electronics

Special Issue Reprint

Advances in Intelligent Data Analysis and Its Applications, 2nd Edition

Edited by
Chao Zhang, Wentao Li, Huiyan Zhang and Tao Zhan

mdpi.com/journal/electronics

MDPI

Advances in Intelligent Data Analysis and Its Applications, 2nd Edition

Advances in Intelligent Data Analysis and Its Applications, 2nd Edition

Guest Editors

Chao Zhang
Wentao Li
Huiyan Zhang
Tao Zhan

MDPI

Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Guest Editors

Chao Zhang
School of Computer and
Information Technology
Shanxi University
Taiyuan
China

Wentao Li
College of Artificial
Intelligence
Southwest University
Chongqing
China

Huiyan Zhang
National Research Base of
Intelligent Manufacturing
Service
Chongqing Technology and
Business University
Chongqing
China

Tao Zhan
School of Mathematics and
Statistics
Southwest University
Chongqing
China

Editorial Office
MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Special Issue, published open access by the journal *Electronics* (ISSN 2079-9292), freely accessible at: https://www.mdpi.com/journal/electronics/special_issues/MPI3V70X1X.

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

ISBN 978-3-7258-3137-1 (Hbk)
ISBN 978-3-7258-3138-8 (PDF)
https://doi.org/10.3390/books978-3-7258-3138-8

Contents

electronics

MDPI

Editorial

Recent Advances in Intelligent Data Analysis and Its Applications, 2nd Edition

Chao Zhang [1,*], Wentao Li [2], Huiyan Zhang [3] and Tao Zhan [4]

[1] School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China
[2] College of Artificial Intelligence, Southwest University, Chongqing 400715, China; drliwentao@gmail.com
[3] National Research Base of Intelligent Manufacturing Service, Chongqing Technology and Business University, Chongqing 400067, China; huiyanzhang@ctbu.edu.cn
[4] School of Mathematics and Statistics, Southwest University, Chongqing 400715, China; zhantao@swu.edu.cn
* Correspondence: czhang@sxu.edu.cn

1. Introduction

The swift growth of cloud computing, the Internet of Things, and the industrial Internet has brought about a surge in complex data analysis tasks that are deeply intertwined with societal and economic progress [1]. To address these intricate challenges, computational intelligence has emerged as a crucial solution, leveraging advanced modeling techniques and cognitive computing [2]. These tools not only tackle the increasing complexity of data, but also play a vital role in shaping informed strategies for development and innovation.

In the realm of tackling challenges related to intelligent data analysis [3], one of the core dilemmas that emerges is how to effectively manage, model, and process the vast and varied datasets generated by the adoption of cutting-edge technologies. This challenge underscores the necessity for exploring and developing robust models and methodologies that harness the power of computational intelligence to advance intelligent data analysis and drive impactful applications [4,5]. In today's rapidly evolving landscape, a growing number of scholars and practitioners from diverse disciplines have come together to create a comprehensive body of work on intelligent data analysis, each contributing unique perspectives. These contributions span various domains, including data mining, social networks, natural language processing, granular computing, cognitive computing, machine learning, and a variety of other interdisciplinary and hybrid approaches.

Data mining [6] has become an indispensable tool for uncovering hidden patterns and trends within large datasets, facilitating data-driven decisions across various domains. Social networks [7], by revealing user behaviors, interaction patterns, and information dissemination pathways, has become a crucial tool for understanding and optimizing online social platforms, marketing strategies, and public opinion monitoring. Natural language processing [8] enables the extraction of meaningful insights from unstructured textual data, expanding the scope of data analysis into linguistically rich contexts. Granular computing [9] provides a structured approach to handling data at varying levels of abstraction, enhancing flexibility, adaptability, and interpretability in decision-making systems. Cognitive computing, by integrating human-like reasoning and learning capabilities into data analysis systems, facilitates intelligent decision-making and problem-solving in complex environments. Machine learning [10] enhances this process by offering sophisticated algorithms that automate the identification of complex relationships and predictive modeling, empowering industries to optimize their processes. These fields, alongside various interdisciplinary and hybrid approaches, demonstrate the vast potential of intelligent

Received: 3 January 2025
Accepted: 7 January 2025
Published: 8 January 2025

Citation: Zhang, C.; Li, W.; Zhang, H.; Zhan, T. Recent Advances in Intelligent Data Analysis and Its Applications, 2nd Edition. *Electronics* **2025**, *14*, 228. https://doi.org/10.3390/electronics14020228

data analysis to tackle complex problems and contribute transformative solutions across diverse sectors.

In the Big Data era, intelligent data analysis exploration holds crucial importance across a broad spectrum of scenarios [11]. These efforts not only tackle immediate challenges, but also contribute to the advancement of the computer science and engineering fields, paving the way for a future characterized by enhanced data literacy and technological innovation. Such undertakings facilitate the navigation of increasingly complex data landscapes and offer innovative solutions for various industries, driving continuous technological progress and the intelligent transformation of society.

The inaugural issue of this Special Issue has been successfully published, featuring a gathering of superior scholarly papers. Building on this achievement, the objective of this Special Issue is to continue gathering the current advancements in intelligent data analysis and examine their uses across various real-world domains. This Special Issue includes 24 papers, covering fields such as decision-making, machine learning, deep learning, anomaly detection, and more.

2. Overview of Contributions

Flight management systems in modern airliners face challenges related to low path planning efficiency and non-smooth trajectories. To address these issues, Li et al. (Contribution 1) explore an improved A* path planning algorithm. By introducing a new "value table" to replace the traditional open and closed tables, retrieval efficiency is enhanced, and the heap sort algorithm optimizes node sorting. A trajectory smoothing optimization algorithm with constraints on turning angles is introduced to overcome the challenge of complex trajectory tracking. Additionally, the methods for calculating gray cost, cumulative cost, and estimated cost have been refined to more effectively adhere to obstacle avoidance constraints.

A growing focus has been placed on conflict analysis in intelligent decision-making, yet trustworthiness in terms of common agreement and opposition remains under explored. Zhai et al. (Contribution 2) present an L-fuzzy three-way concept lattice and puts forward a hybrid model for conflict analysis, combining it with a knapsack-based strategy to resolve conflicts efficiently while minimizing costs, as demonstrated using a case study.

Event extraction identifies event triggers, classifies events, and generates structured arguments from unstructured text. Nevertheless, a significant challenge arises when triggers and arguments from diverse event types coincide within the same sentence, creating ambiguity. To address this, Chen et al. (Contribution 3) introduce a joint learning framework for overlapping event extraction. The framework incorporates a role pre-judgment module, which utilizes the relationship among event types, roles, and trigger embeddings before argument extraction.

Knowledge graphs often contain missing links, and a primary research area is the prediction of the relationships between entities. To address the high variance caused by unbounded scores in existing models, Wang et al. (Contribution 4) propose RotatE Cosine Similarity, which uses joint cosine similarity of complex vectors to achieve bounded scores and model complex relational patterns.

Overconfidence in deep neural networks (DNNs) can hinder generalization and increase risk, while deep ensemble methods improve robustness and accuracy but are resource-intensive and complex to implement. Cao et al. (Contribution 5) present an efficient deep ensemble approximation strategy that requires no additional cost by sampling and saving several optimal parameters during training using a cycle learning rate strategy. These parameters are used as ensemble weights during testing to enhance performance.

Experiments on static image and dynamic video benchmarks demonstrate that the method reduces calibration error and improves model accuracy.

Chen et al. (Contribution 6) introduce an innovative approach for dynamic topic analysis of traffic trajectories, addressing the limitations of traditional methods in analyzing temporal consistency. By embedding spatial information into trajectory words and leveraging dynamic topic modeling, the method explores the time-dependent changes in trajectory topics while using an enhanced Markov decision process to create representative trajectory sequences. To analyze trajectory topics and their changes, they developed a parallel window-based spatiotemporal visualization model with significant visual symbols. Case studies demonstrated the method's effectiveness in uncovering hidden movement patterns in trajectories.

Prediction models often struggle with small sample sizes and incomplete data, leading to inadequate training and low accuracy. Peng et al. (Contribution 7) explore the SALGAN-CatBoost-SSAGA framework to improve prediction accuracy with small, incomplete datasets. Missing data are interpolated, outliers are detected, and SALGAN generates synthetic samples for training. A hybrid sparrow search algorithm and genetic algorithm optimizes vatboost parameters, enhancing accuracy and stability.

Lightweight neural networks often struggle with accurate 6DoF pose estimation in the presence of scale variation problems. Wang et al. (Contribution 8) propose an improved PVNet-based approach that integrates depth-wise convolution for a lightweight design, along with coordinate attention and Atrous Spatial Pyramid Pooling to enhance robustness and accuracy. This approach reduces network size and computational complexity, enabling effective 6DoF pose estimation from monocular RGB images.

Multi-view graph clustering algorithms utilizing representation learning have garnered interest, but face challenges with high-frequency noise and insufficient view complementarity integration. Zhao et al. (Contribution 9) propose a local–global representation enhancement algorithm to address these issues. A low-pass graph encoder improves local representations by enhancing low-frequency signals for better clustering performance, while an attention mechanism integrates weighted local representations into a global representation. The global representation is refined through neighborhood contrastive loss and reconstruction loss, ensuring higher quality. Subsequently, the K-means algorithm is used to perform clustering on the global representation.

Deep network-based models have excelled at extracting discriminative features using convolutional neural networks and recurrent neural networks, yet their performance often diminishes with data that lack temporal or spatial structures. To tackle these limitations, Du et al. (Contribution 10) conduct a hybrid approach that integrates decision trees and neural networks. The model employs decision forest layers to capture essential features and fully connected layers to enhance them, utilizing gradient-boosted decision trees for effective feature selection. Designed for tasks like classification, regression, and ranking, it is trained efficiently with stochastic gradient descent and delivers strong results across diverse machine learning applications.

In autonomous driving and robotics, particularly in complex and dynamic environments, accurate and reliable environmental perception poses a significant challenge. To address the limitations of traditional vision-based methods, Dai et al. (Contribution 11) proposed a fusion of YOLOv8 deep learning object detection, leveraging the advantages of both technologies. YOLOv8 offers robust real-time object detection and classification, whereas LiDAR offers precise spatial and depth information, unaffected by lighting conditions. This integrated approach improves environmental awareness by aligning 3D LiDAR data with image-based detections, filtering ground points, clustering objects, and tracking their positions and distances.

Large-scale group decision-making (LSGDM) involves consolidating diverse opinions from numerous participants to reach collective decisions, often complicated by fuzzy uncertainties, participant diversity, and low consensus levels. Yuan et al. (Contribution 12) present a novel method incorporating dual fine-tuning and online review analysis to address these challenges. A sentiment analysis (SA) is applied to extract and quantify emotional tendencies from online reviews, transforming them into a fuzzy dataset. Decision-makers (DMs) are then clustered using the Louvain algorithm, with enhanced similarity measurements achieved through a combination of Euclidean and Wasserstein distances. The consensus-reaching process (CRP) refines decision scores, first by adjusting representative scores within smaller groups and then by optimizing scores with the lowest consensus levels. The adjusted scores are further analyzed using prospect–regret theory to determine rankings and comprehensive evaluations.

The rapid growth of the e-commerce sector, particularly in online food delivery, underscores the importance of accurate delivery time predictions for customer satisfaction. Although GPS tracking provides courier locations, the lack of real-time traffic and road condition data complicates these estimations. Existing approaches, including machine learning and neural networks, face challenges such as dependency on extensive high-quality data. To address this, Pomykacz et al. (Contribution 13) introduce two Bayesian generalized linear models for delivery time prediction using a linear combination of predictors and the Hamiltonian Monte Carlo sampling technique. Their ability to effectively balance flexibility and generalizability is enhanced by the integration of expert knowledge for tuning.

With the rising popularity of online food delivery services, accurate delivery time predictions have become essential for maintaining customer satisfaction in a competitive market. Gibas et al. (Contribution 14) focus on improving delivery time estimates by enhancing data quality through spatial analysis and preprocessing techniques for Bayesian modeling. Using route data generated via the OSRM API, suspicious results were visualized and analyzed, identifying outliers and refining maximum route distance boundaries. These outliers were predominantly connected to deliveries in areas beyond city limits. By addressing data anomalies and improving data quality, the study aims to enhance the precision of delivery time predictions, supporting better customer experiences.

To tackle the path planning challenges, Dai et al. (Contribution 15) introduce a Hybrid Decentralized and Centralized Training and Execution Strategy designed to enhance computational efficiency and system performance. The approach begins with decentralized path planning using deep Q-networks, enabling robots to independently determine initial paths. A centralized collision detection phase then identifies potential risks, allowing for non-colliding paths to proceed while triggering dynamic re-planning for intersecting routes. During re-planning, robots treat others as moving obstacles to avoid, guaranteeing seamless operation. The optimized paths are subsequently merged with the initial safe routes to create full trajectories.

In the fast-paced digital era, banking firms need to integrate qualitative feedback with quantitative analysis to refine website interfaces, enhancing user experiences and building customer loyalty. Giannakopoulos et al. (Contribution 16) analyze user behavior on the websites of major banking firms by leveraging Big Data, statistical validation, and simulation techniques like agent-based modeling and system dynamics. These methods provided insights into how user interactions and website metrics, including traffic sources and bounce rates, influence key performance indicators such as conversion rates and aid performance. Additionally, integrating neuromarketing data revealed areas for interface and performance optimization, facilitating banking firms in designing more user-focused and intuitive online platforms.

The accurate monitoring and prediction of heart rate (HR) are crucial for optimizing personalized fitness experiences and improving cardiovascular health. As wearable technology continues to grow, tracking HR has become more feasible, but modeling of HR responses to workout intensity remains challenging in real-world applications. Kayange et al. (Contribution 17) introduce a hybrid method combining dynamic Bayesian networks with long short-term memory networks to model HR dynamics during exercise. This approach accounts for individual fitness characteristics and external factors, offering personalized HR predictions for future workouts. Additionally, it incorporates an adaptive feature selection module to enhance performance. Experimental results validate the method's ability to predict HR responses accurately and provide real-time, personalized fitness recommendations.

Advancements in object detection have significantly influenced the progress of face detection, with the YOLO series being recognized for its computational efficiency. However, lightweight YOLO-based face detectors face challenges in accurately identifying small faces because of the loss of critical details. Liu et al. (Contribution 18) tackle these challenges through two improvements: replacing the conventional feature pyramid network with a gather-and-distribute mechanism to enhance information fusion, and adding a specialized detection head for small faces.

Botnets using domain generation algorithms challenge network security by evading traditional defenses, with existing detection models often struggling due to limited training data. Tang et al. (Contribution 19) introduce a multi-scale feature fusion model that integrates transformer and rapid selective kernel network architectures. This approach integrates adaptive receptive field selection, optimized multi-scale convolution, and an enhanced feature pyramid network for improved feature fusion.

Patents offer valuable insights into the technologies developed in various fields. Park et al. (Contribution 20) focus on analyzing patent documents to extract keywords using text mining techniques, which were then used to build a keyword-document matrix for further analysis. However, the prevalence of zero values in the matrix makes it difficult for conventional statistical techniques, like linear regression, to effectively process the data. To address this issue, this article proposes a regression model derived from the quantile cumulative distribution function, specifically designed to manage the zero-inflation issue in patent keyword analysis.

Wang et al. (Contribution 21) present a model reduction technique using singular perturbation theory to simplify spacecraft electrical system models, addressing the challenge of balancing simulation efficiency and accuracy. The proposed method significantly reduces simulation time while maintaining high accuracy, making it suitable for practical applications like digital twins. This approach provides an efficient solution for fast-paced simulations of spacecraft electrical systems and shows promising potential for broader use.

Kim et al. (Contribution 22) explore the relationship among SAs of environmental, social, and governance (ESG) news and the financial outcomes of companies in innovative industries. By analyzing a significant number of ESG-related articles using advanced machine learning models, the study conducted a SA to extract crucial ESG terms and their relevance across different industries. The analysis linked sentiment outcomes with financial indicators like profitability, cash flow, and stability a span of three years. Additionally, ESG ratings from Morgan Stanley Capital International were incorporated to enrich the study. The findings indicate that the influence of sentiment on financial performance differs by industry, with positive sentiment aligning with financial success in mobility and renewable energy, whereas consumer goods companies frequently display positive sentiment despite lower environmental ESG scores. This article emphasizes the importance of tailoring ESG strategies to specific industries, particularly in rapidly evolving sectors, and calls for further research to refine ESG SA.

False data injection attacks pose significant risks to the stability of power systems, as they can deceive detection systems by closely mimicking normal data. Existing detection methods often struggle to accurately identify these attacks, especially when the data are contaminated by ambient noise. To tackle these problems, Luo et al. (Contribution 23) proposed an innovative approach using an attentional convolutional neural network driven by distinction improvements and information integration. The approach first employs an autoencoder designed to minimize reconstruction and discrimination losses, making it particularly effective at distinguishing normal data. Then, the model calculates the association matrix of original and reconstructed data, enhancing the distinction between legitimate and fraudulent data. Additionally, to improve the robustness of feature extraction and mitigate the impact of noise, the network includes a convolutional block attention module, which helps prioritize crucial features.

Amid the swift expansion of the digital economy and the increasing participation in social networks (SNs), decision making and CRP has turned more complex. Xu et al. (Contribution 24) present an enhanced CRP mechanism designed to address the challenges of LSGDM in SNs. As the quantity of DMs rises, consensus efficiency declines, and minority views and non-cooperative behaviors complicate decision-making processes. The proposed solution integrates the hippopotamus optimization algorithm (HOA) with Leiden clustering to speed up community division, known as HOAL, and introduces a two-stage opinion adjustment strategy to manage minority views, non-cooperative behaviors, and fine-tune subgroup perspectives. Trust relationships among DMs are incorporated into both the clustering and opinion adjustment procedures, creating the HOAL-DFT-MOH framework.

Author Contributions: C.Z., W.L., H.Z. and T.Z. worked together in the whole editorial process of the Special Issue, 'Advances in Intelligent Data Analysis and Its Applications, 2nd Edition', published by the journal *Electronics*. H.Z. and T.Z. drafted this editorial introduction. C.Z. and W.L. reviewed, edited, and finalized the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This editorial was supported in part by the Central Government Guides Local Science and Technology Innovation (No. YDZJSX2024D015), the China Postdoctoral Science Foundation (No. 2023T160401), the Natural Science Foundation of Chongqing (No. CSTB2023NSCQ-MSX0152), the Cultivate Scientific Research Excellence Programs of Higher Education Institutions in Shanxi (CSREP) (No. 2019SK036), the Wenying Young Scholars of Shanxi University, the Special Fund for Science and Technology Innovation Teams of Shanxi (No. 202204051001015), the Science and Technology Research Program of Chongqing Education Commission (Nos. KJON202300202; KJON202100205; KJON202100206), and the Training Program for Young Scientific Researchers of Higher Education Institutions in Shanxi.

Conflicts of Interest: The authors declare no conflicts of interest.

List of Contributions

1. Li, J.; Yu, C.; Zhang, Z.; Sheng, Z.; Yan, Z.; Wu, X.; Zhou, W.; Xie, Y.; Huang, J. Improved A-Star Path Planning Algorithm in Obstacle Avoidance for the Fixed-Wing Aircraft. *Electronics* **2023**, *12*, 5047.
2. Zhai, Y.; Jia, Z.; Li, D. Resolving Agent Conflicts Using Enhanced Uncertainty Modeling Tools for Intelligent Decision Making. *Electronics* **2023**, *12*, 4547.
3. Chen, Q.; Yang, K.; Guo, X.; Wang, S.; Liao, J.; Zheng, J. Joint Overlapping Event Extraction Model via Role Pre-Judgment with Trigger and Context Embeddings. *Electronics* **2023**, *12*, 4688.
4. Wang, L.; Luo, J.; Deng, S.; Guo, X. RoCS: Knowledge Graph Embedding Based on Joint Cosine Similarity. *Electronics* **2023**, *13*, 147.

5. Cao, Z.; Li, Y.; Kim, D.; Shin, B. Deep Neural Network Confidence Calibration from Stochastic Weight Averaging. *Electronics* **2024**, *13*, 503.
6. Chen, H.; Wu, Y.; Tang, H.; Lei, J.; Wang, G.; Zhao, W.; Liao, J.; Wang, F.; Wang, Z. Visual Analysis Method for Traffic Trajectory with Dynamic Topic Movement Patterns Based on the Improved Markov Decision Process. *Electronics* **2024**, *13*, 467.
7. Peng, Y.; Liu, Y.; Wang, J.; Li, X. A Novel Framework for Risk Warning That Utilizes an Improved Generative Adversarial Network and Categorical Boosting. *Electronics* **2024**, *13*, 1538.
8. Wang, F.; Tang, X.; Wu, Y.; Wang, Y.; Chen, H.; Wang, G.; Liao, J. A Lightweight 6D Pose Estimation Network Based on Improved Atrous Spatial Pyramid Pooling. *Electronics* **2024**, *13*, 1321.
9. Zhao, X.; Hou, Z.; Wang, J. Local-Global Representation Enhancement for Multi-View Graph Clustering. *Electronics* **2024**, *13*, 1788.
10. Du, L.; Song, H.; Xu, Y.; Dai, S. An Architecture as an Alternative to Gradient Boosted Decision Trees for Multiple Machine Learning Tasks. *Electronics* **2024**, *13*, 2291.
11. Dai, Y.; Kim, D.; Lee, K. An Advanced Approach to Object Detection and Tracking in Robotics and Autonomous Vehicles Using YOLOv8 and LiDAR Data Fusion. *Electronics* **2024**, *13*, 2250.
12. Yuan, X.; Xu, T.; He, S.; Zhang, C. An Online Review Data-Driven Fuzzy Large-Scale Group Decision-Making Method Based on Dual Fine-Tuning. *Electronics* **2024**, *13*, 2702.
13. Pomykacz, J.; Gibas, J.; Baranowski, J. Bayesian Modeling of Travel Times on the Example of Food Delivery: Part 2—Model Creation and Handling Uncertainty. *Electronics* **2024**, *13*, 3418.
14. Gibas, J.; Pomykacz, J.; Baranowski, J. Bayesian Modeling of Travel Times on the Example of Food Delivery: Part 1—Spatial Data Analysis and Processing. *Electronics* **2024**, *13*, 3387.
15. Dai, Y.; Kim, D.; Lee, K. Navigation Based on Hybrid Decentralized and Centralized Training and Execution Strategy for Multiple Mobile Robots Reinforcement Learning. *Electronics* **2024**, *13*, 2927.
16. Giannakopoulos, N.T.; Sakas, D.P.; Migkos, S.P. Neuromarketing and Big Data Analysis of Banking Firms' Website Interfaces and Performance. *Electronics* **2024**, *13*, 3256.
17. Kayange, H.; Mun, J.; Park, Y.; Choi, J.; Choi, J. A Hybrid Approach to Modeling Heart Rate Response for Personalized Fitness Recommendations Using Wearable Data. *Electronics* **2024**, *13*, 3888.
18. Liu, L.; Wang, G.; Miao, Q. ADYOLOv5-Face: An Enhanced YOLO-Based Face Detector for Small Target Faces. *Electronics* **2024**, *13*, 4184.
19. Tang, J.; Guan, Y.; Zhao, S.; Wang, H.; Chen, Y. DGA Domain Detection Based on Transformer and Rapid Selective Kernel Network. *Electronics* **2024**, *13*, 4982.
20. Park, S.; Jun, S. Patent Keyword Analysis Using Regression Modeling Based on Quantile Cumulative Distribution Function. *Electronics* **2024**, *13*, 4247.
21. Wang, L.; Peng, Y.; Luo, J. Model Reduction Method for Spacecraft Electrical System Based on Singular Perturbation Theory. *Electronics* **2024**, *13*, 4291.
22. Kim, M.; Kang, J.; Jeon, I.; Lee, J.; Park, J.; Youm, S.; Jeong, J.; Woo, J.; Moon, J. Differential Impacts of Environmental, Social, and Governance News Sentiment on Corporate Financial Performance in the Global Market: An Analysis of Dynamic Industries Using Advanced Natural Language Processing Models. *Electronics* **2024**, *13*, 4507.
23. Luo, C.; Zhou, Z.; Jiang, R.; Zheng, B. Attentional Convolutional Neural Network Based on Distinction Enhancement and Information Fusion for FDIA Detection in Power Systems. *Electronics* **2024**, *13*, 4862.
24. Xu, T.; He, S.; Yuan, X.; Zhang, C. Enhancing Group Consensus in Social Networks: A Two-Stage Dual-Fine Tuning Consensus Model Based on Adaptive Leiden Algorithm and Minority Opinion Management with Non-Cooperative Behaviors. *Electronics* **2024**, *13*, 4930.

References

1. Huo, R.; Zeng, S.Q.; Wang, Z.H.; Shang, J.J.; Chen, W.; Huang, T. A comprehensive survey on blockchain in industrial internet of things: Motivations, research progresses, and future challenges. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 88–122. [CrossRef]
2. Gupta, S.; Kar, A.K.; Baabdullah, A.; Al-Khowaiter, W.A.A. Big data with cognitive computing: A review for the future. *Int. J. Inf. Manag.* **2018**, *42*, 78–89. [CrossRef]

3. Peres, R.S.; Rocha, A.D.; Leitao, P.; Barata, J. IDARTS–Towards intelligent data analysis and real-time supervision for industry 4.0. *Comput. Ind.* **2018**, *101*, 138–146. [CrossRef]
4. Ferraris, C.; Amprimo, G.; Cerfoglio, S.; Masi, G.; Vismara, L.; Cimolin, V. Machine-Learning-Based Validation of Microsoft Azure Kinect in Measuring Gait Profiles. *Electronics* **2024**, *13*, 4739. [CrossRef]
5. Sun, X.D.; Zhao, L.X.; Chen, J.Q.; Cai, Y.D.; Wu, D.M.; Huang, J.Z. Non-MapReduce computing for intelligent big data analysis. *Eng. Appl. Artif. Intell.* **2024**, *129*, 107648. [CrossRef]
6. Zhang, C.; Zhang, J.J.; Sangaiah, A.K.; Li, D.Y.; Li, W.T. Evaluating edge artificial intelligence-driven supply chain management platforms using collaborative large-scale fuzzy information fusion. *Appl. Soft Comput.* **2024**, *159*, 111686. [CrossRef]
7. Camacho, D.; Panizo-LLedot, A.; Bello-Orgaz, G.; Gonzalez-Pardo, A.; Cambria, E. The four dimensions of social network analysis: An overview of research methods, applications, and software tools. *Inf. Fusion* **2020**, *63*, 88–120. [CrossRef]
8. Sun, S.L.; Luo, C.; Chen, J.Y. A review of natural language processing techniques for opinion mining systems. *Inf. Fusion* **2017**, *36*, 10–25. [CrossRef]
9. Yao, J.T.; Vasilakos, A.V.; Pedrycz, W. Granular computing: Perspectives and challenges. *IEEE Trans. Cybern.* **2013**, *43*, 1977–1989. [CrossRef] [PubMed]
10. Balaji, T.K.; Annavarapu, C.S.R.; Bablani, A. Machine learning algorithms for social media analysis: A survey. *Comput. Sci. Rev.* **2021**, *40*, 100395.
11. Zhang, R.; Meng, Z.H.; Wang, H.L.; Liu, T.H.; Wang, G.; Zheng, L.; Wang, C. Hyperscale data analysis oriented optimization mechanisms for higher education management systems platforms with evolutionary intelligence. *Appl. Soft Comput.* **2024**, *155*, 111460. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

DGA Domain Detection Based on Transformer and Rapid Selective Kernel Network

Jisheng Tang [1], Yiling Guan [1], Shenghui Zhao [1,*], Huibin Wang [1] and Yinong Chen [2]

[1] School of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, China; 2021212074@mail.chzu.edu.cn (J.T.); 2021211908@mail.chzu.edu.cn (Y.G.); wanghuibin@chzu.edu.cn (H.W.)

[2] School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ 85287, USA; yinong@asu.edu

* Correspondence: zsh@chzu.edu.cn

Abstract: Botnets pose a significant challenge in network security by leveraging Domain Generation Algorithms (DGA) to evade traditional security measures. Extracting DGA domain samples is inherently complex, and the current DGA detection models often struggle to capture domain features effectively when facing limited training data. This limitation results in suboptimal detection performance and an imbalance between model accuracy and complexity. To address these challenges, this paper introduces a novel multi-scale feature fusion model that integrates the Transformer architecture with the Rapid Selective Kernel Network (R-SKNet). The proposed model employs the Transformer's encoder to couple the single-domain character elements with the multiple types of relationships within the global domain block. This paper proposes integrating R-SKNet into DGA detection and developing an efficient channel attention (ECA) module. By enhancing the branch information guidance in the SKNet architecture, the approach achieves adaptive receptive field selection, multi-scale feature capture, and lightweight yet efficient multi-scale convolution. Moreover, the improved Feature Pyramid Network (FPN) architecture, termed EFAM, is utilized to adjust channel weights for outputs at different stages of the backbone network, leading to achieving multi-scale feature fusion. Experimental results demonstrate that, in tasks with limited training samples, the proposed method achieves lower computational complexity and higher detection accuracy compared to mainstream detection models.

Keywords: botnet; domain generation algorithm; transformer model; depthwise separable convolution

Citation: Tang, J.; Guan, Y.; Zhao, S.; Wang, H.; Chen, Y. DGA Domain Detection Based on Transformer and Rapid Selective Kernel Network. *Electronics* **2024**, *13*, 4982. https://doi.org/10.3390/electronics13244982

Academic Editors: Wentao Li, Huiyan Zhang, Tao Zhan and Chao Zhang

Received: 16 September 2024
Revised: 28 November 2024
Accepted: 11 December 2024
Published: 18 December 2024

1. Introduction

With the continuous development and iteration of Internet technology, people's lifestyles have become more convenient. However, this progress has also led to a rise in both the number and complexity of malware, giving rise to black-market industries and cybercriminal activities, such as botnets and phishing. Attackers use malware to infect networks, issuing commands to compromised machines or botnets to steal sensitive data, launch Distributed Denial of Service (DDoS) attacks, and carry out other cyberattacks, posing significant challenges to global network security. To evade security checks and maintain control over compromised hosts for extended periods, thereby maximizing economic benefits, modern malware attackers frequently integrate DGA into malware. DGA enables the rapid generation of numerous dynamic domains to connect with Command and Control (C&C) servers. The malware client and attacker both use the same DGA algorithm to generate a list of candidate domains. When preparing for an attack, a subset of these domains is registered to establish communication. Due to the high randomness and dynamic nature of DGA-generated domains, malware can rotate domains throughout its attack cycle, making it nearly impossible for network security tools to block all malicious domains via blacklists. Consequently, accurately detecting DGA-generated domains and disrupting the communication between botnets and C&C servers is crucial for malware defense.

In recent years, the rapid development of deep learning has demonstrated its vast potential across various fields, leading researchers to explore its application in DGA detection. Compared to machine learning methods based on domain traffic characteristics and manually defined statistical features, deep learning eliminates the need for manual feature extraction and significantly improves the accuracy of DGA detection. Therefore, deep learning has become a key technology for enhancing the efficiency and precision of DGA detection.

Current DGA detection models commonly use Transformer encoders to capture complex long-range dependencies between characters within domain names. The resulting feature sequences are then processed through a multi-branch convolutional network architecture to extract features from different receptive fields, followed by a Bidirectional Long Short-Term Memory (Bi-LSTM) network to obtain bidirectional sequence features. The final classification is achieved through feature fusion. However, this network architecture still has several weaknesses:

- While parallel multi-scale convolution benefits from multiple receptive field features, the information between branches remains independent and lacks effective coupling. Additionally, the features extracted from each receptive field are complex, and a simple parallel structure cannot adequately capture comprehensive features across all scales, reducing detection accuracy. Furthermore, although ordinary convolution can extract more comprehensive features, it results in high training costs.
- The outputs of each stage in the backbone network are merely concatenated, which prevents the extraction of hierarchical features at different stages. This leads to feature redundancy and negatively impacts detection accuracy.

To address these weaknesses, this paper proposes R-SKNet and EFAM to deeply extract domain features, enhancing model performance through rapid adaptive selection of receptive field scales and efficient feature fusion across different stages. The contributions of this paper can be summarized as follows:

- This paper proposes R-SKNet, which allows for rapid adaptive selection of multi-scale receptive fields within the network, effectively coupling feature information across branches.
- This paper proposes an improved feature fusion strategy, EFAM, which facilitates efficient multi-scale feature fusion and enhances hierarchical features. It further refines the convolution strategy to capture spatial focus points better.
- Based on Transformer, R-SKNet, and EFAM, this paper proposes a DGA domain detection framework, TransFlexNet, which takes the advantages and the strengths of these three models. It outperforms mainstream models under both large and small sample conditions.

With the growing complexity of network threats, existing DGA detection methods often struggle with effective feature extraction and fusion, making it difficult to counter the evolving tactics of malware attacks. To address these challenges, this paper introduces the TransFlexNet detection framework, designed to improve the precision and efficiency of DGA detection and strengthen network security defenses.

The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 describes the proposed DGA detection model. Section 4 discusses and analyzes the datasets and experimental results. Section 5 concludes the paper with future directions.

2. Related Work

In today's cybersecurity field, DGA detection has become a crucial research topic. With the continuous advancement and application of machine learning and deep learning technologies, these advanced techniques have been integrated into traditional algorithmic models to address the challenges in DGA detection. These challenges include the need for continuous updates to detection methods and models as attackers use various algorithms, the difficulty of feature extraction due to the high similarity between random and legitimate domain names, and the impact of dataset bias on the generalization ability of models. Building on these points, we analyze existing DGA detection techniques.

2.1. Machine-Learning-Based Detection Methods

Traditional machine learning methods rely on large amounts of annotated data, which are difficult to obtain in real-world applications. To address this issue, Yan et al. [1] proposed a semi-supervised learning strategy that constructs domain relationship graphs and utilizes unlabeled data to enhance the identification of DGA family evolution and new variants. This approach maintains high efficiency and accuracy, even with limited labeled resources. Anand et al. [2] integrated C5.0, Random Forest, Gradient Boosting, and CART, and used 44 comprehensive lexical and statistical features, significantly improving the recognition accuracy of DGA domain names and the robustness of the model. Satoh et al. [3] conducted word-level feature analysis, using specific lexical generation patterns to identify dictionary-based DGA malware, highlighting the importance of semantic and linguistic features. Vranken et al. [4] applied TF-IDF techniques to analyze n-grams in domain names, reducing dependence on large-scale annotated data and improving the accuracy of DGA domain name detection.

Although these methods have achieved breakthroughs in specific scenarios, they still have limitations in terms of feature dependence and generalization capabilities. Semi-supervised learning heavily relies on the quality of labeled data, and its performance degrades when the labeling quality is poor. Anand's integrated model relies on complex and time-consuming feature engineering, which does not take the advantages of certain critical features. TF-IDF relies on known vocabularies and performs poorly when facing the novel DGA variants. In contrast, deep learning demonstrates greater advantages through automated feature extraction.

2.2. Deep-Learning-Based Detection Methods

Deep learning techniques offer new solutions to problems in machine learning. Yang et al. [5] developed the N-Trans parallel detection model, which combines the N-gram algorithm and the Transformer model. By adding flag bits, it effectively extracts domain name features, allowing precise differentiation between legitimate and malicious domain names. Namgung et al. [6] proposed an integrated model combining BiLSTM and CNN, utilizing an attention mechanism to more effectively learn both local and global information in domain name sequences, significantly enhancing DGA domain detection performance. Shahzad et al. [7] developed an RNN-based DGA domain classifier that does not require context information or manual features. By analyzing a dataset of over 2 million domain names, they demonstrated the effectiveness of the RNN architecture. Liang et al. [8] found that DGA detection models are sensitive to domain name length and proposed a heterogeneous DGA detection model based on CNN and Random Forest, optimized for domain names of different lengths. Qi et al. [9] introduced the CNN–LSTM model, which combines CNN and LSTM to extract N-gram features through CNN and process them with LSTM, effectively classifying and predicting dictionary-based DGA domain names. Jiang et al. [10] proposed the CNN–GRU–Attention model, which uses CNN to extract spatial features, GRU to extract temporal features, and an attention mechanism to improve detection accuracy.

The aforementioned deep learning methods have achieved promising detection results. However, their use of standard parallel convolution in feature extraction fails to capture comprehensive multi-scale features and incurs high training costs. To address these issues, this paper introduces ECA, which computes attention weights without dimensionality reduction, thereby reducing the model's parameter count.

2.3. Detection Methods Based on Additional Mechanisms and Other Algorithms

In certain scenarios, incorporating additional modules into deep learning frameworks can enhance performance. Ren et al. [11] proposed the ATT–CNN–BiLSTM framework, which combines CNN and BiLSTM with an attention mechanism to extract and optimize domain name features effectively. Pan et al. [12] improved word embedding methods by enhancing character features through the extraction of semantic features and adding additional mechanisms to improve detection accuracy.

Some detection algorithms do not use traditional machine learning or deep learning algorithms. Fang et al. [13] used an improved Word2Vec algorithm and incremental word embedding methods to capture the interactions and time series patterns between domains and endpoint hosts. By automatically learning the features of over 1.9 million domains, they developed a simple classifier to distinguish between malicious and benign domain names.

In summary, traditional machine learning heavily relies on feature engineering, while deep learning offers the advantage of automatic feature extraction. Building upon existing DGA models, our research demonstrates that integrating the ECA module enables efficient multi-scale feature fusion, effectively reduces model parameters, and enhances accuracy.

3. DGA Domain Detection Model

This section presents the domain detection framework, model, its layers and components, as well as their functions.

3.1. Overall Framework

To address the limitations of parallel multiscale convolution in terms of feature coupling and focus aggregation, this paper employs the Selective Kernel Network (SKNet) [14] for feature extraction, optimizing the traditional parallel convolution architecture. SKNet enhances inter-branch feature fusion by implementing an attention mechanism and adaptively adjusts the receptive field weights. However, the dimensional scaling approach used by SKNet may compromise feature integrity and increase training costs. To mitigate this, we introduce the ECA [15] mechanism, which allocates attention across receptive fields through limited cross-channel interactions. Additionally, the paper incorporates Deep Separable Dilated Convolution [16], a method that expands the receptive fields without increasing parameter counts, providing richer semantic information for the backbone network.

To address the issues of insufficient hierarchical feature extraction and feature redundancy resulting from the simple concatenation of outputs from each stage of the backbone network, this paper introduces the FPN [17] to achieve multi-scale feature fusion. By layering up-sampling and lateral connections between different levels, FPN effectively integrates features across scales to enhance detection accuracy. To further enhance fusion efficiency, this paper proposes an improved fusion strategy, EFAM, which reduces feature redundancy and improves aggregation by incorporating an efficient attention mechanism and refining the convolution strategy to better focus on spatial details.

Building on these enhancements, this study introduces a multi-scale feature fusion model utilizing Transformer and R-SKNet. The model consists of three layers: an input layer that converts domain name strings into vectors for word embedding; a feature extraction layer composed of the Transformer Encoder, R-SKNet, and EFAM; and an output layer that employs 1D convolution instead of traditional fully connected layers, outputting the final classification results via a Softmax function. The entire framework is shown in Figure 1.

3.2. Input Layer

The input layer preprocesses raw domain names and converts them into a vector form that machines can interpret. In this study, the length of all domain names is standardized to L. Domain names longer than L are truncated, while those shorter than L are padded with zeros on the right, ensuring that all domain name texts have a uniform length of L, as shown in Figure 2. Let the domain name string be S, where c_i is the i-th character of the domain name, as shown in (1).

$$S = \mathrm{c}_1 c_2 c_3 \cdots c_L \tag{1}$$

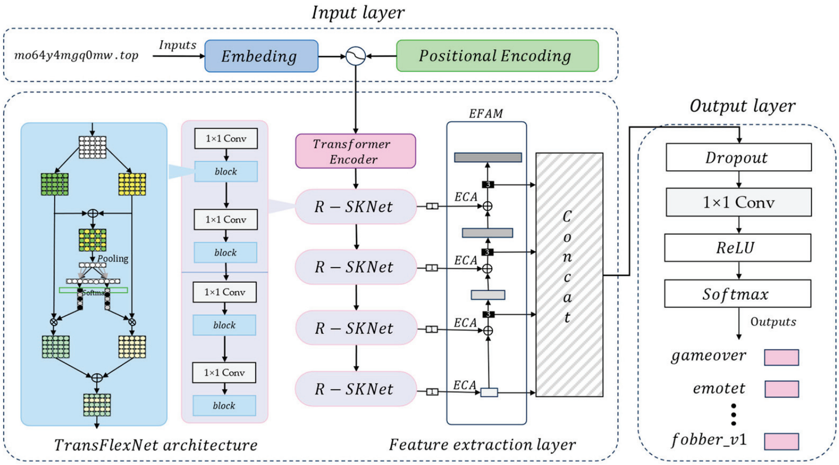


Figure 1. Overall framework.

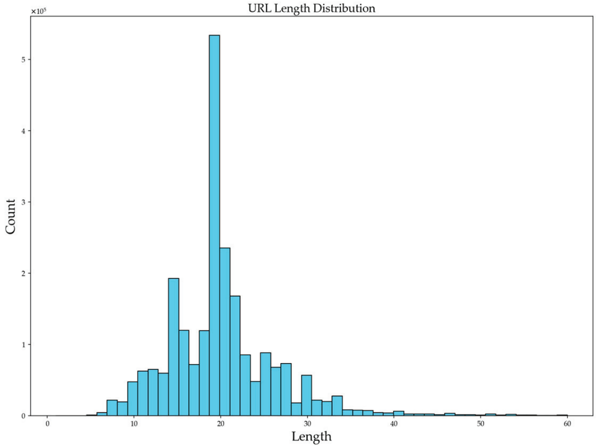


Figure 2. Sample domain length.

As shown in Figure 2, the samples with a length greater than 40 are relatively few. To ensure efficient use of the dataset and to improve data processing efficiency, this study sets the length threshold, L, to 40.

In this paper, a domain name is represented as a unique type of string. The domain name text is split into individual characters, each assigned an index to create an indexed dictionary, D. The original domain name samples are then processed with a tokenizer to generate a list of domain name characters, SD, as shown in (2).

$$D = [D(c_1)D(c_2)\ldots D(c_L)] \tag{2}$$

where $D(c_i)$ is the index value of the domain character c_i in the dictionary D, and $i \in \{1, 2, \ldots, L\}$. The embedding layer's dimension is set to d. According to the embedding layer dimension d and domain name length L, a domain name vector $R^{L\times d}$ is generated by random initialization. For the character vector $x_i \in R^{L\times d}$, it is multiplied by the weight matrix $A^{d\times L}$ to obtain Y_i, and they are concatenated to obtain a new domain name vector E. The weight matrix A is updated during downstream task training to obtain the optimal embedding representation, with d set to 64.

3.3. Feature Extraction Layer

The feature extraction layer integrates the global information encoding capabilities of the Transformer with the rapid adaptive kernel selection mechanism of R-SKNet to target different features effectively. It further incorporates the EFAM efficient fusion attention module to enhance multi-scale feature fusion. This comprehensive design optimizes the model's parameter count and reduces structural complexity while improving the efficiency and accuracy of feature extraction.

3.3.1. Transformer Encoder

The structure of the transformer encoder module is shown in Figure 3.

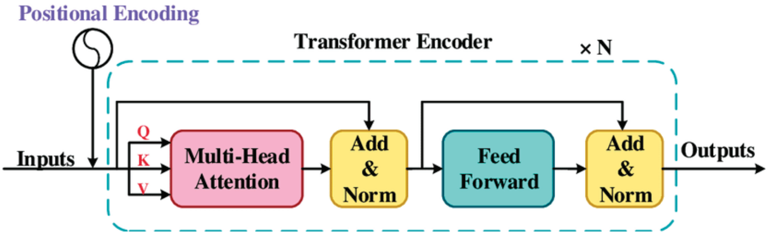


Figure 3. Transformer encoder module.

After processing through the embedding layer, domain names are converted into vector representations, E. These vectors are then passed through the Transformer's multi-head self-attention mechanism, which captures complex interactions between characters by applying multiple attention heads, each focusing on different subspaces of the input. Each head performs linear transformations on the input vector E to generate Q (query), K (key), and V (value). The computation process of the multi-head attention mechanism is expressed as shown in (3).

$$MultiHead(Q,\ K,\ V) = Concat(head_1,\ \cdots,head_h)W^o \tag{3}$$

where each $head_i$ is the output of a single attention head, and W^o is a weight matrix for linear transformation. The calculation process of each $head_i$ is shown in (4).

$$Attention(Q_i, K_i, V_i\ ) = \mathrm{softmax}\left(\frac{Q_i {K_i}^T}{\sqrt{d_k}}\right)V_i \tag{4}$$

where Q_i (query), K_i (key), and V_i (value) matrices are obtained by linear transformation of the original domain name vectors R. The dimension d_k of the key vector is used for normalization, adjusting the softmax distribution to stabilize gradient flow during training.

To ensure that the model adapts to the dynamic changes in input features, the output of the multi-head attention mechanism is normalized through residual connection, as shown in (5).

$$x_i' = \frac{x_i - \mu}{\sigma} \tag{5}$$

where x_i' represents the normalized output and μ and σ represent the mean and standard deviation across the dimensions. Normalization helps to stabilize the training dynamics. The normalized output is then fed into a feed-forward neural network (FFNN), which generates the output of the Transformer encoder, providing a refined feature set for subsequent analysis.

3.3.2. Rapid Selective Kernel Network

In traditional DGA domain detection models, feature gains from different receptive fields are typically achieved through a multi-branch network structure with fixed receptive field sizes. This design leads to an increase in the number of model parameters and higher computational complexity. SKNet introduces additional parameters through its multi-path kernel selection mechanism, which complicates the model, and increases computational demand by applying convolutional kernels of various sizes across multiple branches to enhance feature processing.

To address this, we propose R-SKNet, which integrates an ECA module into SKNet. R-SKNet achieves multi-receptive field feature extraction by adaptively selecting convolution kernel sizes within a single network, simplifying design and optimizing complexity. The ECA module further reduces parameters while enhancing performance, demonstrating improved detection accuracy with a lightweight model. The structure of each R-SK convolution is shown in Figure 4.

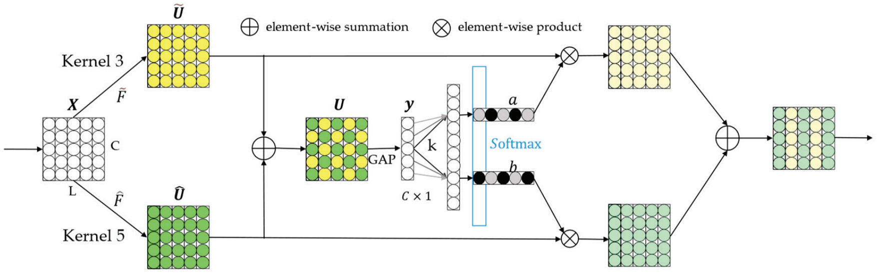


Figure 4. R-SK convolution structure.

In Figure 4, the input domain name matrix $X \in R^{C' \times L'}$ undergoes two different transformation operations, denoted as: $\widetilde{F} : X \rightarrow \widetilde{U} \in R^{C \times L}$ and $\hat{F} : X \rightarrow \hat{U} \in R^{C \times L}$. These operations use one-dimensional convolutional kernels of different sizes (3 and 5) for feature extraction. To enhance model efficiency, traditional convolutions are replaced by depthwise separable dilated convolutions. Batch normalization and the ReLU activation function are applied after each convolution to improve non-linearity and generalization capability. In particular, the 1D convolution with a kernel size of 5 is replaced by a dilated convolution with a kernel size of 3 and a dilation rate of 2.

The depthwise separable convolution technique decomposes standard convolution into depthwise convolution and pointwise convolution, processing each dimension of the input matrix separately. This approach allows each convolution kernel to independently correspond to and process a single dimension, focusing on learning spatial features, while pointwise convolution is responsible for learning features between channels. This decomposition significantly reduces the number of model parameters while preserving learning efficiency.

To gather information from all branches and assign weights to the grouped convolution results, the processing results of each branch are first obtained, and then the features are fused, as shown in (6).

$$U = \widetilde{U} + \hat{U} \tag{6}$$

After obtaining the fused features from different branches, global average pooling is used to generate channel-level statistics. In the DGA domain name detection task, the use of global average pooling (GAP) can achieve better feature aggregation effects. After GAP, the channel-level statistical information $y \in R^{C}$ is generated, as shown in (7).

$$y = \frac{1}{CL} \Sigma_{i=1,j=1}^{C,L} U_{ij} \tag{7}$$

To introduce a channel attention mechanism that adaptively adjusts channel weights, two non-linear fully connected (FC) layers are typically employed, followed by a Sigmoid function to generate the weights. Although this approach enhances the model's ability to capture non-linear cross-channel interactions, it significantly increases model parameters and disrupts the correspondence between channels and weights. To address this, we propose using the ECA module, which replaces traditional FC layers to facilitate local cross-channel interactions without dimensionality reduction. The ECA module effectively facilitates appropriate inter-channel interactions without sacrificing the correspondence between channels, significantly enhancing model performance. The ECA module is shown in Figure 5. The ECA module uses a band matrix W_k to learn channel attention, as shown in Figure 6.

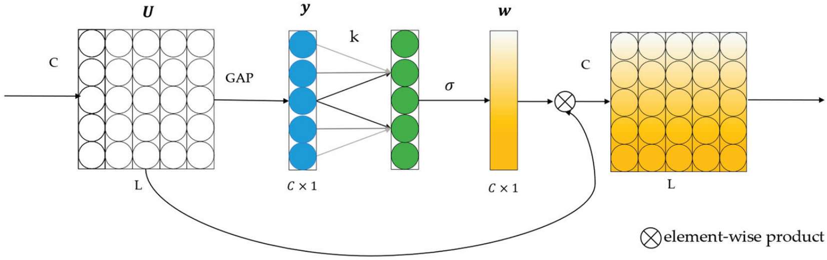


Figure 5. ECA module.

$$\begin{bmatrix} w^{1,1} & \cdots & w^{1,k} & 0 & 0 & \cdots & \cdots & 0 \\ 0 & w^{2,2} & \cdots & w^{2,k+1} & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & w^{C,C-k+1} & \cdots & w^{C,C} \end{bmatrix}$$

Figure 6. Band matrix.

The parameters involved in the band matrix W_k are $K \times C$. Through (8), the weight of y_i considers only the relationship between y_i and its k neighbors. Additionally, all channels share the same learned parameters, enabling local cross-channel interaction. This can be implemented using a 1D convolution with a kernel size of k, as shown in (9).

$$Uw_i = \sigma\left(\sum_{j=1}^{k} w^j y_i^j\right), y_i^j \in \Omega_i^j \tag{8}$$

$$w = \sigma(C1D_k(y)) \tag{9}$$

where Ω_i^k denotes the set of k neighboring channels of y_i, σ represents the Sigmoid function, and $C1D$ denotes the one-dimensional convolution. The value of k is dynamically determined by (10).

$$k = \left| \frac{log_2\left(2^{(C)}\right)}{\gamma} + \frac{b}{\gamma} \right|_{odd} \tag{10}$$

where $|t|_{odd}$ denotes the nearest odd number to t, and C represents the channel dimension. In all experiments, γ and b are set to 2 and 1, respectively. The ECA module, which involves only b parameters, achieves better performance than FC layers, which require more complex operations. Based on the information of w, softmax is performed on the channel dimension to obtain a_c and b_c, as shown in (11).

$$a_c = \frac{e^{\widetilde{U}_c w}}{e^{\widetilde{U}_c w} + e^{\hat{U}_c w}}, b_c = \frac{e^{\hat{U}_c w}}{e^{\widetilde{U}_c w} + e^{\hat{U}_c w}} \tag{11}$$

where a and b are the soft attention vectors of $\widetilde{U}$ and $\hat{U}$ respectively. $\widetilde{U}_c \in R^{C\times 1}$, with a_c being the c-th element of a. Similarly, $\hat{U}_c$ and b_c. The final batch feature map V is derived from the attention weights of each kernel from the branches $\widetilde{U}$ and $\hat{U}$, as shown in (12).

$$\begin{gathered} V_c = a_c \cdot \widetilde{U}_c + b_c \cdot \hat{U}_c, \\ a_c + b_c = 1 \end{gathered} \tag{12}$$

where $V = [V_1, V_2, V_3, \cdots, V_C]$, and $V_c \in R^{C\times 1}$. After obtaining the feature matrix V, a residual connection is made to complete the R-SK convolution operation.

3.3.3. Efficient Fusion Attention Module

The EFAM is a key component of the TransFlexNet architecture. By incorporating the ECA module before both lateral and vertical connections in the FPN and performing convolution after feature fusion, EFAM enhances the efficiency of feature fusion and strengthens feature representation. In the R-SKNet backbone, the output from the last residual block of each R-SK unit, with varying numbers of convolution kernels, is selected as input and passed to the EFAM for feature fusion. The structure of the EFAM is shown in Figure 7.

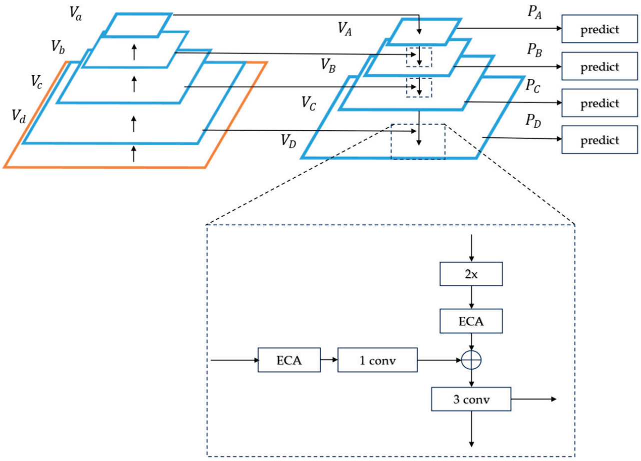


Figure 7. EFAM structure.

Starting with the input feature matrices $\{V_a, V_b, V_c, V_d\}$ from the R-SKNet backbone, each matrix first passes through the ECA module to compute channel-level attention weights. These weights are applied to the original matrices, enhancing the network's response to key features and improving representation capabilities. A 1D convolution with a kernel size of 1 is then used to adjust the channel number of each weighted matrix to 128, resulting in $\{V_a, V_b, V_c, V_d\}$.

To build a multi-scale feature representation hierarchy and promote bottom-up information flow, pooling operations are applied to these adjusted matrices. Specifically, max pooling with a kernel size of 3 and stride of 2 is applied to V_b, average pooling with a kernel size of 3 and stride of 4 to V_c, and average pooling with a kernel size of 3 and stride of 8 to V_d. These operations create feature maps at different scales, with strides set to $\{5, 10, 20, 40\}$. By upsampling higher-level features, the EFAM generates feature matrices with coarser spatial resolution but richer semantic information, establishing a top-down feedback path.

Lateral connections are then used to fuse top-down and bottom-up feature paths. After each fusion, the ECA module readjusts channel attention, followed by a 1D convolution with a kernel size of 3 to extract features, resulting in $\{V_a, V_b, V_c, V_d\}$, which match the original input dimensions. These matrices $\{V_a, V_b, V_c, V_d\}$ serve as the output of the feature extraction layer and the input for the next layer in the pyramid, further strengthening features in the top-down path.

3.3.4. Output Layer

First, the outputs of the feature extraction layer $\{P_A, P_B, P_C, P_D\}$ are fused into a comprehensive feature vector P in the output layer, achieving feature information integration, as shown in (13).

$$P = [P_A,\ P_B,\ P_C,\ P_D] \tag{13}$$

Next, the Dropout layer is used to probabilistically discard elements from the feature vector P to improve the generalization ability of the model and to reduce the risk of overfitting. Subsequently, a convolutional layer with a kernel size of 1 is used to replace the traditional fully connected layer, followed by a ReLU layer and a global average pooling layer, which significantly reduces the number of model parameters, adds non-linear activation, and enhances the network's expressive ability. Finally, the softmax function is used to obtain the probability y of each class, achieving the final classification.

4. Experiments and Result Analysis

4.1. Experimental Environment and Datasets

The experimental setup includes an Intel Core i7-11800H processor, NVIDIA RTX3070 GPU, and 32 GB RAM, operating on Windows 11 with Python 3.11 and PyTorch 2.2.1 for neural network modeling.

We have successfully deployed the detection model on low-performance devices in real-world scenarios, demonstrating its practical applicability. To further validate the model, we designed and implemented a prototype system for model deployment and detection. The system was put on the URL: https://github.com/make666999/jishe_DGA (accessed on 20 November 2024). However, due to the limited occurrence of large-scale DGA domain malicious attack incidents in real network environments, this study utilizes multiple publicly available datasets to construct and validate the domain name classification model.

To improve the breadth and accuracy of the research, different types of datasets are used in this experiment. The first dataset, sourced from Alexa [18] and 360NetLab [19], consists of the top 55,000 benign domains from Alexa and malicious domains from 360NetLab's DGA dataset. The second dataset is sourced from Alexa and DGArchive [20], comprising 57,500 benign domains and 71,841 malicious domains, respectively.

Table 1 lists the three initial samples from 360NetLab and DGArchive. Table 2 lists the three initial samples from Alexa.

Table 1. 360NetLab and DGArchive initial samples.

Dataset	DGA Family	Domian	Start Valid Time	End Valid Time
360NetLab	locky	labywvrw.pw	2020-10-26 00:00:00	2020-10-27 23:59:59
	nymaim	jkybwgxpfr.com	2020-10-27 00:00:00	2020-10-27 23:59:59
	qadars	sqgaoag606cm.top	2020-10-27 00:00:00	2020-11-02 23:59:59
DGArchive	bazarloader_dga_bc7bb913	idfuuhom.bazar	2021/1/1 0:00	2021/1/31 23:59
	bazarloader_dga_bc7bb913	biyxtuom.bazar	2021/12/1 0:00	2021/12/31 23:59
	bazarloader_dga_bc7bb913	etywydom.bazar	2021/10/1 0:00	2021/10/31 23:59

Table 2. Alexa initial samples.

Label	Url
BENIGN	facebook.com
BENIGN	youtube.com
BENIGN	www.google.com

Considering that the sample size of some malicious domain families is insufficient to support model training, families with fewer than 200 samples were excluded to support model training, selecting 33 families from 360NetLab and 42 from DGArchive. To mitigate sample imbalances, domains were randomly sampled from these larger families, and all domain names were standardized to 40 characters through padding or truncation. The processed datasets from 360NetLab and DGArchive were combined with the Alexa dataset, resulting in two comprehensive datasets: Alexa and 360NetLab with 107,541 domain instances and Alexa and DGArchive with 129,341 domain instances. Table 3 lists the results of several datasets after preprocessing.

Table 3. Example after preprocessing.

Label	Url
0	facebook.com
14	labywvrw.pw
18	jkybwgxpfr.com

For binary classification, benign domains are labeled as 0 and malicious domains as 1. In multiclassification, the Alexa and 360NetLab dataset contains 33 DGA families and benign domains, while the Alexa and DGArchive dataset includes 42 DGA families and benign domains, with labels assigned sequentially from 0. Both datasets were randomly divided into training and test sets, with test sets accounting for 20% of the total data. Dataset details and experimental hyperparameters are listed in Tables 4 and 5.

Table 4. Detailed information of dataset.

Category	Alexa and 360NetLab	Alexa and DGArchive
Total Number of Domains	107,541	129,341
Benign Domains	55,000	57,500
DGA Domains	52,541	71,841
Domain Length	40	40
Total Number of Categories	34	42

Table 5. Model hyperparameters.

Hyperparameters	Description
Epochs	100
Batch Size	256
Embedding Dimension	64
Learning Rate	0.001
Step Size for Learning Rate Scheduler	20
Gamma for Learning Rate Scheduler	0.5
Test set proportion	20%
Optimizer	Adam
Loss function	CrossEntropyLoss

4.2. Evaluation Metrics

The evaluation metrics for the experimental results include accuracy, precision, recall, F1 score, and detection time. The calculation formulas are shown in (14) to (17).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{17}$$

where TP is the number of true positives, FN is the number of false negatives, and FP is the number of false positives.

4.3. Binary Classification Experimental Results and Analysis

To verify the performance of the proposed method in DGA domain name detection, binary classification experiments and model parameter comparisons are conducted on the experimental sample set using the proposed method, MHA [21], SW-DRN [22], L-PCAL [23], and the multi-feature fusion model (referred to as TMF). The evaluation metrics included accuracy, recall, precision, F1 score, and parameter count. The experimental results for the Alexa and 360NetLab are shown in Table 6 and Figure 8, and the Alexa and DGArchive's results are shown in Table 7.

Table 6. Binary classification results and model parameters for Alexa and 360NetLab.

Method	Accuracy	Recall	F1 Score	# P
MHA	0.9269	0.9448	0.9260	406,306
SW-DRN	0.9679	0.9717	0.9669	1,963,524
L-PCAL	0.9684	0.9754	0.9676	274,326
TMF	0.9766	0.9754	0.9756	2,771,384
Proposed Method	0.9783	0.9834	0.9777	1,783,544

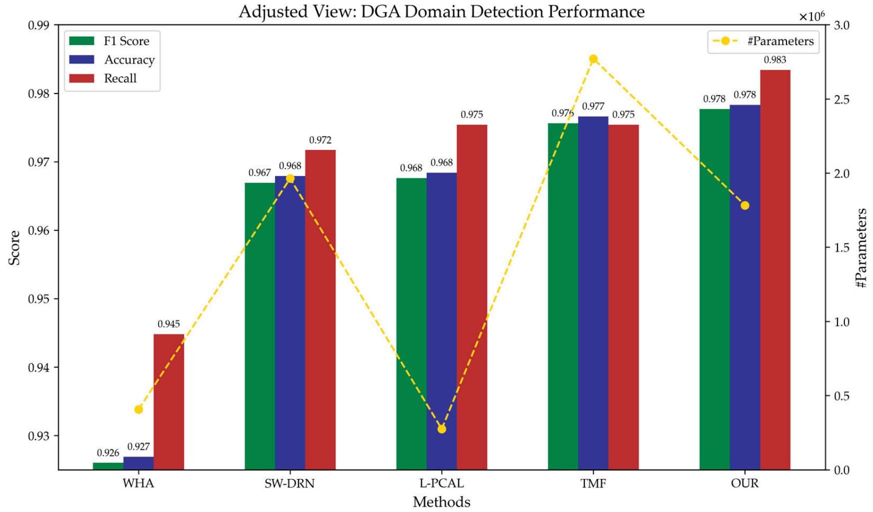


Figure 8. Binary Classification Results and Model Parameters Comparisons.

Table 7. Binary classification results for Alexa and DGArchive.

Method	Accuracy	Recall	Precision	F1 Score
MHA	0.9646	0.9646	0.9634	0.9640
SW-DRN	0.9799	0.9790	0.9802	0.9796
L-PCAL	0.9791	0.9766	0.9808	0.9787
TMF	0.9840	0.9822	0.9852	0.9837
Proposed Method	0.9852	0.9864	0.9852	0.9850

The proposed method integrates the Transformer encoder and SKNet with the ECA module to form the improved R-SKNet. The EFAM combines the ECA module with the FPN structure for multi-scale feature fusion, efficiently extracting feature information. As presented in Table 6, the Alexa and 360NetLab datasets increase the accuracy, recall and F1 score by 1.87%, 1.83%, and 1.66% on average, with peak increases of 5.14%, 3.86%, and 5.17%. Furthermore, the model reduces the parameter count by 35.63% compared to the next-best model. From Table 7, it can be seen that the maximum increase in the Alexa and DGArchive dataset in accuracy, recall, precision, and F1 score reaches 2.12%, 2.06%, 2.18%, and 2.06%, respectively. The method in this paper is more effective and has fewer parameters compared to mainstream models.

4.4. Multi-Classification Experimental Results and Analysis

To validate the effectiveness of the proposed method for DGA domain family classification, we have conducted multi-class domain detection experiments using our method, MHA, SW-DRN, L-PCAL, and TMF on the experimental sample set. The experimental results are shown in Tables 8 and 9.

Table 8. Multi-classification results comparison for Alexa and 360NetLab.

Method	Accuracy	Recall	Precision	F1 Score
MHA	0.8937	0.8937	0.8908	0.8883
SW-DRN	0.9225	0.9225	0.9196	0.9206
L-PCAL	0.9315	0.9315	0.9330	0.9310
TMF	0.9335	0.9335	0.9334	0.9332
Proposed Method	0.9391	0.9391	0.9377	0.9378

Table 9. Multi-classification results comparison for Alexa and DGArchive.

Method	Accuracy	Recall	Precision	F1 Score
MHA	0.9045	0.9045	0.9022	0.8993
SW-DRN	0.9113	0.9113	0.9111	0.9102
L-PCAL	0.9234	0.9234	0.9228	0.9202
TMF	0.9214	0.9214	0.9207	0.9205
Proposed Method	0.9251	0.9251	0.9233	0.9210

From the results in Tables 8 and 9, it is clear that the accuracy, recall, precision, and F1 score for the proposed detection method on the Alexa and 360NetLab dataset are 93.91%, 93.91%, 93.77%, and 93.78%, respectively. On the Alexa and DGArchive dataset, the respective metrics are 92.51%, 92.51%, 92.33%, and 92.10%. The multiclassification detection performance of this method exceeds that of other approaches, with improvements noted across all evaluation metrics.

From the above comparison results, it can be seen that the proposed method performs better in detecting and classifying DGA domain families with fewer samples compared to the other four mainstream methods.

The MHA method uses an LSTM-encoded Transformer detection method, obtaining the dependency between individual characters and the entire domain name, but lacks the learning of multi-scale features and cannot extract deep semantic information, leading to poor detection performance.

The SW-DRN method uses multi-scale convolution to obtain domain name features but does not capture the dependency between characters and uses one-hot encoding, resulting in huge memory consumption.

The L-PCAL method uses multi-scale parallel convolution and LSTM with an attention mechanism to extract short-distance dependencies, but the extraction effect of long-distance dependencies is poor.

The TMF method also uses parallel convolution, and the multi-scale relationships between parallel branch convolutions are isolated and cannot be effectively coupled. The multi-branch feature fusion is simply summed, causing branch feature generalization and leading to reduced accuracy.

The experimental results indicate that, compared to other methods, the method presented in this paper performs best and requires fewer parameters. The main reasons can be divided into three aspects:

- The Transformer encoder part provides a global perspective on domain name character dependencies;
- The R-SKNet efficiently couples multi-scale feature branches, allowing the model to perform information interaction between branches and enhancing the ability to extract deep features;
- The EFAM effectively fuses the outputs of each stage of the backbone network, further enhancing the model's ability to extract domain name feature information.

4.5. Ablation Study

To evaluate the effectiveness of each module in the proposed TransFlexNet architecture, we have conducted an ablation study comparing various configurations of existing DGA domain detection models with and without the use of Transformer (TR), SKNet or R-SKNet, and FPN, or EFAM. The binary and multiclassification results for the Alexa and 360NetLab dataset are displayed in Tables 10 and 11, respectively.

Table 10. Binary classification performance for different model configurations: Alexa and 360NetLab.

Model Configuration	Accuracy	Recall	Precision	F1 Score
TR	0.9221	0.9476	0.8970	0.9216
SKNet	0.9526	0.9535	0.9487	0.9511
SKNet + FPN	0.9501	0.9580	0.9400	0.9489
SKNet + EFAM	0.9515	0.9607	0.9403	0.9504
R-SKNet	0.9498	0.9475	0.9486	0.9480
R-SKNet + FPN	0.9516	0.9610	0.9403	0.9505
R-SKNet + EFAM	0.9500	0.9519	0.9451	0.9485
TR + SKNet	0.9746	0.9773	0.9703	0.9738
TR + SKNet + FPN	0.9753	0.9765	0.9724	0.9745
TR + SKNet + EFAM	0.9761	0.9813	0.9697	0.9755
TR + R-SKNet	0.9761	0.9784	0.9724	0.9754
TR + R-SKNet + FPN	0.9760	0.9778	0.9727	0.9753
Proposed Method	0.9783	0.9834	0.9812	0.9777

Table 11. Multi-classification performance for different model configurations: Alexa and 360NetLab.

Model Configuration	Accuracy	Recall	Precision	F1 Score
TR	0.8838	0.8838	0.8806	0.8787
SKNet	0.9007	0.9007	0.8891	0.8943
SKNet + FPN	0.8989	0.8989	0.8922	0.8944
SKNet + EFAM	0.9031	0.9031	0.8990	0.9004
R-SKNet	0.9007	0.9007	0.8957	0.8973
R-SKNet + FPN	0.9011	0.9011	0.8959	0.8978
R-SKNet + EFAM	0.9066	0.9066	0.9016	0.9034
TR + SKNet	0.9300	0.9300	0.9214	0.9254
TR + SKNet + FPN	0.9365	0.9365	0.9354	0.9356
TR + SKNet + EFAM	0.9352	0.9352	0.9348	0.9345
TR + R-SKNet	0.9352	0.9352	0.9333	0.9339
TR + R-SKNet + FPN	0.9366	0.9366	0.9354	0.9357
Proposed Method	0.9391	0.9391	0.9377	0.9378

As indicated by Table 10, through detailed experimental comparisons, our method achieves the best results in all metrics for binary classification. The improved R-SKNet outperforms the original SKNet, EFAM surpasses FPN, and the use of these improved module combinations yields even better results.

As shown in Table 11, the method presented in this paper achieves the best results in accuracy, recall, precision, and F1 score for multiclassification tasks. Compared to the original SKNet and FPN structures, R-SKNet and EFAM exhibit superior performance under various module combinations.

5. Conclusions

To address the difficulties in extracting a small number of domain features in existing domain detection models, which had certain accuracy but a large amount of redundancy, this paper proposed a multi-scale feature fusion model based on Transformer and R-SKNet. In the input layer, the domain text was split into individual characters and indexed to achieve optimal word embedding and to reduce the difficulty of feature extraction. The feature extraction layer applied the TransFlexNet architecture, which used multi-scale feature fusion to flexibly handle long-distance dependencies and the dynamic changing local feature. The design resulted in complexity reduction. The output layer underwent multiple processing layers to optimize the number of parameters. Compared with existing models, the multi-feature model proposed in this paper achieved better feature extraction results, simplified structure, and improved accuracy. Experimental results showed that in both binary and multi-class classification experiments, the evaluation metrics of this model are mostly improved compared to existing methods, providing a better solution for dealing with botnets and malicious domains.

In future work, we will focus on optimizing the model architecture to enhance the efficiency of individual components and reduce overall complexity. Techniques such as quantization and model pruning will be employed to mitigate the computational overhead introduced by the Transformer architecture, enabling efficient operation on embedded devices and expanding the application scenarios for DGA detection. To further enhance model interpretability, we plan to leverage methods such as Local Surrogate models to provide clearer explanations and insights into the model's decision-making process.

Author Contributions: Conceptualization, J.T. and S.Z.; methodology, J.T., Y.G., S.Z. and Y.C.; validation, J.T. and Y.G.; formal analysis, J.T., Y.G. and H.W.; investigation, J.T., Y.G. and H.W.; data curation, J.T. and Y.G.; writing—original draft preparation, J.T. and Y.G.; writing—review and editing, J.T. Y.G., S.Z., H.W. and Y.C.; funding acquisition, S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the University Synergy Innovation Program of Anhui Province (grant number GXXT-2023-052), and the Chuzhou Science and Technology Plan Project (grant number 2023CI001).

Data Availability Statement: The data presented in this study includes the Alexa and 360NetLab dataset and the Alexa and DGArchive dataset, they are all publicly available on the Internet.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yan, F.; Liu, J.; Gu, L.; Chen, Z. A Semi-Supervised Learning Scheme to Detect Unknown DGA Domain Names Based on Graph Analysis. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December 2020; pp. 1578–1583.
2. Anand, P.M.; Kumar, T.G.; Charan, P.S. An ensemble approach for algorithmically generated domain name detection using statistical and lexical analysis. *Procedia Comput. Sci.* **2020**, *171*, 1129–1136. [CrossRef]
3. Satoh, A.; Fukuda, Y.; Kitagata, G.; Nakamura, Y. A word-level analytical approach for identifying malicious domain names caused by dictionary-based DGA malware. *Electronics* **2021**, *10*, 1039. [CrossRef]
4. Vranken, H.; Alizadeh, H. Detection of DGA-generated domain names with TF-IDF. *Electronics* **2022**, *11*, 414. [CrossRef]
5. Yang, C.; Lu, T.; Yan, S.; Zhang, J.; Yu, X. N-trans: Parallel detection algorithm for DGA domain names. *Future Internet* **2022**, *14*, 209. [CrossRef]
6. Namgung, J.; Son, S.; Moon, Y.-S. Efficient deep learning models for DGA domain detection. *Secur. Commun. Netw.* **2021**, *2021*, 8887881. [CrossRef]
7. Shahzad, H.; Sattar, A.R.; Skandaraniyam, J. DGA domain detection using deep learning. In Proceedings of the 2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP), Zhuhai, China, 8–10 January 2021; pp. 139–143.
8. Liang, J.; Chen, S.; Wei, Z.; Zhao, S.; Zhao, W. HAGDetector: Heterogeneous DGA domain name detection model. *Comput. Secur.* **2022**, *120*, 102803. [CrossRef]
9. Qi, G.; Mao, J. An Improved CNN-LSTM Algorithm for Detection of DGA Domain Name. In Proceedings of the 2023 7th International Conference on Electronic Information Technology and Computer Engineering, Xiamen China, 20–22 October 2023; pp. 1293–1298.
10. Jiang, Y.; Jia, M.; Zhang, B.; Deng, L. Malicious domain name detection model based on CNN-GRU-attention. In Proceedings of the 2021 33rd Chinese control and decision conference (CCDC), Kunming, China, 22–24 May 2021; pp. 1602–1607.
11. Ren, F.; Jiang, Z.; Wang, X.; Liu, J. A DGA domain names detection modeling method based on integrating an attention mechanism and deep neural network. *Cybersecurity* **2020**, *3*, 4. [CrossRef]
12. Pan, R.; Chen, J.; Ma, H.; Bai, X. Using extended character feature in Bi-LSTM for DGA domain name detection. In Proceedings of the 2022 IEEE/ACIS 22nd International Conference on Computer and Information Science (ICIS), Zhuhai, China, 26–28 June 2022; pp. 115–118.
13. Fang, X.; Sun, X.; Yang, J.; Liu, X. Domain-embeddings based DGA detection with incremental training method. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–6.
14. Li, X.; Wang, W.; Hu, X.; Yang, J. Selective kernel networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 510–519.
15. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
16. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
17. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the 2017 IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
18. Liang, Y.; Deng, J.; Cui, B. Bidirectional LSTM: An innovative approach for phishing URL Identification. In *Innovative Mobile and Internet Services in Ubiquitous Computing*; Springer: Cham, Switzerland, 2020; Volume 994, pp. 326–337.
19. Xu, Y.; Yan, X.; Wu, Y.; Hu, Y.; Liang, W.; Zhang, J. Hierarchical bidirectional RNN for safety-enhanced B5G heterogeneous networks. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 2946–2957. [CrossRef]
20. Plohmann, D.; Yakdan, K.; Klatt, M.; Bader, J.; Gerhards-Padilla, E. A comprehensive measurement study of domain generating malware. In Proceedings of the 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, USA, 10–12 August 2016; pp. 263–278.
21. Zhang, X.; Cheng, H.; Fang, Y. A DGA domain name detection method based on Transformer. *Comput. Eng. Sci.* **2020**, *42*, 411–417.
22. Zhang, B.; Liao, R.J. Malicious domain name detection model based on CNN and LSTM. *J. Electron. Inf. Technol.* **2021**, *43*, 2944–2951.
23. Liu, X.Y.; Liu, J.M.; Liu, C.; Zhang, Y.H. Novel Botnet DGA domain detection method based on character level sliding window and deep residual network. *Acta Electron. Sin.* **2022**, *50*, 250–256.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

Article

Enhancing Group Consensus in Social Networks: A Two-Stage Dual-Fine Tuning Consensus Model Based on Adaptive Leiden Algorithm and Minority Opinion Management with Non-Cooperative Behaviors

Tingyu Xu, Shiqi He, Xuechan Yuan and Chao Zhang *

School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China;
202201001163@email.sxu.edu.cn (T.X.); 202201006111@email.sxu.edu.cn (S.H.);
202201001169@email.sxu.edu.cn (X.Y.)
* Correspondence: czhang@sxu.edu.cn

Abstract: The rapid growth of the digital economy has significantly enhanced the convenience of information transmission while reducing its costs. As a result, the participation in social networks (SNs) has surged, intensifying the mutual influence among network participants. To support objective decision-making and gather public opinions within SNs, the research on the consensus-reaching process (CRP) has become increasingly important. However, CRP faces three key challenges: first, as the number of decision-makers (DMs) increases, the efficiency of reaching consensus declines; second, minority opinions and non-cooperative behaviors affect decision outcomes; and third, the relationships among DMs complicate opinion adjustments. To address these challenges, this paper introduces an enhanced CRP mechanism. Initially, the hippopotamus optimization algorithm (HOA) is applied to update the initial community division in Leiden clustering, which accelerates the clustering process, collectively referred to as HOAL. Subsequently, a two-stage opinion adjustment method is proposed, combining minority opinion handling (MOH), non-cooperative behavior management, and dual-fine tuning (DFT) management, collectively referred to as DFT-MOH. Moreover, trust relationships between DMs are directly integrated into both the clustering and opinion management processes, resulting in the HOAL-DFT-MOH framework. The proposed method proceeds by three main steps: (1) First, the HOAL clusters DMs. (2) Then, in the initial CRP stage, DFT manages subgroup opinions with a weighted average to synthesize subgroup perspectives; and in the second stage, MOH addresses minority opinions, a non-cooperative mechanism manages uncooperative behaviors, and DFT is used when negative behaviors are absent. (3) Third, the prospect-regret theory is applied to rank decision alternatives. Finally, the approach is applied to case analyses across three different scenarios, while comparative experiments with other clustering and CRP methods highlight its superior performance.

Keywords: social network; Leiden clustering; hippopotamus optimization algorithm; minority opinion handling; non-cooperative behavior

Citation: Xu, T.; He, S.; Yuan, X.; Zhang, C. Enhancing Group Consensus in Social Networks: A Two-Stage Dual-Fine Tuning Consensus Model Based on Adaptive Leiden Algorithm and Minority Opinion Management with Non-Cooperative Behaviors. *Electronics* **2024**, *13*, 4930. https://doi.org/10.3390/electronics13244930

Academic Editor: Domenico Ursino

Received: 25 November 2024
Revised: 10 December 2024
Accepted: 12 December 2024
Published: 13 December 2024

1. Introduction

With the rapid development of digitalization and the emergence of the digital economy, social media platforms such as Facebook, Twitter, and LinkedIn have gradually entered the public eye, becoming an indispensable part of daily life [1,2]. Moreover, short video applications such as TikTok and Snapchat have also enriched social functions, making information exchange among users more convenient [3]. In sociology, groups associated with decision-makers (DMs) influence their opinion adjustments, and social networks (SNs) play an important role in the consensus-reaching process (CRP) [4–6]. Since social network analysis (SNA) can reveal the relationships and influence among DMs, integrating SNA with CRP allows for the consideration of SNs' impacts on group opinion adjustments [7,8].

As a crucial step in group decision-making, CRP plays a significant role in decision-related issues [9]. While combining SNA with CRP allows for a thorough consideration of mutual influence among DMs in the decision process; the ease of online communication and reduced communication costs has introduced inevitable challenges to group decision-making. First, the number of DMs has surged, and when it exceeds 20, it is generally considered as a large-scale group decision-making (LSGDM) problem [10–12]. This increase in scale also raises the complexity of performing CRP within LSGDM [13]. Second, incorporating SNA allows for the formulation of rational opinion adjustment strategies [14]. Third, the complexity of user opinions in the DM communication process makes it crucial to obtain a comprehensive evaluation that balances individual opinions as much as possible.

This paper optimizes the Leiden clustering using the hippopotamus optimization algorithm (HOA) with collaborative indicators based on weight and opinion similarities for the individual classification [15]. Leveraging SNA, the proposed HOAL-DFT-MOH group consensus adjustment mechanism effectively analyzes social relationships and interaction characteristics among individuals within an SN group. By integrating the group consensus adjustment mechanism, the model considers mutual influences among DMs and applies tailored opinion adjustment strategies, closely simulating real-life decision scenarios. Furthermore, as an essential component of LSGDM, the group consensus adjustment enhances the rationality and realism of decision outcomes, advancing the development of LSGDM.

In what follows, Section 1.1 introduces existing methods for partial clustering and highlights the advantages of the clustering approach used in this paper. Section 1.2 presents LSGDM, with a focus on the CRP that considers minority opinions and non-cooperative behaviors. Section 1.3 provides a summary of challenges and contributions of the method proposed in this paper.

1.1. Clustering in LSGDM Based on SNs

Clustering is an unsupervised learning algorithm that classifies objects in a dataset based on certain features within samples [16]. In LSGDM, the efficiency of traditional group consensus adjustments declines significantly, falling short of requirements [17,18]. Due to the simplicity and flexibility of clustering, it has been widely applied in LSGDM, where increasing the number of adjustment iterations reduces the scope of consensus adjustments, thereby enhancing decision efficiency [19,20].

There has been extensive research on integrating clustering with group consensus adjustments. In this paper, clustering methods are roughly classified into three categories: first, further improvements and optimizations based on classic clustering [21,22]; second, the combination of clustering with SNA to achieve more intuitive grouping results [23–25]; and third, the integration of biological or medical knowledge to simulate certain biological behaviors or mechanisms for clustering analysis [26].

Recently, Harshavardhan et al. [27] utilized the biogeography-based optimization to improve fuzzy C-means clustering, while Guo et al. [28] proposed a new three-way clustering based on K-means clustering. Although these approaches enhanced the clustering efficiency by refining the foundational clustering, they failed to consider applying clustering analysis within the context of SNs [25]. Both Louvain and Leiden clustering can perform community divisions on complex SNs [10,29]; however, Leiden clustering demonstrates superior efficiencies and performances [30]. Simulating certain biological behaviors offers stronger spatial search capabilities and better clustering outcomes than classic clustering, such as genetic algorithms [31], ant colony optimization [32], and wolf pack algorithms [33]. Many researchers have also combined the above algorithms with traditional clustering in hopes of obtaining better solutions. For instance, Zeebaree et al. [34] combined the genetic algorithm with the K-means clustering method to address the sensitivity of K-means clustering to initial conditions through the genetic algorithm. Compared to these works, the HOA exhibits stronger spatial search capabilities [15].

In this paper, the HOA is combined with Leiden clustering to enhance spatial search abilities, reduce computational complexity, and accelerate optimal solution discovery.

The HOA's robust spatial search ability provides better initial community divisions for Leiden clustering, while Leiden clustering addresses the need for extensive exploration points when using HOA for clustering. This combination leverages the strengths of both approaches, effectively reducing time in actual adjustment processes and improving algorithm efficiencies.

1.2. CRP in LSGDM for Different Decision Scenarios

The rise of social media and the increasing complexity of SNs have led to the phenomenon where DMs' opinions align not only with prevailing popular views during the CRP but are also influenced by interpersonal relationships and potential user behaviors [24]. As a result, several researchers have emphasized potential benefits of integrating SNA with group consensus adjustment processes [25].

LSGDM is a process in which a group of individuals collectively analyze, discuss, and make decisions on various alternatives. In recent years, with the growth of social media, LSGDM has gained significant momentum and is now widely applied across diverse fields. It has two key characteristics: first, the number of participants in decision-making processes exceeds 20 [35,36]; and second, DMs may exhibit undesirable behaviors during the process [37,38]. The CRP, as both a strategy and process aimed at fostering agreement among members, plays a critical role in LSGDM. Through various technical methods, the opinions and preferences of group members are aligned to a satisfactory level of consensus, and much research has been devoted to addressing this challenge.

Research has shown that SNA can enhance the provision of references, advice, and other forms of support based on trust [39]. Therefore, this study integrates SNA with CRP, using the trust network among DMs as the foundation for influencing opinion adjustments.

One of the challenges in combining SNA with CRP is how to effectively categorize and assess the diverse composition of DMs, identify their professional backgrounds and personality traits, and use this information to leverage trust networks among them. This approach aims to balance the opinions of each DM while minimizing the impact of disruptive opinions on the final decision. To address this challenge, three distinct opinion management mechanisms are proposed: minority opinion handling (MOH), non-cooperative behavior management, and dual-fine tuning (DFT).

1.2.1. MOH

The ancient Greek philosopher Plato once stated, "Truth may lie in the hands of the few", while Lenin similarly remarked, "Truth often resides with the minority". In the process of opinion adjustments in daily life, certain individuals may be fewer in number but own greater influence [40]. Their opinions tend to be more reliable and can significantly shape the direction of opinion adjustments for others. In such instances, when DMs are reluctant to revise their views, this phenomenon has already been explored in relevant research [41].

In common CRP, adjustments are typically made with the primary consideration of the majority's opinions [42]. While this approach achieves apparent fairness, it often fails to fully utilize the valuable input of a minority of experts, which results in the unfair treatment of these individuals. To address this issue, Xu et al. [43] proposed the MOH method, and Shen et al. [25] focused on analyzing trust networks in SNs to incorporate minority opinions into the decision process.

However, although the methods proposed by these researchers consider the role of trust networks in decision-making and use mutual trust among individuals as a criterion for opinion adjustments, they overlook the impact of individual opinion adjustment weights on changes in opinions. In detecting minority opinions using standards such as participant weighting, this paper draws from psychological theories on expert influences and the effects of SNA on opinion adjustment tendencies [44]. By integrating both trust levels with weight influences as synergistic adjustment indicators, the proposed method determines the pace of adjustment in a way that better aligns with the psychology of DMs, leading to more realistic and fair decision outcomes.

1.2.2. Handling of Non-Cooperative Behaviors Caused by Overconfidence

Non-cooperative behaviors refer to situations in a group, where one or more members fail to effectively communicate, coordinate, or collaborate with others, stubbornly adhering to their own opinions. In such cases, their refusal to adjust their views can hinder the achievement of consensus and may even cause deviations in decision outcomes, ultimately rendering decisions less reliable.

This phenomenon has been recognized by many researchers, who have proposed various adjustment strategies. Liu et al. [45] dynamically adjusted networks based on trust relationships and associated risks. Fu et al. [46] introduced a moderate adjustment intervention, leveraging trust networks to facilitate quicker consensus building. Liu et al. [47] proposed a self-managed weight correction approach to address non-cooperative behaviors.

To minimize the impact of opinions from individuals who refuse to cooperate and to fully leverage the influence of trust networks on DMs, this study employs a dual adjustment strategy for both weights and opinions. The trust network provides opinion outcomes for individuals involved in adjustment processes, while human intervention is applied to reduce the weights of non-cooperative individuals, thereby minimizing their influence on final decisions.

1.2.3. Handling of Conventional Cases Using DFT

A key step in the CRP is adjusting the opinions of DMs to reach a collective agreement. The strategy used to select DMs for opinion adjustments not only affects the efficiency of the CRP but also influences the final decision outcomes to some extent. DFT is a selection strategy designed for adjusting the opinions of DMs. Specifically, during the opinion adjustment process, it selects the two DMs whose opinions deviate most from those of the rest of the group for further adjustment. Applying DFT in the CRP step of LSGDM can enhance the efficiency of reaching a final decision without causing over-adjustment. This is achieved by appropriately increasing the number of DMs adjusted in each round of the process.

Previous studies have proposed various methods for adjusting DMs' choices, with the most common approach being to adjust the opinions of all DMs. Zhou et al. [35] and Liu et al. [45] adjusted the opinions of all DMs in each iteration. While this method is efficient and can achieve group consensus in fewer rounds, it is prone to over-adjustment, which may reduce the level of consensus. In contrast, Shen et al. [25] changed only the DMs with the lowest consensus in each round. While this reduces the likelihood of over-adjustment, it significantly lowers adjustment efficiency. To combine the advantages of both strategies, Yuan et al. [19] proposed a compromise approach using DFT, in which the two DMs with the lowest consensus are selected for adjustments, thereby improving efficiency while mitigating the risk of over-adjustment.

In this paper, we further optimize the DFT method proposed by Yuan et al. [19], incorporating an additional consideration of the trust relationships between DMs to determine the pace of opinion adjustments.

1.3. Summary of Our Method

The main objective of this paper is to introduce a decision-making method, HOAL-MOH-DFT. By integrating SNA into LSGDM, it provides more reasonable community divisions and more objective decision outcomes.

1.3.1. Challenges Addressed by Our Method

The implementation of this method faces several key challenges:

(1) How to account for both the trust relationships and opinion similarity among DMs in clustering analysis to form more rational groupings.

(2) How to detect the presence of undesirable behaviors among DMs and identify the specific types of such behaviors.

(3) How to devise appropriate adjustment strategies for minority opinions and non-cooperative behaviors driven by overconfidence during consensus adjustments.
(4) How to integrate two potential types of misconducts with trust relationships among DMs, regulate the adjustment pace, and manage opinions to derive final decisions.

1.3.2. Motivation Behind Our Method

The main motivation for this method is as follows:

(1) The context of LSGDM has become more complex with the growth of social media and an increase in participants, making the effective partitioning of DMs into subgroups crucial for improving decision efficiency.
(2) Social media has fostered closer connections among DMs, with mutual trust playing a critical role in the opinion adjustment process, making trust relationships essential for more realistic adjustments.
(3) In real-world scenarios, truth is often held by a small group of experts, so it is vital to consider all perspectives, not just follow the majority opinion, when making decisions.
(4) DMs may propose opinions that significantly deviate from the group, lacking authority or accuracy; effectively minimizing the impact of such disruptive opinions is crucial for ensuring reliable outcomes.

1.3.3. Contributions of Our Method

In addressing the challenges mentioned above, this paper makes the following contributions:

(1) A comprehensive collaborative index, combining weights with opinion similarities, serves as a foundation for community divisions.
(2) A mechanism has been developed to detect misconduct and provide corresponding adjustment strategies by utilizing the distance between DMs and the group, as well as their proportions and weights.
(3) MOH strategies, non-cooperative management strategies, and DFT strategies are employed to address minority opinions, manage non-cooperative behaviors driven by overconfidence, and handle situations devoid of negative behaviors.
(4) The adjustment coefficient is calculated based on the trust relationship and weight similarity between DMs, while the adjustment step size is determined by the level of group opinions.

The remainder of the paper is arranged below, Section 2 of this paper outlines the fundamental concepts of LSGDM, SNA, and the prospect-regret theory. Section 3 details the specific implementation steps of the HOAL. Section 4 introduces the consensus measurement criteria and provides an in-depth description of the two-stage CRP adjustment process. In Section 5, the HOAL-MOH-DFT method is applied to three real-world scenarios. Finally, Section 6 presents comparative experiments, simulation studies, and sensitivity analysis to assess the methods proposed in this paper. Section 7 discusses the applicability of the proposed method. Section 8 provides a summary of the paper.

2. Preliminaries

This section reviews the basic implementation steps of LSGDM, the three basic representations of SNs, and the implementation process of the prospect-regret theory in existing research.

2.1. Basic Steps for Implementing LSGDM

In LSGDM problems, multiple DMs evaluate alternative solutions across various attributes to reach a collective decision [48,49]. LSGDM is particularly effective for managing the inherent complexity and uncertainty in LSGDM scenarios [50,51]. The following outlines the specific steps for implementing LSGDM.

Step 1: Data standardization. To ensure the comparability of evaluations from different experts, especially when they may use varying units or scales, it is essential to standardize the decision matrix. Each expert e_k provides an evaluation matrix $Q^k = (q_{ij}^k)_{n*m}$, where q_{ij}^k

represents the evaluation of an alternative x_i with respect to an attribute a_j. The standardization process converts these evaluations into a standard decision matrix $P^k = (p_{ij}^k)_{n*m}$. The process of data normalization is implemented as follows:

(1) Determination of the maximum and minimum values for each attribute across all experts and alternatives.

(2) Standardization of each evaluation is applied using the equation: $p_{ij}^k = \frac{q_{ij}^k - \min(q_{ij}^k)}{\max(q_{ij}^k) - \min(q_{ij}^k)}$, where q_{ij}^k denotes the evaluations provided by an expert e_k for all alternatives x_i concerning an attribute a_j.

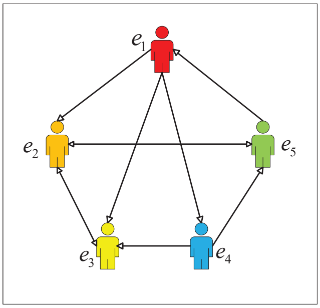
Step 2: Expert classification. To streamline computations and simplify the LSGDM problem, DMs are often categorized into multiple subgroups based on their similar opinions. If the larger group is divided into b subgroups, labeled as $C = \{C_1, C_2, \ldots, C_b\}$. In this study, the decision matrix for a subgroup C_h is denoted as $V^h = (v_{ij}^h)_{n*m}$, which is derived using the weighted average method.

Step 3: Aggregation process. The next step is to aggregate the standardized evaluations across all DMs for each alternative. This can be achieved using several aggregation techniques, such as arithmetic mean, weighted mean, or other consensus methods. The updated group evaluation matrix $G = (g_{ij})_{n*m}$ is obtained through further calculations. $g_{ij} = WAA(v_{ij}^1, v_{ij}^2, \ldots, v_{ij}^b) = \sum\limits_{h=1}^{b} \lambda_h \times v_{ij}^h$, where λ_h is the weight of C_h, $\sqrt{0 \leq \lambda_h \leq 1}$, and $\sqrt{\sum\limits_{i=1}^{m} \lambda_h = 1}$.

2.2. SNA

In this subsection, three different representations of trust networks are introduced as shown in Table 1 below.

Table 1. Different representation schemes for SNs.

Graph Theory	Algebraic Method	Sociometric Matrix
	e_1Re_2 e_1Re_3 e_1Re_4 e_2Re_3 e_2Re_5 e_3Re_2 e_4Re_3 e_4Re_5 e_5Re_1 e_5Re_2	$\begin{bmatrix} - & 1 & 1 & 1 & 0 \\ 0 & - & 1 & 0 & 1 \\ 0 & 1 & - & 0 & 0 \\ 0 & 0 & 1 & - & 1 \\ 1 & 1 & 0 & 0 & - \end{bmatrix}$

Graph theory: To represent the trust network among members, a graph is employed, composed of interconnected nodes with directed edges. Specifically, an edge $e1 \rightarrow e2$ in the graph indicates the existence of a straightforward trust relation from the node $e1$ to $e2$.

Algebraic method: Different types of relationships can be distinguished by this approach and combinations of these relationships are displayed.

Sociometric matrix: The relation between members is denoted as a matrix $ST = st_{d \times d}$ in sociometric, which indicates whether there is a trust relationship among the members.

Remark 1. *The degree refers to the number of edges connected to a node, reflecting its level of activity or engagement within the network. Nodes with higher degrees are often considered more influential.*

Centrality measures a node's importance within a network, with common metrics including degree centrality, closeness centrality, betweenness centrality, and clustering coefficient.

Remark 2. *The degree centrality of a node is measured by the number of direct connections it has.*

Remark 3. *Closeness centrality evaluates the speed at which a node can access other nodes in the network, based on the shortest path.*

Remark 4. *Betweenness centrality assesses how often a node serves as a bridge along the shortest path between two other nodes, reflecting its potential to control the flow of information.*

Remark 5. *The clustering coefficient measures the extent to which nodes in a network tend to cluster together, with a high value indicating that nodes are closely connected, forming tight-knit communities.*

2.3. Prospect-Regret Theory

The prospect-regret theory is commonly employed to address decision-making biases driven by psychological factors, such as herd behaviors and other cognitive influences [52,53].

The most important parts of prospect theory are the value function and attribute weights [54]. The following are the steps for applying prospect theory:

Step 1: Calculate the value function for each DM.

$$v(\Delta x_{ij}) = \begin{cases} (\Delta x_{ij})^{\alpha} & \Delta x_{ij} \geq 0 \\ -\lambda(-\Delta x_{ij})^{\beta} & \Delta x_{ij} < 0 \end{cases} \tag{1}$$

where λ represents the loss aversion coefficient, with a higher value signifying that DMs exhibit greater sensitivity to losses.

Step 2: Calculate the prospect value matrix V_{ij} and joy value matrix R_{ij}.

$$V_{ij} = v(\Delta x_{ij})w \tag{2}$$

where w represents attribute weights, and both the maximum prospect value V_i^+ and the minimum prospect value V_i^- can be derived from the following equation:

$$Z_i(x) = \sum_j^m (R_{ij}(x) + G_{ij}(x)) \tag{3}$$

Step 3: Calculate the regret value matrix G_{ij}.

$$G_{ij}(x) = 1 - \exp\left[-\delta\left|\frac{V_{ij}(x) - V_{ij}^-(x)}{V_{ij}^+(x) - V_{ij}^-(x)}\right|\right] \tag{4}$$

Regret theory is mainly used to avoid biases that DMs may develop to reduce their level of regret [55,56]. Therefore, regret theory is often used in conjunction with prospect theory, and the main equation is shown as follows:

$$G_{ij}(x) = 1 - \exp\left[-\delta\left|\frac{V_{ij}(x) - V_{ij}^-(x)}{V_{ij}^+(x) - V_{ij}^-(x)}\right|\right] \tag{5}$$

3. Community Detection and Weight Measure

This section focuses on enhancing the Leiden clustering by the use of HOA [15]. Additionally, it presents a method for determining the weights of DMs and communities using trust networks.

3.1. Community Division Based on HOAL Strategies

As the digital economy progresses, the number of participants in decision processes continues to rise. When this number exceeds a certain threshold, the efficiency of reaching a consensus typically decreases. Clustering can help address this by grouping DMs, thereby reducing the scale and complexity of decision process and enabling quicker conclusions.

However, many existing clustering methods either rely on a single metric or fail to incorporate trust networks into the clustering process. This paper proposes an adaptive HOA-Leiden clustering method, called HOAL, which utilizes both the viewpoint similarity and weight proximity as dual criteria for clustering.

3.1.1. Initializing the Community Using the HOA

The HOA is a swarm intelligence optimization technique inspired by the positional adjustments of hippos within a population in response to both internal and external influences. It primarily models three behaviors: position updates, defensive actions against predators, and escape maneuvers. By continuously refining positions, the HOA seeks to optimize the placement of the most advantageous hippos, ultimately leading to better solutions.

With its rapid convergence and high precision, the HOA has proven to be effective across a range of problem types, improving the solution speed. In this study, the HOA is employed to generate superior initial solutions, which are then integrated with Leiden clustering to enhance community detection efficiencies.

To better mimic the survival behaviors of hippo populations, the initial state excludes specific behavioral patterns. Equation (6) is used to initialize the individual positions within the hippo population. Here, N represents the size of the hippopotamus population, while m is determined based on the scale of the problem being addressed. lb refers to the lower bound of the position range for the hippopotamuses, and ub denotes the upper bound of the position range.

$$X_i : x_{ij} = lb_j + r \cdot \left(ub_j - lb_j\right), i = 1, 2, \ldots, N, j = 1, 2, \ldots, m \tag{6}$$

The position matrix of the hippopotamus population, initialized according to Equation (6), is given by Equation (7). Here, X represents the positions of all the hippos in the population, while X_i denotes the current position of the individual hippopotamus i:

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_3 \\ \vdots \\ X_5 \end{bmatrix} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,m} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,m} \end{bmatrix} \tag{7}$$

Figure 1 illustrates the three distinct stages involved in the adjustment of the hippopotamus's position.

In what follows, we provide detailed explanations and implementation steps for the three optimization processes shown in Figure 1.

Stage 1: The renewal of hippopotamus positions in rivers or ponds.

The distribution of hippopotamus populations is influenced by the positions of other members within the same group. Typically, hippopotamuses tend to cluster together, with the other individuals often found near the dominant hippos. Female hippos and their offspring are usually distributed around the dominant male hippos. Adult male hippos may be displaced by more dominant individuals, who either attract females to rebuild their population or secure a position within the group to replace the dominant males.

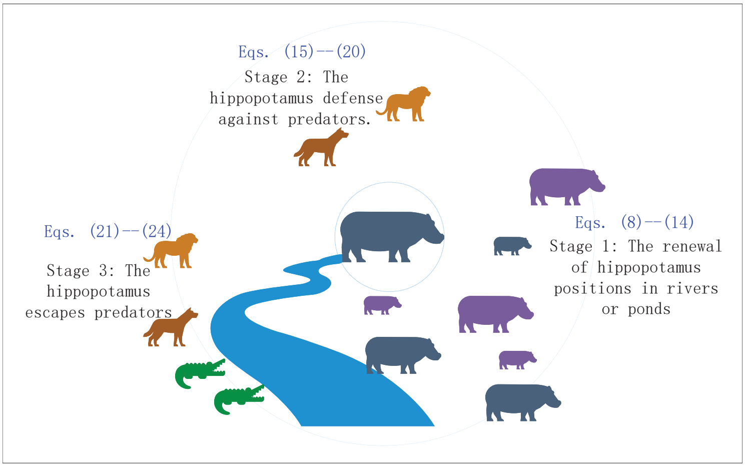

Eqs. (15)--(20)
Stage 2: The hippopotamus defense against predators.
Eqs. (21)--(24)
Stage 3: The hippopotamus escapes predators
Eqs. (8)--(14)
Stage 1: The renewal of hippopotamus positions in rivers or ponds

Figure 1. Hippopotamus position updates across various phases.

Equation (13) provides the equation for updating the position of a male hippopotamus. Here, X represents the updated position of the i-th male hippopotamus. The position update formula for male hippos is shown in Equation (8).

To update the position of female hippopotamuses, use Equations (11) and (12). Then, apply Equation (14) to determine both the original and the updated positions of the female hippos.

In these equations: X_i^{MH} represents the position of the i-th male hippopotamus, Dh denotes the position of the dominant hippopotamus, and y_i is a random number between 0 and 1.

$$X_i^{MH} : x_{ij}^{MH} = x_{ij} + y_{i\cdot}\left(Dh - I_i x_{ij}\right), i = 1, 2, \ldots, \left\lceil \frac{N}{2} \right\rceil, j = 1, 2, \ldots, m \tag{8}$$

I_1 and I_2 represent random numbers between 1 and 2, while $\overrightarrow{r_1}$, $\overrightarrow{r_2}$, $\overrightarrow{r_3}$, and $\overrightarrow{r_4}$ are vectors representing taboo values between 0 and 1. r_5 is a random number between 0 and 1. q_1 and q_2 are binary values, either 0 or 1.

$$h = \begin{cases} I_2 \times \overrightarrow{r_1} + (\sim q_1) \\ 2 \times \overrightarrow{r_2} - 1 \\ \overrightarrow{r_3} \\ I_1 \times \overrightarrow{r_4} + (\sim q_2) \\ r_5 \end{cases} \tag{9}$$

t represents the current iteration number, while τ denotes the maximum number of iterations.

$$T = \exp\left(-\frac{t}{\tau}\right) \tag{10}$$

MG_i represents the average position of a randomly selected subset of the hippopotamus population.

$$X_i^{FB} : x_{i,j}^{FB} = \begin{cases} x_{i,j} + h_1\left(Dh - I_2 MG_i\right) & T > 0.6 \\ \Xi & else \end{cases} \tag{11}$$

r_6 and r_7 represent random numbers between 0 and 1.

$$\Xi = \begin{cases} x_{i,j} + h_2(MG_i - Dh) & r_6 > 0.5 \\ lb_j + r_7\left(ub_j - lb_j\right) & else \end{cases} \tag{12}$$

$$X_i = \begin{cases} X_i^M & F_i^M < F_i \\ X_i & else \end{cases} \tag{13}$$

$$X_i = \begin{cases} X_i^{FB} & F_i^{FB} < F_i \\ X_i & else \end{cases} \tag{14}$$

Stage 2: The hippopotamus defense against predators.

The safety of the hippopotamus populations is crucial for its reproduction and survival. The social behavior of hippos plays an important role in deterring predators. However, young, sick, or elderly hippos may sometimes stray from the group, making them vulnerable to attacks. These weaker individuals are more likely to become targets for predators such as Nile crocodiles, lions, and spotted hyenas.

In defense, hippos often confront predators directly, issuing loud calls to warn them to stay away. At the same time, hippos may engage in aggressive behavior, approaching the predator to force it into retreat.

The current position of the predator is randomly generated as shown in Equation (15). Given that hippos employ different defense strategies based on the distance between themselves and predators, a distance calculation method is provided in Equation (16):

$$\Pr edator : \Pr edator_j = lb_j + \overrightarrow{r_8}\left(ub_j - lb_j\right), j = 1, 2, \ldots, m \tag{15}$$

$$\overrightarrow{D} = \left|\Pr edator_j - x_{i,j}\right| \tag{16}$$

Hippos employ a defensive behavior represented by the factor $F_{\Pr edator}$ to protect themselves from predators. If $F_{\Pr edator}$ is smaller than F_i, it indicates that the distance between the hippopotamus and the predator is very close, prompting the hippopotamus to quickly face the predator and force it to retreat. Conversely, if $F_{\Pr edator}$ exceeds F_i, it suggests that the distance between the hippo and the predator is relatively large. In this case, while the hippopotamus will still move toward the predator, the movement is less aggressive, primarily serving as a warning that the predator has entered its territory.

$$X_i^H : x_{i,j}^H = \begin{cases} \overrightarrow{RL} \oplus \Pr edator_j + \left(\frac{f}{c - d \times \cos(2\pi g)}\right)\left(\frac{1}{\overrightarrow{D}}\right) & F_{\Pr edator} < F_i \\ \overrightarrow{RL} \oplus \Pr edator_j + \left(\frac{f}{c - d \times \cos(2\pi g)}\right)\left(\frac{1}{2 \times \overrightarrow{D} + \overrightarrow{r_9}}\right) & F_{\Pr edator} \geq F_i \end{cases} \tag{17}$$

Below, v and w are random numbers between 0 and 1, with v being a constant equal to 1.5, and Γ denotes the gamma function.

$$Levy(v) = 0.05 \times \frac{w \times \sigma_w}{|v|^{\frac{1}{v}}} \tag{18}$$

$$\sigma_w = \left[\frac{\Gamma(1+v)\sin\left(\frac{\pi v}{2}\right)}{\Gamma\left(\frac{(1+v)}{2}\right)v2^{\frac{(v-1)}{2}}}\right]^{\frac{1}{v}} \tag{19}$$

$$X_i = \begin{cases} X_i^H & F_i^H < F_i \\ X_i & F_i^H < F \end{cases} \quad \overrightarrow{D} = \left|\Pr edator_j - x_{i,j}\right| \tag{20}$$

Stage 3: Hippopotamus escapes predators.

While the hippopotamus's active defense is crucial for its safety, it is not always effective, especially when facing a group of predators or other challenging situations.

In such cases, the hippopotamus will voluntarily flee. Typically, hippos attempt to escape to the nearest lake or pond to avoid being chased by terrestrial predators.

$$lb_j^{local} = \frac{lb_j}{t}, ub_j^{local} = \frac{ub_j}{t}, t = 1, 2, \ldots, T \tag{21}$$

$$X_i^H : x_{ij}^H = x_{ij} + r_{10}\left(lb_j^{local} + o_1\left(ub_j^{local} - lb_j^{local}\right)\right) \tag{22}$$

$$o = \begin{cases} 2 \times \overrightarrow{r_{11}} - 1 \\ r_{12} \\ r_{13} \end{cases} \tag{23}$$

$$X_i = \begin{cases} X_i^H & F_i^H < F_i \\ X_i & F_i^H \geq F_i \end{cases} \tag{24}$$

3.1.2. HOAL

This subsection introduces the method of combining the HOA with Leiden clustering. The process of classical Leiden clustering is shown in Figure 2.

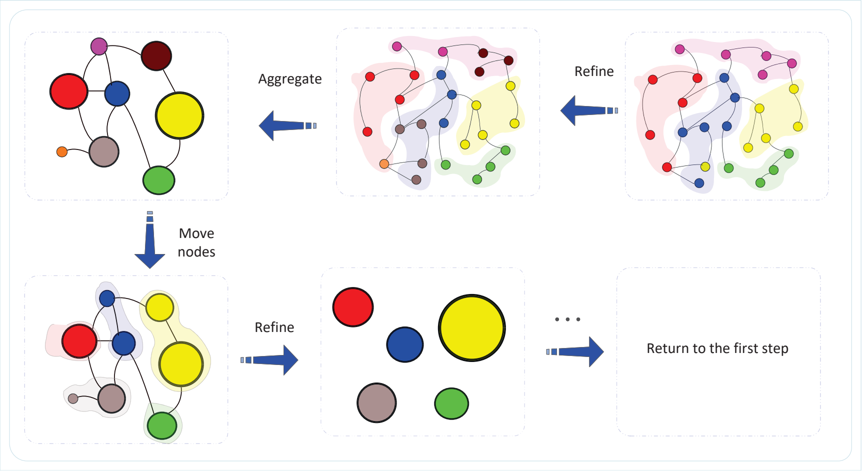


Figure 2. Implementation steps of Leiden clustering.

According to Figure 2 in the appeal, the basic steps of traditional Leiden clustering are shown as follows:

Step 1: Assign each node to its own initial community, treating it as an independent community.

Step 2: Move a single node from one community to another, then evaluate the partition to determine if the change improves it. Retain the move if it enhances the partition; otherwise, reverse it.

Step 3: Refine the newly found partition by dividing nodes into sub-partitions, further optimizing the partition to improve its quality.

Step 4: Generate an aggregation network for the refined partitions, where each node represents a community, and community similarity serves as the connection between nodes.

Step 5: Repeat Steps 3 and 4 by moving nodes within the aggregated network and refining the partitions until no further improvements in partitioning are achievable.

To accelerate and optimize the clustering results based on classical Leiden clustering, the HOA algorithm is used to detect a relatively optimal initial community distribution. This initial distribution is then used for Leiden clustering, reducing the number of required distribution adjustments. Additionally, the HOA is employed for parameter selection in Leiden clustering to obtain better community partitioning results.

The flowchart of the specific implementation steps of the clustering process is shown in Figure 3.

The pseudocodes for the specific implementation of the algorithm can be found in Algorithm 1.

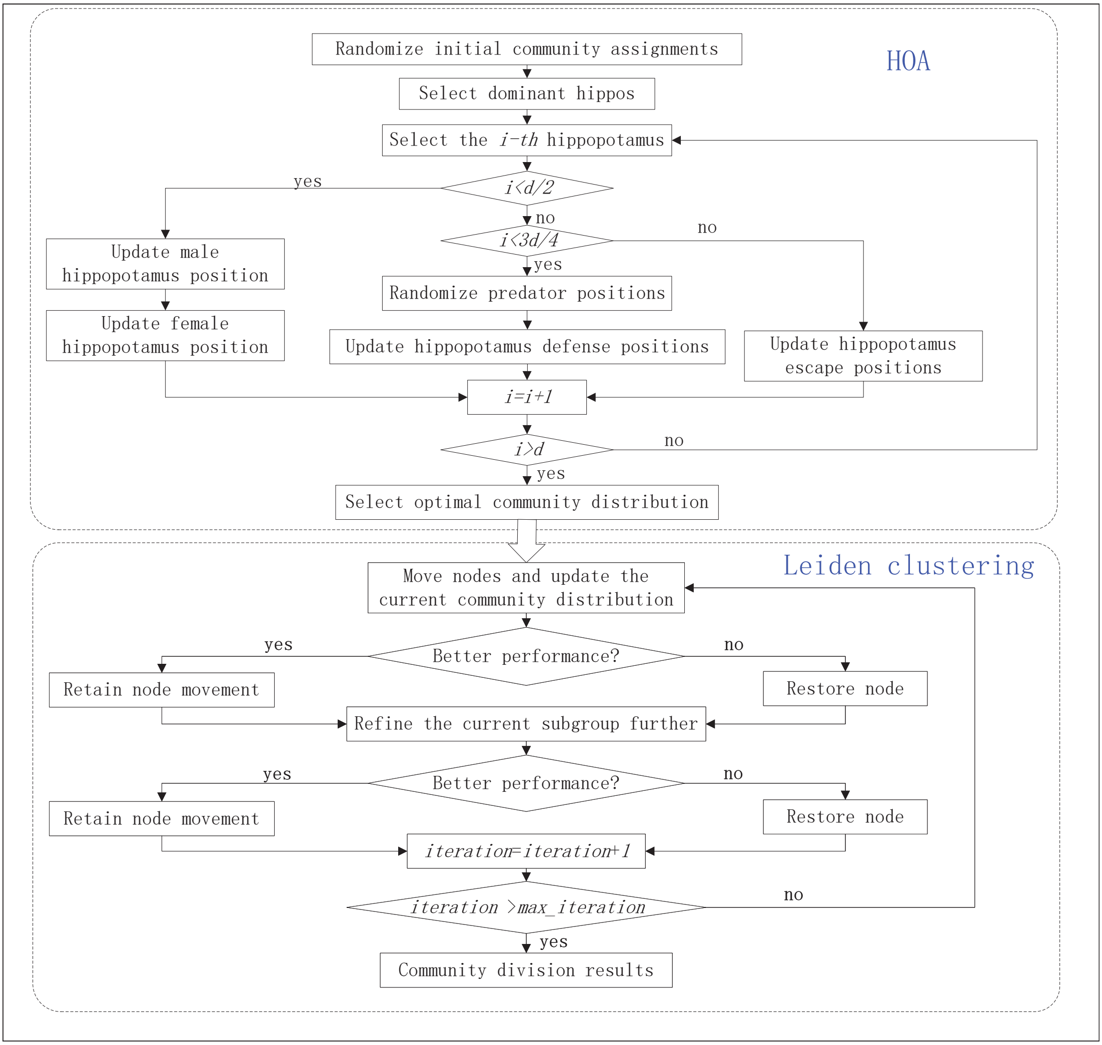

HOA
Randomize initial community assignments
Select dominant hippos
Select the i-th hippopotamus
i<d/2
yes
no
Update male hippopotamus position
Update female hippopotamus position
i<3d/4
yes
no
Randomize predator positions
Update hippopotamus defense positions
Update hippopotamus escape positions
i=i+1
i>d
no
yes
Select optimal community distribution
Leiden clustering
Move nodes and update the current community distribution
Better performance?
yes
no
Retain node movement
Restore node
Refine the current subgroup further
Better performance?
yes
no
Retain node movement
Restore node
iteration=iteration+1
iteration >max_iteration
no
yes
Community division results

Figure 3. Community division implementation flowchart.

The time complexity of the HOAL algorithm consists of several parts. The initialization phase takes $O(num\_hippos \times d)$, where num_hippos is the number of hippos and d is the dimensionality of the vectors. The core update process requires $O(num\_hippos \times max\_iterations \times d)$. Calculating the trust matrix and community partitioning contributes $O(num\_hippos^2)$, while refining community divisions results in $O(max\_iteration \times num\_hippos^2)$. Thus, the overall time complexity is dominated by the community division phase, with a final complexity of $O(max\_iteration \times num\_hippos^2)$.

Algorithm 1: The implementation of Pseudocodes using the HOAL algorithm

```
Input: The trust matrix among DMs, the opinion matrix of DMs.
Output: Community allocation results.
Function Main:
    Data: Initialize parameters num_hippos, max_iterations, community_size_range,
          etc.
    Data: Initialize population of hippos (Each individual hippopotamus is
          composed of a vector of length d).
    for $i = 1$ to $\frac{\text{num\_hippos}}{2}$ do
        for $iteration = 1$ to max_iterations do
            Update the position of male hippopotamus using Equation (8);
            Evaluate the updated individual hippopotamus using Equation (29);
    for $i = \frac{\text{num\_hippos}}{2}$ to d do
        for $iteration = 1$ to max_iterations do
            Update the position of female hippopotamus using Equation (11);
            Evaluate the updated individual hippopotamus using Equation (29);
            if new_position better than current_position then
                Update hippo's position;
    for $i = 1$ to d do
        Randomly generate a predator's position vector of length d;
        for $iteration = 1$ to max_iterations do
            Update the position of male hippopotamus using Equation (20);
            Evaluate the updated individual hippopotamus using Equation (29);
            if new_position better than current_position then
                Update hippo's position;
```

3.2. Weight Measure

The roles of DMs and community weights are crucial in the LSGDM process, affecting not only the opinion adjustment process but also the final outcome. Assigning appropriate and reasonable weights facilitates opinion adjustments during the CRP and the generating of final decision results.

In this paper, a trust network-based weight feedback confirmation mechanism is established. This mechanism comprehensively considers the current DMs' professionalism and credibility within the group. We assume that the DM's weight matrix is $W = [w_1, w_2, \ldots, w_i, \ldots, w_n]$. Assume that the matrix formed by the column of the trust matrix where the ith DM is $T_i = [t_{i1}, t_{i2}, \ldots, t_{in}]$. The matrix of trust values for the DM is $BT = [bt_1, bt_2, \ldots, bt_n]$.

Step 1: Initialize the weights of all DMs to an equal status.

Step 2: Extract the trust degree column for each DM from the decision matrix, and calculate the trust value of each DM by applying a weighted average based on their respective weights.

$$bt_i = \sum_{j=0}^{n} \left(w_j \times t_{ij} \right) \tag{25}$$

Step 3: Normalize the trust values obtained from the previous calculations to generate the updated weights.

$$w_i = \frac{bt_i}{\sum\limits_{j=0}^{n} bt_j} \tag{26}$$

Step 4: Repeat the operations in Step 2 until the difference between the weights before and after the update is less than 0.05.

4. Considering the CRP for Misbehavior Management

Since the clustering process employs a collaborative index of opinion similarities and trust weights, this paper considers that all subgroup consensus adjustments fall under ordinary scenarios. When adjusting consensus between groups, it is necessary to classify the composition states of DMs using indicators and select the appropriate adjustment method accordingly.

4.1. Consensus Measure

When decisions are made by consultation, there is always an expectation to reach a consensus. However, achieving complete agreement in practice can often be challenging. To enhance the acceptance of decision outcomes, it is necessary to establish a level of consensus that satisfies the group. This requires the design of an appropriate consensus measurement method to assess whether consensus has been reached. To this end, this subsection introduces a consensus measurement equation that considers both the degree of consensus between individuals and within the group as a whole.

The distance between two differing opinions, x and y, is measured using the Manhattan distance, which effectively accounts for differences across various attributes and reduces data complexity. The equation for measuring the distance between two opinions is provided in Equation (27):

$$d_{x,y} = \frac{1}{n}\sum_{i=0}^{n}\left(\frac{1}{m}\sum_{j=0}^{m}\left|p_{ij}^{x} - p_{ij}^{y}\right|\right) \tag{27}$$

The calculation of the distance between individual opinions and group opinions is shown in Equation (28):

$$D_x = \frac{1}{d-1}\sum_{i=0}^{d-1}(d_{x,i}) \tag{28}$$

The calculation of the consensus degree of the comprehensive group opinion is shown in Equation (29):

$$GC = \frac{1}{d}\sum_{i=0}^{d} 1 - D_x \tag{29}$$

To calculate the current consensus level of the group, begin by using Equation (27) to compute the opinion distance matrix for every pair of DMs. Next, apply Equation (28) to determine the distance between each DM and the group based on the distance matrix. Finally, use Equation (29) to calculate the average of the consensus levels between each DM and the group, which will provide the overall group consensus level.

4.2. DFT-MOH Group Consensus Achievement Mechanism

To facilitate the introduction of the CRP, this subsection defines the metrics needed to measure consensus and classify DMs during the CRP.

4.2.1. Detection of Minority Opinions and Non-Cooperative Behavior

The mechanism for detecting misbehaviors can be see in Figure 4.

The Pareto principle, also known as the 80/20 rule, was first introduced by the Italian economist Pareto [57]. In his study of wealth distribution, Pareto observed that approximately 80% of wealth is controlled by 20% of the population. A similar situation can be found in decision processes [58]. When a small number of experts are involved, their influence can far outweigh that of the other DMs. When these experts' opinions differ significantly from the group consensus, specific strategies must be adopted to fully consider their input. In this paper, the Pareto principle is used as one of the criteria for detecting minority opinions. It is assumed that when fewer than 20% of the DMs in a community are involved, yet their influence exceeds 80%, a minority opinion group exists. If, at the same time, the consensus level of this minority group is lower than that of other groups, the MOH method should be employed for opinion management.

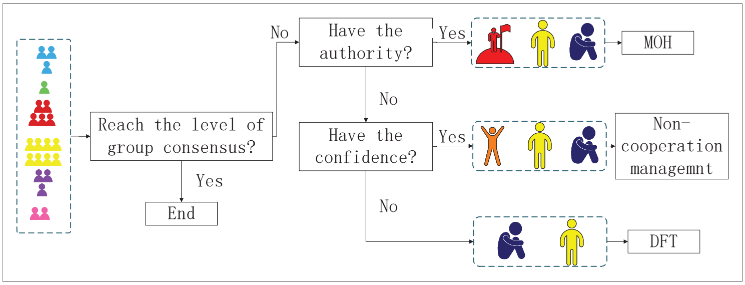


Figure 4. The flowchart of misconduct detection.

Within a group, there are often individuals who refuse to revise their opinions due to overconfidence. While they may hold significant personal weight, it is not on par with that of experts, meaning their opinions are no more valuable than those of other DMs. Despite this, their substantial influence can still have a significant impact on the final decision. To detect such non-cooperative behaviors, this paper proposes the following criteria: when fewer than 20% of individuals in a community possess a weight greater than 50% and their opinions show low consensus with the rest of the group, this is considered the evidence of non-cooperative behaviors driven by overconfidence.

When the decision body does not exhibit any of the issues described above, it is assumed that there are no other misconducts among DMs. In such cases, the standard opinion adjustment strategy for regular DMs, known as DFT, can be applied.

Let us assume that after grouping, the number of DMs in one group is x, and we use this group as an example. Let W_g represent the weight of this group. The classification criteria for this group are provided in Equation (30):

$$\begin{cases} \frac{x}{d} \leq 0.2, w_g \geq 0.8 \quad MOH \\ \frac{x}{d} \leq 0.2, w_g \geq 0.5 \quad Non-Cooperation \\ else \quad DFT \end{cases} \tag{30}$$

4.2.2. Management Mechanism for Minority Opinions

In this subsection, we present the specific management methods for addressing minority opinions discussed in this paper.

Given the inherent high authority and reference value of minority opinions, significant modifications are applied to the opinions of other DMs.

Based on the relative authority and trust among DMs, the corresponding adjustment coefficients for their opinions are defined in Equation (31):

$$TA = \frac{1}{2}(1 - w_i) + \frac{\sum\limits_{j=0}^{n} t_{ij}}{2 \times \sum\limits_{k=0}^{n} \sum\limits_{j=0}^{n} t_{kj}} \tag{31}$$

According to the adjustment coefficient of opinions and the weighted average of current group opinions, we can calculate the pace of adjustment for the DM's opinions by Equation (32):

$$CR_{ij} = \sum_{j=0}^{n} \left(\frac{t_{ij}}{\sum\limits_{k=0}^{n} t_{ik}} o_{ij} \right) \tag{32}$$

Based on the obtained pace, the opinions can be adjusted using Equation (33):

$$o_{ij} = \begin{cases} o_{ij} + TA \times CR_{ij} & o_{ij} < CR_{ij} \\ o_{ij} & o_{ij} = CR_{ij} \\ o_{ij} - TA \times CR_{ij} & o_{ij} > CR_{ij} \end{cases} \tag{33}$$

When minority opinions are detected within the group, the MOH mechanism is activated. First, the current overall opinion of the group is calculated using a weighted average. Subsequently, Equation (31) is employed to determine the adjustment coefficient for each DM, and Equation (32) is used to ascertain the size of the opinion adjustment step for each individual. Thereafter, Equation (33) is utilized to update the opinion of each DM accordingly. Finally, a recalculation of the consensus level among the DMs is performed. If consensus is achieved, the process concludes; if not, the procedure returns to recompute the overall group opinion and the adjustment coefficients for another round of opinion adjustments.

4.2.3. Management Mechanism for Non-Cooperative Behaviors Caused by Overconfidence

Overconfidence-induced non-cooperative behaviors can make it difficult to adjust the opinions of those unwilling to cooperate. To address this, this subsection introduces a dual mechanism for adjusting both opinions and weights. This mechanism reduces the weights of DMs who refuse to modify their opinions, while using group consensus to guide the opinion adjustments of others.

For DMs exhibiting non-cooperative behaviors, their opinions are adjusted according to the group's level of distrust. The adjusted result for these individuals is given in Equation (34):

$$w_{non} = w_{non} C_{non} \tag{34}$$

Equation (35) shows the pace of opinion adjustments for DMs, excluding the opinion adjuster:

$$CR_{ij} = \sum_{j=0}^{n} \left(w_i \times o_{ij} \right) \tag{35}$$

Calculate the final adjusted opinion based on Equation (36):

$$o_{ij} = \begin{cases} o_{ij} + \frac{CR_{ij}}{5} & o_{ij} < CR_{ij} \\ o_{ij} & o_{ij} = CR_{ij} \\ o_{ij} - \frac{CR_{ij}}{5} & o_{ij} > CR_{ij} \end{cases} \tag{36}$$

To manage non-cooperative behaviors within a group and achieve consensus, the following process is used: First, calculate the current group's aggregate opinion by taking a weighted average. Then, apply Equation (35) to determine the adjustment step for the non-cooperative DM's opinion. Next, use Equation (34) to calculate the weight adjustments for the non-cooperative actors. Recalculate the adjusted opinions using Equation (36), and normalize the weight matrix accordingly. Finally, recompute the group consensus level based on the revised opinions. If consensus is reached, the process concludes. If not, evaluate the presence of any detrimental behavior within the group and make the necessary adjustments.

4.2.4. Management Mechanism Under Normal Circumstances

This subsection outlines the management of group opinions under general circumstances. In ordinary situations, the DFT is used for opinion adjustments. For groups that have not yet reached consensus, Equation (32) is applied to assess the consensus degree of each DM relative to the group's aggregate opinion. The two DMs with the lowest consensus levels are selected, and Equation (33) is used to calculate the pace of opinion adjustment. Subsequently, Equation (36) is employed to adjust the opinions. After the adjustments, a consensus measurement is conducted. If consensus is not reached, the process returns to selecting two DMs with the lowest consensus levels for another round of opinion adjustment. If consensus is achieved, the current CRP concludes.

4.2.5. Application of MOH-DFT in Two-Stage Consensus Adjustments

In this subsection, the consensus management process and detection methods discussed above are integrated to present a two-stage CRP.

In the first stage, the collaborative indicator serves as the clustering criterion, and the DFT management approach is applied to adjust opinions within each group. After the opinions within each group are adjusted, the group's representative opinion is determined using a weighted average.

In the second stage, the two potential types of misbehaviors are detected and addressed using the corresponding adjustment strategies.

To provide a clearer illustration of the opinion adjustment process, a flowchart outlining the specific adjustment steps is shown in Figure 5.

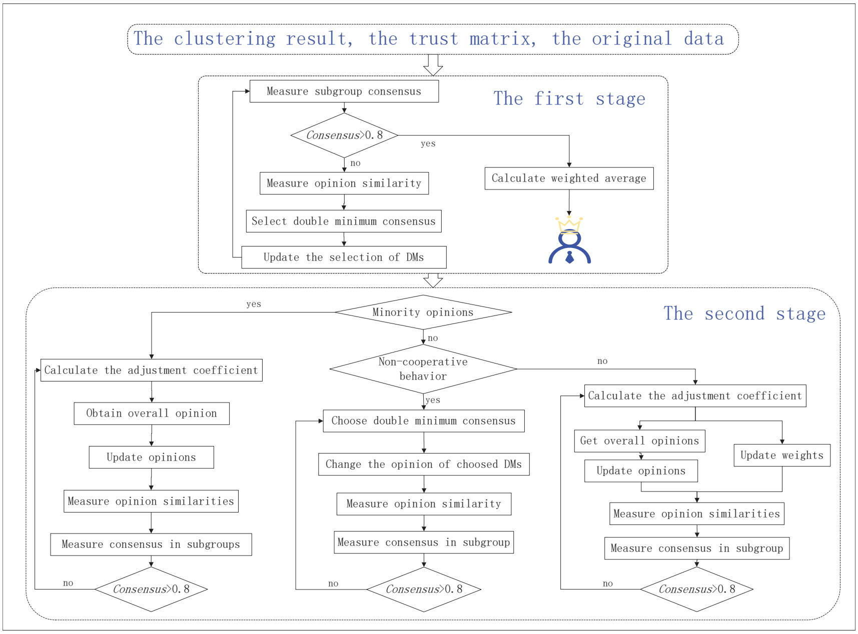

The clustering result, the trust matrix, the original data
The first stage
Measure subgroup consensus
Consensus>0.8
yes
no
Measure opinion similarity
Select double minimum consensus
Update the selection of DMs
Calculate weighted average
The second stage
Minority opinions
yes
no
Non-cooperative behavior
yes
no
Calculate the adjustment coefficient
Obtain overall opinion
Update opinions
Measure opinion similarities
Measure consensus in subgroups
Consensus>0.8
no
Choose double minimum consensus
Change the opinion of choosed DMs
Measure opinion similarity
Measure consensus in subgroup
Consensus>0.8
no
Calculate the adjustment coefficient
Get overall opinions
Update opinions
Update weights
Measure opinion similarities
Measure consensus in subgroup
Consensus>0.8
no

Figure 5. Two-stage opinion adjustments flowchart.

The pseudocodes for the CRP of the algorithm can be seen in Algorithm 2.

The time complexity of the pseudocodes is primarily determined by the consensus iteration phase. The initialization and basic calculations take $O(num\_DMs)$ per subgroup. The consensus iteration phase runs for up to $max\_iterations$, with each iteration involving operations that require $O(num\_subgroups \times num\_DMs)$. Finally, the final calculations have a complexity of $O(num\_subgroups \times num\_DMs)$. Therefore, the overall time complexity is dominated by the consensus iteration phase, resulting in $O(max\_iterations \times num\_subgroups \times num\_DMs)$.

Algorithm 2: Pseudocodes for Managing General Cases in Group Consensus Adjustments

```
Input: The trust matrix among DMs, the opinion matrix of DMs, community division results.
Output: Opinions after reaching consensus.
Function Main:
    for each subgroup do
        Calculate the comprehensive opinion of the current subgroup using weighted average;
        Select the two DMs with the lowest level of consensus in this subgroup;
        Calculate the adjustment coefficients for the two adjusted DMs using Equation (31);
        Adjust DMs' opinions using adjustment coefficients by Equation (36);
        Calculate the current consensus among subgroups using Equation (29);
        while $consensus < 0.8$ do
            Use weighted average to represent subgroup opinions;
            Use Equation (30) to determine the strategy required for grouping representative
              opinions;
            if choose minority opinion management then
                for each subgroup representative opinion do
                    if not the holders of minority opinions then
                        Calculate the adjustment coefficients using Equation (31);
                        Calculate the comprehensive opinion of the current subgroup using
                          weighted average;
                        Adjust DMs' opinions using adjustment coefficients by Equation (31);
                        Calculate the current consensus among subgroups using Equation (29);
            if choose non-cooperative behavior management then
                for each subgroup representative opinion do
                    if not a non-collaborator then
                        Calculate the comprehensive opinion of the current subgroup using
                          weighted average;
                        Calculate the adjustment coefficients using Equation (35);
                        Adjust DMs' opinions using adjustment coefficients by Equation (36);
                    if a non-collaborator then
                        Calculate the adjustment coefficients using Equation (31);
                        Adjust DMs' weight using adjustment coefficients by Equation (34);
                        Calculate the current consensus among subgroups using Equation (29);
            else
                Calculate the comprehensive opinion of the current subgroup using weighted
                  average;
                Select the two DMs with the lowest level of consensus in this subgroup;
                Calculate the adjustment coefficients for the two adjusted DMs using Equation (31);
                Adjust DMs' opinions using adjustment coefficients by Equation (33);
                Calculate the current consensus among subgroups using Equation (29);
    Calculate the value function and loss function of each group's opinions;
    Calculate group comprehensive;
```

5. Application of HOAL-DFT-MOH Method in Different Real-Life Scenarios

One important criterion for evaluating the effectiveness of a method is its applicability. This section demonstrates the universality and flexibility of the HOAL-DFT-MOH method across several application scenarios, as well as the accuracy of SNA in achieving consensus adjustment results. Three scenarios are presented, each involving 20 DMs who collectively make the final decision on four alternative options based on four attributes. The overall flowchart of the proposed method is provided in Figure 6.

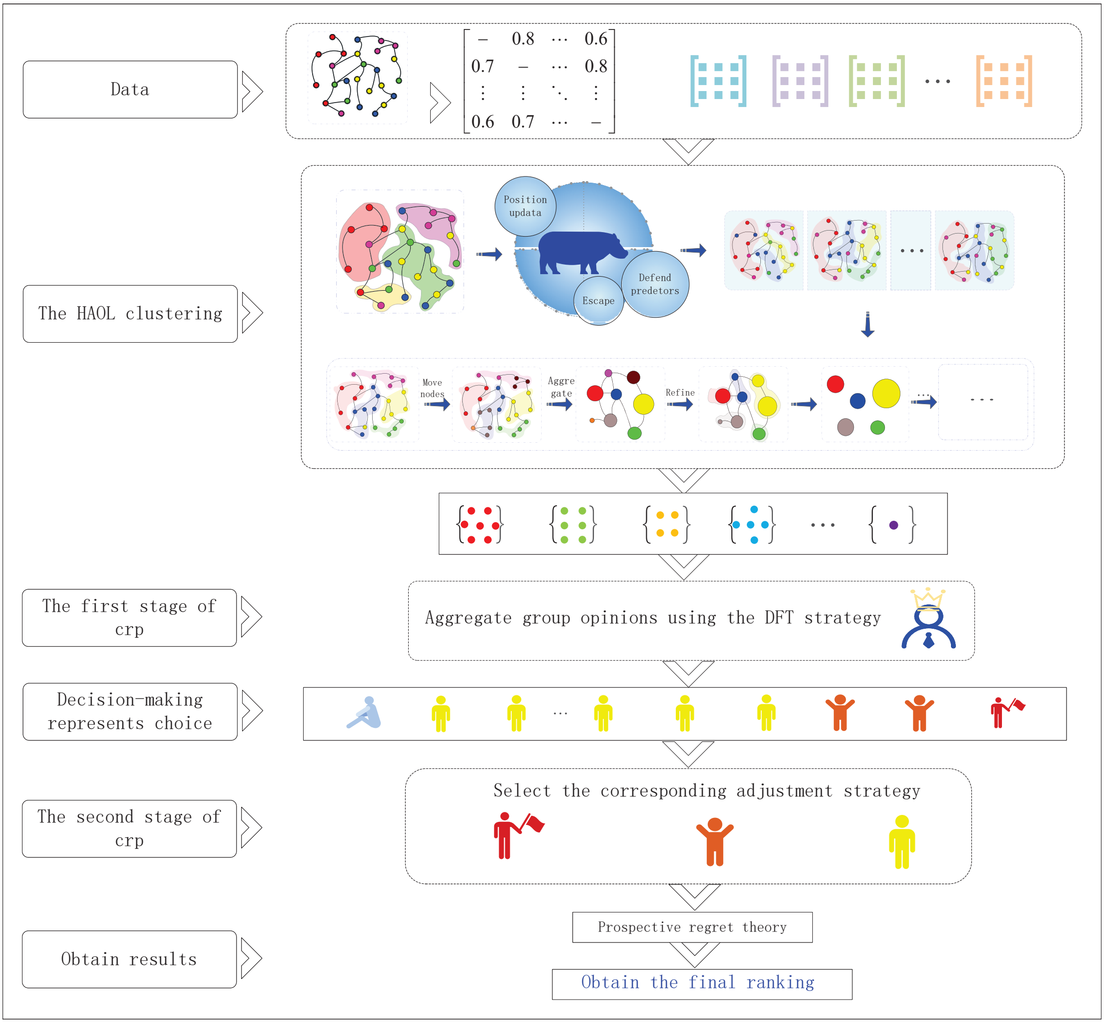

Data
$\begin{bmatrix} - & 0.8 & \cdots & 0.6 \\ 0.7 & - & \cdots & 0.8 \\ \vdots & \vdots & \ddots & \vdots \\ 0.6 & 0.7 & \cdots & - \end{bmatrix}$
The HAOL clustering
Position updata
Defend predetors
Escape
Move nodes
Aggre gate
Refine
The first stage of crp
Aggregate group opinions using the DFT strategy
Decision-making represents choice
The second stage of crp
Select the corresponding adjustment strategy
Prospective regret theory
Obtain results
Obtain the final ranking

Figure 6. The flowchart of our method.

5.1. Introduction to Three Types of Example Scenarios

This paper addresses the trust relationships among DMs and the potential misbehavior of DMs, which are common in everyday decision processes. The proposed method not only enhances the reliability and democratic nature of the results but also holds promise for broader applications as social media and networking platforms continue to evolve. This subsection presents three application scenarios, along with their inputs, to demonstrate how the method can be applied in real-world contexts.

E1: Material Selections in Production

The heat dissipation of electronic components is crucial for the lifespan and performance of electronic devices [59]. Effective heat dissipation methods are essential in electronic device manufacturing, and the use of cooling agents is a common solution. This subsection introduces a method for evaluating the advantages and disadvantages of CPU heat dissipation schemes based on the selection of coolant types. Four types of coolants are considered, i.e., biological nanofluid, water-Al nanofluid, copper oxide water nanofluid, and water. The evaluation criteria include Reynolds number, particle concentration, thermal conductivity, and cost.

E2: Corporate Strategy Decisions

Consider the example of an electronic product research and development company looking to increase investment in its current product line. The company often chooses to refine one aspect of its production process, typically in areas such as technology research and development, talent cultivation, infrastructure investment, and marketing promotion. Before making an investment, company managers need to analyze each aspect and make a final decision. This decision is based on four core evaluation criteria, i.e., deficiency degree, investment effectiveness, return on investment, and investment risk.

E3: Personal Product Choices

Lastly, a personal example of choosing an electronic product is provided. When an individual wants to select a mobile phone brand, they may turn to social media to gather input from a randomly selected group of 20 netizens. The DMs evaluate four brands Huawei, Apple, Samsung, and Xiaomi based on four attributes, i.e., cost performance, portability, additional features, and appearance. Through collaborative negotiation, a comprehensive ranking for each brand is determined, and a consensus result is reached.

5.2. Example Decision Process and Results Display

In this subsection, we show how the experiments are conducted based on the three situations given above, and the final decision-making results are obtained. The trust matrix among DMs is shown in Appendix B, and the opinion matrix of DMs is shown in Appendix A.

Step 1: Group the DMs using the HOAL algorithm and collaborative index, as presented in Table 2. Display the community weights and trust relationships between communities in Table 3.

Table 2. The result of clustering analysis.

E1	E2	E3
$g_1 = \{e_2, e_{10}, e_{12}, e_{14}, e_{15}\}$, $g_2 = \{e_0, e_3, e_5, e_{11}, e_{16}\}$, $g_3 = \{e_8, e_{13}, e_{17}, e_{18}\}$, $g_4 = \{e_1, e_4, e_7, e_{19}\}$, $g_5 = \{e_6, e_9\}$	$g_1 = \{e_2, e_7, e_{11}, e_{12}, e_{15}, e_{16}, e_{17}, e_{19}\}$, $g_2 = \{e_0, e_1, e_9, e_{18}\}$, $g_3 = \{e_4, e_5, e_6, e_8\}$, $g_4 = \{e_3, e_{10}, e_{13}, e_{14}\}$	$g_1 = \{e_1, e_2, e_4, e_5, e_7, e_8, e_{10}, e_{12}\}$, $g_2 = \{e_0, e_3, e_6, e_{11}, e_{13}, e_{19}\}$, $g_3 = \{e_9, e_{14}, e_{15}, e_{16}, e_{17}, e_{18}\}$

Step 2: Assign weights to subgroups based on the degree of mutual trust, similarity of opinions, and the number of people in each group. Calculate the level of mutual trust between different groups based on the level of mutual trust between individuals in each group. The results are shown in Table 3.

Table 3. Indicators within subgroups.

E1	gw_1= 0.249	1.0000	0.4680	0.4300	0.4950	0.6200
	gw_2= 0.245	0.5240	1.0000	0.5950	0.3950	0.5000
	gw_3= 0.187	0.4950	0.3700	1.0000	0.4625	0.3625
	gw_4= 0.205	0.4650	0.4550	0.4750	1.0000	0.6875
	gw_5= 0.114	0.5000	0.6600	0.5000	0.5125	1.0000
E2	gw_1= 0.396	1.0000	0.5156	0.5031	0.5594	
	gw_2= 0.201	0.5250	1.0000	0.3875	0.4625	
	gw_3= 0.188	0.4406	0.5625	1.0000	0.5063	
	gw_4= 0.215	0.4719	0.4125	0.4875	1.0000	
E3	gw_1= 0.387	1.0000	0.5167	0.5646		
	gw_2= 0.303	0.4583	1.0000	0.4917		
	gw_3= 0.310	0.4833	0.4972	1.0000		

Step 3: Using the DFT algorithm, obtain the integrated opinions of each subgroup as shown in Appendix C. The consensus within the subgroup at this time is as follows:

$A : SGC = \{0.8296, 0.8112, 0.8050, 0.8288, 0.9325\};$

$B : SGC = \{0.9025, 0.8375, 0.8381, 0.8338\};$

$C : SGC = \{0.9038, 0.8031, 0.8050, 0.9050\}.$

Step 4: Utilize detection mechanisms to adjust opinions between groups and detect the presence of inappropriate behavior. After testing, there are no adverse behaviors found in E1 , E2, and E3.

Step 5: Determine the management approach for opinions based on the detection of negative behaviors and their types. In cases where no misbehaviors are identified, employ DFT for opinion adjustments. Upon reaching consensus, calculate the final consensus degree for each instance, with the results shown as follows:

$E1 : GC = 0.9211;$

$E2 : GC = 0.9248;$

$E3 : GC = 0.9309.$

Step 6: Use the prospect theory of regret to derive the final scores and rankings for the products. The results are shown in Table 4.

Table 4. The final score matrix.

E1's Final Score Matrix	$\begin{bmatrix} 0.9578 & 1.4023 & 0.9465 & 0.4048 \\ 1.1662 & 1.2253 & 0.3390 & 1.0806 \\ 0.8883 & 0.8035 & 0.7030 & 1.4903 \\ 1.3005 & 0.8839 & 0.8119 & 1.4191 \end{bmatrix}$
E2's Final Score Matrix	$\begin{bmatrix} 1.5809 & 0.1127 & 0.6476 & 1.7655 \\ 1.2870 & 1.2274 & 1.6484 & 0.1804 \\ 1.8256 & 0.6592 & 0.6380 & 1.2040 \\ 0.5150 & 1.0076 & 0.6507 & 1.5515 \end{bmatrix}$
E3's Final Score Matrix	$\begin{bmatrix} 0.0923 & 0.3883 & 1.1255 & 0.8513 \\ 1.5825 & 0.4193 & 0.9414 & 1.5143 \\ 0.9868 & 0.5149 & 1.1291 & 0.4115 \\ 0.7501 & 0.5996 & 1.8331 & 0.5951 \end{bmatrix}$

Step 7: Use the weighted average of different attribute scores to calculate the final score of the alternative plan, and use a line graph to represent the score. The results of the three cases are shown in Table 5 below.

Table 5. The score ranking result.

E1	**E2**	**E3**
$X_4 \succ X_3 \succ X_2 \succ X_1$	$X_4 \succ X_3 \succ X_2 \succ X_1$	$X_2 \succ X_1 \succ X_4 \succ X_3$

6. Further Analysis

In this section, sensitivity experiments, simulation experiments, and comparative experiments with other methods are conducted to verify the usability and superiority of the HOAL-MOH method proposed in this paper.

6.1. Comparison Analysis

Conduct comparative experiments between the method proposed in this paper and existing methods to further demonstrate the advantages of the proposed method.

6.1.1. A Comparison of Our Method and Its Sub-Methods

To better demonstrate the necessity of each sub-method used in the approach presented in this paper, we explain the role each plays within the overall method. By applying a control variable approach, we systematically remove each sub-method and observe its impact on certain process indicators.

The summary of the deformation methods based on the approach proposed in this paper is provided in Table 6.

Table 6. The summary of required comparison methods.

Our method	m0
Our method removes HOA optimization in clustering	m1
Our method removes the Leiden clustering	m2
Our method replaces DFT with single-DM adjustment	m3
Our method removes MOH	m4
Our method removes non-cooperative behavior management	m5

The methods m0, m1, and m2 are compared to explore the impact of improvements made during the clustering process on the results. The comparison is based on four aspects: total execution time, clustering execution time, consensus within subgroups, and trust within subgroups. Each method is executed 500 times, with the aforementioned metrics recorded, and the average values for each metric are provided. The overall results are shown in Table 7.

Table 7. The analysis of the necessity of sub-methods in clustering.

Methods	The Total Execution Time	The Clustering Execution Time	The Consensus Within Subgroups	The Trust Within Subgroups
m0	0.0135	0.0100	0.941	0.843
m1	0.0210	0.1755	0.940	0.831
m2	0.0106	0.794	0.702	0.726

The execution efficiency of method m0 lies between that of m2 and m3, being closer to m2. The clustering results of m0 and m3 are very similar. However, the clustering performance of m2 significantly differs from that of m0 and m1. Overall, although method m0 does not outperform the others in any particular metric, it effectively combines the advantages of the other methods, mitigates their weaknesses, and provides an efficient solution.

In the opinion adjustment process without undesirable behaviors, DFT plays a crucial role. To provide more convincing evidence, we randomly generated initial data and executed the process using both the m0 and m3 methods. The procedure was repeated over 100 times, and the required metric values were recorded and averaged. The results are shown in Table 8.

Table 8. The analysis of the necessity of DFT in CRP.

Methods	The Group Consensus	The Adjustment Rounds	The Execution Time
m0	0.961	4	0.0012
m3	0.912	6	0.0019

As shown in Table 8, after transforming the DFT to focus on opinion adjustments for individual DMs, the impact of each adjustment round on the overall group consensus decreases. Therefore, to achieve the same adjustment effect, both the number of adjustment rounds and the adjustment time have to increase.

To validate the importance of the consensus adjustment strategy, Table 9 compares method m0 with methods m4 and m5. Since different behaviors occur with certain probabilities in real-world scenarios, our method provides manually selected data for each comparison method to more clearly reflect the strategy's impact. For instance, in the case of methods lacking the MOH mechanism, the data should include minority opinions. Consensus and final opinion adjustment results are used as reference metrics for statistical analysis. The values outside the parentheses represent the metrics for the respective method, while the values inside the parentheses represent the metrics for m0 with the same dataset. The overall results are shown in Table 9.

Table 9. The analysis of the necessity of MOH and non-cooperative management in CRP.

Methods	The Adjustment Rounds	The Execution Time	The Group Consensus	Decision Otcomes
m4(m0)	3(5)	0.0014(0.0021)	0.951(0.934)	$x2 \succ x1 \succ x4 \succ x3(x4 \succ x1 \succ x2 \succ x3)$
m5(m0)	4(6)	0.0010(0.0015)	0.938(0.933)	$x3 \succ x1 \succ x2 \succ x4(x1 \succ x3 \succ x2 \succ x4)$

After removing the behavior detection for certain minority DMs, the opinion adjustment process becomes one where the opinions of the minority DMs move closer to those of the majority. As a result, the scale of the adjustments required is reduced, which is reflected in the decrease in both adjustment time and iterations shown in Table 9.

Additionally, the consensus level of methods m4 and m5 is higher than that of method m0, as the impact of minority DMs on the final group consensus is minimized. This leads to a result that is closer to the collective opinion of the group.

However, despite the removal of non-cooperative behavior management, the method does not fully account for the opinions of each individual DM. Consequently, the final decision deviates from the true objective scenario, creating only the illusion of improved efficiency and correct outcomes.

6.1.2. Comparison Between HOAL and Other Methods

This subsection compares the clustering method proposed in this paper with the clustering methods of others, and the results are shown in Table 10.

As shown in Table 10, the method proposed in this paper not only performs clustering based on SN but also incorporates a collaborative indicator into the community detection criteria. In contrast, the method by Guo et al. [28] only considers opinion similarity as the clustering criterion, without integrating the trust between DMs or considering the decision-making context within the SN. As a result, the consensus and average trust levels within subgroups are the lowest among the four methods compared.

Table 10. The comparative analysis of clustering outcomes.

Methods	Indicators	Based on SNs	Consensuses in Subgroups	The Average Trust Level
Guo et al. [28]	Opinion similarity	No	0.423	0.611
Song et al. [23]	Opinion similarity and trust relation	No	0.548	0.791
Shen et al. [25]	Opinion similarity and trust relation	Yes	0.654	0.725
Our method	Opinion similarity and trust relation	Yes	0.873	0.854

Although Song et al. [23] improved their clustering criterion by incorporating a collaborative indicator, they still do not take into account the decision-making context within the SN. Therefore, while their method shows some improvement in terms of the clustering indicator, it still cannot surpass the methods by Shen et al. [25] and the one proposed in this paper.

Compared to the method by Shen et al. [25], the method in this paper not only uses the Leiden clustering algorithm, which outperforms the Louvain algorithm, but also further optimizes the traditional Leiden algorithm.

In summary, compared to other methods, the partitioning results obtained by our approach exhibit higher consensus and trust levels within subgroups. The clustering outcomes align more closely with real-life expectations, grouping DMs with high opinion similarities and mutual trust into the same community. This effectively reduces the number of subsequent adjustment steps.

6.1.3. Qualitative Analysis of Our Method and Other Methods

To illustrate the advantages and disadvantages of different consensus models, Table 11 selects the basic characteristics of the formula model to compare with existing methods, highlighting the strengths of the approach proposed in this paper. For ease of the comparisons, the models involved in the comparison are divided into three categories: traditional LSGDM models (TL), LSGDM models combined with SN (SNL), and LSGDM models that manage misbehaviors (ML).

Table 11. The comparative analysis of consensus adjustment methods: a qualitative approach.

Types	References	MOH	Non-Cooperative Behaviors	Opinion Management Reference Trust	Clustering Based on Opinion Similarities	Clustering Based on Trust Relationships
TL	Yuan et al. [19]	No	No	No	Yes	No
TL	Chen et al. [60]	No	No	No	Yes	No
TL	Meng et al. [42]	No	No	No	Yes	No
SNL	Liang et al. [61]	No	No	Yes	Yes	No
SNL	Du et al. [39]	No	No	Yes	Yes	No
ML	Liu et al. [45]	Yes	Yes	Yes	Yes	No
ML	Xu et al. [43]	Yes	Yes	Yes	Yes	No
ML	Song et al. [23]	No	Yes	Yes	Yes	Yes
ML	Shen et al. [25]	Yes	No	Yes	Yes	Yes
ML	Our method	Yes	Yes	Yes	Yes	Yes

As shown in Table 11, the given TL method does not account for potential undesirable behaviors that may arise during decision-making in the CRP, nor does it consider the trust relationships among DMs in the management of opinions. Additionally, it uses a single indicator for clustering.

In contrast, the SNL method adds the trust relationship between DMs to guide the opinion adjustment process, compared to the TL approach. The ML methods, such as those by Liu et al. [45] and Xu et al. [43], rely solely on a single clustering indicator. Song et al. [23] do not consider the MOH, while Shen et al. [25] do not address non-cooperative management.

In this paper, the proposed method not only considers two types of undesirable behavior management but also integrates opinion adjustment with a trust matrix. Furthermore, it uses a collaborative indicator based on both trust relationships and opinion similarities in the clustering analysis.

Therefore, the method presented in this paper covers more possible scenarios, takes into account a broader range of influencing factors, and is more aligned with real-life situations compared to the other methods.

The above method provides a qualitative comparison between existing approaches and the proposed method. To enable a more objective comparison of the LSGDM method and present more convincing results, Table 12 presents a quantitative analysis in comparison with methods from ML.

Table 12. The comparative analysis of consensus adjustment methods: a quantitative approach.

Methods	Average Number of Rounds Adjusted in The First Stage	Final Average Consensus of Subgroup	Number of Rounds Adjusted in The Second Stage	Final Consensus Degrees	Decision Outcomes
Liu et al. [45]	6	0.864	4	0.854	$x2 \succ x1 \succ x4 \succ x3$
Song et al. [23]	4	0.971	5	0.791	$x2 \succ x1 \succ x4 \succ x3$
Shen et al. [25]	0	0.543	8	0.875	$x2 \succ x1 \succ x4 \succ x3$
Our method	3	0.901	2	0.920	$x2 \succ x1 \succ x4 \succ x3$

From Table 12, it can be seen that the decision-making method proposed in this paper not only has fewer adjustment rounds, but also has a higher level of consensus, which is more in line with the public opinion level.

As shown in Table 12, the proposed method requires only three adjustments in the first round. Except for Shen et al. [25], which does not involve opinion adjustments in the first round, the average number of adjustments for the proposed method is lower than that of other methods. The average group consensus within subgroups is second only to that achieved by Song et al. [23]. In the second round, the number of adjustments reduces to two, while other methods require more than twice as many adjustments. Ultimately, the overall consensus achieved by the proposed method exceeds that of other methods, effectively integrating the opinions of all DMs.

6.2. Simulation Analysis

In this subsection, simulation tests are conducted to verify the feasibility of the proposed method on a limited scale. The simulation experiment is conducted using PyCharm Community Edition 2023.3.2, executed on a personal computer equipped with an Intel Core i7-12700 processor from the 12th generation, with a running frequency of 3.90 GHz.

6.2.1. Feasibility Simulation Within a Limited Scale

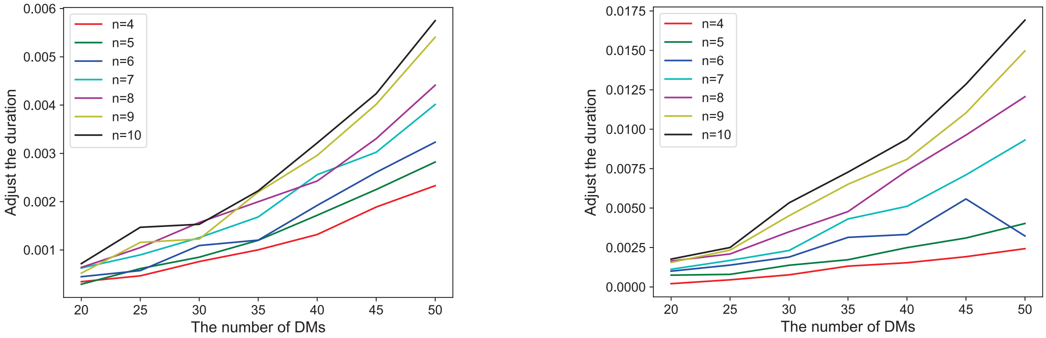
Choose seven different scales for DMs aged 20 to 50 and change the number of alternative solutions for different DM scales. Adjust the above conditions separately, execute 500 times for each situation to obtain the time for a single execution, and record the time required to reach a final consensus on the two stages of opinion adjustment. Figure 7 shows a line chart of the time required for different adjustments in two stages.

As shown in Figure 7, both the adjustment time increases more steeply with the number of DMs in both the first and second stages. Additionally, the increase in the size of the decision replacement options also leads to an increase in adjustment time. It can be observed that, in both adjustment rounds, when the number of DMs is the same, larger decision replacement option sizes are positioned higher than smaller ones.

In the first stage of opinion management, when the number of DMs reaches 50 and there are 10 alternative options, consensus can still be reached within 0.006 s. For the same scale, the consensus adjustment in the second stage can be completed within 0.0175 s.

Within a certain scale range, adjustments in both stages can be completed within a limited time, thus meeting the adjustment requirements.

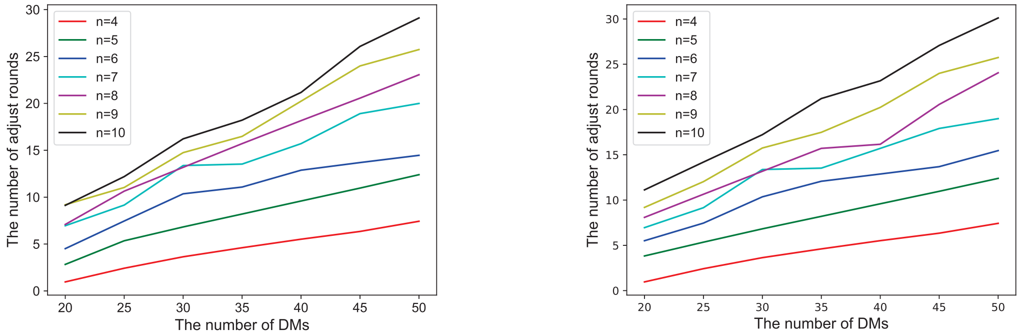
To provide a more intuitive demonstration of the efficiency of the proposed method, seven different scales of DMs were set within the range of 20 to 50, corresponding to 4 to 10 different alternative solutions. Each configuration was executed 500 times. The number of adjustment rounds required to achieve the desired formula in different stages was recorded, and the adjustment round charts for the first and second stages are shown in Figure 8.



(a) The First Stage

(b) The Second Stage

Figure 7. Two-stage adjustment time statistics.



(a) The First Stage

(b) The Second Stage

Figure 8. The statistics of adjustment rounds in two stages.

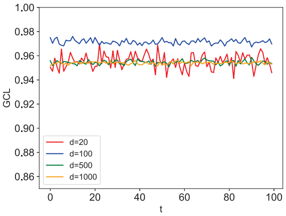
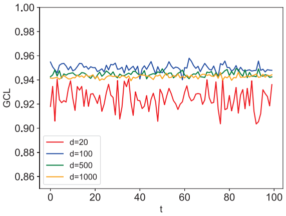
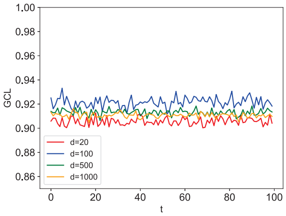
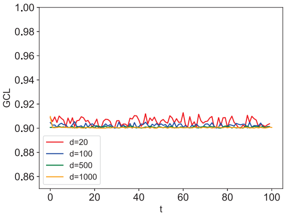
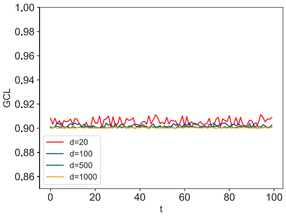
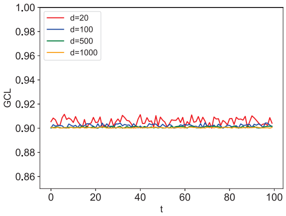
Although the trend of increasing adjustment rounds becomes steeper as the number of DMs increases, as shown in Figure 8, the scale of adjustment rounds increases more steadily. Similarly, an increase in the scale of decision replacement schemes, which corresponds to the number of adjustment rounds, also leads to an increase in adjustment rounds. However, the difference between adjustment rounds is more significant than that of adjustment time.

When the DM size is 50 and the number of alternative options is 20, the average number of adjustments in the first stage does not exceed 30, and the average number of adjustments in the second stage is also less than 30.

6.2.2. Simulation of Effectiveness for Large-Scale Opinion Management

In order to verify whether the DM and the increase in the size of the decision-making problem will have an impact on the consensus level of the final decision, the DM size is taken as $d = 20, d = 200, d = 500$, and $d = 1000$. The number of alternative options for the decision is the same as the number of attributes for each option, and six values from three to eight are taken to plot the impact on the consensus level of decisions, as shown in Table 13.

Table 13. The decision performance statistics under different scales.

$n = 3$, $m = 3$	$n = 4$, $m = 4$	$n = 5$, $m = 5$
		
$n = 6$, m =6	$n = 7$, $m = 7$	$n = 8$, $m = 8$
		

Analyzing all the line charts in Table 13, it can be observed that as the scale of decision alternatives and reference attributes increases, the final consensus level gradually decreases. However, the differences in final consensus levels caused by the increase in the number of DMs become smaller. This is because, when the decision problem itself becomes more complex, the complexity of opinions increases, leading to a greater variety of opinions. At this point, the growth in the number of DMs is no longer the primary factor influencing the consensus level.

From an application perspective, even when the scale increases, the final consensus level still exceeds 0.9, indicating that the decision outcomes maintain a high level of collective agreement and coherence, demonstrating the robustness of the decision process.

6.3. Sensitivity Analysis

This section conducts sensitivity testing experiments on the three parameters α, β, and λ in the theory of prospect-regret. Take the values of α as 1, 2, 3, 4, and 5. Take the values of β as 0.88, 1.76, 2.64, 3.52, and 4.40. Take the values of λ as 2.25, 4.50, 6.75, 9.00, and 11.25. Change the values of the above three parameters separately to obtain the final score of alternative solutions and draw it as a line graph, as shown in Table 14. The ranking results of alternative solutions under different values are shown in Table 15 through comparisons.

Based on the results in Tables 14 and 15, it can be seen that the ranking order of alternative solutions does not change when the value of α is greater than 2. The change in the values of β and λ will not cause a change in the ranking order of alternative solutions. Thus, in the method presented in this paper, we choose a value of 2 for α, 0.88 for β, and 2.25 as the values for λ.

Through the application of case studies in various scenarios, it can be observed that the method proposed in this paper is adaptable to a variety of LSGDM problems under different contexts. Furthermore, simulation experiments validated the feasibility of this method for a certain scale, consistently yielding final decision results within a limited time frame. Subsequently, comparisons are made with other methods. In the comparison of clustering methods, the proposed method not only incorporates more reference indicators but also achieves higher trustworthiness in the final clustering result. Regarding the comparison with CRP, it is evident that this method not only integrates

SNA with CRP but also requires fewer opinion adjustments and shorter adjustment times, further demonstrating its superiority. Finally, sensitivity analysis is conducted to verify the stability of results.

Table 14. The sensitivity analysis result for α, β, and λ.

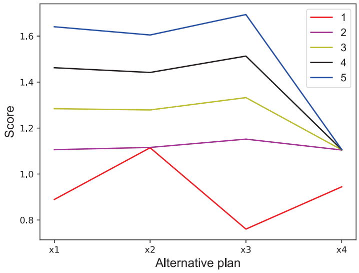
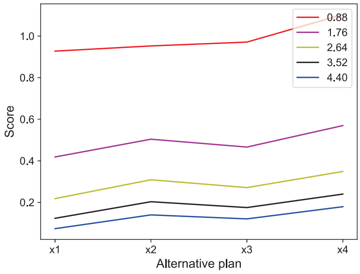
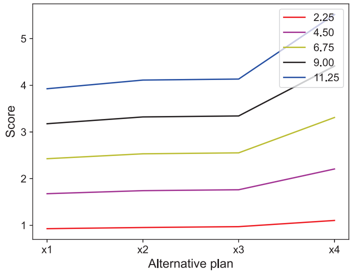
α	β	λ
		

Table 15. The ranking result in sensitivity analysis.

α	β	λ
$\alpha = 1, \beta = 0.88, \lambda = 2.25$ $x2 \succ x1 \succ x4 \succ x3$	$\alpha = 2, \beta = 0.88, \lambda = 2.25$ $x2 \succ x1 \succ x4 \succ x3$	$\alpha = 2, \beta = 0.88, \lambda = 2.25$ $x2 \succ x1 \succ x4 \succ x3$
$\alpha = 2, \beta = 0.88, \lambda = 2.25$ $x3 \succ x2 \succ x1 \succ x4$	$\alpha = 2, \beta = 1.76, \lambda = 2.25$ $x2 \succ x1 \succ x4 \succ x3$	$\alpha = 2, \beta = 0.88, \lambda = 4.50$ $x2 \succ x1 \succ x4 \succ x3$
$\alpha = 3, \beta = 0.88, \lambda = 2.25$ $x3 \succ x2 \succ x1 \succ x4$	$\alpha = 2, \beta = 2.64, \lambda = 2.25$ $x2 \succ x1 \succ x4 \succ x3$	$\alpha = 2, \beta = 0.88, \lambda = 6.75$ $x2 \succ x1 \succ x4 \succ x3$
$\alpha = 4, \beta = 0.88, \lambda = 2.25$ $x3 \succ x2 \succ x1 \succ x4$	$\alpha = 2, \beta = 3.52, \lambda = 2.25$ $x2 \succ x1 \succ x4 \succ x3$	$\alpha = 2, \beta = 0.88, \lambda = 9.00$ $x2 \succ x1 \succ x4 \succ x3$
$\alpha = 5, \beta = 0.88, \lambda = 2.25$ $x3 \succ x2 \succ x1 \succ x4$	$\alpha = 2, \beta = 4.40, \lambda = 2.25$ $x2 \succ x1 \succ x4 \succ x3$	$\alpha = 2, \beta = 0.88, \lambda = 11.25$ $x2 \succ x1 \succ x4 \succ x3$

7. Discussion

This section provides a further discussion of the method presented in this paper, including the problem contexts it effectively addresses and recommendations for executing each step of the method.

7.1. Discussion on Problem Types Effectively Addressed

Our method in this paper provides a solution to decision-making problems by considering the influence of interactions between DMs, as well as managing minority opinions and non-cooperative behaviors resulting from overconfidence, particularly in LSGDM contexts. The problem scenarios addressed are more complex and offer broader applications.

Our method also offers an approach to further analyze the authority and roles of DMs within the entire social network, based on their interpersonal relationships. This provides valuable insights for determining the weights when constructing SNs between DMs and others.

First, the method can be applied to traditional expert system problem-solving. In this context, initial expert weights are assigned based on the authority of each expert, and experts are required to evaluate each other's trust levels and vote on decision alternatives. Based on this initial data, a reasonable decision can be reached.

Furthermore, the method is applicable in situations where DMs have diverse professional backgrounds, such as in public voting systems that ensure fairness. In these cases, the professionalism of the participants cannot be guaranteed, and non-expert opinions may undermine the accuracy of the final decision. Unlike traditional methods that rely on experts, the proposed approach includes mechanisms to detect expertise and uses the MOH mechanism for opinion management.

Finally, the method can be extended to scenarios where DMs have more complex personality traits, such as trust relationships and evaluations derived from social media. In such cases, it cannot be assumed that all participants will offer their opinions in a serious and objective manner. The proposed method detects non-cooperative behaviors arising from overconfidence and provides strategies to mitigate their impact, leading to more reliable decision outcomes.

7.2. Discussion on Applying Our Method to Solve the Problem

Applying our method to solve real-world problems is simple and cost-effective. The method offers high decision-making efficiency, meaning that the computational demands on the processor are not strict. The code can be implemented in any programming language the user is familiar with. The application process consists of the following steps:

Step 1: Data collection, where DMs provide their trust relationships and initial opinions. If initial weights are available, they can be included; otherwise, equal weights are used for initialization.

Step 2: The initial data are input into the program, which generates outputs such as DM weights, groupings, opinion adjustments, and final adjustments within each group, along with the decision results.

Step 3: Based on the results, DMs can proceed with a series of related decisions, policy formulations, product comparisons, and other actions.

8. Conclusions

The complexity of the DMs' composition often leads to undesirable behaviors during the CRP, resulting in situations where decisions fail to reach consensus or yield incorrect outcomes. This study proposes the HOAL-MOH-DFT consensus adjustment model, which addresses both the handling of minority opinions and non-cooperative opinions arising from overconfidence. The following contributions are made:

(1) A collaborative indicator that combines trust relationships with opinion similarity among DMs is used as an evaluation metric for grouping quality, optimizing the initial community using HOA. Compared to other clustering methods, the proposed clustering method not only achieves a better clustering performance but also enhances the clustering efficiency.

(2) This paper proposes a new decision opinion processing mechanism, which integrates the distance between the group's comprehensive opinions, the proportion of group members, the size of DMs, and the group's weight. Three different opinion adjustment mechanisms are provided.

(3) Trust relationships and weight proximity are combined to obtain adjustment coefficients, and the current comprehensive opinion is used to manage the opinions within each group. This opinion management method ensures that all opinions are considered, making them more reasonable.

(4) MOH incorporates everyone's opinions by adjusting DMs' opinions in a balanced way. In the case of non-cooperative behaviors caused by overconfidence, a dual adjustment of weight and opinions is applied, reducing the influence of non-collaborators on the decision process while steering the opinions toward consensus. In the absence of misconduct, DFT is used for management, striking a balance between minimizing the adjustment range and reducing the number of adjustment rounds to achieve optimal results.

With the rapid development of social media and its increasing role in daily life, integrating these SN with LSGDM problems and formulating corresponding adjustment strategies has become a prominent research focus. Although the method described in this paper combines SNA with clustering and CRP while considering two types of misbehaviors during adjustment processes, there is still room for improvement in the following aspects:

(1) Individuals within an SN are not always interconnected, and there may be gaps in the trust matrix of SNs. It is therefore essential to explore an efficient and accurate method for filling these gaps.

(2) The reference metric for the closeness of social relationships is not limited to trust. During the opinion adjustment process and clustering analysis, more metrics from SNs should be considered to provide results that better reflect realities.

(3) During the opinion adjustment process, in addition to the two types of misconduct proposed in this paper, other misbehaviors such as interest-driven adjustments and herd behaviors should also be considered within the scope of opinion adjustments.

Author Contributions: Conceptualization, T.X. and C.Z.; Methodology, C.Z.; Validation, T.X., S.H. and X.Y.; Data curation, T.X.; Writing—original draft, T.X., S.H. and X.Y.; Writing—review and editing, C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Central Government Guides Local Science and Technology Innovation (YDZJSX2024D015), the Special Fund for the Training Program for Young Scientific Researchers of Higher Education Institutions in Shanxi, the Wenying Young Scholars of Shanxi University, the Science and Technology Innovation Teams of Shanxi (202204051001015), the 22nd Undergraduate Innovation and Entrepreneurship Training Program of Shanxi University (202410108008), and the Cultivate Scientific Research Excellence Programs of Higher Education Institutions in Shanxi (2019SK036).

Data Availability Statement: The authors declare that all data supporting the findings of this study are available within this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

E1′s Original Data

$$
\begin{bmatrix} 0.7 & 0.2 & 0.0 & 0.9 \\ 0.1 & 0.1 & 0.6 & 0.4 \\ 0.6 & 0.3 & 0.1 & 0.5 \\ 0.7 & 0.2 & 0.8 & 0.7 \end{bmatrix}
\begin{bmatrix} 0.8 & 0.6 & 0.2 & 0.6 \\ 0.8 & 0.1 & 0.6 & 0.6 \\ 0.4 & 0.8 & 0.8 & 0.3 \\ 0.0 & 1.0 & 0.1 & 0.3 \end{bmatrix}
\begin{bmatrix} 0.0 & 0.0 & 0.7 & 0.1 \\ 0.4 & 0.1 & 0.8 & 0.8 \\ 0.6 & 0.8 & 0.3 & 0.5 \\ 0.3 & 0.7 & 0.6 & 0.6 \end{bmatrix}
\begin{bmatrix} 0.6 & 0.7 & 0.7 & 0.5 \\ 1.0 & 0.2 & 0.5 & 0.1 \\ 0.1 & 0.7 & 0.1 & 0.1 \\ 1.0 & 0.8 & 0.9 & 0.0 \end{bmatrix}
$$

$$
\begin{bmatrix} 0.3 & 0.6 & 0.2 & 0.6 \\ 0.9 & 0.9 & 0.3 & 0.9 \\ 0.5 & 0.4 & 1.0 & 0.4 \\ 0.1 & 0.6 & 0.5 & 0.2 \end{bmatrix}
\begin{bmatrix} 0.6 & 0.2 & 0.7 & 0.8 \\ 0.8 & 0.4 & 0.9 & 0.8 \\ 0.4 & 0.3 & 0.2 & 0.5 \\ 0.1 & 0.8 & 0.8 & 0.5 \end{bmatrix}
\begin{bmatrix} 0.3 & 0.2 & 0.8 & 0.5 \\ 0.1 & 0.8 & 0.9 & 0.1 \\ 0.7 & 0.8 & 0.6 & 0.6 \\ 0.9 & 0.1 & 0.2 & 0.4 \end{bmatrix}
\begin{bmatrix} 0.7 & 0.7 & 0.2 & 0.7 \\ 0.2 & 0.2 & 0.5 & 0.8 \\ 0.8 & 1.0 & 0.4 & 0.0 \\ 0.1 & 0.8 & 1.0 & 0.1 \end{bmatrix}
$$

$$
\begin{bmatrix} 0.8 & 0.1 & 0.6 & 0.5 \\ 0.8 & 0.4 & 0.9 & 0.3 \\ 0.2 & 0.1 & 0.3 & 0.1 \\ 0.3 & 0.1 & 0.7 & 0.6 \end{bmatrix}
\begin{bmatrix} 0.3 & 0.0 & 0.4 & 0.5 \\ 0.3 & 0.9 & 0.7 & 0.0 \\ 1.0 & 0.8 & 0.7 & 0.5 \\ 1.0 & 0.0 & 0.7 & 0.8 \end{bmatrix}
\begin{bmatrix} 0.1 & 0.4 & 0.6 & 1.0 \\ 0.0 & 0.2 & 0.5 & 0.8 \\ 0.6 & 0.8 & 0.9 & 0.5 \\ 0.2 & 0.5 & 0.1 & 0.5 \end{bmatrix}
\begin{bmatrix} 0.9 & 0.6 & 0.2 & 0.4 \\ 0.6 & 0.3 & 0.5 & 0.9 \\ 0.6 & 0.4 & 0.1 & 0.4 \\ 0.2 & 0.8 & 1.0 & 0.2 \end{bmatrix}
$$

$$
\begin{bmatrix} 0.2 & 0.8 & 0.6 & 0.2 \\ 0.5 & 0.1 & 0.5 & 0.0 \\ 0.4 & 0.5 & 0.8 & 0.8 \\ 0.4 & 0.8 & 0.6 & 0.1 \end{bmatrix}
\begin{bmatrix} 0.6 & 0.5 & 0.8 & 0.3 \\ 0.2 & 0.3 & 0.1 & 0.1 \\ 0.2 & 0.3 & 0.9 & 0.1 \\ 0.7 & 0.4 & 0.0 & 0.1 \end{bmatrix}
\begin{bmatrix} 0.0 & 0.2 & 0.4 & 0.6 \\ 0.1 & 0.8 & 0.6 & 1.0 \\ 0.5 & 0.3 & 0.9 & 0.8 \\ 0.2 & 0.4 & 0.5 & 0.4 \end{bmatrix}
\begin{bmatrix} 0.4 & 0.6 & 0.2 & 0.9 \\ 0.1 & 0.0 & 0.7 & 0.4 \\ 0.4 & 0.5 & 0.8 & 0.5 \\ 0.3 & 0.8 & 0.5 & 0.8 \end{bmatrix}
$$

$$
\begin{bmatrix} 0.6 & 0.9 & 0.5 & 0.4 \\ 0.7 & 0.5 & 0.0 & 0.6 \\ 0.7 & 0.0 & 0.1 & 0.7 \\ 0.0 & 0.6 & 0.7 & 0.5 \end{bmatrix}
\begin{bmatrix} 0.8 & 0.3 & 0.6 & 0.5 \\ 0.5 & 0.6 & 0.5 & 0.2 \\ 0.1 & 0.2 & 0.0 & 0.2 \\ 0.0 & 0.2 & 0.1 & 0.2 \end{bmatrix}
\begin{bmatrix} 0.6 & 1.0 & 0.7 & 0.9 \\ 0.3 & 0.2 & 0.1 & 1.0 \\ 0.0 & 0.2 & 0.8 & 0.8 \\ 0.9 & 0.9 & 0.2 & 0.2 \end{bmatrix}
\begin{bmatrix} 0.2 & 0.1 & 0.2 & 0.1 \\ 0.7 & 0.9 & 0.8 & 0.2 \\ 0.6 & 0.3 & 0.7 & 0.0 \\ 0.1 & 0.6 & 0.3 & 0.2 \end{bmatrix}
$$

E2's Original Data

$$\begin{array}{cccc}
\begin{bmatrix} 0.7 & 0.2 & 0.0 & 0.9 \\ 0.1 & 0.1 & 0.6 & 0.4 \\ 0.6 & 0.3 & 0.1 & 0.5 \\ 0.7 & 0.2 & 0.8 & 0.7 \end{bmatrix} & \begin{bmatrix} 0.8 & 0.6 & 0.2 & 0.6 \\ 0.8 & 0.1 & 0.6 & 0.6 \\ 0.4 & 0.8 & 0.8 & 0.3 \\ 0.0 & 1.0 & 0.1 & 0.3 \end{bmatrix} & \begin{bmatrix} 0.0 & 0.0 & 0.7 & 0.1 \\ 0.4 & 0.1 & 0.8 & 0.8 \\ 0.6 & 0.8 & 0.3 & 0.5 \\ 0.3 & 0.7 & 0.6 & 0.6 \end{bmatrix} & \begin{bmatrix} 0.6 & 0.7 & 0.7 & 0.5 \\ 1.0 & 0.2 & 0.5 & 0.1 \\ 0.1 & 0.7 & 0.1 & 0.1 \\ 1.0 & 0.8 & 0.9 & 0.0 \end{bmatrix} \\
\begin{bmatrix} 0.3 & 0.6 & 0.2 & 0.6 \\ 0.9 & 0.9 & 0.3 & 0.9 \\ 0.5 & 0.4 & 1.0 & 0.4 \\ 0.1 & 0.6 & 0.5 & 0.2 \end{bmatrix} & \begin{bmatrix} 0.6 & 0.2 & 0.7 & 0.8 \\ 0.8 & 0.4 & 0.9 & 0.8 \\ 0.4 & 0.3 & 0.2 & 0.5 \\ 0.1 & 0.8 & 0.8 & 0.5 \end{bmatrix} & \begin{bmatrix} 0.3 & 0.2 & 0.8 & 0.5 \\ 0.1 & 0.8 & 0.9 & 0.1 \\ 0.7 & 0.8 & 0.6 & 0.6 \\ 0.9 & 0.1 & 0.2 & 0.4 \end{bmatrix} & \begin{bmatrix} 0.7 & 0.7 & 0.2 & 0.7 \\ 0.2 & 0.2 & 0.5 & 0.8 \\ 0.8 & 1.0 & 0.4 & 0.0 \\ 0.1 & 0.8 & 1.0 & 0.1 \end{bmatrix} \\
\begin{bmatrix} 0.8 & 0.1 & 0.6 & 0.5 \\ 0.8 & 0.4 & 0.9 & 0.3 \\ 0.2 & 0.1 & 0.3 & 0.1 \\ 0.3 & 0.1 & 0.7 & 0.6 \end{bmatrix} & \begin{bmatrix} 0.3 & 0.0 & 0.4 & 0.5 \\ 0.3 & 0.9 & 0.7 & 0.0 \\ 1.0 & 0.8 & 0.7 & 0.5 \\ 1.0 & 0.0 & 0.7 & 0.8 \end{bmatrix} & \begin{bmatrix} 0.1 & 0.4 & 0.6 & 1.0 \\ 0.0 & 0.2 & 0.5 & 0.8 \\ 0.6 & 0.8 & 0.9 & 0.5 \\ 0.2 & 0.5 & 0.1 & 0.5 \end{bmatrix} & \begin{bmatrix} 0.9 & 0.6 & 0.2 & 0.4 \\ 0.6 & 0.3 & 0.5 & 0.9 \\ 0.6 & 0.4 & 0.1 & 0.4 \\ 0.2 & 0.8 & 1.0 & 0.2 \end{bmatrix} \\
\begin{bmatrix} 0.2 & 0.8 & 0.6 & 0.2 \\ 0.5 & 0.1 & 0.5 & 0.0 \\ 0.4 & 0.5 & 0.8 & 0.8 \\ 0.4 & 0.8 & 0.6 & 0.1 \end{bmatrix} & \begin{bmatrix} 0.6 & 0.5 & 0.8 & 0.3 \\ 0.2 & 0.3 & 0.1 & 0.1 \\ 0.2 & 0.3 & 0.9 & 0.1 \\ 0.7 & 0.4 & 0.0 & 0.1 \end{bmatrix} & \begin{bmatrix} 0.0 & 0.2 & 0.4 & 0.6 \\ 0.1 & 0.8 & 0.6 & 1.0 \\ 0.5 & 0.3 & 0.9 & 0.8 \\ 0.2 & 0.4 & 0.5 & 0.4 \end{bmatrix} & \begin{bmatrix} 0.4 & 0.6 & 0.2 & 0.9 \\ 0.1 & 0.0 & 0.7 & 0.4 \\ 0.4 & 0.5 & 0.8 & 0.5 \\ 0.3 & 0.8 & 0.5 & 0.8 \end{bmatrix} \\
\begin{bmatrix} 0.6 & 0.9 & 0.5 & 0.4 \\ 0.7 & 0.5 & 0.0 & 0.6 \\ 0.7 & 0.0 & 0.1 & 0.7 \\ 0.0 & 0.6 & 0.7 & 0.5 \end{bmatrix} & \begin{bmatrix} 0.8 & 0.3 & 0.6 & 0.5 \\ 0.5 & 0.6 & 0.5 & 0.2 \\ 0.1 & 0.2 & 0.0 & 0.2 \\ 0.0 & 0.2 & 0.1 & 0.2 \end{bmatrix} & \begin{bmatrix} 0.6 & 1.0 & 0.7 & 0.9 \\ 0.3 & 0.2 & 0.1 & 1.0 \\ 0.0 & 0.2 & 0.8 & 0.8 \\ 0.9 & 0.9 & 0.2 & 0.2 \end{bmatrix} & \begin{bmatrix} 0.2 & 0.1 & 0.2 & 0.1 \\ 0.7 & 0.9 & 0.8 & 0.2 \\ 0.6 & 0.3 & 0.7 & 0.0 \\ 0.1 & 0.6 & 0.3 & 0.2 \end{bmatrix}
\end{array}$$

E3's Original Data

$$\begin{array}{cccc}
\begin{bmatrix} 0.7 & 0.2 & 0.0 & 0.9 \\ 0.1 & 0.1 & 0.6 & 0.4 \\ 0.6 & 0.3 & 0.1 & 0.5 \\ 0.7 & 0.2 & 0.8 & 0.7 \end{bmatrix} & \begin{bmatrix} 0.8 & 0.6 & 0.2 & 0.6 \\ 0.8 & 0.1 & 0.6 & 0.6 \\ 0.4 & 0.8 & 0.8 & 0.3 \\ 0.0 & 1.0 & 0.1 & 0.3 \end{bmatrix} & \begin{bmatrix} 0.0 & 0.0 & 0.7 & 0.1 \\ 0.4 & 0.1 & 0.8 & 0.8 \\ 0.6 & 0.8 & 0.3 & 0.5 \\ 0.3 & 0.7 & 0.6 & 0.6 \end{bmatrix} & \begin{bmatrix} 0.6 & 0.7 & 0.7 & 0.5 \\ 1.0 & 0.2 & 0.5 & 0.1 \\ 0.1 & 0.7 & 0.1 & 0.1 \\ 1.0 & 0.8 & 0.9 & 0.0 \end{bmatrix} \\
\begin{bmatrix} 0.3 & 0.6 & 0.2 & 0.6 \\ 0.9 & 0.9 & 0.3 & 0.9 \\ 0.5 & 0.4 & 1.0 & 0.4 \\ 0.1 & 0.6 & 0.5 & 0.2 \end{bmatrix} & \begin{bmatrix} 0.6 & 0.2 & 0.7 & 0.8 \\ 0.8 & 0.4 & 0.9 & 0.8 \\ 0.4 & 0.3 & 0.2 & 0.5 \\ 0.1 & 0.8 & 0.8 & 0.5 \end{bmatrix} & \begin{bmatrix} 0.3 & 0.2 & 0.8 & 0.5 \\ 0.1 & 0.8 & 0.9 & 0.1 \\ 0.7 & 0.8 & 0.6 & 0.6 \\ 0.9 & 0.1 & 0.2 & 0.4 \end{bmatrix} & \begin{bmatrix} 0.7 & 0.7 & 0.2 & 0.7 \\ 0.2 & 0.2 & 0.5 & 0.8 \\ 0.8 & 1.0 & 0.4 & 0.0 \\ 0.1 & 0.8 & 1.0 & 0.1 \end{bmatrix} \\
\begin{bmatrix} 0.8 & 0.1 & 0.6 & 0.5 \\ 0.8 & 0.4 & 0.9 & 0.3 \\ 0.2 & 0.1 & 0.3 & 0.1 \\ 0.3 & 0.1 & 0.7 & 0.6 \end{bmatrix} & \begin{bmatrix} 0.3 & 0.0 & 0.4 & 0.5 \\ 0.3 & 0.9 & 0.7 & 0.0 \\ 1.0 & 0.8 & 0.7 & 0.5 \\ 1.0 & 0.0 & 0.7 & 0.8 \end{bmatrix} & \begin{bmatrix} 0.1 & 0.4 & 0.6 & 1.0 \\ 0.0 & 0.2 & 0.5 & 0.8 \\ 0.6 & 0.8 & 0.9 & 0.5 \\ 0.2 & 0.5 & 0.1 & 0.5 \end{bmatrix} & \begin{bmatrix} 0.9 & 0.6 & 0.2 & 0.4 \\ 0.6 & 0.3 & 0.5 & 0.9 \\ 0.6 & 0.4 & 0.1 & 0.4 \\ 0.2 & 0.8 & 1.0 & 0.2 \end{bmatrix} \\
\begin{bmatrix} 0.2 & 0.8 & 0.6 & 0.2 \\ 0.5 & 0.1 & 0.5 & 0.0 \\ 0.4 & 0.5 & 0.8 & 0.8 \\ 0.4 & 0.8 & 0.6 & 0.1 \end{bmatrix} & \begin{bmatrix} 0.6 & 0.5 & 0.8 & 0.3 \\ 0.2 & 0.3 & 0.1 & 0.1 \\ 0.2 & 0.3 & 0.9 & 0.1 \\ 0.7 & 0.4 & 0.0 & 0.1 \end{bmatrix} & \begin{bmatrix} 0.0 & 0.2 & 0.4 & 0.6 \\ 0.1 & 0.8 & 0.6 & 1.0 \\ 0.5 & 0.3 & 0.9 & 0.8 \\ 0.2 & 0.4 & 0.5 & 0.4 \end{bmatrix} & \begin{bmatrix} 0.4 & 0.6 & 0.2 & 0.9 \\ 0.1 & 0.0 & 0.7 & 0.4 \\ 0.4 & 0.5 & 0.8 & 0.5 \\ 0.3 & 0.8 & 0.5 & 0.8 \end{bmatrix} \\
\begin{bmatrix} 0.6 & 0.9 & 0.5 & 0.4 \\ 0.7 & 0.5 & 0.0 & 0.6 \\ 0.7 & 0.0 & 0.1 & 0.7 \\ 0.0 & 0.6 & 0.7 & 0.5 \end{bmatrix} & \begin{bmatrix} 0.8 & 0.3 & 0.6 & 0.5 \\ 0.5 & 0.6 & 0.5 & 0.2 \\ 0.1 & 0.2 & 0.0 & 0.2 \\ 0.0 & 0.2 & 0.1 & 0.2 \end{bmatrix} & \begin{bmatrix} 0.6 & 1.0 & 0.7 & 0.9 \\ 0.3 & 0.2 & 0.1 & 1.0 \\ 0.0 & 0.2 & 0.8 & 0.8 \\ 0.9 & 0.9 & 0.2 & 0.2 \end{bmatrix} & \begin{bmatrix} 0.2 & 0.1 & 0.2 & 0.1 \\ 0.7 & 0.9 & 0.8 & 0.2 \\ 0.6 & 0.3 & 0.7 & 0.0 \\ 0.1 & 0.6 & 0.3 & 0.2 \end{bmatrix}
\end{array}$$

Appendix B

E1's Trust Matrix

$$\begin{pmatrix}
1.0 & 0.7 & 0.4 & 0.3 & 0.4 & 0.9 & 0.7 & 0.2 & 0.9 & 0.5 & 0.9 & 1.0 & 0.1 & 0.9 & 0.3 & 0.6 & 0.5 & 0.9 & 0.9 & 0.2 \\
0.1 & 1.0 & 0.1 & 0.3 & 0.3 & 0.9 & 0.7 & 0.0 & 0.3 & 0.9 & 0.1 & 0.7 & 0.2 & 0.0 & 0.3 & 0.7 & 0.7 & 0.6 & 0.2 & 0.9 \\
1.0 & 0.6 & 1.0 & 0.2 & 0.3 & 0.9 & 0.7 & 0.6 & 1.0 & 0.3 & 0.2 & 0.1 & 0.7 & 0.6 & 0.1 & 0.7 & 0.5 & 0.6 & 0.6 & 0.0 \\
0.9 & 0.4 & 0.8 & 1.0 & 0.3 & 0.0 & 0.8 & 0.0 & 0.4 & 0.2 & 0.4 & 0.1 & 1.0 & 0.4 & 0.4 & 0.1 & 0.2 & 0.0 & 0.3 & 0.0 \\
0.3 & 0.1 & 0.7 & 0.3 & 1.0 & 0.3 & 0.3 & 0.9 & 0.4 & 0.4 & 0.0 & 0.3 & 0.7 & 0.7 & 0.3 & 0.7 & 0.2 & 0.5 & 0.4 & 0.1 \\
0.9 & 0.8 & 0.8 & 0.5 & 0.3 & 1.0 & 0.5 & 1.0 & 0.5 & 0.2 & 0.8 & 0.1 & 0.9 & 1.0 & 0.7 & 0.3 & 0.8 & 0.6 & 0.7 & 0.7 \\
0.8 & 0.2 & 0.3 & 0.8 & 0.5 & 0.8 & 1.0 & 0.9 & 0.9 & 0.8 & 0.9 & 1.0 & 0.8 & 0.3 & 0.4 & 0.7 & 1.0 & 0.6 & 1.0 & 0.2 \\
0.3 & 0.2 & 0.6 & 0.7 & 0.4 & 0.5 & 1.0 & 1.0 & 0.4 & 1.0 & 0.1 & 0.1 & 0.6 & 0.3 & 0.6 & 0.3 & 0.5 & 0.8 & 0.8 & 0.8 \\
0.2 & 0.1 & 0.4 & 0.9 & 1.0 & 0.3 & 0.2 & 0.7 & 1.0 & 0.2 & 0.3 & 0.0 & 0.2 & 0.8 & 0.8 & 1.0 & 0.7 & 0.6 & 0.2 & 0.4 \\
0.7 & 0.5 & 0.1 & 0.2 & 0.2 & 0.1 & 0.4 & 0.9 & 0.2 & 1.0 & 0.6 & 0.8 & 0.4 & 0.1 & 0.3 & 0.5 & 0.4 & 0.2 & 0.7 & 0.7 \\
0.7 & 0.5 & 0.6 & 0.4 & 0.6 & 0.5 & 0.3 & 0.8 & 0.1 & 0.1 & 1.0 & 0.9 & 0.2 & 0.0 & 0.3 & 0.8 & 0.2 & 0.2 & 0.2 & 0.9 \\
0.5 & 0.2 & 0.3 & 0.6 & 0.1 & 0.4 & 0.5 & 0.6 & 0.8 & 0.3 & 0.8 & 1.0 & 0.3 & 0.8 & 0.3 & 0.2 & 0.4 & 0.8 & 0.6 & 0.3 \\
0.8 & 0.4 & 0.8 & 0.4 & 0.2 & 0.1 & 0.9 & 0.7 & 0.8 & 0.5 & 0.7 & 0.1 & 1.0 & 0.1 & 0.7 & 0.3 & 0.9 & 0.2 & 0.9 & 0.6 \\
0.5 & 0.3 & 0.4 & 0.1 & 0.4 & 0.1 & 0.6 & 0.7 & 0.6 & 0.4 & 0.6 & 0.6 & 0.9 & 1.0 & 0.5 & 0.6 & 0.6 & 0.5 & 0.3 & 0.5 \\
0.5 & 0.8 & 0.0 & 0.9 & 0.0 & 0.3 & 0.7 & 0.3 & 0.4 & 1.0 & 0.3 & 0.3 & 0.8 & 0.3 & 1.0 & 0.6 & 0.2 & 0.1 & 0.4 & 0.4 \\
0.6 & 0.9 & 0.1 & 0.7 & 0.1 & 0.3 & 0.7 & 0.4 & 0.6 & 1.0 & 0.2 & 0.1 & 0.3 & 0.0 & 0.2 & 1.0 & 0.1 & 0.6 & 0.9 & 0.8 \\
0.1 & 0.1 & 1.0 & 0.5 & 0.9 & 0.3 & 0.7 & 0.7 & 0.1 & 0.6 & 0.5 & 0.7 & 0.1 & 0.4 & 0.6 & 0.5 & 1.0 & 0.1 & 0.8 & 0.0 \\
0.1 & 0.2 & 0.1 & 0.2 & 0.9 & 0.5 & 0.0 & 0.3 & 0.2 & 0.4 & 0.9 & 0.9 & 0.1 & 0.1 & 0.2 & 0.8 & 0.2 & 1.0 & 0.4 & 0.4 \\
0.4 & 0.9 & 0.2 & 0.5 & 0.4 & 0.1 & 0.8 & 0.1 & 0.7 & 0.3 & 0.2 & 0.0 & 0.6 & 0.6 & 0.3 & 0.8 & 0.5 & 0.2 & 1.0 & 0.1 \\
0.5 & 0.1 & 0.7 & 0.5 & 0.0 & 0.9 & 0.8 & 0.8 & 0.9 & 0.4 & 0.8 & 0.2 & 0.3 & 0.1 & 0.5 & 1.0 & 0.8 & 0.4 & 0.8 & 1.0
\end{pmatrix}$$

E2's Trust Matrix
$\begin{pmatrix} 1.0 & 0.1 & 0.2 & 0.4 & 1.0 & 0.0 & 0.6 & 0.9 & 0.6 & 0.9 & 0.8 & 0.9 & 0.9 & 0.6 & 0.1 & 0.5 & 0.0 & 0.5 & 0.4 & 0.7 \\ 0.9 & 1.0 & 0.1 & 0.6 & 0.8 & 0.6 & 0.6 & 1.0 & 0.6 & 0.2 & 0.3 & 0.3 & 0.4 & 0.9 & 0.2 & 0.8 & 0.9 & 0.2 & 0.0 & 0.8 \\ 0.5 & 0.3 & 1.0 & 0.9 & 1.0 & 1.0 & 0.6 & 0.8 & 0.7 & 0.6 & 1.0 & 0.7 & 0.3 & 0.3 & 0.2 & 0.9 & 0.0 & 0.9 & 0.1 & 1.0 \\ 0.6 & 0.2 & 0.7 & 1.0 & 0.9 & 0.1 & 0.4 & 0.2 & 0.4 & 0.2 & 0.8 & 0.4 & 0.9 & 0.5 & 0.3 & 0.5 & 0.0 & 0.7 & 0.1 & 0.3 \\ 0.7 & 0.7 & 0.2 & 0.3 & 1.0 & 0.3 & 0.5 & 0.5 & 0.3 & 0.9 & 0.8 & 0.9 & 0.4 & 0.8 & 0.2 & 0.4 & 0.9 & 0.3 & 0.3 & 0.6 \\ 0.9 & 0.9 & 0.1 & 0.5 & 0.8 & 1.0 & 0.9 & 0.2 & 1.0 & 0.5 & 0.7 & 0.3 & 0.4 & 1.0 & 0.3 & 0.7 & 0.1 & 0.6 & 0.9 & 0.3 \\ 0.4 & 0.7 & 0.2 & 0.1 & 0.1 & 0.2 & 1.0 & 0.2 & 0.6 & 0.9 & 0.4 & 0.7 & 0.4 & 0.0 & 0.8 & 0.5 & 0.0 & 0.7 & 0.3 & 0.7 \\ 0.7 & 0.5 & 0.0 & 0.6 & 0.0 & 0.2 & 0.4 & 1.0 & 0.8 & 0.7 & 0.7 & 0.8 & 0.3 & 0.4 & 0.6 & 0.6 & 0.6 & 0.0 & 0.5 & 0.2 \\ 0.6 & 0.0 & 0.2 & 0.5 & 0.4 & 0.4 & 0.2 & 0.7 & 1.0 & 0.1 & 0.4 & 0.7 & 0.7 & 0.8 & 0.5 & 0.1 & 0.6 & 0.6 & 0.2 & 0.2 \\ 0.3 & 0.3 & 0.2 & 0.0 & 0.5 & 0.1 & 0.1 & 0.7 & 0.0 & 1.0 & 0.0 & 0.6 & 0.9 & 0.5 & 0.3 & 0.1 & 0.9 & 0.5 & 0.9 & 0.1 \\ 0.6 & 0.7 & 0.4 & 0.7 & 0.4 & 0.3 & 0.8 & 0.2 & 0.7 & 0.4 & 1.0 & 0.1 & 0.5 & 0.3 & 1.0 & 0.3 & 0.5 & 0.4 & 0.4 & 0.7 \\ 0.5 & 0.9 & 0.3 & 0.1 & 1.0 & 0.2 & 0.5 & 0.6 & 0.1 & 0.9 & 0.0 & 1.0 & 1.0 & 0.6 & 0.3 & 0.3 & 0.3 & 0.2 & 0.0 & 0.8 \\ 0.4 & 0.4 & 0.9 & 0.5 & 0.5 & 0.9 & 0.9 & 1.0 & 0.3 & 0.8 & 0.7 & 0.7 & 1.0 & 0.9 & 0.8 & 0.7 & 0.3 & 1.0 & 0.2 & 0.2 \\ 0.0 & 0.4 & 0.2 & 0.5 & 0.3 & 0.9 & 0.5 & 0.1 & 0.3 & 0.7 & 0.4 & 0.7 & 0.2 & 1.0 & 0.8 & 0.9 & 0.5 & 0.9 & 0.3 & 0.6 \\ 0.2 & 0.8 & 0.2 & 0.8 & 0.4 & 1.0 & 0.2 & 0.0 & 0.2 & 0.7 & 0.9 & 0.9 & 0.7 & 0.9 & 1.0 & 0.4 & 0.7 & 0.7 & 0.3 & 0.6 \\ 1.0 & 0.2 & 0.2 & 0.0 & 0.7 & 0.1 & 0.1 & 1.0 & 0.4 & 0.8 & 0.8 & 0.1 & 0.0 & 0.9 & 0.3 & 1.0 & 0.7 & 0.2 & 0.9 & 0.9 \\ 0.4 & 0.1 & 0.1 & 0.3 & 0.6 & 0.8 & 0.5 & 0.6 & 0.0 & 0.2 & 0.0 & 0.4 & 0.5 & 0.9 & 0.6 & 0.3 & 1.0 & 0.8 & 0.9 & 0.1 \\ 0.5 & 0.9 & 0.2 & 0.9 & 0.8 & 0.5 & 0.4 & 0.2 & 0.0 & 0.4 & 0.9 & 0.1 & 0.3 & 1.0 & 0.7 & 0.9 & 0.8 & 1.0 & 0.7 & 0.4 \\ 0.6 & 0.3 & 0.7 & 0.6 & 0.4 & 0.0 & 0.2 & 0.3 & 0.1 & 0.7 & 0.7 & 0.2 & 0.9 & 1.0 & 0.4 & 0.2 & 0.3 & 0.7 & 1.0 & 0.4 \\ 0.6 & 0.3 & 0.7 & 0.6 & 0.4 & 0.0 & 0.2 & 0.3 & 0.1 & 0.7 & 0.7 & 0.2 & 0.9 & 1.0 & 0.4 & 0.2 & 0.3 & 0.7 & 0.0 & 1.0 \end{pmatrix}$

E3's Trust Matrix
$\begin{pmatrix} 1.0 & 0.5 & 0.3 & 0.8 & 0.2 & 0.5 & 0.1 & 0.2 & 0.6 & 0.9 & 0.5 & 0.2 & 0.6 & 0.4 & 0.5 & 0.7 & 0.9 & 0.0 & 0.7 & 0.5 \\ 0.0 & 1.0 & 0.9 & 1.0 & 0.1 & 0.8 & 0.8 & 0.3 & 0.7 & 0.5 & 0.4 & 0.2 & 0.1 & 0.8 & 0.9 & 0.8 & 0.3 & 0.1 & 0.7 & 0.0 \\ 0.9 & 0.5 & 1.0 & 0.2 & 0.0 & 0.7 & 0.9 & 0.5 & 0.7 & 0.0 & 1.0 & 0.4 & 0.8 & 0.9 & 1.0 & 0.2 & 0.3 & 0.6 & 0.5 & 0.6 \\ 1.0 & 0.3 & 0.0 & 1.0 & 0.2 & 0.5 & 0.6 & 0.9 & 0.4 & 0.4 & 0.5 & 0.2 & 0.5 & 0.2 & 0.1 & 0.4 & 0.2 & 0.8 & 0.1 & 0.2 \\ 1.0 & 0.8 & 0.1 & 0.5 & 1.0 & 0.6 & 0.0 & 0.8 & 0.2 & 0.9 & 0.6 & 0.1 & 0.5 & 0.2 & 0.2 & 0.1 & 0.6 & 0.4 & 0.8 & 0.5 \\ 0.5 & 0.1 & 0.0 & 0.5 & 0.9 & 1.0 & 0.6 & 0.1 & 0.2 & 0.8 & 1.0 & 0.9 & 0.0 & 0.3 & 0.1 & 0.7 & 0.9 & 0.3 & 0.2 & 1.0 \\ 0.2 & 0.1 & 0.7 & 0.7 & 0.9 & 0.1 & 1.0 & 0.5 & 0.8 & 0.5 & 0.5 & 0.6 & 0.9 & 0.9 & 0.9 & 0.8 & 0.2 & 1.0 & 0.2 & 0.6 \\ 0.9 & 0.9 & 0.8 & 0.6 & 0.2 & 0.5 & 0.6 & 1.0 & 0.7 & 0.1 & 0.1 & 0.5 & 0.7 & 0.3 & 0.8 & 0.7 & 0.5 & 0.4 & 0.4 & 1.0 \\ 0.2 & 0.5 & 0.8 & 0.0 & 0.6 & 1.0 & 0.7 & 0.1 & 1.0 & 0.4 & 0.6 & 0.3 & 0.5 & 0.8 & 0.9 & 0.7 & 0.4 & 0.9 & 0.5 & 0.4 \\ 0.9 & 0.1 & 0.1 & 0.0 & 0.7 & 0.2 & 0.4 & 0.5 & 0.5 & 1.0 & 0.9 & 0.7 & 0.9 & 0.5 & 0.3 & 0.1 & 0.6 & 0.6 & 0.2 & 0.2 \\ 0.6 & 0.6 & 0.2 & 0.4 & 0.1 & 0.4 & 0.1 & 0.6 & 0.1 & 0.7 & 1.0 & 0.5 & 0.2 & 0.5 & 0.7 & 0.5 & 0.9 & 0.8 & 0.0 & 0.4 \\ 0.6 & 0.8 & 0.2 & 0.0 & 0.2 & 0.3 & 0.4 & 0.5 & 0.7 & 0.0 & 0.8 & 1.0 & 1.0 & 0.4 & 0.8 & 0.3 & 0.1 & 0.9 & 0.5 & 0.7 \\ 0.3 & 0.3 & 0.6 & 0.7 & 0.4 & 0.5 & 0.5 & 0.6 & 0.5 & 0.8 & 0.6 & 0.8 & 1.0 & 0.6 & 0.8 & 0.7 & 1.0 & 1.0 & 0.6 & 0.3 \\ 0.4 & 0.1 & 0.2 & 0.6 & 0.6 & 0.3 & 0.7 & 0.6 & 0.8 & 0.8 & 0.4 & 0.4 & 0.4 & 1.0 & 0.2 & 0.6 & 0.9 & 0.8 & 0.2 & 1.0 \\ 0.5 & 0.4 & 0.3 & 0.6 & 0.9 & 0.5 & 0.3 & 0.7 & 0.6 & 0.1 & 0.2 & 0.3 & 0.6 & 0.7 & 1.0 & 0.1 & 0.4 & 0.2 & 0.9 & 0.1 \\ 0.6 & 0.0 & 0.5 & 0.1 & 0.6 & 0.3 & 0.8 & 0.6 & 0.8 & 0.9 & 0.5 & 0.4 & 1.0 & 0.9 & 0.0 & 1.0 & 0.4 & 0.4 & 0.7 & 0.7 \\ 0.6 & 0.3 & 0.5 & 0.9 & 0.3 & 1.0 & 0.8 & 0.8 & 0.8 & 0.4 & 0.7 & 0.4 & 0.3 & 0.6 & 0.2 & 0.1 & 1.0 & 0.5 & 0.9 & 0.5 \\ 0.1 & 0.8 & 0.1 & 0.4 & 0.2 & 0.5 & 0.8 & 0.6 & 0.1 & 0.2 & 0.3 & 0.7 & 0.6 & 0.2 & 1.0 & 0.5 & 0.7 & 1.0 & 0.0 & 0.6 \\ 0.2 & 0.1 & 0.1 & 0.3 & 0.1 & 0.2 & 0.9 & 0.8 & 0.5 & 0.8 & 0.2 & 0.1 & 0.9 & 0.1 & 0.6 & 0.1 & 0.2 & 0.1 & 1.0 & 0.6 \\ 0.5 & 0.1 & 0.7 & 0.5 & 0.0 & 0.9 & 0.8 & 0.8 & 0.9 & 0.4 & 0.8 & 0.2 & 0.3 & 0.1 & 0.5 & 1.0 & 0.8 & 0.4 & 0.8 & 1.0 \end{pmatrix}$

Appendix C

E1	E2	E3
$\begin{array}{c} \begin{bmatrix} 0.1400 & 0.4000 & 0.5000 & 0.5600 \\ 0.2200 & 0.2400 & 0.6200 & 0.6000 \\ 0.5000 & 0.5800 & 0.7400 & 0.6200 \\ 0.2800 & 0.6400 & 0.4600 & 0.4800 \end{bmatrix} \\ \begin{bmatrix} 0.6800 & 0.5200 & 0.4200 & 0.6000 \\ 0.6400 & 0.3000 & 0.5000 & 0.5600 \\ 0.4800 & 0.3400 & 0.1200 & 0.4400 \\ 0.4000 & 0.6400 & 0.8400 & 0.3800 \end{bmatrix} \\ \begin{bmatrix} 0.7000 & 0.4750 & 0.6750 & 0.5500 \\ 0.4500 & 0.3750 & 0.4000 & 0.4000 \\ 0.1250 & 0.2000 & 0.5000 & 0.3000 \\ 0.4750 & 0.4000 & 0.2500 & 0.2750 \end{bmatrix} \\ \begin{bmatrix} 0.5000 & 0.5000 & 0.2000 & 0.5000 \\ 0.6500 & 0.5250 & 0.5500 & 0.6250 \\ 0.5750 & 0.6250 & 0.7250 & 0.1750 \\ 0.0750 & 0.7500 & 0.4750 & 0.2000 \end{bmatrix} \\ \begin{bmatrix} 0.3000 & 0.1000 & 0.6000 & 0.5000 \\ 0.2000 & 0.8000 & 0.8000 & 0.0500 \\ 0.8500 & 0.8000 & 0.6500 & 0.5500 \\ 0.9500 & 0.0500 & 0.4500 & 0.6000 \end{bmatrix} \end{array}$	$\begin{array}{c} \begin{bmatrix} 0.5208 & 0.6100 & 0.5770 & 0.6664 \\ 0.4119 & 0.6472 & 0.3852 & 0.4623 \\ 0.6163 & 0.3272 & 0.7855 & 0.3519 \\ 0.3848 & 0.3925 & 0.3221 & 0.4622 \end{bmatrix} \\ \begin{bmatrix} 0.5250 & 0.5500 & 0.8500 & 0.2500 \\ 0.7750 & 0.2000 & 0.2250 & 0.6500 \\ 0.1250 & 0.7000 & 0.7000 & 0.5000 \\ 0.7750 & 0.3500 & 0.4250 & 0.2000 \end{bmatrix} \\ \begin{bmatrix} 0.2500 & 0.6250 & 0.4250 & 0.2750 \\ 0.3500 & 0.3750 & 0.2250 & 0.8500 \\ 0.4250 & 0.6500 & 0.5500 & 0.3750 \\ 0.3750 & 0.7750 & 0.6500 & 0.2000 \end{bmatrix} \\ \begin{bmatrix} 0.3750 & 0.9250 & 0.4250 & 0.3750 \\ 0.3500 & 0.6250 & 0.7750 & 0.6500 \\ 0.3750 & 0.5500 & 0.2500 & 0.7000 \\ 0.4750 & 0.2000 & 0.4750 & 0.4750 \end{bmatrix} \end{array}$	$\begin{array}{c} \begin{bmatrix} 0.3350 & 0.3810 & 0.4419 & 0.4493 \\ 0.2349 & 0.5938 & 0.4826 & 0.7519 \\ 0.4407 & 0.4557 & 0.5697 & 0.5746 \\ 0.4463 & 0.2744 & 0.4487 & 0.5028 \end{bmatrix} \\ \begin{bmatrix} 0.3667 & 0.7167 & 0.5833 & 0.4500 \\ 0.7833 & 0.6667 & 0.4333 & 0.3167 \\ 0.4667 & 0.4167 & 0.2500 & 0.4167 \\ 0.6833 & 0.6176 & 0.4167 & 0.5667 \end{bmatrix} \\ \begin{bmatrix} 0.7546 & 0.7082 & 0.4576 & 0.6671 \\ 0.3179 & 0.5563 & 0.6815 & 0.3947 \\ 0.3635 & 0.6030 & 0.3972 & 0.5008 \\ 0.3486 & 0.7718 & 0.1798 & 0.4852 \end{bmatrix} \end{array}$

References

1. Freire, Y.; Gomez Sanchez, M.; Sanchez Ituarte, J.; Frias Senande, M.; Diaz-Flores Garcia, V.; Suarez, A. Social media impact on students' decision-making regarding aesthetic dental treatments based on cross-sectional survey data. *Sci. Rep.* **2024**, *14*, 21626. [CrossRef] [PubMed]
2. Bokolo, B.G.; Liu, Q. Artificial intelligence in social media forensics: A comprehensive survey and analysis. *Electronics* **2024**, *13*, 1671. [CrossRef]
3. Hsieh, Y.; Lu, L.; Ku, Y. Review evaluation for hotel recommendation. *Electronics* **2023**, *12*, 4673. [CrossRef]
4. Li, H.; Duan, J.; Sun, Q. A strategy adaptive evolution approach based on the public goods game. *Electronics* **2022**, *11*, 2006. [CrossRef]
5. Yang, J.; Xiu, P.; Sun, L.; Ying, L.; Muthu, B. Social media data analytics for business decision making system to competitive analysis. *Inform. Process. Manag.* **2022**, *59*, 102751. [CrossRef]
6. Alonso, S.; Perez, I.J; Cabrerizo, F.J.; Herrera-Viedma, E. A linguistic consensus model for Web 2.0 communities. *Appl. Soft Comput.* **2013**, *13*, 149–157. [CrossRef]
7. Freire, M.; Antunes, F.; Costa, J.P. Enhancing decision-making support by mining social media data with social network analysis. *Soc. Netw. Anal. Min.* **2023**, *13*, 86. [CrossRef] [PubMed]
8. Du, Z.; Yu, S.; Luo, H.; Lin, X. Consensus convergence in large-group social network environment: Coordination between trust relationship and opinion similarity. *Knowl.-Based Syst.* **2021**, *217*, 106828. [CrossRef]
9. Li, Y.; Kou, G.; Li, G.; Peng, Y. Consensus reaching process in large-scale group decision making based on bounded confidence and social network. *Eur. J. Oper. Res.* **2022**, *303*, 790–802. [CrossRef]
10. Qin, J.; Wang, D.; Liang, Y. Social network-driven bi-level minimum cost consensus model for large-scale group decision-making: A perspective of structural holes. *Inform. Sci.* **2023**, *649*, 119678. [CrossRef]
11. Herrera-Viedma, E.; Martinez, L.; Mata, F.; Chiclana, F. A consensus support system model for group decision-making problems with multigranular linguistic preference relations. *IEEE Trans. Neural Netw.* **2005**, *13*, 644–658. [CrossRef]
12. Amin, E.; Choi, J.; Choi, G. Advanced community identification model for social networks. *CMC-Comput. Mater. Contin.* **2024**, *69*, 1687–1707. [CrossRef]
13. Wang, Y.; Song, H.; Dutta, B.; Garcia-Zamora, D.; Martinez, L. Consensus reaching in LSGDM: Overlapping community detection and bounded confidence-driven feedback mechanism. *Inform. Sci.* **2024**, *679*, 121104. [CrossRef]
14. Zhou, L.; You, X.; Zhao, S.; You, Z. A geometric-based lsgdm method for tourism project decision optimization with trust–distrust relationships. *Entropy* **2022**, *24*, 588. [CrossRef]
15. Amiri, M.H.; Mehrabi Hashjin, N.; Montazeri, M.; Mirjalili, S.; Khodadadi, N. Hippopotamus optimization algorithm: A novel nature-inspired optimization algorithm. *Sci. Rep.* **2024**, *14*, 5032. [CrossRef]
16. Xu, R.; Wunsch, D. Survey of clustering algorithms. *IEEE Trans. Neural Netw.* **2005**, *16*, 645–678. [CrossRef] [PubMed]
17. Zhang, C.; Hou, H.; Sangaiah, A.K.; Li, D.; Li, W. Enhancing high temperature prediction via six-fold strategy consensus-reaching processes: A case study using FY-3E spatio-temporal remote sensing satellite data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 16377–16391. [CrossRef]
18. Zhang, C.; Wang, B.; Li, W.; Li, D. Incorporating artificial intelligence in detecting crop diseases: Agricultural decision-making based on group consensus model with MULTIMOORA and evidence theory. *Crop Prot.* **2024**, *179*, 106632. [CrossRef]
19. Yuan, X.; Xu, T.; He, S.; Zhang, C. An online review data-driven fuzzy large-scale group decision-making method based on dual fine-tuning. *Electronics* **2024**, *13*, 2702. [CrossRef]
20. Teng, F.; Liu, X.; Liu, P. Overlapping community-driven dynamic consensus reaching model of large-scale group decision making in social network. *Inform. Sci.* **2024**, *685*, 121290. [CrossRef]
21. Zhu, J.; Ma, X.; Martinez, L.; Zhan, J. A probabilistic linguistic three-way decision method with regret theory via fuzzy C-means clustering algorithm. *IEEE Trans. Fuzzy Syst.* **2023**, *31*, 2821–2835. [CrossRef]
22. Zhang, R.; Ma, X.; Zhan, J.; Yao, Y. 3WC-D: A feature distribution-based adaptive three-way clustering method. *Appl. Intell.* **2023**, *53*, 15561–15579. [CrossRef]
23. Song, H.; Dutta, B.; Garcia-Zamora, D.; Wang, Y.; Martinez, L. Managing non-cooperative behaviors in consensus reaching process: A novel multi-stage linguistic LSGDM framework. *Expert Syst. Appl.* **2024**, *240*, 122555. [CrossRef]
24. Zhou, H.; Li, W.; Zhang, C.; Zhan, T. Dynamic maintenance of updating rough approximations in interval-valued ordered decision systems. *Appl. Intell.* **2023**, *53*, 22161–22178. [CrossRef]
25. Shen, Y.; Ma, X.; Zhang, H.; Zhan, J. Fusion social network and regret theory for a consensus model with minority opinions in large-scale group decision making. *Inform. Fusion* **2024**, *112*, 102548. [CrossRef]
26. Li, X.; Liao, H. A large-scale group decision making method based on spatial information aggregation and empathetic relationships of experts. *Inform. Fusion* **2023**, *632*, 503–515. [CrossRef]
27. Harshavardhan, A; Boyapati, P.; Neelakandan, S; Abdul-Rasheed Akeji, A.A.; Singh Pundir, A.K.; Walia, R. LSGDM with Biogeography-Based Optimization (BBO) model for healthcare applications. *J. Healthc. Eng.* **2022**, *13*, 149–157. [CrossRef] [PubMed]
28. Guo, L.; Zhan, J.; Xu, Z.; Alcantud, J.C.R. A consensus measure-based three-way clustering method for fuzzy large group decision making. *Inform. Sci.* **2023**, *632*, 144–163. [CrossRef]

29. Guo, L.; Zhan, J.; Kou, G.; Martinez, L. A sentiment analysis and dual trust relationship-based approach to large-scale group decision-making for online reviews: A case study of China Eastern Airlines. *Inform. Sci.* **2024**, *667*, 120515. [CrossRef]
30. Hewitt, M.; Ortmann, J.; Rei, W. Decision-based scenario clustering for decision-making under uncertainty. *Ann. Oper. Res.* **2022**, *315*, 747–771. [CrossRef]
31. Maulik, U.; Bandyopadhyay, S. Genetic algorithm-based clustering technique. *Pattern Recogn.* **2000**, *33*, 1455–1465. [CrossRef]
32. Inkaya, T.; Kayaligil, S.; Ozdemirel, N.E. Ant colony optimization based clustering methodology. *Appl. Soft Comput.* **2015**, *28*, 301–311. [CrossRef]
33. Yu, X.-W.; Yu, H.; Liu, Y.; Xiao, R.-R. A clustering routing algorithm based on wolf pack algorithm for heterogeneous wireless sensor networks. *Comput. Netw.* **2020**, *167*, 106994.
34. Zeebaree, D.Q.; Haron, H.; Abdulazeez, A.M.; Zeebaree, S. Combination of K-means clustering with genetic algorithm: A review. *Inter. J. Appl. Eng. Res.* **2017**, *12*, 14238–14245.
35. Zhou, X.; Li, S.; Wei, C. Consensus reaching process for group decision-making based on trust network and ordinal consensus measure. *Inform. Fusion* **2024**, *101*, 101969. [CrossRef]
36. Hua, Z.; Jing, X.; Martinez, L. Consensus reaching for social network group decision making with ELICIT information: A perspective from the complex network. *Inform. Sci.* **2023**, *627*, 71–96. [CrossRef]
37. Zha, Q.; Cai, J.; Gu, J.; Liu, G. Information learning-driven consensus reaching process in group decision-making with bounded rationality and imperfect information: China's urban renewal negotiation. *Appl. Intell.* **2023**, *53*, 10444–10458. [CrossRef]
38. Wang, Y.; Zhan, J.; Zhang, C.; Deveci, M. Considering personalized individual semantics with ordinal and cardinal consensus reaching processes via three-way decision and regret theory. *Appl. Soft Comput.* **2024**, 111851. [CrossRef]
39. Du, Z.; Yu, S.; Cai, C. Constrained community detection and multistage multicost consensus in social network large-scale decision-making. *IEEE T. Comput. Soc. Syst.* **2023**, *11*, 997–1012. [CrossRef]
40. Gardikiotis A. Minority influence. *Soc. Personal. Psychol. Compass* **2011**, *5*, 679–693. [CrossRef]
41. Yang, G.; Wang, X.; Ding, R.; Lin, S.; Lou, Q.; Herrera-Viedma, E. Managing non-cooperative behaviors in large-scale group decision making based on trust relationships and confidence levels of decision makers. *Inform. Fusion* **2023**, *97*, 101820. [CrossRef]
42. Meng, F.; Tang, J.; An, Q. Cooperative game based two-stage consensus adjustment mechanism for large-scale group decision making. *Omega* **2023**, *117*, 102842. [CrossRef]
43. Xu, X.; Du, Z.; Chen, X. Consensus model for multi-criteria large-group emergency decision making considering non-cooperative behaviors and minority opinions. *Decis. Support Syst.* **2015**, *79*, 150–160. [CrossRef]
44. Pingle M. Submitting to authority: Its effect on decision-making. *J. Econ. Psychol.* **1997**, *18*, 45–68. [CrossRef]
45. Liu, Z.; Wang, W.; Zhang, X.; Liu, P. A dynamic dual-trust network-based consensus model for individual non-cooperative behaviour management in group decision-making. *Inform. Sci.* **2024**, *674*, 120750. [CrossRef]
46. Fu, Y.; Liang, D.; Xu, Z.; Duan, W. Exploiting game equilibrium mechanisms towards social trust-based group consensus reaching. *Inform. Fusion* **2024**, *112*, 102558. [CrossRef]
47. Liu, Y.; Zhang, X.; Rodriguez, R.M.; Martinez, L. Managing non-cooperative behaviors in consensus reaching processes: A comprehensive self-management weight generation mechanism. *Appl. Intell.* **2024**, *54*, 2673–2702. [CrossRef]
48. Li, W.; Deng, C.; Pedrycz, W.; Castillo, O.; Zhang, C.; Zhan, T. Double-quantitative feature selection approach for multi-granularity ordered decision systems. *IEEE Trans. Artif. Intell.* **2023**, *5*, 2385–2396. [CrossRef]
49. Li, W.; Zhan, T. Multi-granularity probabilistic rough fuzzy sets for interval-valued fuzzy decision systems. *Int. J. Fuzzy Syst.* **2023**, *25*, 3061–3073. [CrossRef]
50. You, X.; Hou, F.; Chiclana F. Consensus reaching process with noncooperative behaviors in large-scale group social network environment. *Appl. Soft Comput.* **2023**, *144*, 110454. [CrossRef]
51. Zhou, M.; Zhou, Y.J.; Liu, X.B.; Wu, J.; Fujita, H.; Herrera-Viedma, E. An adaptive two-stage consensus reaching process based on heterogeneous judgments and social relations for large-scale group decision making. *Inform. Sci.* **2023**, *644*, 119280. [CrossRef]
52. Liu, X.; Yang, Y.; Jiang, J. The behavioral topsis based on prospect theory and regret theory. *Int. J. Inf. Tech.* **2023**, *22*, 1591–1615. [CrossRef]
53. Bisht, G.; Pal, A.K. Prospect-regret theory based decision-making approach for incomplete probabilistic hesitant fuzzy environment: An application to medical field. *Expert Syst. Appl.* **2024**, *250*, 123906. [CrossRef]
54. Deng, J.; Zhan, J.; Ding, W.; Liu, P.; Pedrycz, W. A novel prospect-theory-based three-way decision methodology in multi-scale information systems. *Artif. Intel. Rev.* **2023**, *56*, 6591–6625. [CrossRef]
55. Zhang, N.; Zhou, Y.; Liu, J.; Wei, G. VIKOR method for Pythagorean hesitant fuzzy multi-attribute decision-making based on regret theory. *Eng. Appl. Artif. Intel.* **2023**, *126*, 106857. [CrossRef]
56. Wang, Y.; Zhan, J.; Zhang, C.; Xu, Z. A group consensus model with prospect theory under probabilistic linguistic term sets. *Inform. Sci.* **2024**, *653*, 119800. [CrossRef]
57. Backhaus, J. The pareto principle. *Anal. Krit.* **1980**, *2*, 146–171. [CrossRef]
58. Kritik, J. The Pareto principle in organizational decision making. *Mange. Decis.* **2002**, *40*, 729–733.
59. Zhan, T.; Li, W.; Zhang, C. Discrete impulsive signal observer for fractional order control systems and its consumer electronic circuit application. *IEEE Trans. Consum. Electron.* **2023**, *70*, 341–3348. [CrossRef]

60. Chen, G.; Wei, L.; Fu, J.; Li, C.; Zhao, G. A large group emergency decision-making method based on uncertain linguistic cloud similarity method. *Math. Comput. Appl.* **2022**, *27*, 101. [CrossRef]
61. Liang, X.; Guo, J.; Liu, P. A large-scale group decision-making model with no consensus threshold based on social network analysis. *Inform. Sci.* **2022**, *612*, 361–383. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Attentional Convolutional Neural Network Based on Distinction Enhancement and Information Fusion for FDIA Detection in Power Systems

Chuan Luo [1], Zhiyong Zhou [2,*], Rui Jiang [1] and Bochuan Zheng [3,*]

[1] School of Computer Science, China West Normal University, Nanchong 637009, China; rotry@stu.cwnu.edu.cn (C.L.); 212022083500001@stu.cwnu.edu.cn (R.J.)
[2] Education and Information Technology Center, China West Normal University, Nanchong 637009, China
[3] Institute of Artificial Intelligence, China West Normal University, Nanchong 637009, China
[*] Correspondence: zhouzy@cwnu.edu.cn (Z.Z.); zhengbc@cwnu.edu.cn (B.Z.)

Abstract: A false data injection attack (FDIA) is one of the major threats to power systems, and identifying false data is critical to the stable operation of power systems. However, false data that closely resemble normal data hinder the accuracy of existing detection methods, and their performance further declines when exposed to ambient noise. To address these challenges, this paper proposes an attentional convolutional neural network based on distinction enhancement and information fusion (DEIF-ACNN) for FDIA detection. Firstly, by minimizing the loss of reconstruction and discrimination, this paper designed an autoencoder with a discriminator for normal data (NAE), which had the characteristic of producing a small loss for normal data. Secondly, the trained NAE is utilized to compute the feature correlation matrix between the original and reconstructed data to enhance the distinction between normal and false data. Finally, to enhance feature extraction and suppress ambient noise interference in detection, DEIF-ACNN incorporates a convolutional block attention module (CBAM) to emphasize key feature channels and highlight crucial regions in the feature matrix. Experimental results show that DEIF-ACNN outperforms other FDIA detection methods on IEEE-9, IEEE-14, and IEEE-118 bus power systems, achieving an accuracy of 99.22%, 99.83%, and 100.00%, respectively. In addition, the method exhibits the best robustness under different noise environments, and its accuracy is maintained at about 80%.

Keywords: power system; false data injection attack; CNN; autoencoder; attention mechanism

Citation: Luo, C.; Zhou, Z.; Jiang, R.; Zheng, B. Attentional Convolutional Neural Network Based on Distinction Enhancement and Information Fusion for FDIA Detection in Power Systems. *Electronics* **2024**, *13*, 4862. https://doi.org/10.3390/electronics13244862

Academic Editor: Ahmed Abu-Siada

Received: 12 November 2024
Revised: 6 December 2024
Accepted: 9 December 2024
Published: 10 December 2024

1. Introduction

The advancement of information technology in power systems has significantly enhanced convenience; however, it has also increased the vulnerability of power systems to various cyberattacks [1–4]. Among these risks, the false data injection attack (FDIA) is one of the most challenging cyberattacks in power systems [5]. Figure 1 illustrates the process of the false data injection attack (FDIA). Phasor measurement units (PMUs) are essential for the measurement, control, and protection of power grids. However, installing PMUs at every bus in a power system is cost-prohibitive, leading to the need for partial PMU placement that maintains system observability with a minimal number of units [6,7]. A downside of this cost-effective strategy is that it makes the system more susceptible to FDIAs. Attackers utilize the system topology structure and prior knowledge to tamper with the data transmitted by PMUs, effectively bypassing the bad data detection mechanisms [8,9] and causing erroneous decision-making [10]. Such attacks can result in far-reaching consequences, including widespread power outages [11], as well as severe disruptions like traffic congestion [12] and substantial market losses [13], posing significant risks to both society and the economy.

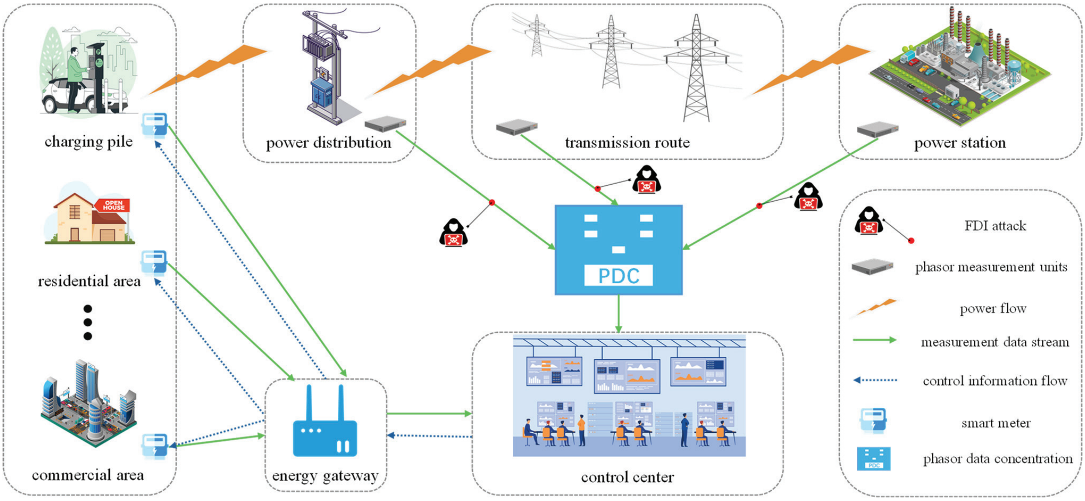


Figure 1. The FDIA of power systems.

Currently, FIDA detection methods are mainly categorized into model-based and data-driven approaches [14]. Model-based detection methods rely on the physical model and prior knowledge of power systems. Therefore, these methods have low scalability [15]. Data-driven detection methods rely on historical measurement data to enable real-time or near real-time inspection through rapid data processing and analysis [16]. However, with the continuous evolution of attack methods, the false data generated by attackers becomes extremely like normal data in terms of features, which may lead to false alarms or omissions in detection results. In addition, since PMUs are easily disturbed by transmission delays or environmental factors, the measurement data are often contaminated by noise, further reducing the detection efficiency. To address these challenges, the DEIF-ACNN model is proposed to identify FDIA. The proposed method improves the detection efficiency of identifying FDIA by using advanced feature extraction techniques driven by deep learning models to learn and adapt subtle patterns in the data.

Firstly, DEIF-ACNN is used to enhance the distinction of data by constructing an NAE that produces a smaller loss for normal data inputs and a larger loss for false data inputs. Subsequently, DEIF-ACNN fuses the information from the original data and the enhanced data to generate a feature correlation matrix, thus preserving the original information and adapting it to subsequent detection tasks. Finally, DEIF-ACNN introduces a CBAM to compute the attention weight matrix from both channel and spatial perspectives, which enhances the key feature channels and highlights the important regions while suppressing the noise interference, thus improving the efficiency of false data detection. The main contributions of this paper are as follows:

1. A data-driven FDIA detection method is proposed, which utilizes an autoencoder with a discriminator for normal data to enhance the distinction between normal data and false data to improve the model detection performance.
2. To suppress the interference of noise on the detection results, the attention mechanism is added to the convolutional neural network to improve the robustness of the detection model.
3. The method is completely data-driven and has excellent scalability without considering the physical model and topology of the power system.
4. The proposed method is validated by simulation on IEEE-9, IEEE-14, and IEEE-118 bus systems, considering the interference of measured data in real environments.

The rest of the paper is organized as follows: Section 2 describes the related work on FDIA detection techniques in recent years; Section 3 describes the fundamentals of power system state estimation and FDIA attacks; Section 4 describes the proposed methodology of this paper in detail; Section 5 demonstrates the simulation results and performance evaluations on three IEEE standard test systems; and Section 6 concludes the work of this paper.

2. Related Work

The current detection methods can be categorized into model-based and data-driven detection methods.

2.1. Detection Methods Based on Model

The authors of [17] improved the detection process by focusing on preventing outages rather than just the denial of attacks, and evaluated the impact of cyberattacks on the PMU state estimation process. At the beginning of the study, Duan J et al. used a weighted least squares method to identify FDIAs in power systems, using state estimation to compare with actual measurements in order to detect deviations [18]. Li B's team proposed a proactive defense approach (PAMA) to protect the grid configuration information and raw measurement data by encrypting the raw measurement data while protecting the grid configuration information [19]. M. G. Kallitsis et al. proposed an adaptive statistical detection method that uses measurements from trusted nodes to detect other anomalous nodes [20]. In order to improve the real-time performance of FDIA detection, R. Moslemi et al. exploited the near-singular sparsity of the power system to develop an efficient framework for solving the associated maximum likelihood (ML) estimation problem, and then decomposed the FDIA detection method into several local ML estimation problems [21]. With the introduction of dynamic modeling, K. Manandhar et al. proposed a detection method based on the Kalman Filter (KF); their team studied the mathematical model of the power system and used the KF to estimate the variables of the various state processes in the model, and then the estimation results and the system data were fed back to the detector to obtain the detection results [22]. The GoDec algorithm was proposed by B. Li et al.; their approach relies on the low-rank property of the measurement matrix and the sparsity of the attack matrix to reformulate FDIA detection as a matrix separation problem, and the approach was shown to be capable of handling measurement noise and applicable to large-scale attacks [23]. S. Li et al. proposed a generalized likelihood ratio-based sequence detector to solve the problems of the low robustness of the detector and computational inefficiency by considering the sequential detection of FDIA [24].

2.2. Detection Methods Based on Data-Driven Approaches

With the advent of machine learning and big data technologies, data-driven approaches have emerged as a promising alternative for FDIA detection. Y. He et al. proposed a deep belief network (DBN) based on a restricted Boltzmann machine for detecting damaged data in DC power systems [25]. Foroutan S A et al. proposed a semi-supervised method based on Gaussian Mixture Models; they used Gaussian Mixture Models to learn the features of the positive samples and select the appropriate thresholds on the mixed dataset to evaluate the unlabeled data [26]. Similarly, C. Wang et al. used automatic coding to learn the intrinsic features of normal data and predicted positive anomalies by calculating the reconstruction loss, and their approach effectively overcame the challenge of unbalanced data samples in power systems [27]. A convolutional neural network (CNN) is widely used in pattern recognition image processing, and the ability of the CNN to extract sample features has become a very promising algorithm for FDIA detection. Min Lu et al. constructed an FDIA model based on an improved CNN, which extracts spatial–temporal features of the data by adding a gate recursion unit (GRU) before the fully connected layer of the CNN to achieve efficient, real-time FDIA detection [28].

3. State Estimation and FDIA

This section will introduce the state estimation and BDD technology of power systems, sort out the FDIA execution process, and analyze the similarity problem between normal and false data.

3.1. State Estimation

The state estimation problem in power systems involves estimating the state variables (e.g., bus voltage magnitudes and phase angles) from measured data. It is typically formulated as a nonlinear optimization problem, where the goal is to minimize the error between the measured data and the predicted values, derived from the state variables. The state estimation process can be represented as follows [10]:

$$z = h(x) + e \tag{1}$$

where z denotes the measurement data, usually the bus voltage, the bus active and reactive power injections, and the branch active and reactive power flows of the power system [11]; x denotes the state vector; e denotes the measurement noise obeying a Gaussian distribution $e \sim N(0, \sigma^2)$; $h(\cdot)$ denotes the functional dependence between the measurements and the state variables, which is usually determined by the system parameters and topology. The nonlinear optimization involved here is typically unconstrained, meaning that there are no explicit physical constraints (e.g., voltage limits or power flow boundaries) enforced during the optimization process. The objective is to minimize the difference between the actual measurements z and the estimated measurements $h(x)$ represented by the following cost function:

$$J(x) = \frac{1}{2}(z - h(x))^T W (z - h(x)) \tag{2}$$

where $W = diag(\sigma_i^{-2}, 0)$, σ_i^2 is the variance of the measurement error of the *i-th* meter. The optimization problem is solved iteratively using methods like gradient descent or the Gauss–Newton method, where the focus is solely on minimizing the measurement error without considering additional physical constraints.

Due to the large amount of computation involved in solving the state estimation through multiple iterations, Equation (1) is simplified to the following:

$$z = Hx + e \tag{3}$$

where $H = \frac{\partial h(x)}{\partial x}$ is an $M \times N$ Jacobi matrix related to the system parameters and topology. Similarly, solving for the state variable estimates $\hat{x}$ can be performed in the following way:

$$\begin{aligned} \hat{x} &= \min J(x) \\ &= (z - Hx)^T W (z - Hx) \end{aligned} \tag{4}$$

The solution of the state estimation is obtained as follows:

$$\hat{x} = \left(H^T W H\right)^{-1} H^T W z \tag{5}$$

3.2. BDD Technology

During the operation of the power system, the bad data detection mechanism is used to determine whether the current system-generated measurements contain false data. The basic principle of this mechanism is to substitute the state variable estimate $\hat{x}$ into the residual calculation, a process that can be expressed as follows:

$$r = z - H\hat{x} \tag{6}$$

where r represents the difference between the actual measured value and the estimated value. In the ideal case, r should be close to zero. However, due to measurement errors and potentially bad data, the residual may be larger than a certain threshold. Thus, potential bad data can be identified by residual analysis as shown in the following equation:

$$||r||_2 > \tau \tag{7}$$

where τ is a threshold set in advance. At this point, it can be determined that the system is experiencing bad data.

3.3. False Data Injection Attack

The FDIA will inject attack vectors into the measurement data z without changing the $||r||_2$:

$$z_a = z + a \tag{8}$$

where z_a denotes the measurement vector after the attack. At this point, the system state variable estimate can be expressed as follows:

$$\begin{aligned} \hat{x}_a &= \left(H^T W H\right)^{-1} H^T W z_a \\ &= \left(H^T W H\right)^{-1} H^T W (z + a) \\ &= \hat{x} + \left(H^T W H\right)^{-1} H^T W a \\ &= \hat{x} + c \end{aligned} \tag{9}$$

At this point, the residual can be computed as follows:

$$\begin{aligned} r_a &= z_a - H\hat{x}_a \\ &= z + a - H(\hat{x} + c) \\ &= z - H\hat{x} + (a - Hc) \\ &= r + (a - Hc) \end{aligned} \tag{10}$$

If $a = Hc$, it is guaranteed not to change the value of $||r||_2$ to inject the attack vector into the measurement data. Note that the injection vector a can also be obtained without explicitly asking for c. In [11], it was shown that the relation $a = Hc$ can be transformed into an equivalent form without explicitly using c, and from this a is generated. Since the FDIA constructed through the above is less than τ as the residual generated in the normal case, then false data can perfectly bypass the BDD and pose a threat to the stable operation of the power system.

3.4. Problem Analysis

The main reason why the false data behaves very similarly to the normal data is that the attacker utilizes an in-depth understanding of the power system model. The attacker has access to the parameters and topology of the system and uses the Jacobi matrix H to design an attack vector that makes the injected false data behave very similarly to the normal data in terms of features. To quantify this similarity, this paper uses kernel density estimation to estimate the distribution of normal and false data. The kernel density estimation formula is as follows:

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{11}$$

where $\hat{f}(x)$ is the estimated probability density function, $K(\cdot)$ is the kernel function; h is the bandwidth parameter, which controls smoothing; and x_i is the sample point.

The feature distribution of normal data is usually more centralized, showing typical measurements of the system in steady-state operation, such as bus voltages and branch active and reactive power flows. False data are injected into the system via attack vectors, which are designed to be highly overlapped with the normal data in terms of features by using the Jacobi matrix H to ensure that they satisfy $a = Hc$. The false data successfully bypasses the BDD mechanism by modifying the measurement data z without changing the residuals r. In the probability density distribution plots of normal data and false data shown in Figure 2, the distribution curves of false data are highly overlapped with the normal data curves, and the feature overlap rate reaches 96.27%, where Class 0 and Class 1 indicate normal and false data, respectively. This means that the false data are very similar to the normal data in the feature space, causing the existing detection methods to be difficult to distinguish. Therefore, ability to the detect false data must be improved by enhancing the differentiation between the two.

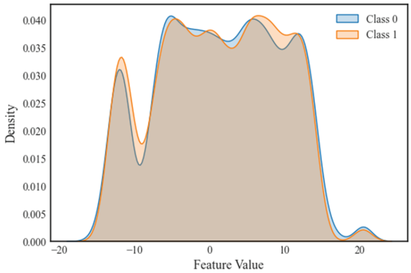


Figure 2. The distribution of normal and false data.

4. Proposed Detection Method

The proposed method is described in this section. Section 4.1 describes the overall flow of the method; Section 4.2 introduces distinction enhancement and information fusion; Section 4.3 introduces attention convolutional neural networks.

4.1. Method Overview

The method proposed in this paper is shown in Figure 3. The proposed method consists of two main modules.

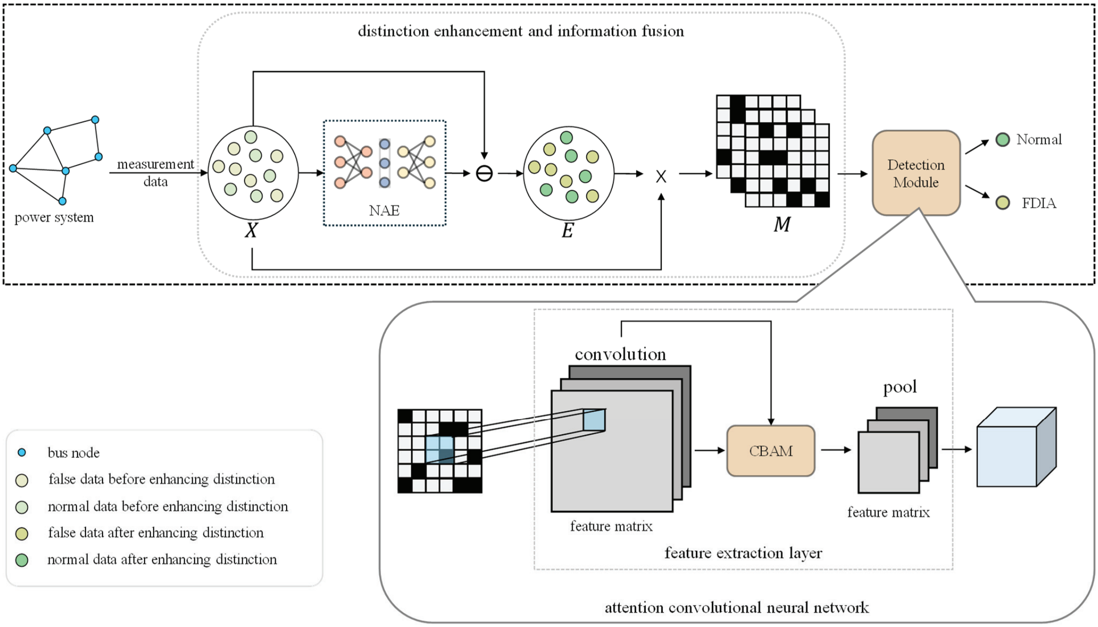

distinction enhancement and information fusion
power system
measurement data
X
NAE
E
M
Detection Module
Normal
FDIA
convolution
feature matrix
CBAM
pool
feature matrix
feature extraction layer
attention convolutional neural network
bus node
false data before enhancing distinction
normal data before enhancing distinction
false data after enhancing distinction
normal data after enhancing distinction

Figure 3. The structure of the DEIF-ACNN model.

The first module is distinction enhancement and information fusion. In this module, the NAE has the property that it has a small reconstruction error for normal data and a large reconstruction error for false data. This property of the NAE is exploited to enhance the distinction between normal and false data. Then, the enhanced data features are associated with the original data to obtain the feature matrix. This process is performed to retain the original data features and adapt to the subsequent detection module.

The second module is the attention convolutional neural network. The purpose of this module is to detect if the current data are false data or not. Since convolutional neural network has better performance in extracting data features, it can better extract features in the feature matrix. At the same time, the attention mechanism can suppress the interference of noise on the detection results.

4.2. Distinction Enhancement and Information Fusion

In the proposed DEIF-ACNN method, the NAE is constructed to enhance the distinction between the normal and false data, and its specific structure is shown in Figure 4.

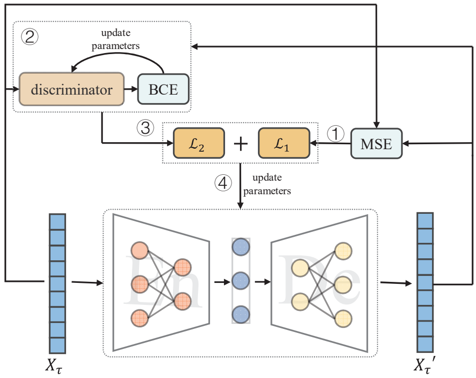


Figure 4. The structure of the NAE.

When training the NAE, only the normal data are taken as input and the reconstructed features are computed by the NAE; the process can be described as follows:

$$z = f_{enc}(X_\tau; \theta_{enc}) \tag{12}$$

$$X'_\tau = f_{dec}(z; \theta_{dec}) \tag{13}$$

where $X_\tau \in \mathbb{R}^m$ denotes the m-dimensional normal data features; $f_{enc}(\cdot)$ denotes the data encoding computation; $f_{dec}(\cdot)$ denotes the data decoding computation; θ_{enc} and θ_{dec} denotes the parameters of the encoder and decoder, respectively; and $X'_\tau \in \mathbb{R}^m$ denotes the m-dimensional reconstructed feature vector. The mean square error is chosen as the loss function for the reconstruction error and the calculation process is shown in Equation (14):

$$\mathcal{L}_1 = -\frac{1}{n}\sum_{i=1}^{n}\left(X_\tau^{(i)} - X_\tau'^{(i)}\right)^2 \tag{14}$$

where n denotes the number of samples per small batch.

To better ensure that the reconstructed data features belong to the type of normal data, the NAE introduces a discriminator. This discriminator is trained using the normal data and the loss is computed using a binary cross-entropy loss function in the following procedure:

$$\hat{y}_\tau = f_{clf}\left(X_\tau; \theta_{clf}\right) \tag{15}$$

$$BCELoss(\hat{y}_\tau) = -\frac{1}{n}\sum_{i=1}^{n}\left[\log\left(1 - \hat{y}_\tau^{(i)}\right)\right] \tag{16}$$

where θ_{clf} denotes the internal parameters of the discriminator; $f_{clf}(\cdot)$ denotes the discriminator; and $\hat{y}_\tau$ denotes the prediction result of the discriminator. By minimizing the loss, the internal parameters of the discriminator are updated until the loss converges. At this point, the reconstructed data features are used as inputs to the discriminator to obtain the prediction results, which are calculated as follows:

$$\hat{y}'_\tau = f_{clf}(X'_\tau; \theta_{clf}) \tag{17}$$

where $\hat{y}'_\tau$ denotes the prediction result of the discriminator on the reconstructed data. Then, the discrimination loss of the discriminator for the reconstructed features is calculated as follows:

$$\mathcal{L}_2 = -\frac{1}{n}\sum_{i=1}^{n}\left[\log\left(1 - \hat{y}_\tau'^{(i)}\right)\right] \tag{18}$$

Combining Equations (13) and (17) yields the final loss value $\mathcal{L}$ for the NAE:

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 \tag{19}$$

The NAE parameters θ_{enc} and θ_{dec} are updated by minimizing $\mathcal{L}$. In this way, the NAE is equipped with the characteristic of incurring smaller losses for normal data and larger losses for false data.

Next, DEIF-ACNN takes the hybrid data $X \in \{X_\tau, X_\alpha\}$ as input and computes the reconstructed data features X' through the trained NAE. Due to the characteristic of the NAE, the following inequality will be satisfied:

$$\mathcal{L}(X, X')\begin{cases} \approx 0, & if \cdot X \in X_\tau \\ > 0, & if \cdot X \in X_\alpha \end{cases} \tag{20}$$

where X_α denotes false data. According to Equation (19), the distinction-enhanced feature vectors can be computed by Equation (20):

$$E = X \ominus X' \tag{21}$$

where $\ominus$ denotes the corresponding positional subtraction of the feature vector and E denotes the enhanced data feature. Figure 5 shows the distribution of normal data and false data after distinction enhancement. Compared with Figure 2, it can be clearly seen that the distribution overlap of the two types of data is significantly reduced, and the overlap rate is reduced to 87.03%.

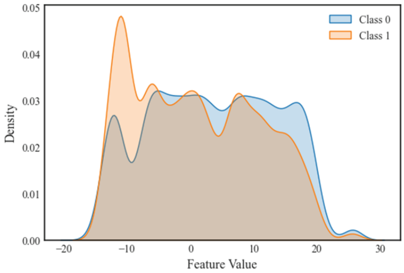


Figure 5. The distribution of normal and false data after distinction enhancement.

To ensure that the original data feature information is not lost during enhancing the distinction process and to satisfy the input requirements of the subsequent detection modules, DEIF-ACNN generates the feature correlation matrix by feature vector correlation after distinction enhancement. The process is described as follows:

$$M = X \times E^T \tag{22}$$

Therefore, the flow of the DEIF-ACNN model in the distinction enhancement and information fusion phase is shown in Algorithm 1.

Algorithm 1: Distinction Enhancement and Information Fusion.

```
Input: Normal data $X_\tau$; hybrid data $X$; learning rate $\alpha$; initial parameter of
discriminator $\theta_1$; initial parameter of the NAE $\theta_2$;
Output: Feature correlation matrix.
1. Perform the data preprocessing.
2. Train the discriminator.
3.   Repeat 4 to 8, if the loss $BCELoss$ is reduced substantially.
4.     Obtain the normal data $\left\{x_\tau^{(1)}, \cdots, x_\tau^{(n)}\right\}$ from $X_\tau$.
5.     Calculate the prediction of the discriminator using Equation (14): $\hat{y}_\tau$.
6.     Calculate the loss using Equation (15): $BCELoss(\hat{y}_\tau)$.
7.     Calculate the gradient: $g_1 \leftarrow \nabla_{\theta_1} BCELoss(\hat{y}_\tau)$ .
8.     Parameter update: $\theta_1 \leftarrow \theta_1 - \alpha \times \mathrm{g}_1$ .
9. Train the NAE.
10. Repeat 11 to 17, if the loss $\mathcal{L}$ is reduced substantially.
11.   Obtain the normal data $\left\{x_\tau^{(1)}, \cdots, x_\tau^{(n)}\right\}$ from $X_\tau$.
12.   Calculate the NAE's reconstruction features using Equations (11) and (12): $X'_\tau$
13.   Calculate the reconstruction loss using Equation (13): $\mathcal{L}_1$.
14.   Calculate the discrimination loss using Equation (17): $\mathcal{L}_2$.
15.   Calculate the loss using Equation (18): $\mathcal{L}$
16.   Calculate the gradient: $g_2 \leftarrow \nabla_{\theta 2} \mathcal{L}$ .
17.   Update the parameter: $\theta_2 \leftarrow \theta_2 - \alpha \times \mathrm{g}_2$ .
18. Distinction enhancement and information fusion
19. Calculate the distinction-enhanced data using Equation (20): $E$.
20. Calculate the dataset after information fusion using Equation (21): $M$.
21. Return feature correlation matrix: $M$
```

4.3. Attention Convolutional Neural Network

After processing in Section 4.1, each sample in the original data has been transformed into a feature correlation matrix. To improve the accuracy of FDIA detection, this method uses a CNN with a built-in attention mechanism module for better extraction of the data features of the feature correlation matrix. The structure is shown in the detection module of Figure 3. The ACNN consists of multiple feature extraction layers, each of which includes a convolutional layer, a CBAM module, and a pooling layer.

In the ACNN, the convolutional layer of each feature extraction layer consists of multiple learnable convolutional kernels. The key operations of the convolutional layer include local correlation and window sliding. Local correlation means that each convolution kernel is considered as a window filter, and during neural network training, the convolution kernels are convolved with the input data according to a customized size. Then, by controlling the step size, the sliding distance of the window filter is determined so that features are extracted in each localized region. The convolution process is essentially the multiplication of two matrices, and after convolution, the dimensions of the input data are reduced.

In the convolutional layer, in addition to the convolution operation, there are two key parameters: the activation function and the learnable bias vector. The role of the bias vector $c_{i,j}$ is to perform a linear addition of the convolved data, i.e., the convolution result is added to the bias vector, as shown in Equation (22). The activation function is used to enhance the nonlinear capability of the network by activating the data to remove useless information:

$$c_{i,j} = ReLU\left(z * h_{i,j}\right) + b_{i,j} \tag{23}$$

where $c_{i,j}$ is the feature mapping of the input feature matrix z after the convolution operation of the *i-th* feature extraction layer; $h_{i,j}$ is the *j-th* convolution kernel of the *i-th* convolution layer; $*$ denotes the convolution operation; $b_{i,j}$ is the *j-th* bias vector of the *i-th* convolution layer; $ReLU(\cdot)$ denotes the activation function.

Although PMUs provide high-precision measurements, they are not immune to noise. The noise in PMU data mainly comes from synchronization errors (due to GPS signal interference) and communication delays (due to network delays), which may compromise the accuracy of the collected data. To extract the locally important information more effectively and suppress the noise interference, the ACNN introduces the CBAM module based on the attention mechanism after the convolutional layer, and its specific structure is shown in Figure 6. The CBAM is a lightweight module with low memory requirements and computational cost, conditions that are extremely favorable for power system applications. The CBAM consists of two sub-modules: the channel attention module (CAM) and the spatial attention module (SAM), which help to emphasize relevant information from different perspectives. The CAM highlights channels associated with change while suppressing irrelevant channels; the SAM amplifies the difference between changed and unchanged feature elements in the spatial dimension. In this way, the CBAM more accurately identifies extremely important regions of the feature matrix.

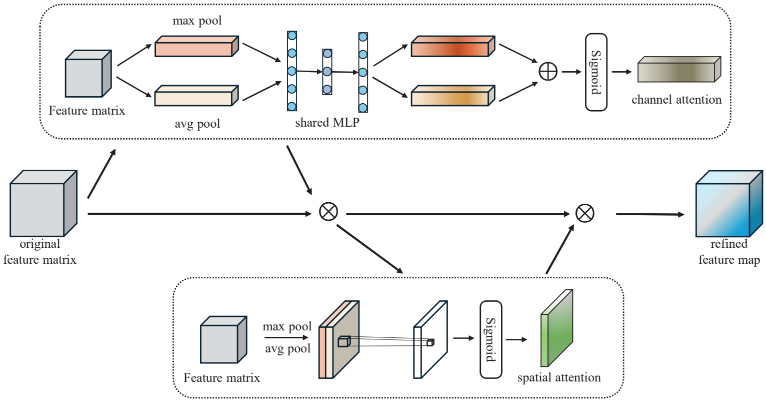


Figure 6. The internal structure of the CBAM module.

The CBAM takes the output of the previous convolutional layer as input, computes the attention weight matrix in channel and spatial dimensions, and finally multiplies the input feature matrix with the attention weight matrix to obtain a new feature matrix. The process is described as follows:

$$\begin{aligned} F' &= M_c(F) \otimes F \\ F'' &= M_s(F') \otimes F' \end{aligned} \tag{24}$$

$M_c(\cdot)$ indicates that the CAM sub-module computes the channel attention weight matrix; $M_s(\cdot)$ indicates that the SAM sub-module computes the spatial attention weight matrix; F, F', and F'' denote the input feature matrix, the feature matrix after the channel attention, and the final output feature matrix, respectively.

In computing the channel attention weights, the CAM sub-module takes the original feature matrix as input and applies maximum pooling and average pooling, respectively.

Subsequently, the pooling results are processed through a multilayer perceptron (MLP) based on an encoder–decoder architecture. Finally, the channel attention feature matrix is generated by the Sigmoid activation function. The process is shown in Equation (25):

$$\begin{aligned} M_c(F) &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\ &= \sigma(W_1W_0(AvgPool(F)) + W_1W_0(MaxPool(F))) \end{aligned} \tag{25}$$

where W_0 and W_1 denote the weights of the first and second hidden layers of the multilayer perceptron, respectively; $\sigma(\cdot)$ denotes the Sigmoid activation function. The new feature matrix F' is obtained by multiplying the channel attention weight matrix with the input feature matrix.

In computing the spatial attention weight matrix, the SAM sub-module first performs average pooling and maximum pooling operations on the feature matrix F' along the channel axis, respectively; then, these two pooled feature maps are spliced onto the channel axis to form a combined feature matrix; subsequently, a 7∗7 convolution operation is performed on the combined feature matrix; and finally, the spatial attention weight matrix is generated by the Sigmoid activation function. The process is shown in Equation (26):

$$M_s(F') = \sigma\Big(f^{7*7}([AvgPool(F'); MaxPool(F')])\Big) \tag{26}$$

where $f^{7*7}(\cdot)$ denotes a convolution operation using a 7∗7 convolution kernel. The spatial attention weight matrix is multiplied with the output F' of the CAM sub-module to finally obtain the output F'' of the CBAM.

The CBAM module further captures important information about the output results of the convolutional layer and suppresses the interference of noise in the feature matrix. To reduce the subsequent computational complexity, the ACNN adds a pooling layer after the CBAM module, aiming at compressing the data and the number of parameters and reducing the risk of overfitting, while preserving the most important features in the data. The pooling layer operates as follows:

1. Feature invariance: the pooling operation removes unimportant information in data features, while the retained information is scale invariance and still representative of the data features before pooling.
2. Feature dimensionality reduction: the pooling layer reduces the data dimensionality by removing redundant information and extracting only the most important features, thus reducing the amount of computation and preventing overfitting to a certain extent.

The last layer of the ACNN is the linear layer, whose main function is to compute the final prediction probability from the features extracted from the feature extraction layer through the fully connected network, whose structure is shown in Figure 7. First, the feature matrices are converted to feature vectors by a flatten layer, since the input to the linear layer requires one-dimensional feature vectors rather than matrices. Next, the feature vectors are processed through a fully connected network to generate predictions. Finally, the model is optimized by calculating the loss between the ACNN predictions and the true labels. The loss is calculated using the cross-entropy loss function as follows:

$$\mathcal{L}_{CCES} = -\frac{1}{n}\sum_{i=1}^{n}(y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)) \tag{27}$$

where n denotes the amount of data in each small batch; y_i denotes the true label of the *i-th* sample; $\hat{y}_i$ denotes the probability that the *i-th* sample is predicted to have a positive label. The internal parameters of the ACNN are adjusted to optimize the model performance by minimizing the loss values calculated by Equation (26).

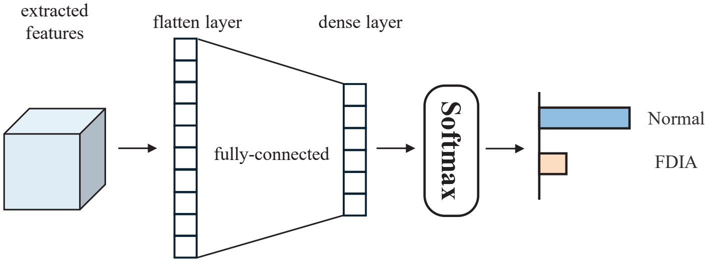


Figure 7. The fully connected linear layers.

Therefore, the process of training the DEIF-ACNN model to predict false data is shown in Algorithm 2.

Algorithm 2: DEIF-ACNN predicts false data.

```
Input: Feature correlation matrix: $M$; learning rate $\gamma$; initial parameters of the ACNN $\omega$.
Output: The parameter matrix of the ACNN: $\omega$.
1. Train the ACNN.
2.  Repeating 1.1 to 1.4, if the loss $\mathcal{L}_{CCES}$ is reduced substantially.
3.      Obtain mixed data $\left\{M^{(1)}, \cdots, M^{(n)}\right\}$ from $M$; the label of $M^{(i)}$ is $y^{(i)}$.
4.      Calculate the loss using Equation (26): $\mathcal{L}_{CCES}\left(y^{(i)}, \hat{y}^{(i)}\right)$.
5.      Calculate the gradient: $g_3 \leftarrow \nabla_\omega \mathcal{L}_{CCES}\left(y^{(i)}, \hat{y}^{(i)}\right)$.
6.      Update the parameter $\omega \leftarrow \omega - \gamma \times \mathrm{g}_3$.
7. Return the parameter matrix of the ACNN: $\omega$.
```

5. Experiments

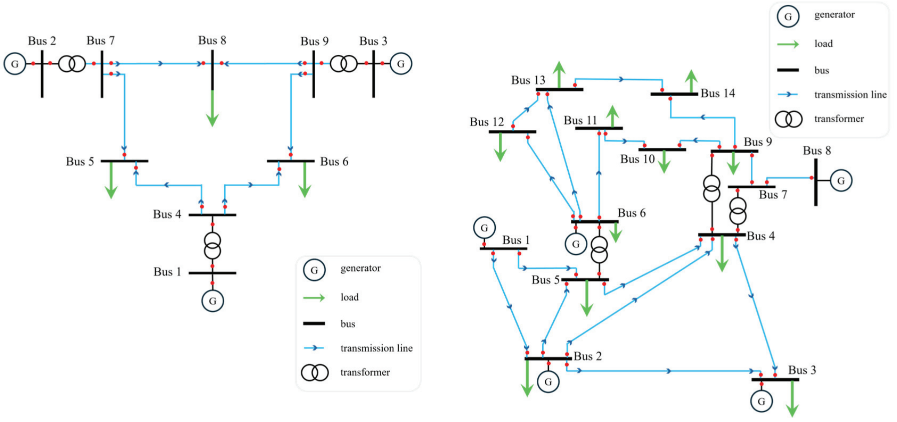
In this section, the effectiveness of the proposed method on the IEEE-9, IEEE-14, and IEEE-118 bus power system standard test systems is analyzed in detail and the performance of the method in a noisy environment is explored. The GUP used in this experiment is RTX 3060, the CPU is Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50 GHz, and the memory is 32 G. The others use pytorch 2.2.1, Python3.8, Cuda 11.2, etc.

5.1. Dataset Setup

The datasets used in this experiment were collected from the IEEE-9, IEEE-14, and IEEE-118 standard test case simulations. These test cases are from PYPOWER, a derivative of MATPOWER [29].

5.1.1. Simulation Case Description

The IEEE-9 bus is a test case consisting of nine buses, three generators, and nine branches; the IEEE-14 bus consists of fourteen buses, five generators, and twenty branches; and the IEEE-118 bus consists of one hundred eighteen buses, fifty-four generators, and one hundred eighty-six branches. The topologies of IEEE-9 and IEEE-14 are shown in Figure 8. Due to the huge IEEE-118 system, the topology diagram of IEEE-118 bus is omitted in this paper, and the details can be seen in [30]. The main system parameters for the test case are shown in Table 1, where the PF and QF of the branch are only included in the power flow output and are ignored when used as inputs to the power flow.



(**a**) IEEE-9 bus

(**b**) IEEE-14 bus

Figure 8. The topology structure.

Table 1. The main system parameters of the test case.

Name	Column	Description
bus	PD QD	real power demand reactive power demand
generator	PG QG	real power output reactive power output
branch	PF QF	real power injected at “from” bus end reactive power injected at “from” bus end

5.1.2. Data Generation

Since PYPOWER has only one test case for the IEEE-9, IEEE-14, and IEEE-118 bus, the generated data are insufficient to support the whole experiment. Therefore, to be able to obtain more data, each bus system will be assigned about 100 k load data [31], each of which includes both active and reactive loads. The process of generating measurement data is as follows:

Step 1: Modify the active power demand (PD) and reactive power demand (QD) of the simulated system using this load data. The modifications are as follows:

$$\begin{aligned} PD &\leftarrow load\_active \\ QD &\leftarrow load\_reactive \end{aligned} \tag{28}$$

where *load_active* denotes active load and *load_reactive* denotes reactive load. Each modification becomes a new test case. The new test case is input into the AC-PF of PYPOWER for power flow calculation, and measurement data are taken from the results of the calculation. Among the measured data are the bus active power injection (pi), the bus reactive power injection (qi), the active power injection at the branch “from” end (pf), and the reactive power injection at the branch “from” end (qf). The calculation process of the measured data is as follows:

$$z = concatenate([pf, pi, qf, qi]) \tag{29}$$

The number of measurements of the measured data in the three test cases is shown in Table 2.

Table 2. The number of measurements of the measurement data.

Test Case	Type	Number	Total
IEEE-9 bus	pi qi pf qf	9 (9 buses) 9 (9 buses) 9 (9 branch) 9 (9 branch)	36
IEEE-14 bus	pi qi pf qf	14 (14 buses) 14 (14 buses) 20 (20 branches) 20 (20 branches)	68
IEEE-118 bus	pi qi pf qf	118 (118 buses) 118 (118 buses) 186 (186 branches) 186 (186 branches)	608

Step 2: A random number α from 0 to 1 is generated to determine whether the current measurement data are needed for a false data injection attack by using Equation (8). The conditions are as follows:

$$z = \begin{cases} z, & if \quad \alpha < 0.5 \\ z_a, & if \quad \alpha > 0.5 \end{cases} \tag{30}$$

Step 3: By using Equations (2)–(7), determine whether the measurement data z obtained by the current system satisfy the residual analysis. If the residual analysis is satisfied, add it to the dataset collection; if not, return to Step 1.

Repeat Step 1 to Step 3 until all load data are used up.

5.1.3. Data Preprocessing

To maximize the performance of all models used in the experiment, the collected dataset will be preprocessed.

1. Balancing Process. From the generated dataset, 15 k normal data and 15 k false data were randomly selected to form the sample balanced dataset; details are given in Table 3.

Table 3. Specifics of the datasets.

Bus Name	Bus-9	Bus-14	Bus-118
Normal data	15,000	15,000	15,000
False data	15,000	15,000	15,000
Number of features	36	68	608
Total	30,000	30,000	30,000

2. Normalization Process. The features of the dataset are deflated to between [0,1]. The deflation process is as follows:

$$X = \frac{X - \min}{\max - \min} \tag{31}$$

where min and max are the minimum and maximum values of the sample data, respectively.

3. Dataset splitting. Throughout the experiments, each dataset was divided into 60% training set, 20% validation set, and 20% test set.

5.2. Evaluation Metrics

Accuracy, precision, recall, F1-Score, and cross-entropy loss are considered as evaluation metrics to measure the performance of the classification models, and they are defined below:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{32}$$

$$Precision = \frac{TP}{TP + FP} \tag{33}$$

$$Recall = \frac{TP}{TP + FN} \tag{34}$$

$$F_1 - Score = 2 \times \frac{precision \times recall}{precision + recall} \tag{35}$$

where TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative. $F_1 - Score$ can comprehensively evaluate the performance of the model, and, therefore, the evaluation metric is the focus of comparison in this paper. The cross-entropy loss reflects the convergence of the model during the training process.

5.3. FDIA Detection Performance

To assess the detection efficiency and reliability of the method in this paper, a comparative experiment was conducted with several baseline models.

5.3.1. Ideal Environment

In the experimental evaluation, this paper first conducts a comparison experiment in an ideal environment (no noise condition). The experimental results are shown in Table 4, where the No-CBAM is the ablation model of the proposed method in this paper, specifically, the removal of the CBAM module in the ACNN.

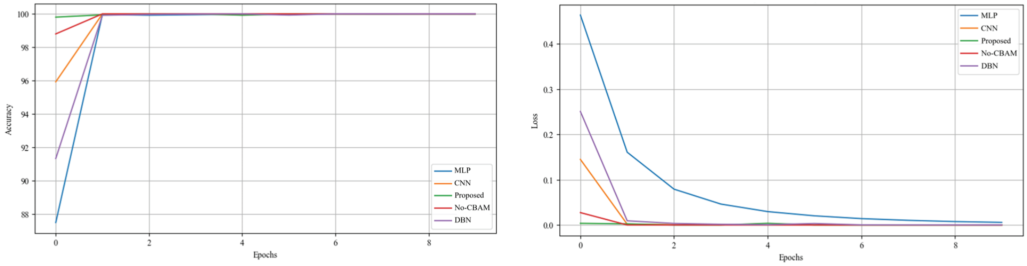
In the IEEE-9 bus system, DEIF-ACNN has an accuracy of 99.22%, a precision of 99.97%, a recall of 98.48%, and an F1-Score of 99.22%. Compared with other optimal models (e.g., DT), the F1-Score improves by 0.42%. This smaller boost reflects the fact that DT already performs quite well when dealing with a system of this size. However, the distinction enhancement and information fusion strategies of DEIF-ACNN enable it to better mine the subtle differences in the data; combined with the optimized extraction of features by the CBAM module, it further improves the comprehensive performance of the model. In contrast, weaker models such as the AE perform significantly worse in this system, with an F1-Score of only 88.20%. This is mainly because the AE is not able to extract the important features as DEIF-ACNN does when dealing with complex data features; thus, the performance gap between the two is significant.

In the IEEE-14 bus system, DEIF-ACNN has an accuracy of 99.83%, a precision of 100%, a recall of 99.67%, and an F1-Score of 99.83%. Compared with other better-performing models (e.g., CNN), the F1-Score is only improved by 0.43%. The CNN has already demonstrated a fairly good performance in this system due to its powerful feature extraction capabilities. However, DEIF-ACNN further enhances the capture of key features by incorporating the CBAM of the attention mechanism, thus achieving a small but significant performance improvement over the already high level. In contrast, the F1-Score of the AE is only 84.36%, which further indicates that the feature extraction ability of the AE is much less than that of DEIF-ACNN when dealing with more complex system structures.

In the IEEE-118 bus system, DEIF-ACNN achieves 100% accuracy, precision, recall, and F1-Score, which is an excellent performance. The accuracy is improved by 41.2% compared to the worst KNN. This is because KNN classifies the data by calculating the Euclidean distance, which makes it difficult to perform well with high-dimensional data. However, some other models (e.g., MLP, CNN, DBN, and No-CBAM) also achieve the same high scores. This situation is difficult to compare further. Therefore, the advantage of the model is evaluated by observing the change in cross-entropy loss during the training process. As shown in Figure 9, DEIF-ACNN converges faster during the training process, indicating that it has significant advantages in learning efficiency and training stability. This also implies that DEIF-ACNN may have stronger generalization ability and higher reliability in more complex practical application scenarios.

Table 4. Comparative experimental results in ideal environment.

System	Method	Accuracy	Precision	Recall	F1-Score
IEEE-9 bus	Proposed	**99.22**	99.97	98.48	**99.22**
	KNN	94.65	100.00	89.31	94.35
	DT	98.80	98.95	98.66	98.80
	MLP	94.25	100.00	88.58	93.95
	CNN	95.58	99.93	91.30	95.42
	GRU	95.92	99.93	91.96	95.78
	DBN	90.38	99.96	80.94	89.45
	AE	89.33	99.62	79.12	88.20
	No-CBAM	98.05	100.00	96.13	98.03
IEEE-14 bus	Proposed	**99.83**	100.00	99.67	**99.83**
	KNN	94.44	100.00	88.88	94.11
	DT	99.40	99.38	99.42	99.40
	MLP	95.62	100.00	91.30	95.45
	CNN	98.38	99.73	97.05	98.37
	GRU	96.15	99.96	92.39	96.03
	DBN	91.87	97.42	86.14	91.43
	AE	85.87	95.29	75.68	84.36
	No-CBAM	99.32	99.60	99.04	99.32
IEEE-118 bus	Proposed	**100.00**	100.00	100.00	**100.00**
	KNN	58.80	100.00	17.58	29.90
	DT	99.84	99.89	99.79	99.84
	MLP	100.00	100.00	100.00	100.00
	CNN	100.00	100.00	100.00	100.00
	GRU	100.00	100.00	100.00	100.00
	DBN	100.00	100.00	100.00	100.00
	AE	89.63	100.00	79.42	88.53
	No-CBAM	100.00	100.00	100.00	100.00

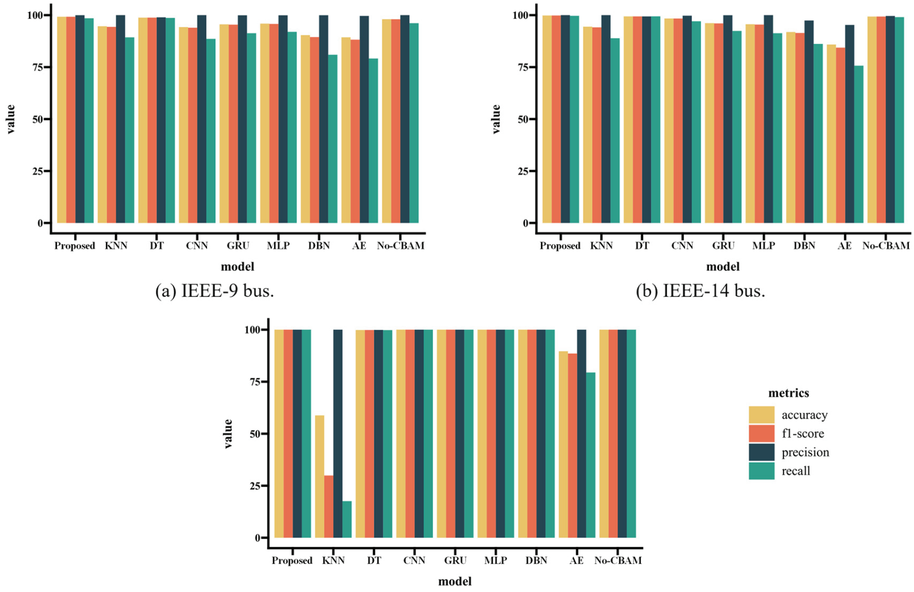


(**a**) Training accuracy.

(**b**) Training loss.

Figure 9. Changes in accuracy and loss of training on IEEE bus-118.

In summary, although the improvement in DEIF-ACNN is limited when compared with better-performing models, its stable performance and superiority in complex systems still reflect its strong detection ability. Compared with the poorer-performing models, DEIF-ACNN significantly improves the detection efficiency, mainly due to its effective combination of distinction enhancement, information fusion, and attention mechanism. The overall results of the comparison experiments are shown in Figure 10.



(a) IEEE-9 bus.

(b) IEEE-14 bus.

(c) IEEE-118 bus.

Figure 10. Comparing and contrasting the bar graphs of the experimental results.

5.3.2. Noise Environment

In practical application environments, the interference of noise for measurement data such as electromagnetic waves can significantly affect the detection efficiency of the detection model. Therefore, this section evaluates the detection efficiency under different noise levels. In this experiment, Gaussian noise is used to simulate the interference to the measurement data in the real environment. The process of adding noise is as follows:

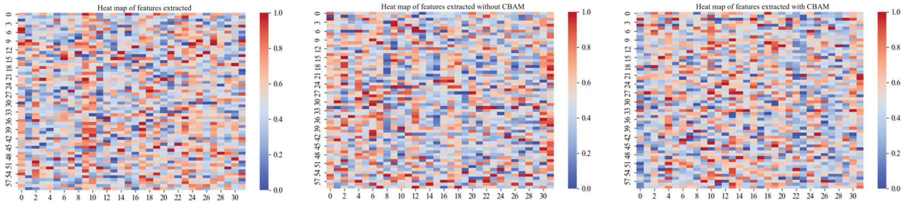
$$X_{noise} = X + noise \tag{36}$$

where $X \in \mathbb{R}^m$ denotes the m-dimensional original data; $noise \in \mathbb{R}^m$ denotes the m-dimensional noise vector obeying a Gaussian noise $noise \sim N\left(0, \lambda^2\right)$; X_{noise} indicates data after adding noise. Different values of λ indicate the level of noise; the higher the value of λ, the higher the noise level and the stronger the interference. The interference of different levels of noise on the detection results is modeled by varying the magnitude of λ; the values of λ are specifically shown in Table 5.

Table 5. The values of σ under the different datasets.

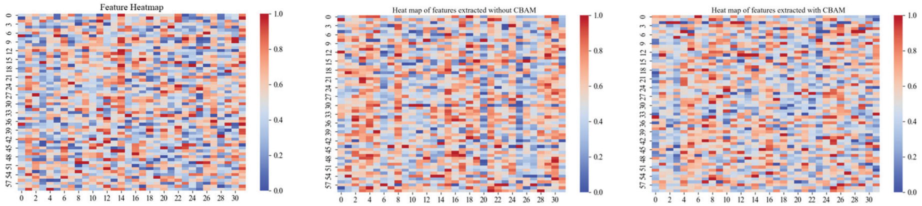
Dataset	λ (Noise Level)						
IEEE-9 bus	0	0.1	0.2	0.3	0.4	0.5	0.6
IEEE-14 bus	0	0.2	0.4	0.6	0.8	1.0	1.2
IEEE-118 bus	0	0.5	1.0	1.5	2.0	2.5	3.0

In the process of extracting important features, DEIF-ACNN employed the CBAM to mitigate noise interference in the detection process. The heatmaps of extracted features were plotted on the three datasets for original data, noisy data, and data with noise suppressed, respectively. For visualization purposes, 60 samples were randomly selected as observations in the test set and the extracted features were nonlinearly mapped to a low-dimensional space (32 dimensions) by an autoencoder. The corresponding heatmaps are shown in Figures 11–13. The horizontal axis of each heatmap represents the individual feature of the samples, and the vertical axis represents each sample. The color scale from blue to red indicates the magnitude of the eigenvalue. Comparing (a) and (b) of Figures 11–13, the heatmaps drawn by features extracted from the noisy data have changed significantly; this situation may lead to a decrease in accuracy. As shown in Figure 14, by utilizing the CBAM to suppress the interference of noise, the overall pattern of the heatmap after the noise is suppressed is close to the original data, as can be seen in (c) of Figures 11–13. This is because the CBAM adaptively adjusts the contribution of features by calculating attention weights. Thus, the effect of noise on the detection performance of DEIF-ACNN is reduced. Overall, the noise suppression strategy of DEIF-ACNN is effective and greatly restores the features of the noisy data.



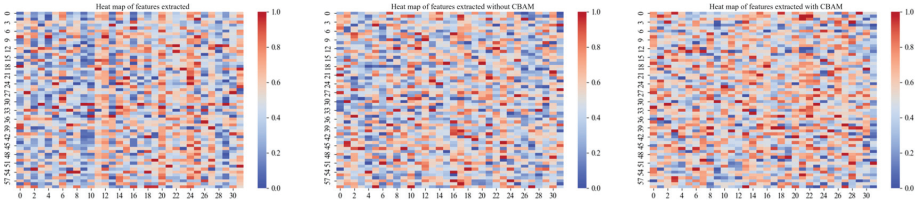
(**a**) Original data. (**b**) Noisy data. (**c**) Noise-suppressed data.

Figure 11. Heatmap of the extracted features on IEEE-9 bus.



(**a**) Original data. (**b**) Noisy data. (**c**) Noise-suppressed data.

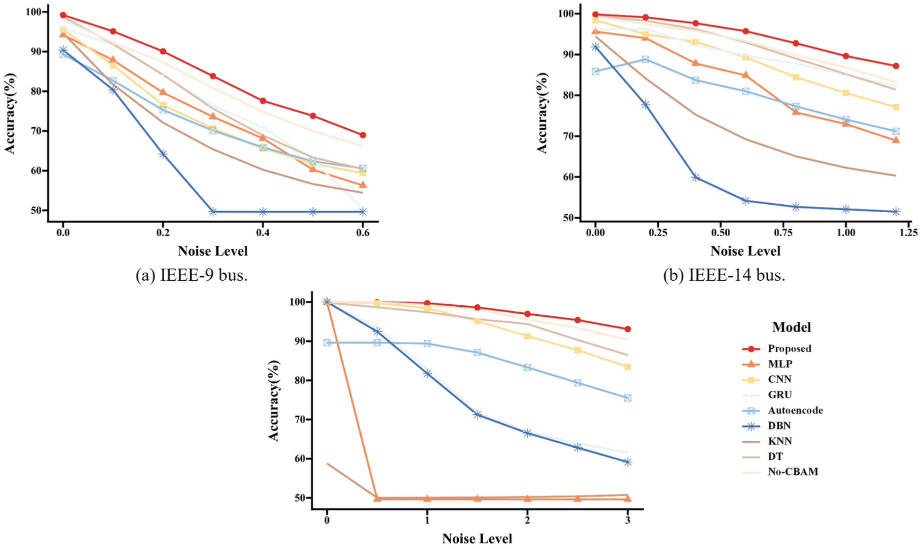
Figure 12. Heatmap of the extracted features on IEEE-14 bus.



(**a**) Original data. (**b**) Noisy data (**c**) Noise-suppressed data.

Figure 13. Heatmap of the extracted features on IEEE-118 bus.

To further illustrate the effectiveness of DEIF-ACNN in suppressing noise, the classification accuracies of DEIF-ACNN were compared with several other detection models under different noise conditions. As shown in Figure 14a, the proposed method outperforms the other models in terms of accuracy under different noise conditions. With the increasing noise level, the accuracy of DEIF-ACNN remains around 80%. However, for DBN, the accuracy decreases to about 50%. This suggests that these models have difficulty dealing with complex feature relationships in noisy environments. Observing Figure 14b, although the accuracy of the AE improves by 5% within the noise level from 0 to 0.2, the accuracy of the AE decreases to about 75% as the noise level continues to increase. This is because moderate noise acts as a regularization factor and enhances the generalization ability of the model. The results on the IEEE-14 bus dataset show that DEIF-ACNN still has the strongest resistance to noise interference. On the IEEE-118 bus dataset, as shown in Figure 14c, as the noise level increases, the accuracy decreases from 60% to 50% since the KNN does not have the feature extraction capability. MLP has a weaker feature extraction capability, and the accuracy of MLP decreases from 100% to 50%. Similarly, the same situation as for the IEEE-14 bus occurs at a noise level of around 1 on this dataset. This further indicates that moderate noise enhances the generalization ability of the model.



(a) IEEE-9 bus.

(b) IEEE-14 bus.

(c) IEEE-118 bus.

Figure 14. Accuracy of different noise levels.

Overall, the proposed method consistently maintains high classification accuracy across all noise levels, particularly in high-noise environments, where it demonstrates stronger robustness and anti-interference capabilities. DEIF-ACNN is compared with No-ACNN on all datasets. The comparison results further demonstrate the effectiveness of DEIF-ACNN for combating noise interference. In comparing No-CBAM with other methods, it is illustrated that DEIF-ACNN still enhances the distinction between normal and false data in noisy environments. All experimental results demonstrate the ability of DEIF-ACNN to improve the distinction between normal and false data, as well as verifying the ability of DEIF-ACNN in resisting noise interference.

5.3.3. Analysis of Ablation Experiments

In an ideal environment, by comparing the method proposed in this paper with No-CBAM, due to the existence of the attention mechanism, the features of the data can be better extracted, so that the detection performance of DEIF-ACNN is not only better than No-CBAM, but also better than the other models. Comparing No-CBAM with the other models, because the NAE is utilized to enhance the distinction between normal data and false data, the detection performance of No-CBAM is only weaker than that of DT. In noisy environment, the detection performance of the proposed method is better than that of No-CBAM under different noise conditions, especially high noise. Meanwhile, the detection performance of No-CBAM is better than the other models.

In summary, the utilization of both the NAE and CBAM can improve the detection performance of the model, and they have different emphases. The NAE is used to enhance the distinction between normal data and false data, while the CBAM is used to better extract data features and suppress the interference of noise on detection results. The combination of the two can maximize the better detection performance.

6. Conclusions

In this paper, a novel FDIA detection method based on DEIF-ACNN is proposed. First, DEIF-ACNN enhances data distinction by constructing an NAE and employs information fusion techniques to transform both the original and enhanced data into a feature correlation matrix, thereby improving FDIA detection performance. Second, to better extract data features and effectively suppress noise interference, the model incorporates the CBAM module, which leverages an attention mechanism. This module significantly enhances the model's robustness in noisy environments, successfully mitigating noise and improving detection accuracy. Finally, experimental results on IEEE-9, IEEE-14, and IEEE-118 bus systems show that the FDIA detection effect of DEIF-ACNN is better than that of other methods, and the accuracy rates are 99.22%, 99.83%, and 100.00%, respectively. In addition, the proposed method shows the best robustness in different noise environments, and its accuracy remains around 80%. However, the current approach is primarily suited for simpler binary classification tasks, and future work will focus on adapting it to more complex system environments to enhance its detection capacity and broader applicability.

Author Contributions: C.L.: conceptualization, methodology, software, validation, data collection, writing—original draft, writing—review and editing, and supervision; Z.Z.: methodology, software, and resources; R.J.: writing—review and editing; B.Z.: writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Chinese Government Guidance Fund on Local Science and Technology Development of Sichuan Province (24ZYRGZN0018).

Data Availability Statement: Data available on request due to restrictions (e.g., privacy, legal or ethical reasons).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dehghanpour, K.; Wang, Z.; Wang, J.; Yuan, Y.; Bu, F. A Survey on State Estimation Techniques and Challenges in Smart Distribution Systems. *IEEE Trans. Smart Grid* **2019**, *10*, 2312–2322. [CrossRef]
2. Fang, X.; Misra, S.; Xue, G.; Yang, D. Smart Grid—The New and Improved Power Grid: A Survey. *IEEE Commun. Surv. Tutor.* **2012**, *14*, 944–980. [CrossRef]
3. Spanò, E.; Niccolini, L.; Pascoli, S.D.; Iannacconeluca, G. Last-Meter Smart Grid Embedded in an Internet-of-Things Platform. *IEEE Trans. Smart Grid* **2015**, *6*, 468–476. [CrossRef]
4. Sridhar, S.; Hahn, A.; Govindarasu, M. Cyber–Physical System Security for the Electric Power Grid. *Proc. IEEE* **2012**, *100*, 210–224. [CrossRef]
5. Deng, R.; Xiao, G.; Lu, R.; Liang, H.; Vasilakos, A.V. False Data Injection on State Estimation in Power Systems—Attacks, Impacts, and Defense: A Survey. *IEEE Trans. Ind. Inform.* **2017**, *13*, 411–423. [CrossRef]
6. Theodorakatos, N.P.; Babu, R.; Moschoudis, A.P. The Branch-and-Bound Algorithm in Optimizing Mathematical Programming Models to Achieve Power Grid Observability. *Axioms* **2023**, *12*, 1040. [CrossRef]
7. Theodorakatos, N.P.; Babu, R.; Theodoridis, C.A.; Moschoudis, A.P. Mathematical Models for the Single-Channel and Multi-Channel PMU Allocation Problem and Their Solution Algorithms. *Algorithms* **2024**, *17*, 191. [CrossRef]
8. Li, F.; Qiao, W.; Sun, H.; Wan, H.; Wang, J.; Xia, Y.; Xu, Z.; Zhang, P. Smart Transmission Grid: Vision and Framework. *IEEE Trans. Smart Grid* **2010**, *1*, 168–177. [CrossRef]
9. Yu, Z.-H.; Chin, W.-L. Blind False Data Injection Attack Using PCA Approximation Method in Smart Grid. *IEEE Trans. Smart Grid* **2015**, *6*, 1219–1226. [CrossRef]
10. Liang, G.; Zhao, J.; Luo, F.; Weller, S.R.; Dong, Z.Y. A Review of False Data Injection Attacks Against Modern Power Systems. *IEEE Trans. Smart Grid* **2017**, *8*, 1630–1638. [CrossRef]
11. Liu, Y.; Ning, P.; Reiter, M.K. False data injection attacks against state estimation in electric power grids. *ACM Trans. Inf. Syst. Secur.* **2011**, *14*, 13. [CrossRef]
12. Wang, H.; Fang, Y.P.; Zio, E. Resilience-oriented optimal post-disruption reconfiguration for coupled traffic-power systems. *Reliab. Eng. Syst. Saf.* **2022**, *222*, 108408. [CrossRef]
13. Xiao, D.; Chen, H.; Cai, W.; Wei, C.; Zhao, Z. Integrated risk measurement and control for stochastic energy trading of a wind storage system in electricity markets. *Prot. Control Mod. Power Syst.* **2023**, *8*, 1–11. [CrossRef]

14. Li, X.; Hu, L.; Lu, Z. Detection of false data injection attack in power grid based on spatial-temporal transformer network. *Expert Syst. Appl.* **2024**, *238*, 121706. [CrossRef]
15. Li, Y.; Wang, Y. Developing graphical detection techniques for maintaining state estimation integrity against false data injection attack in integrated electric cyber-physical system. *J. Syst. Archit.* **2020**, *105*, 101705. [CrossRef]
16. Musleh, A.S.; Chen, G.; Dong, Z.Y. A Survey on the Detection Algorithms for False Data Injection Attacks in Smart Grids. *IEEE Trans. Smart Grid* **2020**, *11*, 2218–2234. [CrossRef]
17. Alexopoulos, T.A.; Korres, G.N.; Manousakis, N.M. Complementarity reformulations for false data injection attacks on PMU-only state estimation. *Electr. Power Syst. Res.* **2020**, *189*, 106796. [CrossRef]
18. Duan, J.; Zeng, W.; Chow, M.-Y. Resilient Distributed DC Optimal Power Flow Against Data Integrity Attack. *IEEE Trans. Smart Grid* **2018**, *9*, 3543–3552. [CrossRef]
19. Li, B.; Lu, R.; Xiao, G.; Su, Z.; Ghorbani, A. PAMA: A Proactive Approach to Mitigate False Data Injection Attacks in Smart Grids. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6. [CrossRef]
20. Kallitsis, M.G.; Bhattacharya, S.; Stoev, S.; Michailidis, G. Adaptive statistical detection of false data injection attacks in smart grids. In Proceedings of the 2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Washington, DC, USA, 7–9 December 2016; pp. 826–830. [CrossRef]
21. Moslemi, R.; Mesbahi, A.; Velni, J.M. A Fast, Decentralized Covariance Selection-Based Approach to Detect Cyber Attacks in Smart Grids. *IEEE Trans. Smart Grid* **2018**, *9*, 4930–4941. [CrossRef]
22. Manandhar, K.; Cao, X.; Hu, F.; Liu, Y. Detection of Faults and Attacks Including False Data Injection Attack in Smart Grid Using Kalman Filter. *IEEE Trans. Control Netw. Syst.* **2014**, *1*, 370–379. [CrossRef]
23. Li, B.; Ding, T.; Huang, C.; Zhao, J.; Yang, Y.; Chen, Y. Detecting False Data Injection Attacks Against Power System State Estimation with Fast Go-Decomposition Approach. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2892–2904. [CrossRef]
24. Li, S.; Yılmaz, Y.; Wang, X. Quickest Detection of False Data Injection Attack in Wide-Area Smart Grids. *IEEE Trans. Smart Grid* **2015**, *6*, 2725–2735. [CrossRef]
25. He, Y.; Mendis, G.J.; Wei, J. Real-Time Detection of False Data Injection Attacks in Smart Grid: A Deep Learning-Based Intelligent Mechanism. *IEEE Trans. Smart Grid* **2017**, *8*, 2505–2516. [CrossRef]
26. Foroutan, S.A.; Salmasi, F.R. Detection of false data injection attacks against state estimation in smart grids based on a mixture Gaussian distribution learning method. *IET Cyber-Phys. Syst. Theory Appl.* **2017**, *2*, 161–171. [CrossRef]
27. Wang, C.; Tindemans, S.; Pan, K.; Palensky, P. Detection of False Data Injection Attacks Using the Autoencoder Approach. In Proceedings of the 2020 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), Liege, Belgium, 18–21 August 2020; pp. 1–6. [CrossRef]
28. Lu, M.; Wang, L.; Cao, Z.; Zhao, Y.; Sui, X. False data injection attacks detection on power systems with convolutional neural network. *J. Phys. Conf. Ser.* **2020**, *1633*, 012134. [CrossRef]
29. Zimmerman, R.D.; Murillo-Sanchez, C.E.; Thomas, R.J. MATPOWER: Steady-State Operations, Planning and Analysis Tools for Power Systems Research and Education. *IEEE Trans. Power Syst.* **2011**, *26*, 12–19. [CrossRef]
30. Information Trust Institute Grainger College of Engineering. IEEE 118-Bus System. 2024. Available online: https://icseg.iti.illinois.edu/ieee-118-bus-system/ (accessed on 25 September 2024).
31. Trindade, A. ElectricityLoadDiagrams20112014 [Dataset]. UCI Machine Learning Repository. 2015. Available online: https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams20112014 (accessed on 18 April 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Differential Impacts of Environmental, Social, and Governance News Sentiment on Corporate Financial Performance in the Global Market: An Analysis of Dynamic Industries Using Advanced Natural Language Processing Models †

Minjoong Kim [1,‡], Jinseong Kang [2,‡], Insoo Jeon [3], Juyeon Lee [4], Jungwon Park [5], Seulgi Youm [6], Jonghee Jeong [7], Jiyoung Woo [2] and Jihoon Moon [1,2,3,*]

[1] Department of ICT Convergence, Soonchunhyang University, Asan 31538, Republic of Korea; wooni3804@sch.ac.kr
[2] Department of AI and Big Data, Soonchunhyang University, Asan 31538, Republic of Korea; iboy115@sch.ac.kr (J.K.); jywoo@sch.ac.kr (J.W.)
[3] Department of Medical Science, Soonchunhyang University, Asan 31538, Republic of Korea; jis601@sch.ac.kr
[4] Graduate School of Mass Communication, Chung-Ang University, Seoul 06974, Republic of Korea; ljy9712@cau.ac.kr
[5] Platform Team, DTaaS, Seoul 06627, Republic of Korea
[6] Gasan Digital Center, IBK, Seoul 08506, Republic of Korea
[7] Data Solution Team, Evidnet, Seoul 06258, Republic of Korea
[*] Correspondence: jmoon22@sch.ac.kr

[†] This paper is an extended version of our paper published in Proceedings of the 2024 International Conference on Platform Technology and Service (PlatCon-24), Jeju, Republic of Korea, 26–28 August 2024.
[‡] These authors contributed equally to this work.

Citation: Kim, M.; Kang, J.; Jeon, I.; Lee, J.; Park, J.; Youm, S.; Jeong, J.; Woo, J.; Moon, J. Differential Impacts of Environmental, Social, and Governance News Sentiment on Corporate Financial Performance in the Global Market: An Analysis of Dynamic Industries Using Advanced Natural Language Processing Models. *Electronics* **2024**, *13*, 4507. https://doi.org/10.3390/electronics13224507

Academic Editor: Ping-Feng Pai

Received: 19 October 2024
Revised: 14 November 2024
Accepted: 15 November 2024
Published: 17 November 2024

Abstract: This study examines how sentiment analysis of environmental, social, and governance (ESG) news affects the financial performance of companies in innovative sectors such as mobility, technology, and renewable energy. Using approximately 9828 general ESG articles from Google News and approximately 140,000 company-specific ESG articles, we performed term frequency-inverse document frequency (TF-IDF) analysis to identify key ESG-related terms and visualize their materiality across industries. We then applied models such as bidirectional encoder representations from transformers (BERT), the robustly optimized BERT pretraining approach (RoBERTa), and big bidirectional encoder representations from transformers (BigBird) for multiclass sentiment analysis, and distilled BERT (DistilBERT), a lite BERT (ALBERT), tiny BERT (TinyBERT), and efficiently learning an encoder that classifies token replacements accurately (ELECTRA) for positive and negative sentiment identification. Sentiment analysis results were correlated with profitability, cash flow, and stability indicators over a three-year period (2019–2021). ESG ratings from Morgan Stanley Capital International (MSCI), a prominent provider that evaluates companies' sustainability practices, further enriched our analysis. The results suggest that sentiment impacts financial performance differently across industries; for example, positive sentiment correlates with financial success in mobility and renewable energy, while consumer goods often show positive sentiment even with low environmental ESG scores. The study highlights the need for industry-specific ESG strategies, especially in dynamic sectors, and suggests future research directions to improve the accuracy of ESG sentiment analysis.

Keywords: environmental social and governance (ESG) news sentiment analysis; mobility sector; corporate financial performance; natural language processing; industry-specific ESG strategies

1. Introduction

The establishment of global sustainability standards and the 2021 Glasgow Climate Pact at the 26th United Nations Climate Change Conference of the Parties (COP26) underscored the importance of environmental, social, and governance (ESG) management for companies worldwide, particularly in sectors such as mobility, technology, and renewable energy [1]. These agreements emphasize the need for companies to actively pursue sustainable value management, aligning environmental and social responsibilities with carbon neutrality goals [2]. The introduction of these standards encourages companies to adopt transparent and responsible management practices, which are expected to significantly impact capital market disclosures and the investment environment, especially in dynamic industries such as the mobility sector.

Reflecting this global trend, South Korea has introduced new requirements for companies listed on the Korea Composite Stock Price Index (KOSPI): those with assets exceeding KRW 2 trillion (South Korean won) will be required to publish sustainability management reports starting in 2025, and this mandate will be extended to all KOSPI-listed companies by 2030 [3]. Although there are currently no legal penalties equivalent to financial reporting standards, these developments highlight the growing importance of ESG management, which requires systematic preparations by companies. However, the lack of clearly defined ESG standards has created confusion among companies and investors, with inconsistencies in the evaluation criteria leading to varying ESG ratings and increasing investor uncertainty [4].

Among various industries, ESG management is particularly critical in the mobility industry due to its significant environmental impacts [5]. With the emergence of innovative technologies such as electric vehicles, autonomous vehicles, and shared mobility services, companies in the sector need to develop targeted ESG strategies that directly address key challenges [6]. From an environmental perspective, companies should prioritize reducing carbon footprints and improving waste management to meet increasing regulatory and consumer demands for sustainable practices. Social responsibility strategies can include fair labor practices and increased community involvement. In addition, strengthening governance through transparent practices, such as implementing regular ESG reporting and establishing board diversity policies, is essential to meet growing stakeholder expectations for accountability [7]. Effective ESG management is inextricably linked to the long-term competitiveness of the mobility sector. To achieve this, it is essential to develop strategies that take into account key ESG drivers such as economic development, the regulatory environment, and responsible investment. Identifying these factors is critical to improving overall ESG practices [8].

This study presents a novel, comprehensive, cross-industry approach to ESG sentiment analysis using multiple natural language processing (NLP) models, building on existing research to explore an innovative method not yet fully developed [9]. By examining ESG sentiment across different sectors, each with unique ESG impacts and challenges, this study addresses a significant gap in the literature, as existing studies typically focus on a single industry or rely on a single-model approach to sentiment analysis [10,11]. In doing so, this study aims to analyze global ESG management trends and investigate the correlation between financial performance and ESG-related news sentiment across various industries, including the mobility, technology, and renewable energy sectors.

To achieve this, we compare and analyze sentiment models such as bidirectional encoder representations from transformers (BERT), decoding-enhanced BERT with disentangled attention (DeBERTa), financial BERT (FinBERT), and RoBERTa, using state-of-the-art NLP techniques to assess correlations with corporate financial indicators [12]. Specifically, we collected, verified, and evaluated data for companies in these sectors from approximately 700 companies listed on the New York Stock Exchange (NYSE), the Nasdaq Stock Market (NASDAQ), and the American Stock Exchange (AMEX) that have received ESG ratings from MSCI. This analysis enables a systematic examination of the impact of ESG-related sentiment on financial performance [13].

Building on previous research [9], the current study extends the analysis by integrating nine advanced NLP and sentiment analysis models along with a comprehensive set of financial indicators across multiple industries. Unlike previous studies, which often rely on limited models or specific financial metrics, this research uses a multimodel, multifaceted approach to provide a more comprehensive view of the impact of ESG sentiment on financial performance [10,11]. This approach not only increases the robustness of our findings but also addresses critical limitations of previous studies by providing a nuanced framework for evaluating ESG strategies across sectors.

Specifically, this study addresses the following research questions:

1. How does positive or negative ESG-related news sentiment affect a company's financial performance across different industries, especially in the mobility, technology, and renewable energy sectors?
2. What specific patterns emerge in each industry regarding the impact of ESG sentiment on financial outcomes, and how do these patterns differ among sectors?
3. How effective are different NLP models in capturing the complex relationships between ESG factors (environmental, social, and governance) and specific financial metrics like profitability, cash flow, and stability?

To address these questions, we offer the following key contributions:

- We develop a novel, multimodel NLP framework that applies nine advanced sentiment analysis models to a large dataset of ESG-related news articles, providing unprecedented depth in understanding the interplay between ESG sentiment and corporate financial performance across multiple industries.
- We uncover and characterize distinct industry-specific patterns in how ESG sentiment affects financial outcomes, demonstrating that the influence of ESG factors is not uniform but varies significantly between sectors, thus emphasizing the importance of customized ESG strategies.
- We critically evaluate the performance of different NLP models in capturing complex ESG–financial relationships, identifying the most effective approaches for nuanced sentiment analysis, and contributing methodological advancements to the field of ESG research.

By integrating a comprehensive cross-industry analysis with advanced NLP techniques, our study offers fresh insights into the multifaceted interactions between ESG sentiment and financial performance, thereby filling critical gaps in the existing literature and informing more effective ESG management strategies [14].

The remainder of this paper is organized as follows: Section 2 provides a review of existing studies on ESG management, sentiment analysis, and its relevance to industries such as mobility, establishing the theoretical framework. Section 3 presents the research findings and discusses key insights regarding ESG management and financial performance across different sectors. Section 4 details the data collection and analysis methodology, introducing the sentiment and financial correlation analysis approaches. Finally, Section 5 summarizes the primary contributions and offers recommendations for future research directions.

2. Related Work

The analysis of ESG factors plays a crucial role in evaluating corporate sustainability and financial performance, and recent studies have increasingly employed NLP techniques to gain deeper insights into ESG data [15]. Existing research primarily focuses on identifying correlations between ESG sentiment and financial performance or examining specific ESG themes using single-model approaches. Table 1 provides a summary of the key contributions from previous studies in ESG sentiment analysis and financial correlation, along with their distinctions from the present study.

Table 1. Overview of prior studies on ESG sentiment analysis and financial correlation in environmental, social, and governance contexts.

Authors	Year	Objective/Purpose	Method and Analytical Tools	Key Findings	Differences from the Present Study
Raman et al. [16]	2020	Analyzed linguistic patterns in ESG topics across corporate earnings calls by industry.	Utilized neural models to classify ESG discourse in earnings calls.	Identified significant industry-specific ESG discourse patterns.	Focused on industry-specific ESG discourse using neural models without a multimodel comparison across ESG factors.
Perazzoli et al. [10]	2022	Examined structural challenges in ESG topics using a systems theory approach.	Applied a systems theory approach to analyze structural ESG issues.	Highlighted challenges within energy and governance themes across industries.	Emphasized single-model analysis of structural ESG issues, lacking multimodel analysis and detailed financial metrics correlation.
Pasch and Ehnes [17]	2022	Enhanced ESG classification performance by fine-tuning transformer models on ESG-related data.	Fine-tuned BERT model on ESG data, achieving higher accuracy.	Achieved 11% higher accuracy than traditional classifiers for ESG sentiment.	Focused on specific model adjustments rather than a comprehensive model comparison across multiple NLP models and industries.
Mehra et al. [11]	2022	Developed an ESG-specific language model to improve document classification accuracy.	Customized an ESG-BERT model fine-tuned on ESG-specific corpora.	Improved accuracy in ESG classification tasks through domain-specific model tuning.	Lacked a multidimensional comparison across ESG domains and did not assess correlations with financial metrics.
Park et al. [18]	2022	Investigated the relationship between public sentiment and corporate resilience using Twitter data.	Analyzed Twitter data to assess public sentiment related to ESG topics.	Found that ESG sentiment on Twitter can be an indicator of corporate resilience during crises.	Utilized a single sentiment analysis model focused on public sentiment without examining detailed financial performance metrics.
Yu et al. [19]	2023	Explored the effect of ESG sentiment on stock price stability.	Examined the correlation between ESG sentiment and stock price volatility.	Identified significant influence of ESG sentiment on stock stability.	Emphasized the correlation with stock stability, limited to specific financial metrics and lacking industry-specific analysis.
Kim et al. [9]	2024	Examined the interconnections between ESG financial trends and sentiment analysis of ESG-related news from 2019 to 2022.	Applied sentiment analysis models to ESG news articles and correlated findings with financial trends.	Identified key relationships between ESG news sentiment and financial performance indicators.	Limited to single-model analysis and specific financial metrics, without a comprehensive cross-industry comparison.
This Study	2024	Utilized nine natural language processing models for ESG sentiment analysis, mapping their relationships to financial performance across industries.	Employed nine NLP models for sentiment analysis of ESG-related news and TF-IDF for key term extraction, examining correlations between sentiment scores and financial performance.	Identified that industry-specific ESG strategies contribute to financial stability, highlighting the importance of ESG practices in sectors like renewable energy and mobility.	Conducted a multimodal comparison, examining diverse correlations between ESG sentiment and detailed financial metrics across multiple industries.

Raman et al. [16] used neural models to analyze ESG-related discourse in corporate earnings calls, identifying how ESG issues vary in importance across industries and examining their direct impact on business operations. However, their focus was primarily on industry-level discourse patterns with a limited scope on financial performance implications, leaving the nuanced financial impact of ESG sentiment unexplored. Perazzoli et al. [10], on the other hand, provided a broad literature analysis of ESG issues, covering structural challenges such as energy management, labor practices, and governance ethics. While this study provided a comprehensive view of ESG challenges, it was limited by its qualitative approach, which lacked model-based sentiment analysis and insights into the quantifiable impact on financial metrics.

Pasch et al. [17] and Mehra et al. [11] contributed to the development of ESG-specific NLP models by creating models, such as ESGBERT, tailored to improve classification accuracy and interpretability in ESG contexts. Pasch's model [17] provided high classification accuracy, while Mehra's work [11] emphasized interpretability. However, both studies are limited by their single-model focus, as they did not perform comparative analyses across multiple models to assess the variability in ESG sentiment capture. This limits their insights into model performance across ESG themes and sectors.

In contrast, studies by Park et al. [18] and Yu et al. [19] examined the role of ESG sentiment in financial stability, highlighting the link between public ESG sentiment and corporate resilience. Park et al. [18] focused on the role of ESG sentiment in corporate resilience during crises, while Yu et al. [19] emphasized its effect on stock price stability. Despite these valuable findings, both studies relied on limited sentiment analysis models that may not capture the diverse and sector-specific sentiments present in different ESG contexts.

In addition, previous research [9] examined ESG sentiment trends and financial implications from 2019 to 2022 using methodologies such as term frequency-inverse document frequency (TF-IDF), latent dirichlet allocation (LDA) topic modeling, valence-aware dictionary and sentiment reasoner (VADER), and BERT. While this research provided basic insights into how ESG sentiment impacts financial metrics, it was limited by single-model analysis and a limited focus on specific financial metrics, thus lacking a broad, comparative perspective.

In summary, previous research has made significant advances in the field of ESG sentiment analysis, but each study is often limited by reliance on a single model, a narrow thematic or sector focus, or qualitative approaches that lack quantifiable insights into financial performance. Most studies emphasize either the development of specific NLP models or the examination of isolated ESG issues, resulting in a partial understanding of ESG impacts.

In contrast, our study adopts a comprehensive multimodel approach, applying nine advanced NLP models, including BERT and RoBERTa, to conduct a thorough analysis of ESG sentiment across multiple industries. This comparative model approach not only allows us to evaluate the effectiveness of different models but also enables a nuanced exploration of the relationships between ESG sentiment and various financial performance indicators across industries.

By including different industries and multiple sentiment models, our study captures the multifaceted impact of ESG on corporate sustainability and fills gaps in model comparison, industry-specific analysis, and the financial impact of ESG sentiment. This research thus fills a critical gap by providing a cross-sector, model-comparative analysis that enhances our understanding of how ESG sentiment impacts financial performance in different contexts. Through this innovative approach, we contribute practical guidance for industry-specific ESG strategy development and a more refined theoretical framework that incorporates sentiment as a central component in assessing ESG impacts on financial sustainability.

3. Methodology

This study takes a quantitative research approach, using statistical and machine learning techniques to analyze the correlation between ESG sentiment and financial performance across industries. By adopting this framework, we aim to provide objective and measurable insights into the impact of ESG sentiment on corporate financial outcomes. Specifically, we apply several NLP models, including advanced models such as BERT, RoBERTa, and so on, to perform sentiment analysis on ESG-related news and examine how sentiment scores correlate with financial indicators across industries. Our methodology consists of three main steps: identifying key ESG themes with TF-IDF analysis, applying multimodel NLP approaches to sentiment classification, and performing correlation analysis to quantify relationships between sentiment scores and financial metrics such as profitability, cash flow, and stability. This structured, data-driven approach enables a robust exploration of ESG impacts across sectors, providing comprehensive cross-industry insights into ESG-related financial dynamics.

3.1. Identifying Key ESG Topics with TF-IDF Analysis

To assess the importance of specific ESG-related keywords in headlines and leadlines, we used the TF-IDF method, which calculates the relevance of each word based on its frequency in the text and rarity across the dataset to accurately extract primary topics [20].

- We cleaned the text data by removing unnecessary symbols, numbers, and stopwords. Stemming and lemmatization were applied to maintain word consistency. This preprocessing improves the quality of the data and ensures that the analysis focuses on meaningful and relevant terms.
- TF-IDF weights were calculated for the preprocessed data, reflecting each word's relative importance in the document. This helped identify the frequency and relevance of the ESG keywords, transforming them into high-weight terms representative of the primary topics.
- Words with high TF-IDF weights were extracted to identify ESG-related themes and corporate reputational topics and served as the basis for subsequent analyses. This extraction allows for a focused examination of the most important ESG issues affecting company performance.

3.2. Sentiment Classification Across Multimodel NLP Approaches

We employed a diverse set of sentiment analysis models to classify the emotional direction of ESG news, employing nine models (BERT, RoBERTa, BigBird, DistilBERT, ALBERT, TinyBERT, ELECTRA, VADER, and TextBlob) to categorize each news item's sentiment as negative, neutral, or positive. To improve efficiency, lightweight models such as DistilBERT, ALBERT, and TinyBERT were applied with binary (positive/negative) classification capabilities, and ELECTRA was included as an additional binary model due to its high performance and fast processing speed in distinguishing fine-grained differences between sentiments.

Larger models such as BERT, RoBERTa, and BigBird, trained on extensive datasets, provide robust contextual understanding, which improves sentiment classification accuracy across multiclass categories [21]. Conversely, the lightweight models, i.e., DistilBERT, ALBERT, and TinyBERT, allow rapid binary classification across large datasets without sacrificing significant accuracy. For consistency, each model was loaded with pretrained parameters, and the prediction classes were determined based on the model output logits. This approach allows for the evaluation of trade-offs between NLP models, supporting informed model selection for future ESG sentiment analysis.

3.2.1. Multiclass Sentiment Analysis Models

- BERT [22]: BERT is a transformer-based model that analyzes text in both directions to understand the context. The model can identify the meaning of words in a sentence through interactions between words. In this study, we chose Bert-base-cased among various BERT models [23].

- RoBERTa [24]: RoBERTa is a model that performs optimized learning on larger datasets based on the underlying structure of BERT. This model is designed to provide a more precise understanding of various linguistic characteristics. In this work, we used the cardiff-nlp/twitter-roberta-base-sentiment model, which boasts high accuracy, especially in emotion analysis [25].
- Big BERT (BigBird) [26]: BigBird was developed to overcome the limitations of conventional transformer models and effectively handle long texts. Using Google/bigbird-roberta-base, this model can identify emotions even when analyzing long and complex ESG news stories without compromising the context. It is particularly effective in emotion analysis in long texts.
- VADER [27]: VADER is a rule-based emotional model suitable for analyzing informal or unstructured text. VADER is particularly strong on informal text, such as social media, and can quickly derive emotional results. This capability enables fast, effective, real-time sentiment analysis of user content.
- TextBlob [28]: TextBlob is a rule-based emotion classification tool that quickly classifies emotions into positives, neutralities, and negatives. In this study, this tool served as a baseline for emotion analysis and provided basic data for comparing performance with pretrained models with VADER.

3.2.2. Binary-Class Sentiment Analysis Models

- Distilled BERT (DistilBERT) [29]: DistilBERT is a lightweight BERT model that provides a faster inference speed while maintaining the performance of BERT. In this study, we chose this model for efficient emotion analysis on large news datasets.
- A Lite BERT (ALBERT) [30]: ALBERT is a model designed to achieve faster processing speeds by reconstructing the parameter structure of BERT. The model is suitable for binary classification in emotion classification tasks, allowing for the fast classification of affirmations or negatives.
- Tiny BERT (TinyBERT) [31]: TinyBERT is a model designed to further reduce the architecture of BERT. The model enables efficient binary emotion classification and performs well in distinguishing between positive and negative emotions.
- Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA) [32]: ELECTRA reduces computational demands using an alternative token prediction method. This model was employed to enhance binary classification performance, specifically in differentiating positive and negative sentiments.

The performance of each model was evaluated in terms of sentiment classification accuracy, speed, and efficiency, allowing us to identify the best models for ESG-related predictions. In addition, we visualized model prediction patterns using uniform manifold approximation and projection (UMAP), a dimensionality reduction technique that simplifies high-dimensional data for better visualization, to understand high-dimensional embedding data across different models [33,34]. This approach enabled us to examine the patterns and variations in sentiment predictions across models, enhancing the interpretability and consistency of the sentiment classification results.

3.3. Correlation Analysis

To explore the relationships between ESG sentiment scores and financial performance across industries, we conducted a correlation analysis to quantify the relationships between sentiment categories (i.e., positive, neutral, and negative) and financial metrics, including profitability, cash flow, and stability. This structured approach facilitates an objective examination of how ESG sentiment can influence financial outcomes across sectors.

- Data aggregation and sector classification: sentiment scores were first categorized by sentiment type and then aggregated by industry to enable sector-specific analysis [35]. In this step, we calculated average sentiment scores for each article within a sector and compared these scores to the financial indicators of companies in that sector. Aggregating the data at the industry level allowed us to capture unique sectoral patterns and interpret the varying impact of ESG sentiment across industries, such as mobility and renewable energy.

- Analytical approach: in this correlation analysis, we compared aggregated sentiment scores to corresponding financial indicators, using a consistent correlation coefficient to measure the alignment between sentiment and financial metrics [36]. This approach provided a robust framework for examining how changes in sentiment may be correlated with financial changes within each sector. By identifying these relationships, we aimed to provide a systematic view of how ESG sentiment correlates with key financial metrics across industries.

This correlation analysis framework thus provides the basis for interpreting the sector-specific findings in the Results section, allowing us to explore the relevance of ESG sentiment to financial outcomes on an industry-specific basis.

4. Data Acquisition and Preparation

This section is divided into two parts. Section 4.1 outlines the methodologies and sources used to obtain ESG-related news and financial data. Section 4.2 describes the procedures employed to purify and consolidate the data for analytical purposes.

4.1. Data Collection

To examine global trends in ESG management, we used the Google News platform with English as the language setting and "ESG finance" as the primary search term [37]. We chose "ESG finance" to capture articles that specifically discuss the intersection of ESG issues and corporate financial performance, ensuring that the collected articles were highly relevant to our research objectives. In the initial data collection phase, all articles matching this keyword were collected without additional filtering or categorization by type (e.g., hard news, editorials, and opinion pieces), which allowed us to collect relevant online news articles from a broad, international perspective for contextual analysis of ESG sentiment related to corporate financial performance [38]. The collection period spanned from June 2019 to May 2022 and was chosen to capture changes in ESG activity and sentiment before, during, and after the COVID-19 pandemic, allowing for analysis of changes in ESG-related news coverage over time [39]. We used a web crawling approach, dividing the collection process by month to minimize data loss associated with large-scale web crawling [40].

This method yielded a total of 9828 English-language news articles, including a variety of article types such as opinion pieces, editorials, and hard news, all written in English, which eliminated the need for translation and reduced the risk of meaning loss that can arise during translation processes. This mix of article types, each with its own unique structure and tone, allowed for more nuanced sentiment analysis that fully represented different perspectives. Recognizing the differences in structure and tone among these types, we acknowledged these variations during our analysis to ensure a comprehensive representation of sentiment across all article categories. For sentiment analysis, we focused primarily on the "headline" and "lead" sections of each article, as these sections typically summarize the main content and sentiment of the article. This focus on headlines and leads allowed us to capture key points that reflect public sentiment on ESG issues, while minimizing the inclusion of general background information. Although we did not explicitly follow the journalistic "inverted pyramid" structure, we prioritized these sections because they are generally the most impactful for understanding sentiment, which is particularly relevant for readers familiar with journalistic practices. This targeted approach allowed us to maintain relevance and clarity by prioritizing the sections most relevant to understanding sentiment.

Sentiment classification was performed using a variety of NLP models to maintain consistency and objectivity across the dataset. For multiclass classification, BERT, RoBERTa, and BigBird were used to classify sentiment into positive, neutral, and negative classes. For binary classification, DistilBERT, ALBERT, TinyBERT, and ELECTRA were used to efficiently distinguish positive from negative sentiment. In addition, VADER and TextBlob provided rule-based sentiment scoring for fast and consistent analysis, especially for unstructured text. By employing these predefined model functions and standardized criteria, sentiment interpretation remained consistent; thus, an additional interrater reliability test was deemed unnecessary due to the automated nature of these models. In addition, this foundational dataset was complemented with news articles specifically related to individual companies' ESG activities to investigate the impact of ESG efforts on financial performance. Throughout the data collection process, we monitored for special cases such as duplicate articles or irrelevant content, addressing them by removing duplicates and filtering out non-relevant articles to enhance the dataset's quality and representativeness. Although the dataset may not encompass every article on corporate financial performance, it provides a substantial and representative cross-sectional view of ESG narratives relevant to financial outcomes over the specified period.

For the company-specific news analysis, we selected 773 U.S.-listed companies with ESG ratings from MSCI [41]. These companies are listed on major exchanges, including the (NYSE), NASDAQ, and AMEX. News articles for each company were gathered from Google News using a combination of the company name and "ESG" as search terms (e.g., "Apple ESG"). This approach was designed to capture a comprehensive view of the ESG-related media coverage. To ensure thorough data acquisition, we employed Selenium WebDriver and ChromeDriver for dynamic web content crawling, retrieving up to 30 pages of news articles per company [42]. Through this process, we acquired over 140,000 articles, which formed a rich dataset on company-specific ESG activities. These data points provided a wide-ranging view of each company's ESG-related actions, which we subsequently analyzed to discern the sentiment direction and examine its correlation with financial performance.

To analyze the financial performance of each company, financial data spanning from 2019 to 2021 were collected, corresponding to the news data collection period to maintain temporal consistency and improve analytical accuracy. Financial data were obtained from Yahoo Finance, focusing on key indicators that provided a comprehensive assessment of a company's profitability, cash flow, and stability [43]. Specifically, for profitability, we gathered revenue, revenue growth rate, and return on assets (ROA). For cash flow, we included earnings before interest, taxes, depreciation, and amortization (EBITDA) and its growth rate. Finally, for stability, we incorporated interest expense, interest expense growth rate, and debt-to-equity ratio. These indicators collectively provided insights into each company's financial health, enabling a multidimensional view of financial performance.

Data extraction from Yahoo Finance was accomplished using the BeautifulSoup library (version 4.12.2) to parse HTML content and extract relevant information systematically [44]. Each company's financial data were then organized for analysis, aligning financial and sentiment data temporally by year to ensure consistency. To assess ESG management performance, we collected ESG rating data from MSCI, including an overall ESG score and specific scores for ESG aspects [45]. These rating data served as an essential benchmark, providing a professional assessment of a company's ESG activities, which we compared against public sentiment derived from news data. Analyzing these ratings relative to sentiment data allowed us to investigate the alignment between public perception and professional ESG evaluations, potentially uncovering discrepancies that may impact investment decisions.

4.2. Data Preprocessing and Integration

Once the data were collected, we performed comprehensive text preprocessing to enhance the quality and relevance of the dataset for sentiment analysis. First, we standardized all text data by converting them to lowercase letters, which reduced case-based inconsistencies [46]. We then employed regular expressions to remove punctuation, numbers, special characters, and non-English characters, thereby ensuring a focus on meaningful words [47]. To address abbreviations, common contractions (e.g., "don't") were expanded to their full forms (e.g., “do not”) to preserve semantic accuracy. Stopwords, which carry minimal semantic value, were removed using the Natural Language Toolkit (NLTK, version 3.9.1) stopword dictionary [48]. Tokenization was performed via NLTK’s word_tokenize function, segmenting the text into individual words for more granular analysis [49]. Finally, where appropriate, stemming or lemmatization was applied to consolidate word forms, grouping variations (e.g., “run” and “running”) under a single root word [50]. This preprocessing stage was vital in enhancing the analytical robustness of the dataset, allowing for more reliable and accurate insights into ESG sentiment trends.

For comparative analysis, the financial and sentiment datasets were integrated at the company level and organized by year. For example, news data from 2020 were analyzed, along with financial data from the same year, facilitating a direct examination of contemporaneous financial and ESG sentiment correlations. Financial metrics were normalized using relative measures, such as revenue growth and EBITDA growth, to account for company size differences and improve cross-company comparability. Recognizing the inherent differences in financial structure across industries, we classified companies into 11 sectors according to the Sustainable Industry Classification System developed by the Sustainability Accounting Standards Board [51,52]. This sectoral categorization enabled industry-specific analysis, accounting for variations in ESG priorities and financial structures that may influence ESG–financial performance relationships. Through these meticulous data acquisition and preparation steps, we established a structured and reliable dataset that formed the foundation for examining the interactions between ESG sentiment, professional ESG assessments, and financial performance.

5. Results and Discussion

This section presents an analysis of the results of our comprehensive study of ESG-related news text. Using TF-IDF analysis, we identified key ESG-related keywords over time, revealed shifts in focus within ESG management, and explored the intricate links between ESG issues and the financial and operational dimensions of companies. For example, the predominance of neutral sentiment observed in the multiclass sentiment analysis is consistent with the findings of Yu et al. [19], who highlighted a trend toward neutrality in ESG news coverage. Our use of both multiclass and binary models allows us to capture more detailed sentiment dynamics across industries, extending the findings of Yu et al. by providing a cross-industry perspective on how neutral sentiment correlates with financial outcomes. In addition, the relationship between neutral sentiment and positive financial outcomes echoes Perazzoli et al.’s [10] findings on how public sentiment can influence corporate reputation, suggesting that balanced or neutral ESG sentiment may promote financial stability by fostering trust and reliability. By linking these findings to previous studies, we provide a broader context for understanding the evolving role of ESG sentiment in financial and operational performance.

5.1. TF-IDF Analysis

First, we performed TF-IDF analysis of the top keywords extracted from headlines and lead text in ESG-related news stories over the entire research period from June 2019 to May 2022. Table 2 shows that environmental terms such as “green”, “greenium”, and “horizon” ranked prominently, highlighting the centrality of environmental concerns in ESG management. Notably, financial terms such as “loans”, “bonds”, “antitrust”, and “investing” also highlight the financial interlinkage in ESG efforts, indicating that ESG management is deeply intertwined with corporate financial practices. This analysis demonstrates that effective ESG strategies integrate environmental and financial practices, which are indispensable for the optimal functioning of a corporation.

Table 2. Summary of key ESG keywords and their TF-IDF weights (June 2019–May 2022).

Rank	Headlines		Leads	
	Keyword	**TF-IDF**	**Keyword**	**TF-IDF**
1	Autos	1	disillusionment	0.886
2	Doctor	1	risk	0.872
3	esg	1	bonds	0.854
4	Green	1	award	0.854
5	greenium	1	excellence	0.820
6	Horizon	1	jbs	0.816
7	Illustrated	1	bgc	0.816
8	Mercy	1	investing	0.812
9	Nordics	1	servicenow	0.808
10	Revealed	1	Loans	0.804
11	Talk	0.990	tigo	0.799
12	antitrust	0.978	materiality	0.798
13	epicenter	0.978	Xylem	0.785
14	abc	0.976	abaxx	0.783
15	operationalize	0.976	ci	0.782

5.1.1. Year-by-Year Analysis of Headlines

The year-by-year breakdown of headline analysis in Table 3 illustrates how the focus of ESG management evolved annually. From June 2019 to May 2020, terms such as “epicenter”, “fails”, and “rip” were prevalent, along with “green”, reflecting initial uncertainty and negative perceptions about early ESG management. Between June 2020 and May 2021, keywords such as “greenium”, “operationalize”, “initiative”, and “importance” became prominent, marking a phase in which ESG management became mainstream. Finally, between June 2021 and May 2022, terms such as “horizon”, “antitrust”, and “tracker” dominated, suggesting that ESG practices were now viewed as not merely initiatives but as impactful factors influencing corporate financial outcomes.

Table 3. Annual trends in ESG keyword emphasis based on headline analysis.

Rank	June 2019–May 2020.05		June 2020–May 2021		June 2021–May 2022	
	Keyword	**TF-IDF**	**Keyword**	**TF-IDF**	**Keyword**	**TF-IDF**
1	autos	1	esg	1	esg	1
2	doctor	1	greenium	1	horizon	1
3	green	1	mercy	1	abc	0.981
4	illustrated	1	nordics	1	antitrust	0.981
5	revealed	1	talk	0.987	putting	0.977
6	epicenter	0.965	crossroads	0.963	tracker	0.977
7	fails	0.962	dna	0.963	accountants	0.974
8	rip	0.959	operationalize	0.963	ready	0.971
9	primer	0.957	initiative	0.954	way	0.967
10	decade	0.955	importance	0.947	war	0.946

5.1.2. Year-by-Year Analysis of Leads

As shown in Table 4, the year-by-year analysis of the lead text revealed governance and financial themes within ESG as major topics, similar to the headlines. From June 2019 to May 2020, governance-related terms such as “disillusionment”, “materiality”, and “investing” featured prominently, indicating the prevalence of governance discussions in the early stages of ESG. Between June 2020 and May 2021, financial terms such as “bonds” and “loans” were dominant, suggesting an increasing intersection between finance and ESG aligned with the emergence of green bonds and ESG investing. This shift underscores the integration of ESG into finance and points to a trend toward sustainable investing.

Table 4. Annual trends in ESG keyword emphasis based on lead analysis.

Rank	June 2019–May 2020		June 2020–May 2021		June 2021–May 2022	
	Keyword	TF-IDF	Keyword	TF-IDF	Keyword	TF-IDF
1	disillusionment	0.874	risk	0.878	award	0.857
2	materiality	0.823	bonds	0.873	servicenow	0.793
3	investing	0.805	loans	0.820	jbs	0.793
4	tigo	0.782	excellence	0.816	director	0.781
5	director	0.775	director	0.783	xylem	0.781
6	trade	0.737	citi	0.769	bgcr	0.761
7	msci	0.734	ci	0.762	lgbtq	0.754
8	ocean	0.731	keamy	0.759	abaxx	0.750
9	stocks	0.718	vale	0.756	wanda	0.749
10	spy	0.717	regulations	0.741	assurance	0.749

This period also saw terms such as “regulations” reflecting heightened discussions on environmental regulation and sustainability legislation. Between June 2021 and June 2022, with the occurrence of the inaugural ESG awards, corporate names (such as ServiceNow, JBS, Xylem, and Abaxx) emerged more frequently than specific keywords related to environment, society, and governance. Social terms such as “LGBTQ” also first appeared, highlighting the expanded scope of ESG into social responsibility. This expansion demonstrates that modern ESG practices are becoming more comprehensive, addressing a broad range of social and corporate responsibilities.

5.1.3. Industry Sentiment Trends in ESG Keywords

The results are consistent with previous studies on the importance of environmental and financial terms in ESG-related discourse. For example, Raman et al. [16] analyzed corporate earnings calls and found a similar emphasis on environmental and financial terms, which is consistent with our findings, where keywords such as “green”, “greenium”, and “bonds” consistently ranked high in TF-IDF scores. This suggests a shared focus in corporate and public discourse on the financial and environmental aspects of ESG, particularly in industries heavily influenced by sustainability demands.

In addition, Pasch and Ehnes [17] highlighted the benefits of fine-tuning NLP models to improve performance in ESG-related text classification. We adopted this approach to improve accuracy in capturing critical ESG terms. While Pasch’s study demonstrated model effectiveness in a specific ESG context, our broader application across multiple industries underscores an even more robust ability to capture ESG themes through TF-IDF analysis. This cross-sector perspective goes beyond the work of Pasch and Ehnes [17] to provide a more comprehensive view of how ESG keywords differ across industries and their impact on financial performance.

In contrast to previous single-focus studies, this analysis incorporates multiple ESG dimensions to account for different sector impacts. For example, the increased prominence of terms such as “operationalize” and “initiative” between 2020 and 2021 echoes the findings of Yu et al. [19], who found that ESG management initiatives are gradually gaining mainstream acceptance. However, our study goes beyond Yu et al. by examining how specific terms evolve and correlate with financial indicators over multiple years, revealing industry trends in response to ESG pressures.

To further interpret these findings, it is important to highlight shifts in sentiment and keyword prominence over time. For example, the rise of terms such as “antitrust” and “tracker” in the later period (2021–2022) reflects a shift in the industry to view ESG not only as a set of initiatives but also as an essential driver of corporate accountability and performance. This shift is consistent with Perazzoli et al. [10], who argued that public sentiment has a significant impact on corporate reputation. Our year-over-year keyword analysis supports this perspective, showing an increasing public and financial focus on accountability through keywords such as “antitrust” and “tracker”.

5.2. Sentiment Analysis

In this study, we leveraged various sentiment analysis models to assess sentiment in ESG-related news headlines and leads. Multiclass models (BERT, RoBERTa, VADER, TextBlob, and BigBird) were used to classify sentiment as negative, neutral, or positive, and binary classification models (DistilBERT, ALBERT, TinyBERT, and ELECTRA) categorized sentiment as either positive or negative.

5.2.1. Multiclass Sentiment Analysis of Headlines

For the headline analysis, we employed multiclass sentiment models, namely, BERT, RoBERTa, VADER, TextBlob, and BigBird, to classify sentiment into negative, neutral, and positive categories. As shown in Figure 1, neutral sentiment was the most common across the models, followed by positive sentiment, with negative sentiment being the least common. Notably, the RoBERTa, TextBlob, and BigBird models exhibited a stronger tendency toward neutral sentiment, indicating a generally neutral tone in ESG news coverage, with an occasional positive emphasis.

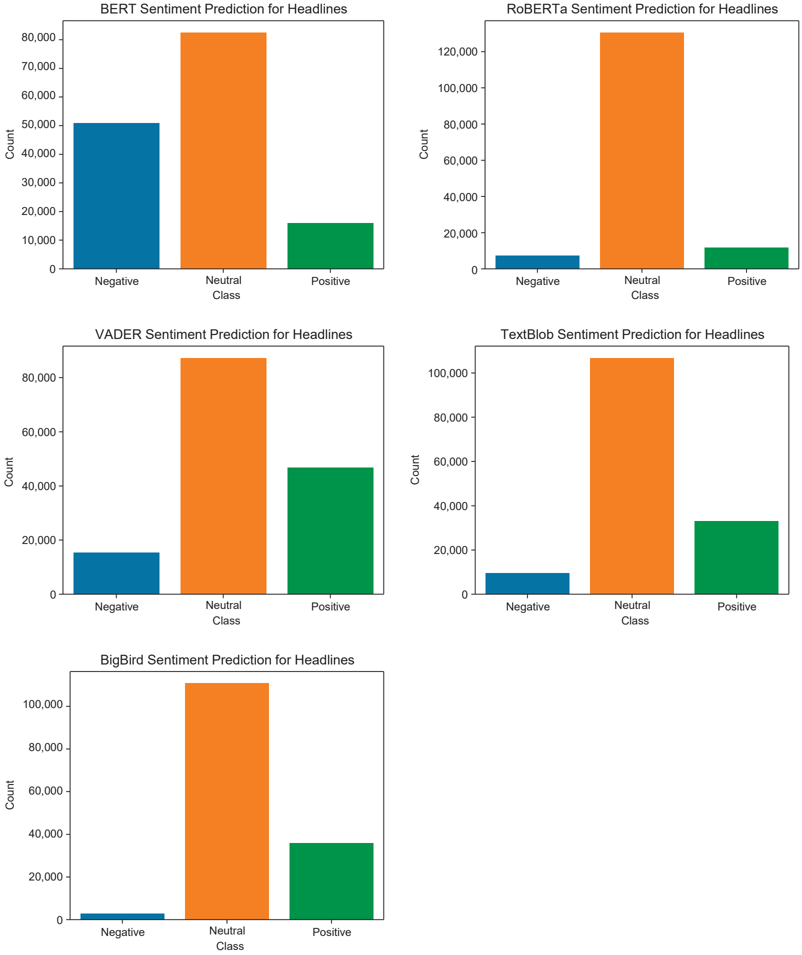


Figure 1. Multiclass sentiment analysis of ESG news headlines.

5.2.2. Binary Sentiment Analysis of Headlines

Binary classification models, namely, DistilBERT, ALBERT, TinyBERT, and ELECTRA, were applied to classify ESG news headlines as positive or negative. As shown in Figure 2, DistilBERT, ALBERT, and TinyBERT exhibited a relatively balanced distribution of positives and negatives, suggesting that they reflect the overall tendency of the text in the context of being classified solely as positive or negative. Conversely, ELECTRA detects negative emotions at a high rate, indicating that different models yield varying emotional outcomes depending on the overall tone of the data.

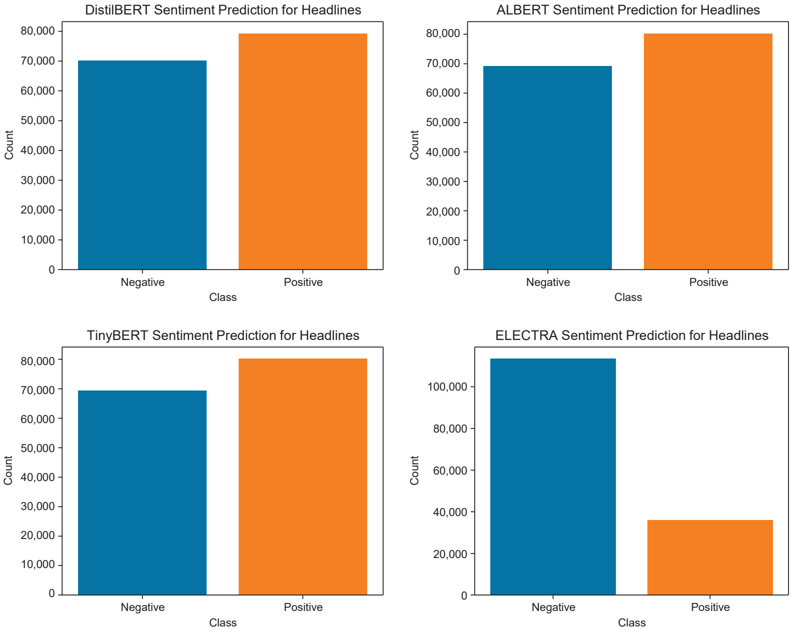


Figure 2. Binary sentiment analysis of ESG news headlines.

5.2.3. Multiclass Sentiment Analysis of Leads

Applying multiclass models to leads revealed a sentiment distribution similar to that of the headlines. As shown in Figure 3, VADER and TextBlob exhibited stronger inclinations toward positive sentiment, whereas BERT and RoBERTa retained higher proportions of neutral sentiment. However, BigBird recorded the highest levels of positive sentiment in the lead text, likely reflecting the additional context within the leads that emphasizes positive ESG-related content.

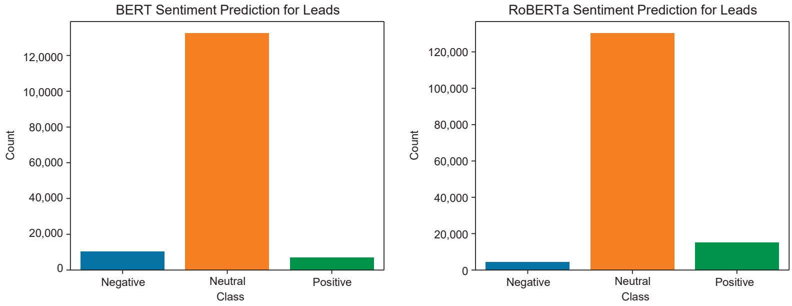


Figure 3. *Cont.*

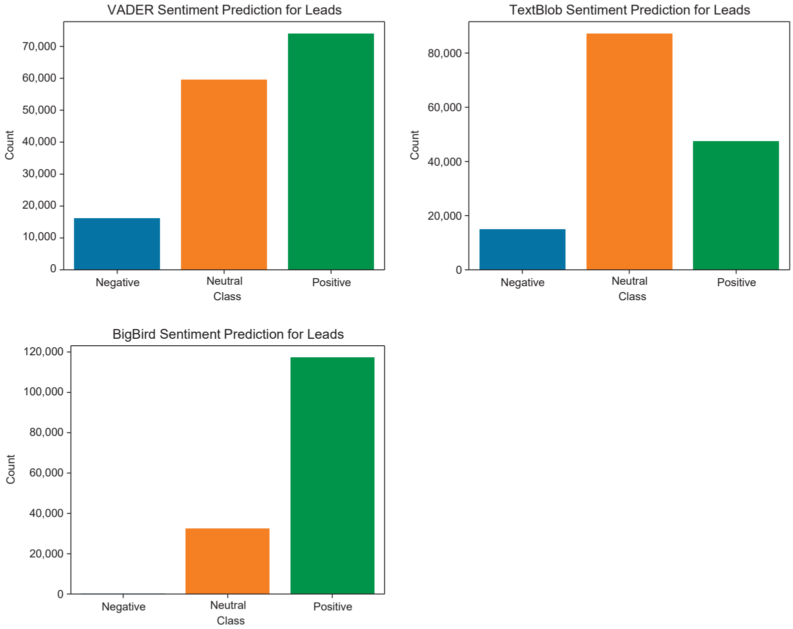


Figure 3. Multiclass sentiment analysis of ESG news leads.

5.2.4. Binary Sentiment Analysis of Leads

The binary sentiment analysis of leads using DistilBERT, ALBERT, TinyBERT, and ELECTRA mirrored the trends observed in headlines. As shown in Figure 4, the ELECTRA model once again exhibited a higher tendency to classify content as negative, possibly because of a lower threshold for neutral categorization. In contrast, DistilBERT and ALBERT demonstrated a relatively balanced sentiment distribution, suggesting that the additional context in the leads typically highlights more positive information.

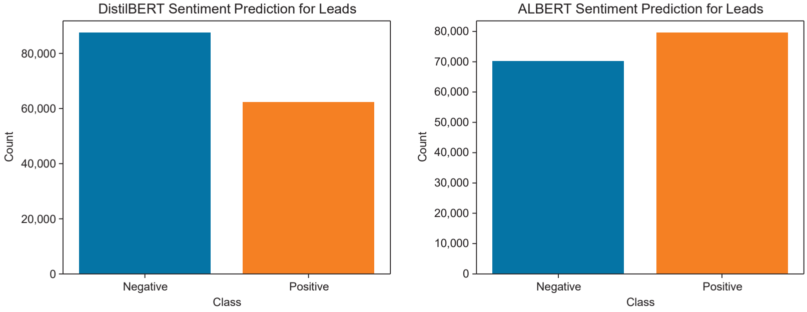


Figure 4. *Cont.*

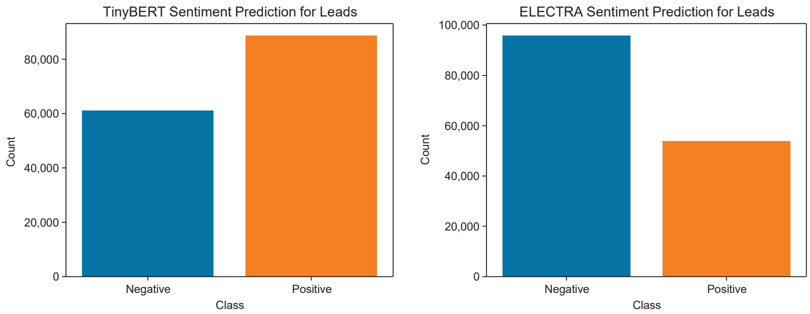


Figure 4. Binary sentiment analysis of ESG news leads.

5.3. Visualize the Financial Link to ESG

This section presents a UMAP visualization to intuitively illustrate the relationships between ESG-related keywords and their financial and environmental implications. UMAP, known for its efficiency in handling large datasets, excels at simplifying complex, high-dimensional data into manageable, low-dimensional representations. This allows us to capture the non-linear data structures in ESG news content and highlight clustering patterns among the extracted keywords.

In this analysis, the UMAP parameters were optimized to enhance the clarity of keyword clustering related to ESG news. Specifically, the n_neighbors parameter was set to three to emphasize local connectivity among ESG and financial keywords, creating tightly clustered groups within specific topics by focusing on the top 100 keywords [53]. In addition, a low min_dist value of 0.01 was selected to encourage close linkages among keywords, revealing distinct clusters that represent various ESG themes.

- The headline UMAP analysis (Figure 5) reveals a clear distribution of keywords across environmental and financial themes. Words such as "sustainable", and "green" are positioned closely to financial terms such as "earnings", "shares", and "stocks", suggesting a close association between ESG management and financial performance. This clustering visually highlights how ESG-related environmental factors are directly connected to corporate financial operations, providing an intuitive overview of their interrelations.
- In the lead UMAP analysis presented in Figure 6, keywords related to ESG and socially responsible management are prominently clustered, along with financial terms such as "equity", "funds", and "assets". This arrangement suggests a potential link between the social and governance aspects of ESG and long-term financial outcomes. For example, keywords such as "sustainability", "social", and "governance" align closely with financial terms, indicating that firms prioritizing social responsibility experience positive financial performance.

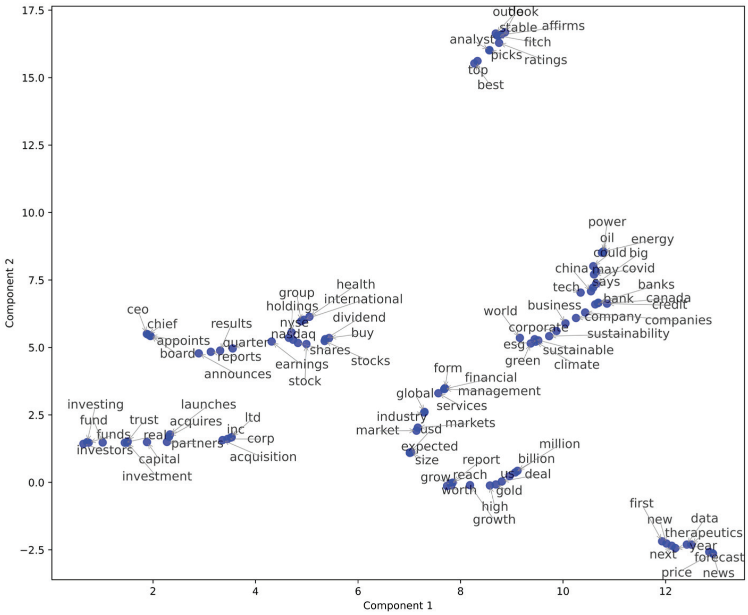


Figure 5. UMAP visualization of ESG keywords in headlines.

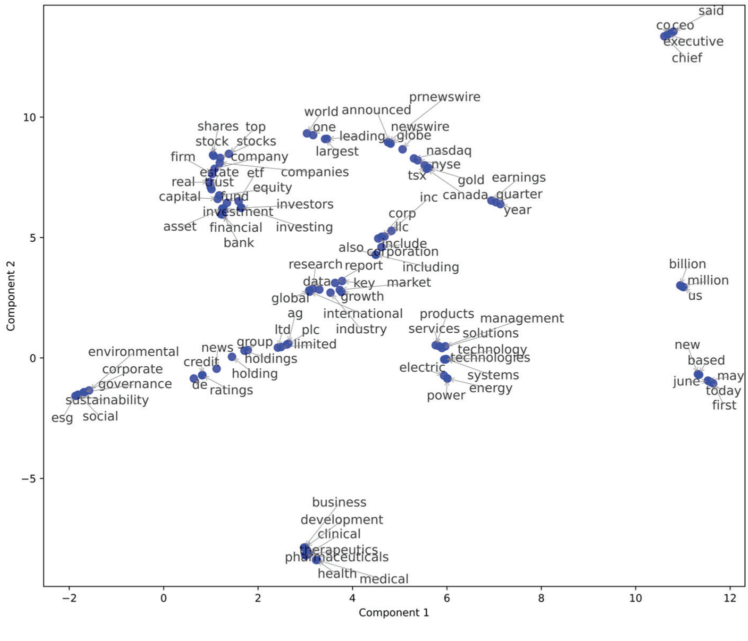


Figure 6. UMAP visualization of ESG keywords in leads.

5.4. Correlation Analysis of ESG Sentiment and Financial Performance

This study analyzes the relationship between sentiment analysis data, ESG ratings (provided by MSCI), and industry-specific financial metrics, focusing on profitability, cash flow, and stability. The primary aim is to identify how ESG-related news sentiment correlates with corporate financial performance across different industries. The analysis explores the impact of ESG ratings on financial indicators and investigates how sentiment analysis results relate to the financial outcomes of companies.

Financial factors are categorized into three areas:

- Profitability: assessed using revenue (2021), revenue growth rate, and ROA (2021) to evaluate a company's earning power. The year 2021 was chosen as the reference year to observe changes in profitability following the pandemic period, allowing a focused analysis of ESG sentiment in the post-pandemic economic environment.
- Cash flow: evaluated through EBITDA and its growth rate (2021), reflecting net cash flow from operations and its growth. Using 2021 data provides insight into operational cash flow stability and growth during the recovery phase post-pandemic.
- Stability: measured using interest expense, interest expense growth rate, and debt-to-equity ratio (2021) as key indicators of financial strength. These metrics indicate a company's ability to manage debt and operational leverage in response to the changing economic conditions after the pandemic.

The analysis spans three years, from 2019 to 2021, with the correlation results visualized through heatmaps in Appendix A (Figures A1–A11). The choice of 2021 as the reference year for these financial metrics is intentional, as it allows us to assess the impact of ESG sentiment on financial performance during a significant post-pandemic economic transition. For Figures A1–A11, each financial variable is coded based on 2021 values to maintain consistency across industries and allow for a clear identification of post-pandemic trends. This approach highlights recent changes in financial performance as companies adapt their ESG practices in response to evolving stakeholder expectations in the post-pandemic period. Growth rates, such as revenue growth and EBITDA growth, reflect annual changes from 2020 to 2021, allowing for year-over-year comparisons in the context of recovery. The ROA metric specifically refers to 2021 values as it provides a focused measure of asset efficiency following the economic disruptions of the pandemic.

The results show different correlations between ESG factors and financial indicators across industries, reflecting the unique characteristics of each sector. Table 5 below provides a consolidated summary of the key findings from our correlation analysis, allowing readers to quickly grasp the key insights. Each industry shows varying degrees of sensitivity to ESG factors; for example, sectors such as mobility and renewable energy show significant impacts from environmental sentiment, while others such as healthcare and financial services show more nuanced relationships. This table allows readers to efficiently locate key findings across sectors. A detailed analysis and visualizations for each sector can be found in Appendix A. These figures (Figures A1–A11) provide in-depth visual insights, including heatmaps and correlation matrices, to illustrate the unique relationships between ESG sentiment and financial performance within each industry.

Table 5. Summary of ESG sentiment and financial correlations by industry sector.

Industry	Key Findings	Figure Reference
Consumer Goods	Positive sentiment; low environmental score; favorable financial indicators	Figure A1
Extractives and Minerals Processing	Strong ESG–governance correlation; profitability; cash flow	Figure A2
Financial	Mixed sentiment effects; positive ESG–financial performance correlation	Figure A3
Food and Beverage	Positive sentiment with ESG; low financial correlation	Figure A4
Healthcare	Positive governance sentiment; limited financial impact	Figure A5
Infrastructure	High environmental score; positive financial correlation; leverage influence	Figure A6
Renewable Resources and Alt. Energy	Strong environmental–financial stability correlation	Figure A7
Resource Transformation	Minimal ESG–financial correlation; high governance score	Figure A8
Services	Varied impacts by service type; mixed ESG correlations	Figure A9
Technology and Communications	Positive environmental–profitability correlation; limited governance impact	Figure A10
Transportation	High ESG–sentiment correlation; environmental sensitivity	Figure A11

5.5. Discussion on Industry-Specific ESG Sentiment and Financial Impacts

This study conducted a sentiment analysis of ESG-related news headlines and leads, examining the relationship between sentiment, company financial performance, and ESG ratings across various industries, including the mobility sector. The analysis highlights correlations between ESG news sentiment and company profitability, cash flow, and stability metrics. The findings indicate that neutral or positive ESG news sentiment is often associated with positive financial outcomes, particularly in sectors where environmental and social factors significantly influence company perceptions, such as mobility and renewable energy. This trend underscores the perspective that positive public sentiment toward ESG can promote corporate resilience and alignment with societal expectations for sustainable practices, thereby fostering accountability and trust among stakeholders. These findings are consistent with previous studies showing that macroeconomic factors and news sentiment significantly influence stock returns [54]. Similarly, social media sentiment around major political events has been found to influence stock markets [55]. The results suggest that the relationship between sentiment and financial performance varies by industry, with environmental and social factors having a significant impact in sectors such as mobility and renewable energy while having a limited impact in other sectors.

From a literature perspective, this study contributes to the growing body of knowledge on ESG sentiment analysis by employing both multiclass and binary models to provide nuanced sentiment metrics across industries. This dual-model approach enhances the depth of ESG analysis and provides new insights into sector-specific sentiment trends, complementing the existing research that typically uses single-model approaches. The link between positive sentiment and positive financial results supports the view that alignment with societal ESG expectations can strengthen both corporate reputation and financial stability. These industry-specific findings reveal distinct patterns of ESG sentiment across industries and complement the work of Park et al. [18], who found that positive public ESG sentiment is correlated with corporate resilience. This study builds on this foundation by examining additional financial metrics, such as cash flow and stability, to provide a deeper understanding of how ESG sentiment influences financial outcomes. In addition, we extend the work of Mehra et al. [11] on the ESGBERT model for improved ESG context capture through our multimodel approach, which improves the accuracy and interpretability of sentiment analysis across different ESG issues. This multimodel comparison addresses the single-model limitation identified in previous research and provides a more comprehensive view of the financial impact of ESG. For example, in the consumer discretionary sector, positive sentiment correlates with strong financial performance despite lower environmental ESG scores, while in the mobility sector, favorable environmental factors directly contribute to financial gains.

From a practical perspective, the use of both multiclass and binary sentiment models allowed for nuanced sentiment metrics, providing deeper insights into the complex impact of ESG sentiment on financial performance that can guide companies in tailoring their ESG practices. The multiclass model identified a high proportion of neutral sentiment, suggesting that ESG news is generally positive or neutral, while the binary model highlighted polarized financial impacts by distinguishing between positive and negative sentiment. This alignment of neutral or positive sentiment with financial stability reflects broader societal priorities for ethical corporate behavior and long-term stability, which is particularly valuable for practitioners seeking to maintain public trust and sustainable growth [56]. The approach is consistent with studies using advanced machine learning techniques for automated stock market trading, demonstrating the value of sophisticated models in financial decision-making. In addition, the TF-IDF analysis uncovered frequently mentioned ESG-related keywords, illustrating the varying importance of ESG factors across industries. Similar to research using machine learning in demand forecasting, this industry-specific analysis improves the effectiveness of ESG strategies by focusing on relevant ESG factors in each sector, helping practitioners develop targeted, impactful strategies [57].

From a societal perspective, these findings underscore the importance of transparent ESG reporting and management for positive public perceptions that can enhance corporate reputation and align with societal goals for sustainability. Such alignment with public expectations for accountability and ethical standards is critical as it builds trust and supports the pursuit of long-term sustainable practices [58]. This highlights the urgent need for companies to tailor their ESG strategies to the specific needs of their industries, particularly in high-impact sectors such as mobility. In addition, the use of advanced machine learning techniques, such as graph neural networks, offers a promising way to improve the predictive power of ESG analysis and provide more accurate and industry-relevant insights [59]. Taken together, these considerations underscore the urgent need for a forward-looking, nuanced approach to ESG management that addresses both societal expectations and industry-specific challenges.

6. Conclusions

This study provides comprehensive insights into the relationship between ESG news sentiment and corporate financial performance, with a particular focus on industries such as mobility. Using both multiclass and binary classification models, we examined how ESG news sentiment affects key financial metrics, including profitability, cash flow, and stability, highlighting the potential for effective ESG management to improve corporate outcomes. Our findings highlight industry-specific variations in the correlation between ESG sentiment and financial performance, underscoring the need for ESG strategies tailored to the unique characteristics of each sector, particularly dynamic sectors such as mobility.

The key findings of this study are as follows:

- Industry-specific effects: different industries show different levels of correlation between ESG sentiment and financial performance. Sectors such as mobility and renewable energy are particularly affected by environmental sentiment, indicating their heightened sensitivity to ESG news and its impact on company results.
- Modeling approach: the use of both multiclass and binary sentiment models allowed for a nuanced analysis of ESG sentiment. The models revealed a high proportion of neutral sentiment in general ESG news while also highlighting the distinct impact of polarized sentiment on financial performance.
- Strategic implications: developing ESG strategies tailored to the unique characteristics of each industry can improve long-term company performance. This is particularly relevant for sectors that are more sensitive to ESG factors, where tailored approaches can better support sustainable growth and stakeholder trust.

For practitioners, this study provides valuable strategic insights for companies seeking to improve long-term performance through tailored ESG management. A key contribution of this study is to demonstrate the feasibility of using sentiment analysis to assess the impact of ESG initiatives. By using sentiment analysis models to examine ESG sentiment indicators across industries and their correlations with financial performance, our findings show that integrating ESG ratings with sentiment analysis can serve as a reliable predictor of corporate outcomes.

In terms of theoretical implications, this study makes several contributions to the ESG and finance literature. First, it demonstrates that ESG sentiment analysis can reveal industry-specific nuances in the relationship between ESG activities and financial outcomes, providing a customized perspective that differs from traditional ESG scoring approaches that may overlook industry-specific factors. This finding extends existing theories of stakeholder perception and corporate performance by incorporating sentiment as a key component in assessing ESG impacts, thereby broadening the theoretical understanding of how sentiment affects corporate success in different contexts.

However, it is important to acknowledge certain limitations of this study. First, the data used for the sentiment analysis were limited to ESG-related news headlines and leads, which may not capture all the nuances of companies' ESG practices. In addition, the observed correlation between ESG sentiment and financial performance does not imply causation, as this analysis relied on contemporaneous data rather than longitudinal datasets. These limitations underscore the need for cautious interpretation of the results and highlight areas for improvement in future research.

To address these limitations, future research should focus on expanding data collection efforts to include a broader range of sentiment sources, such as social media and analyst reports, to provide a more comprehensive view of ESG sentiment. In addition, collecting longitudinal data over longer periods of time would allow researchers to more effectively establish causal relationships between ESG sentiment and financial performance. By incorporating these advanced methodologies, future studies can develop more robust and comprehensive models for ESG analysis, increasing the reliability of insights and contributing to better-informed investment decisions and corporate strategies.

In conclusion, this study presents an analytical methodology that is of practical value to companies developing ESG strategies and represents a significant advance in understanding the complex interplay between ESG sentiment and financial performance, particularly in the mobility industry. Through this research, we highlight the role of sentiment analysis in improving ESG assessments across sectors, ultimately benefiting academia, society, and industry practitioners alike.

Author Contributions: Conceptualization, M.K. and J.K.; methodology, M.K.; software, J.K. and J.L.; validation, I.J. and J.K.; formal analysis, M.K.; investigation, J.P. and S.Y.; resources, J.J. and J.K.; data curation, J.L. and J.P.; writing—original draft preparation, M.K.; writing—review and editing, J.M.; visualization, M.K.; supervision, J.W. and J.M.; project administration, J.W.; funding acquisition, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Research Foundation of Korea (NRF) through the "Regional Innovation Strategy (RIS)" initiative, funded by the Ministry of Education (MOE) (2021RIS-004), and by the Soonchunhyang University Research Fund.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: Jungwon Park, Seulgi Youm, and Jonghee Jeong were employed by the companies, DTaaS, IBK, and Evidnet, respectively. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The companies DTaaS, IBK, and Evidnet had no role in the design of the study, in the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

Appendix A

This section contains comprehensive visualizations that support the findings summarized in the main text. Each figure provides a detailed view of ESG sentiment correlations with financial indicators, tailored to each industry. These figures allow readers to delve into sector-specific patterns and interpret the data within the unique context of each industry's ESG dynamics.

As shown in Figure A1, the consumer goods industry tends to have lower environmental scores (correlation coefficient of -0.12), likely due to the high environmental impacts associated with manufacturing activities. Despite these lower environmental scores, the media coverage of ESG initiatives is overwhelmingly positive, reflecting a positive public perception of companies' ESG practices. Notably, the analysis shows positive correlations between ESG scores and financial indicators such as profitability and cash flow metrics (e.g., correlation coefficients of 0.45 with sales, 0.38 with ROA, and 0.42 with EBITDA), as well as stability metrics. However, the correlation coefficient for stability, specifically the debt ratio, is relatively low (-0.11). This suggests that while ESG scores are generally consistent with financial performance, positive sentiment alone does not ensure financial stability.

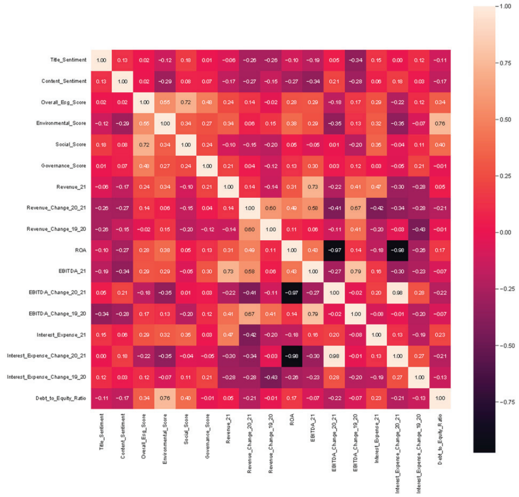


Figure A1. Consumer goods industry correlation analysis.

Furthermore, positive correlations are observed between ESG, environmental, and social scores and the debt ratio (correlation coefficients of 0.34 for ESG, 0.76 for environmental, and 0.40 for social scores), while ESG and environmental scores have an inverse relationship with the interest expense growth rate from 2020 to 2021 (−0.22 and −0.35, respectively). Taken together, these observations suggest that while consumer perceptions of ESG remain positive despite environmental challenges, the relationship between positive sentiment and financial stability is complex and varies across specific financial indicators within the sector.

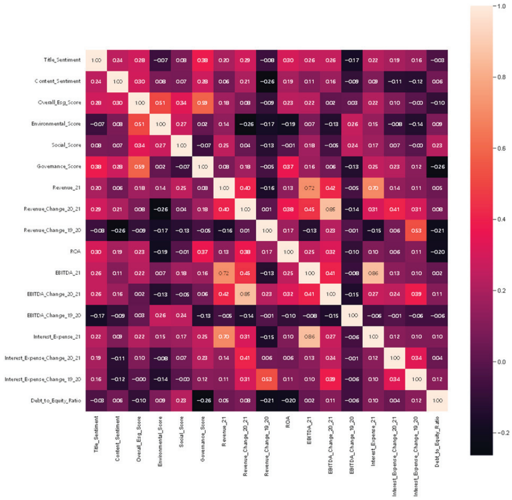


Figure A2. Extractives and minerals processing industry correlation analysis.

In the mining and mineral processing industry, as illustrated in Figure A2, the ESG and governance scores are positively correlated with profitability and cash flow metrics, including sales (correlation coefficient of 0.50), ROA (0.47), and EBITDA (0.52). Higher ESG ratings (average ESG score correlation of 0.28 with positive sentiment) are associated with positive sentiment, indicating that ESG performance, particularly in governance, may influence financial outcomes in this sector. Compared to other industries, this sector exhibits a stronger association between sentiment analysis and profitability and cash flow metrics, as evidenced by the higher correlation coefficients presented in Figure A2. This suggests that companies with higher profitability and cash flow metrics typically have a more positive sentiment direction in news, as reflected by the positive correlations between sentiment scores and financial metrics in Figure A2. Such positive sentiment may influence financial performance in the mining sector, potentially due to stakeholders' emphasis on sustainable practices.

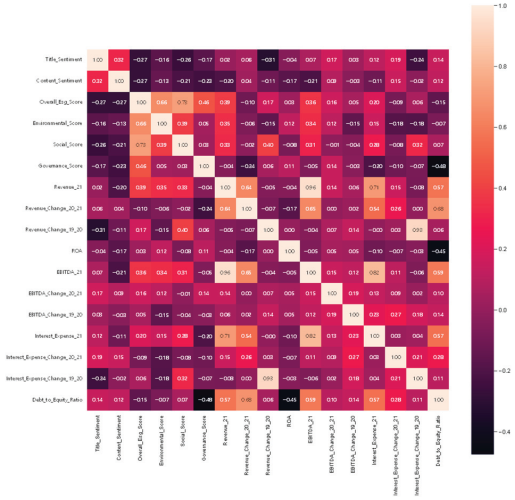


Figure A3. Financial industry correlation analysis.

In the financial industry, as shown in Figure A3, the sector exhibits a negative correlation (correlation coefficient of −0.27) between ESG scores and sentiment analysis scores but a direct positive correlation between overall ESG scores and financial metrics such as profitability and cash flow (correlation coefficients of 0.40 with sales, 0.35 with ROA, and 0.38 with EBITDA). Larger companies tend to exhibit more rigorous ESG management practices, leading to higher ESG scores and improved financial performance. Interestingly, although social and governance scores are central to ESG evaluations, sentiment analysis exhibits negative news sentiment that does not significantly affect social or governance scores (correlation coefficients of −0.26 with social scores and −0.17 with governance scores), indicating that media sentiment has limited influence on actual ESG evaluations in this sector. These associations are depicted in Figure A3, where the negative correlation between ESG scores and sentiment analysis scores contrasts with the positive correlation between ESG scores and financial performance metrics, highlighting the complexity of the relationships in the financial industry.

In the food and beverage industry, as shown in Figure A4, positive correlations are observed between sentiment analysis and ESG, social, and governance indicators (0.40, 0.31, and 0.13, respectively). However, most profitability and cash flow metrics, such as sales, EBITDA, and interest expense, exhibit negative correlations (−0.52, −0.53, and −0.46, respectively). This indicates that although ESG scores may align positively with governance and social aspects, they do not necessarily reflect strong financial outcomes. ESG ratings also exhibit a negative correlation with ROA (−0.20), whereas the debt ratio shows a positive relationship with ESG and environmental ratings (0.35 and 0.36, respectively). Media sentiment aligns with high ESG scores in this consumer-driven sector, although positive news sentiment does not always translate to robust financial performance.

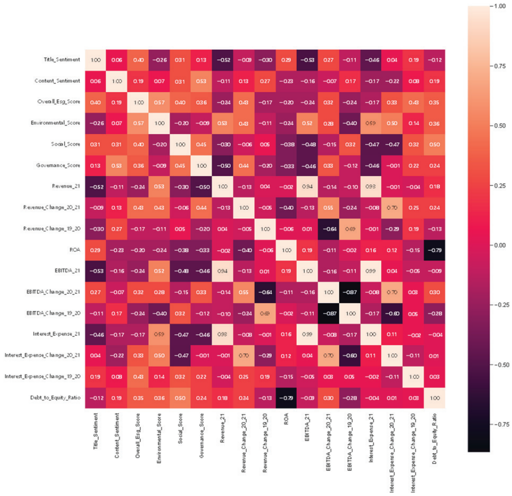


Figure A4. Food and beverage industry correlation analysis.

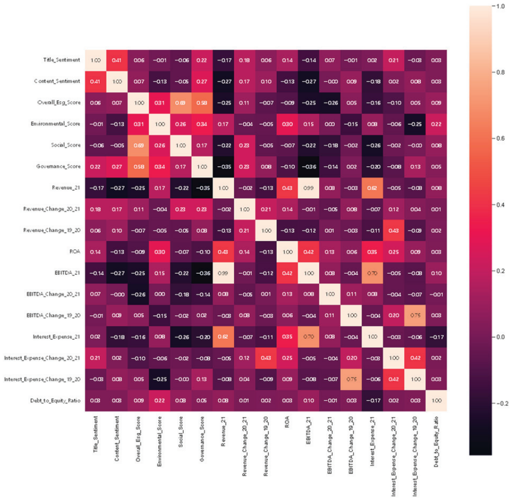


Figure A5. Healthcare industry correlation analysis.

In the healthcare industry, as shown in Figure A5, positive sentiment is closely linked to governance ratings (correlation coefficient of 0.22) but negatively correlated with several profitability and cash flow metrics, including sales (−0.17) and EBITDA (−0.14). This suggests that although positive sentiment may elevate governance ratings, it does not necessarily improve financial performance, highlighting a separation between perceived governance quality and actual financial outcomes in the healthcare sector.

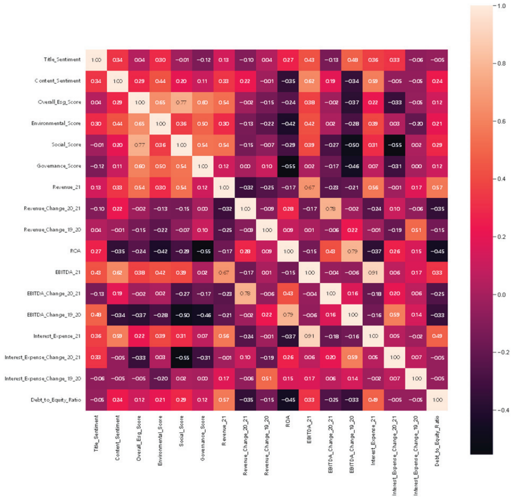


Figure A6. Infrastructure industry correlation analysis.

In the infrastructure industry, as shown in Figure A6, this sector demonstrates a strong positive correlation with environmental scores (correlation coefficient of 0.65), underscoring a focus on environmental management. Higher environmental scores align with positive sentiment and improved financial indicators, such as sales (0.30), EBITDA (0.42), and interest expense (0.39). Although industries with higher leverage exhibit high EBITDA and interest expense metrics, these metrics do not impact the ESG evaluations of the sector, potentially due to the reliance of the industry on high leverage for operations.

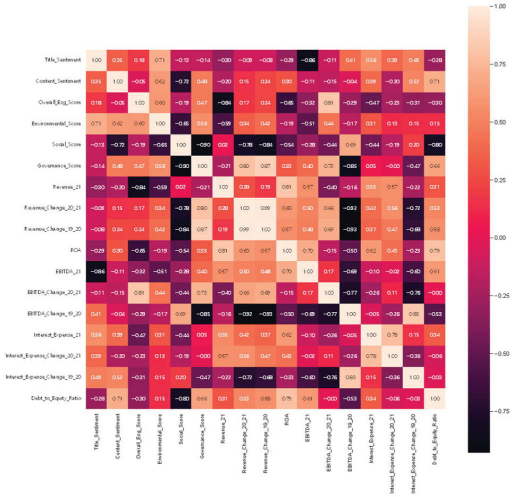


Figure A7. Renewable resources and alternative energy industry correlation analysis.

In the renewable energy industry, as shown in Figure A7, environmental assessments and positive sentiment are strongly correlated (0.71), with positive correlations observed in stability metrics, except for title sentiment and the debt-to-equity ratio (−0.28). This highlights a close relationship between effective environmental management and financial soundness. Although the industry operates with high leverage (reflected in a minimal sentiment correlation with interest expense growth rates), companies with high environmental scores tend to perform well financially, suggesting that environmental management plays a significant role in financial success.

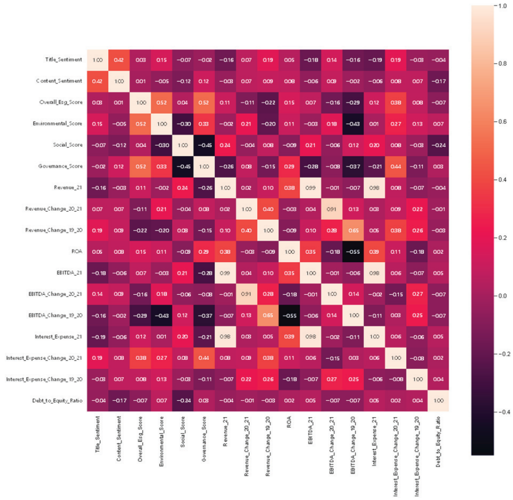


Figure A8. Resource transformation industry correlation analysis.

Within the resource transformation industry, as illustrated in Figure A8, no distinct correlation is observed between ESG ratings and financial metrics, suggesting that this diverse sector lacks a direct relationship between sentiment analysis and financial outcomes. Governance ratings show a negative correlation with cash flow (correlation coefficients of −0.28 with operating cash flow, −0.08 with free cash flow, and −0.37 with cash flow from investments), implying that while governance practices may be well-regarded, they do not necessarily yield strong cash flow in this industry.

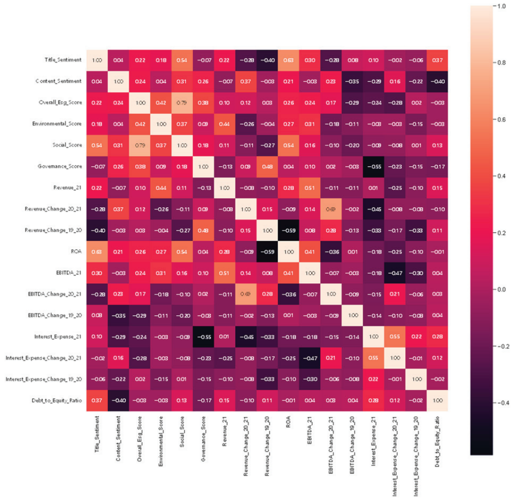


Figure A9. Services industry correlation analysis.

Within the services industry, as shown in Figure A9, the sector exhibits mixed correlations with sentiment analysis and ESG scores, with a notably high ROA value (correlation coefficient of 0.63), depending on service type and operational focus. This variability highlights the heterogeneity in the services industry and suggests that the influence of ESG practices varies significantly across different types of services.

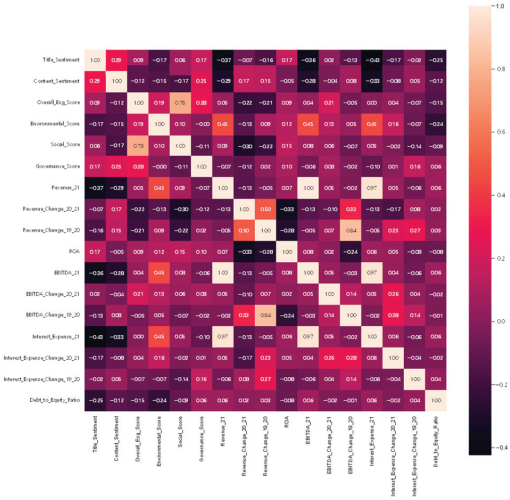


Figure A10. Technology and communications industry correlation analysis.

In the technology and communications industry, as shown in Figure A10, governance scores correlate positively with ESG performance (correlation coefficient of 0.28), and environmental ratings positively affect profitability (correlation coefficient of 0.45). However, there is a negative correlation between profitability metrics (e.g., sales) and governance evaluations (−0.07). Companies with higher environmental scores tend to have favorable financial outcomes even though high governance ratings do not necessarily correlate with profitability, possibly due to the high operational costs associated with governance efforts.

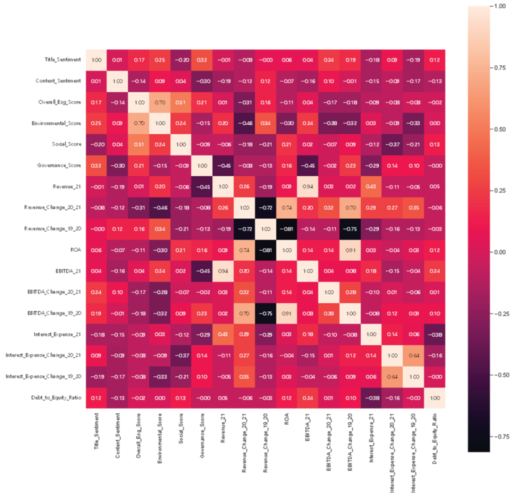


Figure A11. Transportation industry correlation analysis.

As shown in Figure A11, the transportation industry exhibits the highest correlation (0.70) between sentiment analysis scores and ESG ratings among all industries. This likely reflects the characteristics of industries, like renewable energy, that are directly influenced by environmental factors.

References

1. Khan, M.; Serafeim, G.; Yoon, A. Corporate sustainability: First evidence on materiality. *Account. Rev.* **2016**, *91*, 1697–1724. [CrossRef]
2. Kim, S.; Li, Z. Understanding the impact of ESG practices in corporate finance. *Sustainability* **2021**, *13*, 3746. [CrossRef]
3. Sidhoum, A.A.; Serra, T. Corporate sustainable development: Revisiting the relationship between corporate social responsibility dimensions. *Sustain. Dev.* **2018**, *26*, 365–378. [CrossRef]
4. Amel-Zadeh, A.; Serafeim, G. Why and how investors use ESG information: Evidence from a global survey. *Financ. Anal. J.* **2018**, *74*, 87–103. [CrossRef]
5. Dincă, M.S.; Vezețeu, C.-D.; Dincă, D. The relationship between ESG and firm value: Case study of the automotive industry. *Front. Environ. Sci.* **2022**, *10*, 892541. [CrossRef]
6. Whittaker, D.H. *Building a New Economy: Japan's Digital and Green Transformation*; Oxford University Press: Oxford, UK, 2024.
7. Samans, R.; Nelson, J. *Sustainable Enterprise Value Creation: Implementing Stakeholder Capitalism Through Full ESG Integration*; Springer Nature: Cham, Switzerland, 2022; p. 289.
8. Daugaard, D.; Ding, A. Global drivers for ESG performance: The body of knowledge. *Sustainability* **2022**, *14*, 2322. [CrossRef]
9. Kim, M.; Kim, S.; Kim, Y.; Moon, J. Analyzing the financial impact of ESG news sentiment on ESG finance trends. In Proceedings of the International Conference on Platform Technology and Service (PlatCon-24), Jeju, Republic of Korea, 26–28 August 2024.
10. Perazzoli, S.; Joshi, A.; Ajayan, S.; de Santana Neto, J.P. Evaluating Environmental, Social, And Governance (ESG) From a Systemic Perspective: An Analysis Supported by Natural Language Processing. 2022. Available online: https://ssrn.com/abstract=4244534 (accessed on 10 October 2024).
11. Mehra, S.; Louka, R.; Zhang, Y. ESGBERT: Language model to help with classification tasks related to companies' environmental, social, and governance practices. *arXiv* **2022**, arXiv:2203.16788.
12. Cambria, E.; White, B. Jumping NLP curves: A review of natural language processing research. *IEEE Comput. Intell. Mag.* **2014**, *9*, 48–57. [CrossRef]
13. Fatemi, A.; Glaum, M.; Kaiser, S. ESG performance and firm value: The moderating role of disclosure. *Glob. Financ. J.* **2018**, *38*, 45–64. [CrossRef]
14. Oprean-Stan, C.; Oncioiu, I.; Iuga, I.C.; Stan, S. Impact of sustainability reporting and inadequate management of ESG factors on corporate performance and sustainable growth. *Sustainability* **2020**, *12*, 8536. [CrossRef]
15. Friede, G.; Busch, T.; Bassen, A. ESG and financial performance: Aggregated evidence from more than 2000 empirical studies. *J. Sustain. Financ. Investig.* **2015**, *5*, 210–233. [CrossRef]
16. Raman, N.; Bang, G.; Nourbakhsh, A. Mapping ESG trends by distant supervision of neural language models. *Mach. Learn. Knowl. Extr.* **2020**, *2*, 453–468. [CrossRef]
17. Pasch, S.; Ehnes, D. NLP for responsible finance: Fine-tuning transformer-based models for ESG. In Proceedings of the 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 17–20 December 2022; pp. 3532–3536.
18. Park, J.; Choi, W.; Jung, S.-U. Exploring trends in environmental, social, and governance themes and their sentimental value over time. *Front. Psychol.* **2022**, *13*, 890435. [CrossRef]
19. Yu, H.; Liang, C.; Liu, Z.; Wang, H. News-based ESG sentiment and stock price crash risk. *Int. Rev. Financ. Anal.* **2023**, *88*, 102646. [CrossRef]
20. Aizawa, A. An information-theoretic perspective of tf–idf measures. *Inf. Process. Manag.* **2003**, *39*, 45–65. [CrossRef]
21. Cho, S.; Moon, J.; Bae, J.; Kang, J.; Lee, S. A framework for understanding unstructured financial documents using RPA and multimodal approach. *Electronics* **2023**, *12*, 939. [CrossRef]
22. Koroteev, M.V. BERT: A review of applications in natural language processing and understanding. *arXiv* **2021**, arXiv:2103.11943.
23. Shi, P.; Lin, J. Simple BERT models for relation extraction and semantic role labeling. *arXiv* **2019**, arXiv:1904.05255.
24. Delobelle, P.; Winters, T.; Berendt, B. RobBERT: A Dutch RoBERTa-based language model. *arXiv* **2020**, arXiv:2001.06286.
25. Marimuthu, V.K.; Jayaraman, S.; Theik, A.T.; Maple, C. Behavioural analysis of COVID-19 vaccine hesitancy survey: A machine learning approach. In Proceedings of the International Conference on AI and the Digital Economy (CADE 2023), Venice, Italy, 26–28 June 2023; pp. 4–9.
26. Zaheer, M.; Guruganesh, G.; Dubey, K.A.; Ainslie, J.; Alberti, C.; Ontañón, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. Big Bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; pp. 17283–17297.
27. Hutto, C.; Gilbert, E. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media, Ann Arbor, MI, USA, 1–4 June 2014; pp. 216–225.
28. Aljedaani, W.; Rustam, F.; Mkaouer, M.W.; Ghaleb, A.; Rupara, V.; Washington, P.B.; Lee, E.; Ashraf, I. Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry. *Knowl.-Based Syst.* **2022**, *255*, 109780. [CrossRef]

29. Dogra, V.; Singh, A.; Verma, S.; Kavita; Jhanjhi, N.Z.; Talib, M.N. Analyzing DistilBERT for sentiment classification of banking financial news. In Proceedings of the Intelligent Computing and Innovation on Data Science, Kota Bharu, Malaysia, 19–20 February 2021; pp. 501–510.
30. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv* **2020**, arXiv:1909.11942.
31. Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; Li, L.; Wang, F.; Liu, Q. TinyBERT: Distilling BERT for natural language understanding. *arXiv* **2019**, arXiv:1909.10351.
32. Mala, J.B.; SJ, A.A.; SM, A.R.; Rajan, R. Efficacy of ELECTRA-based language model in sentiment analysis. In Proceedings of the 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCS), Coimbatore, India, 7–8 April 2023; pp. 682–687.
33. McInnes, L.; Healy, J.; Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv* **2018**, arXiv:1802.03426.
34. Leem, S.; Oh, J.; So, D.; Moon, J. Towards data-driven decision-making in the Korean film industry: An XAI model for box office analysis using dimension reduction, clustering, and classification. *Entropy* **2023**, *25*, 571. [CrossRef] [PubMed]
35. Salhin, A.; Sherif, M.; Jones, E. Managerial Sentiment, Consumer Confidence and Sector Returns. *Int. Rev. Financ. Anal.* **2016**, *47*, 24–38. [CrossRef]
36. Mandas, M.; Lahmar, O.; Piras, L.; De Lisa, R. ESG in the Financial Industry: What Matters for Rating Analysts? *Res. Int. Bus. Financ.* **2023**, *66*, 102045. [CrossRef]
37. Google News. Available online: https://news.google.com/home?hl=en-US&gl=US&ceid=US:en (accessed on 11 November 2024).
38. Valova, I.; Mladenova, T.; Kanev, G.; Halacheva, T. Web scraping—State of art, techniques and approaches. In Proceedings of the 2023 31st National Conference with International Participation (TELECOM), Sofia, Bulgaria, 26–27 October 2023; pp. 1–4.
39. Cohen, L.E.; Spiro, D.J.; Viboud, C. Projecting the SARS-CoV-2 transition from pandemicity to endemicity: Epidemiological and immunological considerations. *PLoS Pathog.* **2022**, *18*, e1010591. [CrossRef]
40. Weng, Y.; Wang, X.; Hua, J.; Wang, H.; Kang, M.; Wang, F.-Y. Forecasting horticultural products price using ARIMA model and neural network based on a large-scale data set collected by web crawler. *IEEE Trans. Comput. Soc. Syst.* **2019**, *6*, 547–553. [CrossRef]
41. Capizzi, V.; Gioia, E.; Giudici, G.; Tenca, F. The divergence of ESG ratings: An analysis of Italian listed companies. *J. Financ. Manag. Mark. Inst.* **2021**, *9*, 2150006. [CrossRef]
42. García, B.; Muñoz-Organero, M.; Alario-Hoyos, C.; Kloos, C.D. Automated driver management for Selenium WebDriver. *Empir. Softw. Eng.* **2021**, *26*, 107. [CrossRef]
43. Habrosh, A.A. Impact of cash flow, profitability, liquidity, and capital structure ratio on predict financial performance. *Adv. Sci. Lett.* **2017**, *23*, 7177–7179. [CrossRef]
44. Patel, J.M. Web scraping in Python using Beautiful Soup library. In *Getting Structured Data from the Internet*; Apress: Berkeley, CA, USA, 2020; pp. 31–84.
45. Serafeim, G. Public sentiment and the price of corporate sustainability. *Financ. Anal. J.* **2020**, *76*, 26–46. [CrossRef]
46. Palomino, M.A.; Aider, F. Evaluating the effectiveness of text pre-processing in sentiment analysis. *Appl. Sci.* **2022**, *12*, 8765. [CrossRef]
47. Chai, C.P. Comparison of text preprocessing methods. *Nat. Lang. Eng.* **2022**, *29*, 1–27. [CrossRef]
48. Sarica, S.; Luo, J. Stopwords in technical language processing. *PLoS ONE* **2021**, *16*, e0254937. [CrossRef] [PubMed]
49. Kim, Y.; Moon, J.; Hwang, E. Constructing differentiated educational materials using semantic annotation for sustainable education in IoT environments. *Sustainability* **2018**, *10*, 1296. [CrossRef]
50. Pramana, R.; Subroto, J.J.; Gunawan, A.A.S. Systematic literature review of stemming and lemmatization performance for sentence similarity. In Proceedings of the 2022 IEEE 7th International Conference on Information Technology and Digital Applications (ICITDA), Yogyakarta, Indonesia, 21–22 September 2022; pp. 1–6.
51. Financial Services Commission. FSC Publishes Korean Translation of SASB Standards to Aid Companies in Preparing for Sustainability Disclosure Standardization. 2021. Available online: https://fsc.go.kr/eng/pr010101/76850 (accessed on 10 October 2024).
52. IFRS Foundation. SASB Standard-Setting Archive. Available online: https://sasb.ifrs.org/standards/archive/ (accessed on 10 October 2024).
53. Zhang, Q.; Liu, Y.; Fang, H. Manifold learning-based UMAP method for geochemical anomaly identification. *Geochemistry* **2024**, 126157. [CrossRef]
54. Ansari, Y.; Yasmin, S.; Naz, S.; Zaffar, H.; Ali, Z.; Moon, J.; Rho, S. A deep reinforcement learning-based decision support system for automated stock market trading. *IEEE Access* **2022**, *10*, 127469–127501. [CrossRef]
55. Jabeen, A.; Yasir, M.; Ansari, Y.; Yasmin, S.; Moon, J.; Rho, S. An empirical study of macroeconomic factors and stock returns in the context of economic uncertainty news sentiment using machine learning. *Complexity* **2022**, *2022*, 4646733. [CrossRef]
56. Shah, S.S.; Asghar, Z. Individual Attitudes Towards Environmentally Friendly Choices: A Comprehensive Analysis of the Role of Legal Rules, Religion, and Confidence in Government. *J. Environ. Stud. Sci.* **2024**, 1–23. [CrossRef]

57. Yasir, M.; Ansari, Y.; Latif, K.; Maqsood, H.; Habib, A.; Moon, J.; Rho, S. Machine learning–assisted efficient demand forecasting using endogenous and exogenous indicators for the textile industry. *Int. J. Logist. Res. Appl.* **2022**, 1–20. [CrossRef]
58. Kandpal, V.; Jaswal, A.; Santibanez Gonzalez, E.D.; Agarwal, N. Corporate Social Responsibility (CSR) and ESG Reporting: Redefining Business in the Twenty-First Century. In *Sustainable Energy Transition: Circular Economy and Sustainable Financing for Environmental, Social and Governance (ESG) Practices*; Springer Nature: Cham, Switzerland, 2024; pp. 239–272.
59. Han, Y.J.; Moon, J.; Woo, J. Prediction of churning game users based on social activity and churn graph neural networks. *IEEE Access* **2024**, *12*, 101971–101984. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Model Reduction Method for Spacecraft Electrical System Based on Singular Perturbation Theory

Lifeng Wang [1], Yelun Peng [2] and Juan Luo [1,*]

[1] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China; rick@hnu.edu.cn

[2] College of Electrical and Information Engineering, Hunan University, Changsha 410082, China; pengyelun@hnu.edu.cn

* Correspondence: juanluo@hnu.edu.cn

Abstract: Accurate and efficient modeling and simulation of spacecraft electrical systems are crucial because of their complexity. However, existing models often struggle to balance simulation efficiency and accuracy. This paper introduces a model reduction method based on singular perturbation theory to simplify the full-order model of spacecraft electrical systems. The experimental results show that the reduced-order simplified model saves 50% of the simulation time with almost no degradation in the simulation accuracy and can be applied to real-world scenarios, such as digital twins. This method offers a new approach for rapid simulation of spacecraft electrical systems and has broad application prospects.

Keywords: spacecraft electrical system; singular perturbation reduction; stability analysis

Citation: Wang, L.; Peng, Y.; Luo, J. Model Reduction Method for Spacecraft Electrical System Based on Singular Perturbation Theory. *Electronics* **2024**, *13*, 4291. https://doi.org/10.3390/electronics13214291

Academic Editor: Davide Astolfi

Received: 26 September 2024
Revised: 19 October 2024
Accepted: 30 October 2024
Published: 31 October 2024

1. Introduction

Spacecraft, as a typical cyber-physical system (CPS), plays a critical role in various fields, such as navigation [1,2], military reconnaissance [3,4], geological exploration [5,6], and meteorological observation [7,8], greatly advancing technological progress and social development. As a core component of spacecraft, electrical systems feature complex structures and functions, integrating both discrete and continuous characteristics and involving multi-level, multi-component, and multi-variable couplings and interactions. Therefore, how to effectively model and simulate these systems has become a key issue in spacecraft design, operation, maintenance, and health management.

Although existing electrical system modeling and simulation methods [9–11] have made significant contributions, they still show notable deficiencies when addressing different simulation requirements. For different simulation tasks, the input/output parameters and the level of detail needed for the analysis vary. For example, in the simulation of energy balance in an S3R architecture [12,13], the focus is on satellite operational cycles, orbits, solar panel current generation, and the regulation of electrical charge and discharge, while details like battery charge–discharge curves and distribution switches can be simplified. Conversely, when analyzing a bus short-circuit fault, elements like protection control, distribution switches, and filtering circuits become critical, while solar array variations can be treated as a constant current source.

While a high-precision full-order model can accurately describe the system's dynamic behavior, its complexity leads to low simulation efficiency, making it unsuitable for fast-response scenarios. Conversely, simplified models, though improving efficiency, may lose important dynamic features, compromising accuracy. For example, Mariani et al. [14] treated the state variables of the output filter and control inner loop module as fast variables for model order reduction, but overlooked their significant impact on system stability, which compromised the accuracy of the reduced-order model. Luo and Dhople [15] considered the state variables of the transmission line and load module as fast variables for reduction; however, this dynamic characteristic has a considerable influence on the system's dominant modes, and neglecting it may lead to inaccuracies in the analysis. This conflict is particularly prominent in the multi-level simulation needs of spacecraft electrical systems, making it imperative to develop a model order-reduction method that balances both accuracy and efficiency.

To address this challenge, this paper proposes a spacecraft electrical system model reduction method based on singular perturbation theory [16,17]. By identifying the system's behavior and simulation objectives, we apply singular perturbation theory to simplify the detailed mathematical model of the electrical system, ensuring both the retention of critical dynamic characteristics and a significant improvement in simulation efficiency.

Our main contributions are summarized as follows:

- We proposed a model reduction method based on singular perturbation theory to improve the simulation efficiency of spacecraft electrical systems.
- We develop a reduced-order model of the spacecraft electrical system, preserving key dynamic characteristics.
- We implement the model reduction algorithm in C language and verifiy its feasibility.
- Compared to the full-order model, the reduced-order model generated by our method decreases simulation time by 50% with little degradation in the simulation accuracy.

2. Full-Order Mathematical Model of Spacecraft Electrical System

The primary function of the spacecraft electrical system is to convert various forms of energy (such as solar, chemical, and nuclear energy) into electrical power. This power is then distributed and regulated based on the needs of the spacecraft's subsystems to ensure the successful completion of on-orbit missions. Currently, most spacecraft utilize a power distribution system based on a solar array and storage battery. As shown in Figure 1, the system is primarily composed of modules including the solar array, storage battery, shunt regulator, charge regulator, and discharge regulator.

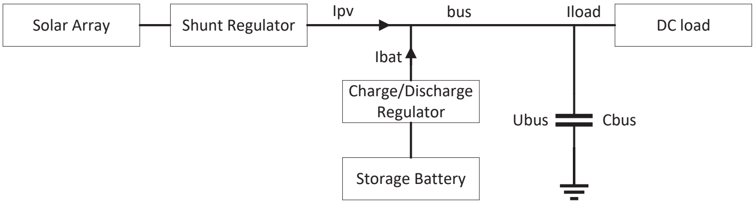


Figure 1. Spacecraft electrical systems.

2.1. Solar Array

The solar array consists of multiple solar cells arranged in a specific configuration to efficiently collect solar energy and convert it into electrical power efficiently. In spacecraft, the solar array serves as the primary power source, and its performance is crucial for the stability and reliability of the overall system.

This paper uses triple-junction gallium arsenide solar cells to construct the solar array. For a single solar cell, its steady-state equivalent circuit [18] is shown in Figure 2.

In the figure, I_{ph} represents the photocurrent, I_d represents the diode current of the semiconductor PN junction, R_s represents the series resistance composed of the internal resistance of the semiconductor material and the electrode resistance, and R_{sh} represents the parallel resistance caused by impurities at the edges of the semiconductor material or inherent defects. When the light source is constant, the solar cell can be regarded as an ideal constant current source.

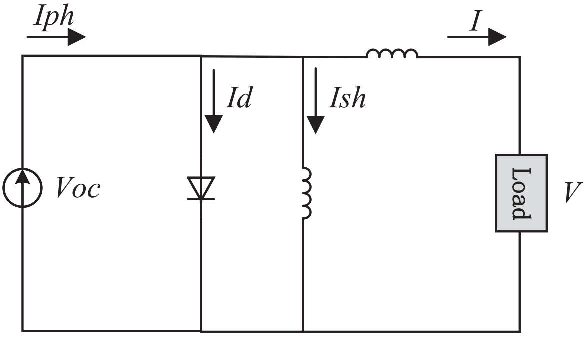


Figure 2. Equivalent circuit of a solar cell.

To accurately simulate the output characteristics of the solar array, we have established a simulation model. The model considers several input variables, including solar irradiance, temperature, incident angle, the number of series and parallel connections of the solar cells, and the performance parameters of the solar cells. By adjusting these input variables, the simulation model can calculate the output power of the solar array and generate its output I-V curve. This data is crucial for evaluating the performance of the solar array and optimizing its design.

2.2. Storage Battery

In spacecraft electrical systems, storage batteries typically use lithium-ion batteries. In the actual working process, these batteries exhibit hysteresis and rebound effects. To simulate these characteristics accurately, a comprehensive equivalent model consisting of an SOC calculation model, equivalent voltage source model, and equivalent impedance model is established in this paper. In addition, the relationship between the remaining battery charge (SOC) and the internal state is considered during the modelling process to facilitate circuit simulation and analysis. The overall circuit structure of the battery model is shown in Figure 3, where V represents the voltage across the battery terminals and I represents the charge and discharge current.

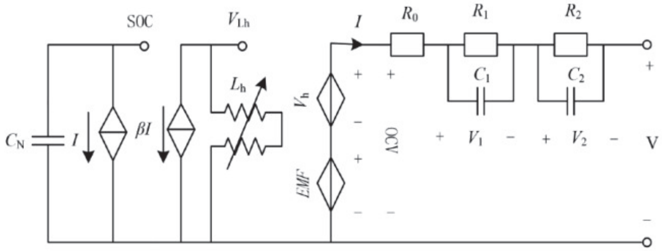
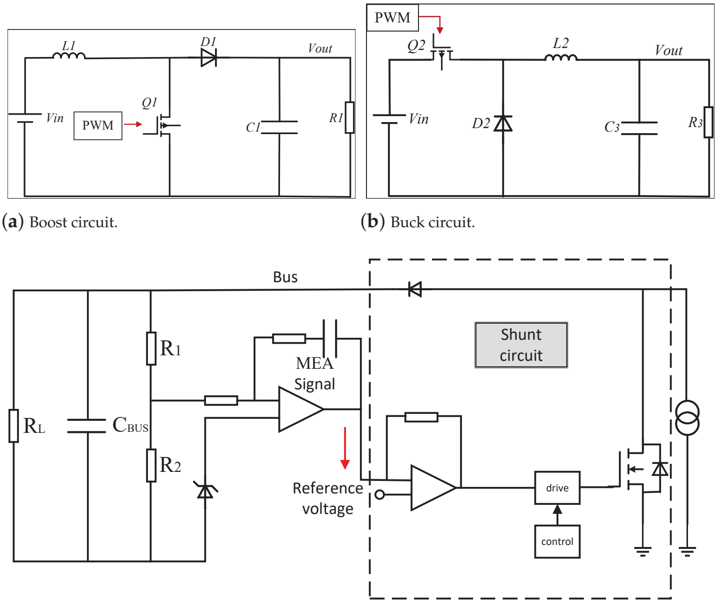


Figure 3. Equivalent circuit model of a battery.

2.3. Power Conditioning and Distribution Unit

The power conditioning and distribution unit (PCDU) in spacecraft electrical systems manages the energy conversion between the solar array and the storage battery. As shown in Figure 4, the PCDU comprises a shunt regulator, a discharge regulator, and a charging regulator. The shunt regulator stabilizes the bus voltage by adjusting the output current from the solar array, maintaining voltage stability by modulating the shunt state during load variations. The discharge regulator uses a buck circuit to step down the battery voltage to the required level, while the charging regulator employs a boost circuit [19] to raise the input voltage to an appropriate level for battery charging, ensuring stable power supply and charging.



(**a**) Boost circuit.

(**b**) Buck circuit.

(**c**) Shunt regulation circuit.

Figure 4. Power conditioning and distribution unit.

2.4. DC Load

The DC load comprises a resistive load and a constant power load (CPL), with the topology shown in Figure 5.

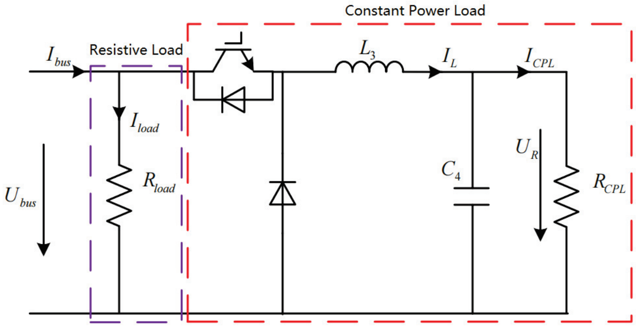


Figure 5. Equivalent circuit model of a DC Load.

The resistive load is represented by a resistor, while the CPL consists of a buck circuit connected in series with a resistive load. The buck circuit operates with constant voltage control to stabilize the voltage across the CPL, maintaining the load power $P = \frac{U_R^2}{R_{CPL}}$ as a fixed value. To adjust the load power, it is sufficient to change the value of R_{CPL}.

2.5. Complete Model of Spacecraft Electrical System

Based on the modular modeling approach, we developed the state space equations for the solar array, storage battery, power controller and distribution unit, and DC load models. By integrating these individual models, we formed a comprehensive full-order mathematical model of the spacecraft electrical system:

$$U_{oc} = f(SOC) \tag{1}$$

$$SOC = SOC_0 + \int I_{bat} dt \tag{2}$$

$$U_{bus} = U_{oc} + I_{bat} R_0 + U_1 + U_2 \tag{3}$$

$$I_{bat} = I_{pv} - I_{load} - I_c \tag{4}$$

$$\frac{\mathrm{d}U_1}{\mathrm{d}t} = -\frac{1}{R_1 C_1} U_1 + \frac{1}{R_1 C_1} U_{bus} \tag{5}$$

$$\frac{\mathrm{d}U_2}{\mathrm{d}t} = -\frac{1}{R_2 C_2} U_2 + \frac{1}{R_2 C_2} U_{bus} \tag{6}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = U_{bus} - U_{bus_{ref}} \tag{7}$$

$$I_{pv} = I_{pv_{max}} [U_{mea_{max}} - K_p(u_{ref} - U_{bus}) - k_i y] \times \frac{1}{U_{mea_{max}} - U_{mea_{min}}} \tag{8}$$

where K_p and K_i are the proportional and integral coefficients of the PI controller. R_0, R_1, and R_2 represent the resistances, C_1 and C_2 represent the capacitances. I_{pv} represents the output current of the solar array after shunt regulation, I_{load} represents the load current, I_c represents the charging or discharging current of the capacitor, and I_{bat} represents the current of the battery. U_{oc} is the open-circuit voltage of the storage battery, U_{bus} is the DC bus voltage, $U_{bus_{ref}}$ is the reference voltage of the bus, and U_1 and U_2 usually represent the voltages across the two capacitors.

In this electrical system, the order of the mathematical model is primarily influenced by the charge and discharge regulator model, resulting in a nonlinear system that includes two second-order equations and multiple first-order differential equations. This model comprehensively reflects the dynamic characteristics and operational state of the spacecraft electrical system, providing a solid foundation for subsequent simulation analysis and model order reduction.

3. Methodology

The simulation analysis of spacecraft electrical systems plays a vital role in space missions. Compared with ground AC power grids, spacecraft electrical systems have flexible and diverse structures and many nonlinear elements, which lead to a large number of mathematical models of electrical systems. Additionally, the significant differences in the dynamic responses of various components in the electrical system result in a mathematical model containing numerous variables with different time constants. Some of these variables have the characteristics of fast change and high frequency, while others have the characteristics of slow change and low frequency. In order to ensure the accuracy of the solution of the time domain simulation, the calculation step size of the simulation needs to be adjusted to a very small size, which leads to a significant increase in the calculation time, posing a great challenge to the demand for rapid response. Therefore, it is urgent to introduce an effective model reduction simplification method.

3.1. Principles of Singular Perturbation Theory

Singular perturbation theory is an effective tool for analyzing and simplifying multi-time-scale systems, especially for complex systems with different dynamic response times.

In the mathematical model of DC electrical systems, the dynamic response time of each component is different, resulting in different time scales for the entire system. The spacecraft electrical system we studied contains at least two time scales, which can be described by the following differential equations:

$$\dot{x} = f(t, x, y, \varepsilon) \tag{9}$$

$$\varepsilon\dot{y} = g(t, x, y, \varepsilon) \tag{10}$$

where $(t, x, y, \varepsilon) \in [0, t_1] \times D_x \times D_y \times [0, \varepsilon_0], D_x \in R^n, D_y \in R^m$ is the domain containing the origin, f and g are both continuously differentiable functions, and ε is a singular perturbation parameter. The state variables of the system in Equations (9) and (10) are divided into the dominant slow state variable x and the dominant fast state variable y, which correspond to the slow time scale variables and fast time scale variables of the system, respectively. Equations (1) and (2) in the above equations divide the state variables of the system into the dominant slow state variable x and the dominant fast state variable y, which correspond to the slow time scale variables and fast time scale variables of the system, respectively.

According to Tikhonov's theorem [20,21], for any point x_0 on the system's motion trajectory, if all the characteristic roots of the matrix $\partial g/\partial y|_{x=x_0}$ are in the left half of the complex plane s, then when the perturbation parameter $\varepsilon = 0$, the constraint manifold g is attractive to y. So, Equation (10) can be transformed into the steady-state equation $g(t, x, y, \varepsilon) = 0$, and the steady-state solution of y is obtained:

$$y = h(t, x) \tag{11}$$

Substituting the steady-state solution shown in Equation (11) into Equation (9), the original $(n + m)$th order system can be reduced to an nth order system as shown in Equation (12):

$$\dot{x} = f(t, x, h(t, x), 0) \tag{12}$$

3.2. Model Reduction Method Based on Singular Perturbation Theory

Before performing model reduction, it is necessary to categorize the state variables of the full-order model according to their time scales to distinguish between fast and slow state variables. So, a participating factor analysis (sensitivity analysis) should be performed to evaluate the sensitivity of model parameters on system outputs and identify the parameters that significantly impact the system's state or output. Based on the analysis results, state variables are categorized into different time scales, differentiating between fast and slow variables.

According to singular perturbation theory, the complete nonlinear model of the spacecraft electrical system is represented by

$$\dot{x} = f(x, \mu) \tag{13}$$

where x represents the state variable of the system and μ represents the system parameter.

The nonlinear equation is linearized at the stable point.

$$\dot{x} = Ax \tag{14}$$

where A is the Jacobian matrix.

By calculating the left eigenvector u and the right eigenvector v of the system in its steady state, a participation matrix P can be constructed to assess the degree of interaction between state variables and modes.

$$p = \begin{bmatrix} u_{11}^T v_{11} & \cdots & u_{1i}^T v_{1i} & \cdots & u_{1n}^T v_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ u_{j1}^T v_{j1} & \cdots & u_{ji}^T v_{ji} & \cdots & u_{jn}^T v_{jn} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ u_{n1}^T v_{n1} & \cdots & u_{ni}^T v_{ni} & \cdots & u_{nn}^T v_{nn} \end{bmatrix} \tag{15}$$

$$\sum_{i=1}^{n} u_{ji}^T v_{ji} = \sum_{j=1}^{n} u_{ji}^T v_{ji} = 1 \tag{16}$$

where p_{ij} is called the participation factor, which is used to measure the degree of interaction between the mode and the state variable. The larger the participation factor p_{ij}, the higher the degree of participation of the i-th mode in the j-th state variable. When the participation factor of a state variable is large, it can be identified as a slow state variable, and the rest are fast state variables. This division can effectively determine the time scale of the model state variables and provide a basis for subsequent order reduction.

In order to establish comparability between eigenvalues of different magnitudes, we normalize the eigenvalues. So, the participation factor p_{ij} can be expressed as

$$p_{ij} = \frac{|u_{ij}^T||v_{ij}|}{\sum_{k=1}^{N} |u_{kj}^T||v_{jk}|} \tag{17}$$

In the full-order mathematical model, U_1, U_2, U_{bus}, y, I_{pv}, I_c, I_{bat}, and SOC are selected as state variables. When U_{bus} is the dominant mode, the participation factors for the other seven variables are calculated using MATLAB/Simulink (version: R2022a). The results are as follows: $U_1 = 0.000000554$, $U_2 = 0.000000119$, $y = 0.490558440$, $I_{pv} = 0.000000012$, $I_c = 0.000245279$, $I_{bat} = 0.000000002$ and $SOC = 0.018637169$.

Based on the results of the participation factor analysis, the state variables (y and SOC) with larger participation factors are classified as slow state variables x, while the remaining variables are classified as fast state variables y.

The reduction method based on singular perturbation theory successfully simplifies the full-order mathematical model of the spacecraft electrical system into a first-order differential system, significantly reducing computational complexity while maintaining the dynamic characteristics of the system. This method provides an innovative approach for the rapid simulation of spacecraft electrical systems and demonstrates broad application prospects, particularly in real-world scenarios such as digital twins.

4. Low-Order Simplified Model of Spacecraft Electrical System

4.1. Solar Array

The inputs involved in the solar array model—temperature, solar irradiance, incident angle, etc.—are all slow variables. For the solar array model, the model is a slowly varying subsystem. When its perturbation coefficient is set to zero, we can obtain

$$I_{pv} = nI_{pv0} \tag{18}$$

$$I_{pv0} = I_{ph} - I_o[\mathrm{e}^{\frac{\mathrm{qV_{oc}}}{\mathrm{AKT}}} - 1] - \frac{\mathrm{V_{oc}}}{\mathrm{R_{sh}}} \tag{19}$$

where I_{pv} is the actual solar cell array output current, n is the number of branches, and I_{pv0} is the single branch current.

4.2. Storage Battery

For the storage battery model, factors such as temperature and impedance are considered slow variables, while other factors are fast variables. Therefore, in simulations over large time scales, it is necessary to distinguish between the effects of slow and fast variables. Under specific simulation conditions, the changes in slow variables can ignored. The specific simplification is shown in Figure 6.

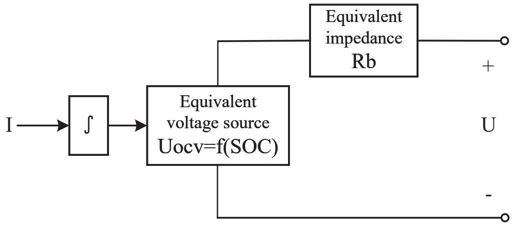


Figure 6. A reduced−order simplified model of the battery pack. $U_{ocv} = f(SOC)$ is a specific curve under certain environmental factors.

4.3. Power Conditioning and Distribution Unit

The simplification approach for the charge regulator model is similar to that of the discharge regulator model. It uses a switch dynamic equivalent model to describe the state changes of switches in the DC/DC circuit.

For the discharge regulator, the PI regulator that controls the on-off of the switch tube in its discharge topology is always MEA. Therefore, only the voltage signal of the MEA needs to be considered as the input. After considering the reduced-order simplification process with the perturbation coefficient being zero, the model can be degenerated into the following formula:

$$L_1 \frac{\mathrm{d}I_{in}}{\mathrm{d}t} = U_{bat} - U_{out} \tag{20}$$

$$I_{in} = (1 - D)(I_{pv} - I_{load}) \tag{21}$$

For the charge regulator, both BEA and MEA regulators are available. Although the basic logic is the same for both, the proportional integral coefficients of the voltage signal u_i are different. The voltage signal of MEA is

$$u_i = k_p(\Delta U) + k_i a_1 \tag{22}$$

The voltage signal of BEA is

$$u_i = k_{pb}(I_{in} - I_{ref}) + k_{ib} a_2 \tag{23}$$

The control signal for the charging regulator is chosen as the smaller value between MEA and BEA. After considering a perturbation coefficient of zero, the model can be simplified to:

$$L_2 \frac{\mathrm{d}DI_{in}}{\mathrm{d}t} = U_{bat} - U_{out} \tag{24}$$

$$U_{in} = DU_{bus} \tag{25}$$

$$I_{in} = I_{pv} - I_{load} \tag{26}$$

where D is described by the normalized voltage control signal:

$$D = \frac{u_i}{u_{dm} - u_{dl}} \tag{27}$$

The shunt regulator consists of multiple switching devices. During operation, its output is modulated by the control signal. This process can be modeled using an average duty cycle model. The average output power P of the duty cycle can be expressed as

$$P = \frac{T_{on} P_L}{T} \tag{28}$$

where P_l is the average power during the on-time period, T is the switching period of the shunt regulator, and T_{on} is the on-time. The average power model assumes that the input power P_L remains constant during the on-time and there is no power output during the off-time.

4.4. DC Load

The changes in the load model mainly exist in the simulation process. The resistive sensitivity of the load is mainly manifested in dynamic changes on a short time scale, whereas spacecraft models are generally simulated over a wide time scale. In this simulation background, the system participation factor of the fast variable is significantly smaller than the system participation factor of the slow variable. Therefore, for the establishment of the spacecraft DC load model, the factors involved in the model that influence the dynamic transients of the system can be approximately simplified.

Constant current and constant impedance loads can be equivalently represented as controlled current sources. A constant current load provides a constant current through the controlled current source, while a constant impedance load provides a current value proportional to the bus voltage through the controlled current source.

$$I_{load} = I_0 \tag{29}$$

$$I_{load} = kU_{bus} \tag{30}$$

The constant power load exhibits a negative incremental impedance characteristic. Modeling constant power loads can be achieved using a controlled current source. Given a specified load power consumption, the load current can be calculated from the load power and voltage values, ensuring that the load current adjusts with changes in voltage to maintain constant load power. The load current is calculated as

$$I_{load} = P_{load}/U_{bus} \tag{31}$$

5. Simulation

In order to fully demonstrate the advantages of model downscaling, we compare the numerical simulation results and numerical computation time.

The electrical model includes the solar array, the PCUD with S3R architecture, the storage battery, and other main modules. The simulated values of state parameters involved in each module are shown in Table 1.

Table 1. The simulated values of state parameters.

Key Component	Parameter	Value
BuS	Bus capacitance (F)	0.00036
	Bus voltage (V)	42
Solar array	Series count	27
	Parallel count	216
	Operating temperature (K)	353
	lrradiance condition (W/m^2)	1300
	Open circuit voltage V$_{oc}$ (mV)	2730
	Operating voltage V$_{mp}$ (mV)	2430
	Short circuit current density J$_{sc}$ (mA/cm^2)	17.2
	Operating current density J$_{mp}$ (mA/cm^2)	16.7
Storage battery	Equivalent Terminal voltage (V)	42
	Equivalent ohmic internal resistance (Ω)	0.013
	Equivalent electrochemical polarization resistance (Ω)	0.007
	Equivalent concentration polarization resistance (Ω)	0.015
Charge regulator	Input inductance (H)	0.006
	Output capacitance (F)	0.0005
	Input resistance (Ω)	0.1
Discharge Regulator	Input Inductance (H)	0.006
	Output Capacitance (F)	0.0002
	Input Resistance (Ω)	0.1

5.1. Comparative Analysis of Numerical Simulation Results

We conducted an energy balance analysis for a satellite over one orbit. Assuming an orbital period of 108 min around the Earth, with 36 min in the Earth's shadow and 72 min in the sunlight. During the sunlight period, the temperature of the photovoltaic panel surface rapidly rises to approximately 80 + 273.15 Kelvin and remains nearly constant. We assume that the solar panels are always oriented toward the sun, with a constant irradiance of 1300 W/m² in the sunlight region.

The simulation results for three orbits around the Earth are shown in Figure 7.

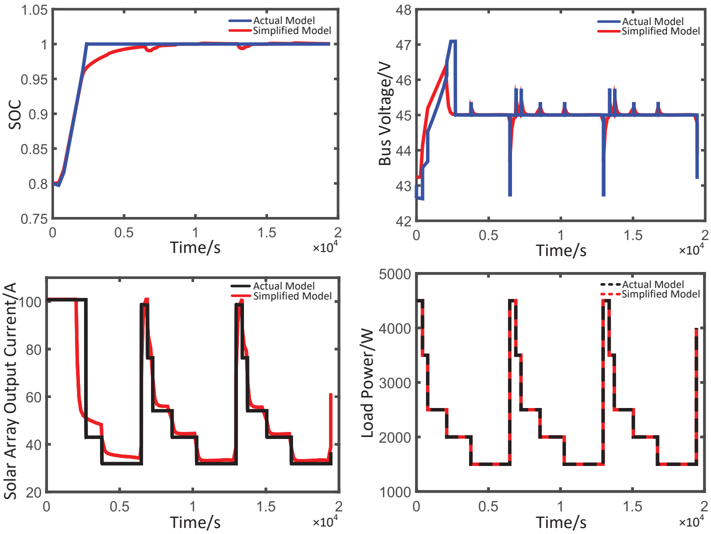


Figure 7. Comparison between reduced-order simplified model and full-order mathematical model.

It can be seen from the simulation results that because of the model reduction, the transient process of the simulation process will be less obvious, but the error size can be controlled within 1%. Regarding the steady-state results, the static error is essentially zero.

5.2. Comparative Analysis of Numerical Calculation Time

To evaluate the efficiency of the reduced-order model, we tested the time required to complete simulations for three orbital periods using different models. The results are shown in Table 2.

Table 2. Simulation time comparison.

Model	Time(s)
Circuit level model	15
Full-order mathematical model	1
Low-order mathematical model	0.5

Compared to the circuit-level model, the full-order mathematical model has slightly lower simulation accuracy and offers a coarser description of bus voltage ripple. However, it significantly reduces the time cost associated with the circuit model. Nonetheless, the full-order model involves numerous algebraic differential equations because of parasitic parameters, such as capacitance and inductance. When integrated into a digital twin simulation environment, the need for steady-state assessment and multiple iterative calculations requires a smaller simulation step size to achieve convergence, which results in lower practical efficiency.

In contrast, the reduced-order mathematical model only decreases the simulation time by 50% compared to the full-order model. However, the complexity of the differential equations is significantly reduced, allowing for larger step sizes during simulation and thereby providing higher efficiency in practical applications.

6. Software Package Development for Reduced-Order Simplified Models

In order to verify whether the simplified electrical system model is suitable for a digital twin simulation, it is deployed in a C language package in a Windows environment for energy balance simulation verification and comparison. The whole simulation flow of the software program is shown in the Figure 8.

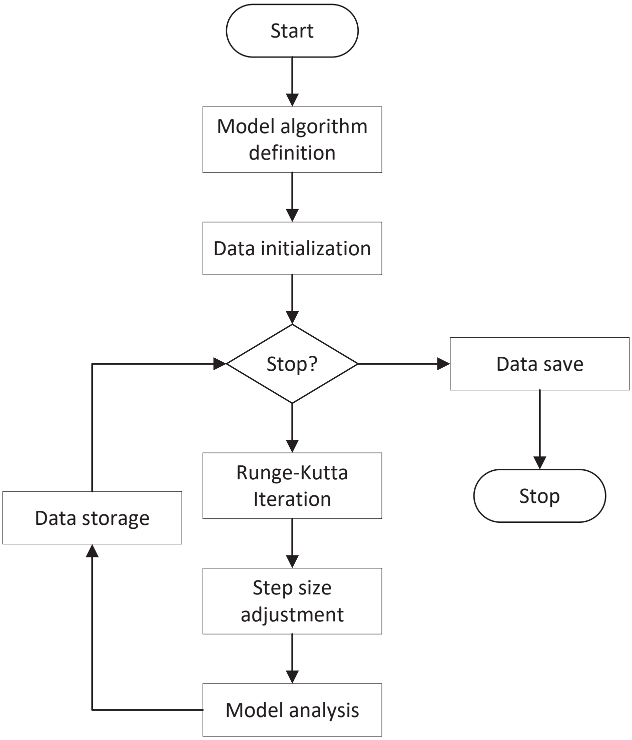


Figure 8. Software program simulation process.

First, set the simulation step size to 1 s. Then, define the solver algorithm to solve the differential equations in the power distribution system. Initial values for the simulation are provided, and the simulation is checked to see if the stopping conditions are met. If not, the step size is adjusted, and the Runge-Kutta solver algorithm [22–24] is reintroduced. This algorithm approximates the solution by calculating the slope at several points in the next interval, then averaging these slopes to obtain a new average slope *k*. This average slope is multiplied by the current step size and added to the initial point value to obtain the data for the next point, which is stored in the data repository. This iterative process continues until the stopping conditions for the simulation are achieved. Finally, all data are saved, and plotting them generates the simulation process curves.

As shown in Figure 9, the computational results from the actual code closely match the MATLAB simulation results, achieving a fast solution for 20,000 data sets in under 5 s. Although the reduced-order simplified model is based on the full-order mathematical model of the spacecraft electrical system, it enables fast and efficient solving, making it suitable for energy balance evaluations and digital twin simulations. Additionally, the model can be extended to study the interaction between the power system and other subsystems, such as attitude and orbit control and satellite computers.

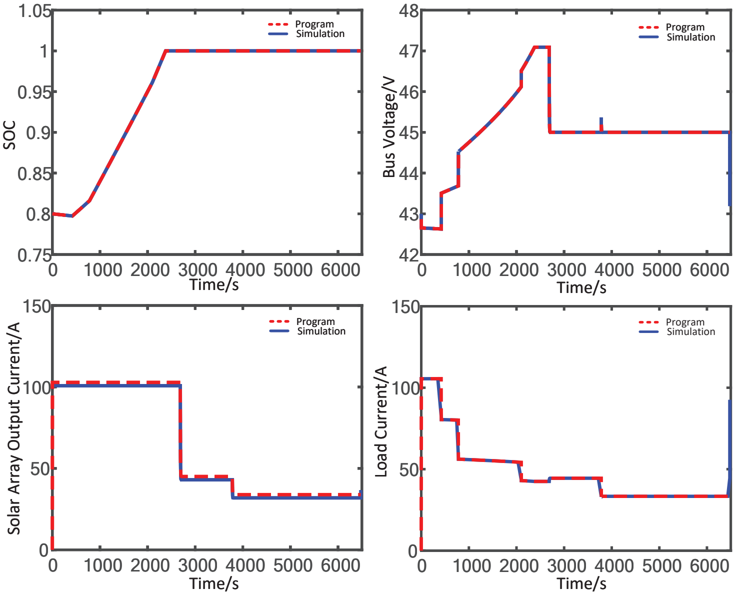


Figure 9. Comparison between program and simulation results. "Program" represents the result calculated in the C program. "Simulation" represents the result calculated in MATLAB.

7. Conclusions

This study presents a model reduction method for spacecraft electrical systems based on singular perturbation theory. The method effectively simplifies complex full-order models, reducing simulation time by 50% while maintaining a comparable level of accuracy. The reduced-order model not only optimizes the simulation process but is also applicable to real-world scenarios such as digital twins.

Furthermore, our research opens avenues for further theoretical developments, such as the application of this method to other complex dynamic systems in various engineering fields. Practically, the ability to achieve efficient and accurate simulations can significantly enhance the design and analysis of spacecraft electrical systems, facilitating improved decision-making and operational efficiency. Overall, our findings highlight the potential for this approach to be adapted for diverse applications, contributing to advancements in both aerospace engineering and other interdisciplinary domains.

Author Contributions: Methodology, L.W.; Validation, L.W.; Resources, J.L.; Writing—original draft, L.W.; Writing—review and editing, Y.P. and J.L.; Supervision, Y.P.; Project administration. J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Reid, T.G.; Chan, B.; Goel, A.; Gunning, K.; Manning, B.; Martin, J.; Neish, A.; Perkins, A.; Tarantino, P. Satellite navigation for the age of autonomy. In Proceedings of the 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS), Portland, OR, USA, 20–23 April 2020; pp. 342–352.
2. Yang, Y.; Mao, Y.; Sun, B. Basic performance and future developments of BeiDou global navigation satellite system. *Satell. Navig.* **2020**, *1*, 1. [CrossRef]

3. Early, B.R.; Gartzke, E. Spying from space: Reconnaissance satellites and interstate disputes. *J. Confl. Resolut.* **2021**, *65*, 1551–1575. [CrossRef]
4. Willbold, J.; Sciberras, F.; Strohmeier, M.; Lenders, V. Satellite Cybersecurity Reconnaissance: Strategies and their Real-world Evaluation. In Proceedings of the 2024 IEEE Aerospace Conference, Big Sky, MT, USA, 2–9 March 2024; pp. 1–13.
5. Epuh, E.E.; Okolie, C.J.; Daramola, O.E.; Ogunlade, F.S.; Oyatayo, F.J.; Akinnusi, S.A.; Emmanuel, E.O.I. An integrated lineament extraction from satellite imagery and gravity anomaly maps for groundwater exploration in the Gongola Basin. *Remote Sens. Appl. Soc. Environ.* **2020**, *20*, 100346. [CrossRef]
6. Shirmard, H.; Farahbakhsh, E.; Müller, R.D.; Chandra, R. A review of machine learning in processing remote sensing data for mineral exploration. *Remote Sens. Environ.* **2022**, *268*, 112750. [CrossRef]
7. Gao, H.; Tang, S.; Han, X. China's Fengyun (FY) meteorological satellites, development and applications. *Sci. Technol. Rev.* **2021**, *39*, 9–22.
8. Zhao, Q.; Yu, L.; Du, Z.; Peng, D.; Hao, P.; Zhang, Y.; Gong, P. An overview of the applications of earth observation satellite data: Impacts and future trends. *Remote Sens.* **2022**, *14*, 1863. [CrossRef]
9. Kuzlu, M.; Abalos, K.; Elma, O.; Popescu, O.; Pipattanasomporn, M. Modeling and simulation of the International Space Station (ISS) electrical power system. *Int. Trans. Electr. Energy Syst.* **2021**, *31*, e12980. [CrossRef]
10. Shangguan, D.; Chen, L.; Ding, J. A digital twin-based approach for the fault diagnosis and health monitoring of a complex satellite system. *Symmetry* **2020**, *12*, 1307. [CrossRef]
11. Beukes, H.; Dowson, K.; Enslin, J. Simulation program for Stellenbosch University satellite (SUNSAT) power system. *Trans. S. Afr. Inst. Electr. Eng.* **2021**, *84*, 32–36.
12. Dou, Y.; Liu, T.; Yang, A.; Dong, B. Stability analysis for S3R-architecture Power Conditioning Units interconnection system using impedance-based methods. *Energy Rep.* **2022**, *8*, 582–591. [CrossRef]
13. Zhu, H.; Zhang, B.; Zhang, D. Overview of architectures for satellite's regulated bus power system. In Proceedings of the 2020 IEEE 1st China International Youth Conference on Electrical Engineering (CIYCEE), Wuhan, China, 1–4 November 2020; pp. 1–8.
14. Mariani, V.; Vasca, F.; Vasquez, J.C.; Guerrero, J.M. Model order reductions for stability analysis of islanded microgrids with droop control. *IEEE Trans. Ind. Electron.* **2014**, *62*, 4344–4354. [CrossRef]
15. Luo, L.; Dhople, S.V. Spatiotemporal model reduction of inverter-based islanded microgrids. *IEEE Trans. Energy Convers.* **2014**, *29*, 823–832. [CrossRef]
16. Hoppensteadt, F. Asymptotic stability in singular perturbation problems. II: Problems having matched asymptotic expansion solutions. *J. Differ. Equ.* **1974**, *15*, 510–521. [CrossRef]
17. Eckhaus, W. *Matched Asymptotic Expansions and Singular Perturbations*; Elsevier: Amsterdam, The Netherlands, 2011.
18. Cheknane, A.; Hilal, H.S.; Djeffal, F.; Benyoucef, B.; Charles, J.P. An equivalent circuit approach to organic solar cell modelling. *Microelectron. J.* **2008**, *39*, 1173–1180. [CrossRef]
19. Babaa, S.E.; El Murr, G.; Mohamed, F.; Pamuri, S. Overview of boost converters for photovoltaic systems. *J. Power Energy Eng.* **2018**, *6*, 16–31. [CrossRef]
20. Dontchev, A.; Donchev, T.; Slavov, I. A Tikhonov-type theorem for singularly perturbed differential inclusions. *Nonlinear Anal. Theory Methods Appl.* **1996**, *26*, 1547–1554. [CrossRef]
21. Tang, Y.; Prieur, C.; Girard, A. Tikhonov theorem for linear hyperbolic systems. *Automatica* **2015**, *57*, 1–10. [CrossRef]
22. Kasdin, N.J. Runge-Kutta algorithm for the numerical integration of stochastic differential equations. *J. Guid. Control. Dyn.* **1995**, *18*, 114–120. [CrossRef]
23. Ahmadianfar, I.; Heidari, A.A.; Gandomi, A.H.; Chu, X.; Chen, H. RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. *Expert Syst. Appl.* **2021**, *181*, 115079. [CrossRef]
24. Schober, M.; Duvenaud, D.K.; Hennig, P. Probabilistic ODE solvers with Runge-Kutta means. In Proceedings of the Advances in Neural Information Processing Systems 27 (NIPS 2014), Montreal, QC, Canada, 8–13 December 2014.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

Article

Patent Keyword Analysis Using Regression Modeling Based on Quantile Cumulative Distribution Function

Sangsung Park and Sunghae Jun *

Department of Data Science, Cheongju University, Chungbuk 28503, Republic of Korea; hanyul@cju.ac.kr
* Correspondence: shjun@cju.ac.kr or statcs@gmail.com; Tel.: +82-10-7745-5677; Fax: +82-43-229-8432

Abstract: Patents contain detailed information of researched and developed technologies. We analyzed patent documents to understand the technology in a given domain. For the patent data analysis, we extracted the keywords from the patent documents using text mining techniques. Next, we built a patent document–keyword matrix using the patent keywords and analyzed the matrix data using statistical methods. Each element of the matrix represents the frequency of a keyword that occurs in a patent document. In general, most of the elements were zero because the keyword becomes a column of the matrix even if it occurs in only one document. Due to this zero-inflated problem, we experienced difficulty in analyzing patent keywords using existing statistical methods such as linear regression analysis. The purpose of this paper is to build a statistical model to solve the zero-inflated problem. In this paper, we propose a regression model based on quantile cumulative distribution function to solve this problem that occurs in patent keyword analysis. We perform experiments to show the performance of our proposed method using patent documents related to blockchain technology. We compare regression modeling based on a quantile cumulative distribution function with convenient models such as linear regression modeling. We expect that this paper will contribute to overcoming the zero-inflated problem in patent keyword analysis performed in various technology fields.

Keywords: patent keyword analysis; quantile cumulative distribution function; regression; patent document; patent–keyword matrix

Citation: Park, S.; Jun, S. Patent Keyword Analysis Using Regression Modeling Based on Quantile Cumulative Distribution Function. *Electronics* **2024**, *13*, 4247. https://doi.org/10.3390/electronics13214247

Academic Editors: Wentao Li, Huiyan Zhang, Tao Zhan and Chao Zhang

Received: 28 September 2024
Revised: 28 October 2024
Accepted: 28 October 2024
Published: 30 October 2024

1. Introduction

Patent keyword analysis (PKA) is important to technology management because a patent contains extensive and detailed information about the developed technology. Using the PKA results, we can build research and development (R&D) plans and strategies for the target technology. In general, for PKA, we extract technology keywords from patent documents using text mining techniques [1,2]. Using the extracted keywords, we construct a patent–keyword matrix for PKA based on statistics and machine learning algorithms. The matrix consists of elements representing the frequency values of keywords that occur in patents [1,3–5]. In most cases, this matrix has a sparse data structure that suffers from the zero-inflated problem [3–5]. This is because a keyword that is included in even just one patent document becomes one column in the matrix [3–5]. The sparse zero-inflated problem reduces the performance of PKA models [4,6]. As such, we have to solve the zero-inflated problem for PKA. Many existing studies rely on statistical models such as the zero-inflated Poisson and negative binomial models to solve the problem [5,7–11]. Recently, studies based on machine learning methods such as generative models have been conducted to solve the zero-inflated problem [4,5]. However, existing models have limitations in that model performance deteriorates as the proportion of zeros included in the data increases [3–5,7,8]. To solve this problem, we consider a regression model based on a quantile Cumulative Distribution Function (CDF) [12–14]. We call this model CDF-based Quantile Regression Model (QRM) in this paper. To verify the performance of the CDF-based QRM, we perform experiments using patent documents related to blockchain technology.

The motivation for this research is to appropriately deal with the zero-inflated problem that occurs in patent keyword data analysis. In particular, we study a method to overcome the extreme zero-inflated problem, where the proportion of zeros in the given data exceeds half. Since the extreme zero-inflated problem is difficult to solve even with existing statistical zero-inflated models, we need to find new methods to solve it.

The remainder of this paper is organized as follows. We survey works related to our research such as regression and zero-inflated models in Section 2. In Section 3, we present the theoretical explanation of our proposed method and the analysis process step by step. In addition, we present the performance evaluation indexes of comparative models in this section. Next, we show the improved performance and validity of our proposed method from the experimental results using patent documents related to blockchain technology in Section 4. In this section, we compare the model performance of the CDF-based QRM with traditional linear regression and statistical zero-inflated models. In the Section 5, we illustrate how the proposed method can be applied to practical tasks in various domains. Lastly, we provide the conclusions and future works related to our research in Section 6.

2. Related Works

Patent analysis has been performed in various technology domains such as photovoltaic, medicine, mountain logistics, climate change, artificial intelligence (AI), surgery, and energy [15–21]. This is because when developers register a technology they have developed as a patent, they are guaranteed exclusive rights to use their technology for a certain period of time. Therefore, we analyze patents to understand these technologies. Also, we use the results of patent analysis for technology management such as Research and Development strategy development. PKA, which we propose in this paper, is also a field of patent analysis. PKA mainly extracts technology keywords from the abstracts and claims contained in collected patent documents and analyzes them. In this process, we use text mining and various data analysis methods based on statistics and machine learning.

The regression model is very popular in machine learning as well as in statistics [12,22–24]. This model consists of independent and dependent variables called X and Y, respectively [22]. Regression analysis is statistical modeling that explores relationships between variables [24]. We can predict Y for a given X using regression analysis [19]. Figure 1 shows a process of regression modeling [23].

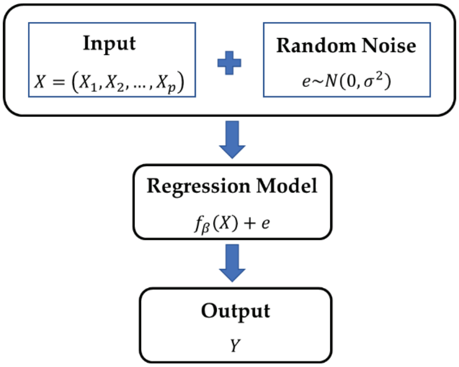


Figure 1. Regression modeling process.

We assume that the response variable Y adds the error e to the explanatory variables X Using X and Y, we create the linear regression model (LRM) as follows.

$$Y = f_\beta(X) + e \quad (1)$$

In Equation (1), $f_{\beta}(X)$ is $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$, and we estimate the model parameters, $\beta = \left(\beta_0, \beta_1, \beta_2, \ldots \beta_p\right)$ that minimize the error using the least squared loss function [12,24]. The error represents a random noise included in observed data and follows a normal distribution with a mean (μ) = 0 and variance = σ^2. This model has provided good performance in exploring the relationship between X and Y in most data, including errors with a mean of 0 and equal variance [24]. However, we have difficulty in using the LRM when the given data does not satisfy the model assumptions [9–11]. In particular, if the given data has many extreme values, we cannot use the LRM [4–6]. To solve the problems of LRM, we can consider the QRM [13,25]. Quantile regression aims to model the impact of explanatory variables on the quantile of the response variable. The QRM finds the conditional quantile of Y just as the regression based on the least square method estimates the conditional mean of Y [9]. We can apply both continuous and count data to QRM. QRM is a model that can be used when the given data do not satisfy the normality assumption and are asymmetric or contain many outliers. In the PKA, we found that the patent–keyword matrix contains zero-inflated data that is sparse and asymmetric. As such, we propose a method to analyze the patent keywords using QRM. In addition, we consider the CDF for our PKA model based on QRM because we aim to predict the specific quantile of each patent keyword.

In statistics, the zero-inflated model is typically used to analyze data that contain a lot of zeros [8,10,11]. This model has been used to solve the zero-inflated problem that occurs in various domains [26–29]. The zero-inflated model based on statistics is defined as follows [9]:

$$P(X = x) = \begin{cases} \pi + (1-\pi) f(x),\ x = 0 \\ (1-\pi) f(x),\ x > 0 \end{cases} \tag{2}$$

In Equation (2), $f(x)$ is a density of random variable $X = x$. In the statistical zero-inflated model, the probability model of X is separated into two parts of zero and non-zero [9–11]. The π represents the probability of zero occurrence. Although the statistical zero-inflated models have been used to overcome the problem that arises in various data analysis processes, they have shown a problem in that model performance deteriorates as the proportion of zeros in the data increases [3–5]. Therefore, in this paper, we propose a PKA method using QRM for analyzing patent keyword data with a high zero ratio.

3. Proposed Method

3.1. Patent–Keyword Matrix

The reason we analyze patent keywords is because patent keywords represent technologies. Many governments and companies around the world are working hard to establish R&D strategies for new and promising technology development in order to survive in the fierce technological competition. For this purpose, understanding of technology is essential, and one of the effective methods for understanding technology is patent keyword analysis. In order to analyze patent keywords, we construct a patent–keyword matrix from patent documents. However, since a significant portion of the elements in this matrix are zero, existing data analysis methods have limitations. For example, if the given data contains too many zeros, the explanatory and predictive power of models built using this data will be reduced. To solve this problem, in this paper, we propose an analytical method for patent keyword data. The patent keyword data are generated from the title and abstract parts of patent documents. In general, we search the patent documents related to the target technology from various patent databases in the world using a keyword search equation. For our PKA, we preprocess the searched patent documents using text mining techniques as follows [1,2].

(TM.1) Searching patent documents related to target technology.
(1-1) Using keyword searching equation, we collect the patent documents related to the target technology.

(1-2) By examining all the retrieved patents, we select valid patents that can be used for analysis.
(TM.2) Building structured patent data by text mining.
(2-1) Using tokenization and normalization, we preprocess the patent documents to create the corpus.
(2-2) By extracting keywords from the corpus, we construct a patent–keyword matrix.

In the first text mining (TM.1) step, we determine the target technology for PKA. In this step, we collect the patent documents related to the target technology from various patent databases. In addition, we select the valid patents from the collected patent documents. Using tokenization and normalization methods such as stemming, lemmatization, lowercasing, and removing stopwords, we preprocess the valid patent documents and make a corpus of them in the second text mining (TM.2) step. Finally, we extract the patent keywords from the corpus and construct the patent–keyword matrix. The rows and columns are patent documents and keywords from the vocabulary, respectively. Also, the matrix values are the frequency values of each keyword in a document. Therefore, through the text mining process of making text corpus, parsing, and constructing text database, we build a patent–keyword matrix [1,3,4]. We define this matrix M using Equation (3).

$$M = \left(Freq_{ij}\right),\ i = 1,\ 2, \ldots, p\ j = 1,\ 2, \ldots,\ n \tag{3}$$

where p and n are the numbers of keywords and patent documents, respectively. Also, $Freq_{ij}$ is frequency value of the jth keyword occurring in the ith patent. The observed data of $Freq_{ij}$ is distributed to Poisson probability distribution with parameter λ as follows.

$$X = Freq_{ij},\ X \sim\ Poisson(\lambda),\ \lambda > 0,\ x = 0,\ 1,\ 2, \ldots \tag{4}$$

In Equation (4), X has values greater than or equal to zero, but most X values are zeros. This is one cause of the deteriorating performance of statistical analysis models in PKA. This is the problem we aim to solve in this paper. Next, we normalize the frequency values of the patent–keyword matrix as follows.

$$Freq_{ij\_nor} = \frac{Freq_{ij} - Min}{Max - Min} \tag{5}$$

Max and Min represent the maximum and minimum among all values of $Freq_{ij}$ in Equation (5). According to Equation (5), the range of values that $Freq_{ij\_nor}$ can have changes the values of $Freq_{ij}$ from an integer greater than 0 to a real number between 0 and 1. In most cases, the patent–keyword matrix has the zero-inflated problem. That is, most of the frequency values are zeros. This becomes a factor that seriously reduces the performance of the analysis model. Therefore, we have to deal with this problem for PKA. We try to solve this problem using the CDF-based QRM in our study.

3.2. Quantile Regression Modeling Based on Cumulative Distribution Function for PKA

The patent–keyword matrix is asymmetric and sparse because of the zero-inflated problem of matrix elements. Most of the elements in the matrix have the value zero. Therefore, the matrix has a very imbalanced data structure. Because of the characteristics of the patent–keyword matrix with such an asymmetric structure, we have difficulty in analyzing patent keywords using statistical techniques. This problem reduces the analytical performance of statistical methods and machine learning algorithms. To solve the problem, we propose a method of PKA using CDF-based QRM. The CDF is defined as follows [9,12].

$$F(y) = P(Y \leq y) \tag{6}$$

In Equation (6), $F(y)$ is the value of CDF of $Y = y$ and is computed by the probability of $Y \leq y$. Also, the q-th quantile of Y is a value of between 0 and 1. Therefore, we have to change the value of Y to a (0,1) interval. In our study, we normalized the frequency value of response keywords to a real value between 0 and 1. This approach is similar to PKA using beta regression modeling. The probability distribution corresponding to a random variable Y with support between 0 and 1 is the beta distribution [9,24]. Though the regression analysis model based on the beta distribution can be used for PKA, as the zero-inflated problem becomes more severe, the beta regression model also shows limitations in model performance like existing regression models [4–6]. Therefore, we use CDF-based QRM to analyze the patent–keyword matrix data. This models the CDF of the response variable (keyword) Y and makes predictions for specific quantiles of Y. In other words, we can use the model to estimate the probability that the response variable is below a certain value. Because the CDF-based QRM has robust characteristics that are not significantly affected by extreme values such as outliers, we use this model for analyzing patent keywords with the sparsity problem of zero inflation. The CDF with a location parameter μ and a dispersion parameter σ is defined with the following equation [13,25].

$$\mathrm{G}(X = x, \mu, \sigma) = F\left[U\left(H^{-1}(x), \mu, \sigma\right)\right],\ 0 \leq x \leq 1,\ -\infty < \mu < \infty,\ 0 < \sigma \tag{7}$$

In Equation (7), the random variable X has the support (0,1) and two parameters, μ and σ. F and H are a CDF and an invertible CDF with supports D_1 and D_2, respectively. U is a suitable transformation from $D_1(-\infty, \infty)$ to $D_2(0, \infty)$ for applying μ and σ. In addition, $H^{-1}(x)$ is a corresponding quantile function from D_1 to D_2. Using this CDF quantile family in Equation (7), we carry out the CDF-based QRM for our PKA. To estimate the parameters, we carry out maximum likelihood estimation (MLE) for all parameters based on a gradient [12,13,23,25]. For the CDF in Equation (7), the probability density function (PDF) is defined as in Equation (8) [24,25].

$$G(X = x, \mu, \sigma) = \frac{q(x) f\left(\frac{H^{-1}(x) - \mu}{\sigma}\right)}{\sigma},\ 0 \leq x \leq 1,\ -\infty < \mu < \infty,\ 0 < \sigma \tag{8}$$

where $f(x)$ and $q(x)$ are the PDF corresponding to F and the quantile density function corresponding to H^{-1}. We differentiate the log of G with respect to μ and σ and dropping q. As such, the regression model has two sub models as follows [13]:

$$L_\mu(\hat{\mu}) = x^T \beta \tag{9}$$

$$L_\sigma(\hat{\sigma}) = z^T \delta \tag{10}$$

where Equations (9) and (10) are the models for location (μ) and dispersion (σ), respectively. x and z are the vectors of predictors. Also, β and δ are the vectors of coefficients. In this paper, we use the link functions of identity and log for L_μ and L_σ. In the next section, we conduct experiments using patents related to blockchain technology to evaluate the model performance. We illustrate our PKA process of CDF-based QRM as follows.

(Step.1) Patent data collection and preprocessing.
(1-1) Searching patent documents from patent databases.
(1-2) Extracting keywords from searched patents using text mining.
(1-3) Constructing the patent–keyword matrix from extracted keywords.
(Step.2) Data preparation and normalization.
(2-1) Determining response and explanatory variables (keywords).
(2-2) Normalizing response variable to a (0,1) interval.
(Step.3) Data analysis.
(3-1) Building the CDF-based QRM.
(3-2) Evaluating model performance using loglikelihood, AIC, and BIC.

Our proposed PKA method consists of three steps. In Step.1, we collect patent documents related to the target technology from patent databases such as the United States

Patent and Trademark Office (USPTO). We extract keywords from the collected patent documents using various text mining techniques. For patent analysis, we construct a matrix consisting of patents and keywords for rows and columns, respectively. Also, each element of the matrix is the frequency of occurrence of each keyword in the patent document. Next, we select response and explanatory variables according to the target technology and the aim of the patent analysis. For CDF-based QRM, we normalize the value of the response variable to a (0,1) interval. In the final step, we use the CDF-based QRM to analyze the patent–keyword matrix, which is sparse and has the zero-inflated problem. In this paper, we compared the CDF-based QRM with the LRM. This is because the LRM is widely used in the field of keyword data analysis. We used three indexes to evaluate the performance between the comparative models. First, we use the loglikelihood, defined as in Equation (11) [9,13,24].

$$L(\theta|x) = \sum_{i=1}^{n} log(f(x_i|\theta)) \tag{11}$$

where θ and n are the model parameter and data size. $f(x_i|\theta)$ is a joint probability density (or mass) function of x_i given θ. The larger the loglikelihood value of a model is, the better the model fits the data. Next, we consider the Akaike information criterion (AIC) to evaluate the performance of model fitting. This is represented in the following equation [23].

$$AIC = -2L(x|\hat{\theta}) + 2k \tag{12}$$

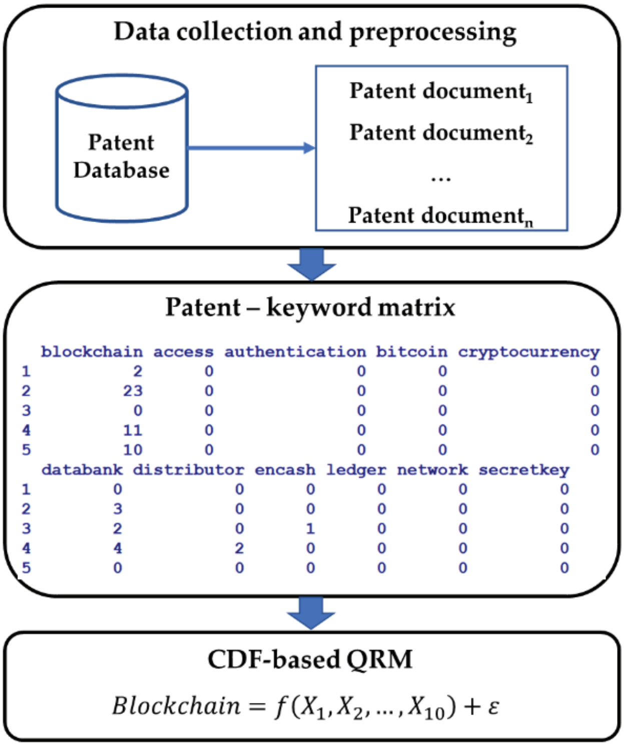
In Equation (12), $\hat{\theta}$ is the maximum likelihood estimate of θ and $L(x|\hat{\theta})$ is the maximum loglikelihood function given x. Also, k is the number of explanatory variables. The smaller the AIC value, the better the fitting performance of the model. Lastly, we apply the Bayesian information criterion (BIC) index to evaluate the performance of comparative models. The value of BIC is computed as follows [23]:

$$BIC = -2L(x|\hat{\theta}) + k \times log(n) \tag{13}$$

The BIC is an index that adds consideration to data size n to the AIC in Equation (13). As with the AIC, in the case of the BIC, the smaller this value is, the better the model performance is. In this paper, we compared the proposed QRM with LRM and a zero-inflated model in terms of explanatory and predictive power. Loglikelihood is an index that measures the explanatory power of the model, and the AIC and BIC are indexes that compare the predictive power between models.

4. Experiments and Results

The experiments were carried out using practical patent documents to illustrate how the proposed method can be applied to real fields. We collected patent documents related to blockchain technology from world patent databases [30,31]. Blockchain technology has been developed by relying on the blockchain-related technologies such as bitcoin and cryptocurrency. So, in this experiment, we provide the technological relations between blockchain technology and other related technologies based on the keywords of blockchain, access, authentication, bitcoin, cryptocurrency, databank, distributor, encash, ledger, network, and secretkey. In this paper, we determined blockchain technology as our target domain. Blockchain is defined as a technology for securely managing data across distributed systems [6]. We select the keyword blockchain as the response variable and use the remaining ten extracted keywords (access, authentication, bitcoin, cryptocurrency, databank, distributor, encash, ledger, network, and secretkey) for explanatory variables ($X_1, X_2, \ldots, X_{10}$). Figure 2 shows the process of our proposed modeling of PKA.



```
  blockchain access authentication bitcoin cryptocurrency
1          2      0              0       0              0
2         23      0              0       0              0
3          0      0              0       0              0
4         11      0              0       0              0
5         10      0              0       0              0
  databank distributor encash ledger network secretkey
1        0           0      0      0       0         0
2        3           0      0      0       0         0
3        2           0      1      0       0         0
4        4           2      0      0       0         0
5        0           0      0      0       0         0
```

Figure 2. Patent keyword analysis process.

First, we collected the patent documents related to blockchain technology using keyword search expression from patent databases across the world [30,31]. Next, we chose the valid patents representing blockchain technology and preprocessed the valid patent documents. In our experiments, we used the R project as a tool for statistical analysis. R is a free, open-source piece of software that supports statistical analysis and visualization [32]. The current version of R has been upgraded to 4.4.2. Up until now, R has been widely used for statistical analysis of data generated in various fields [33]. We also used the tm package of R for text mining [1]. This package provides many functions for preprocessing of text data using natural language processing [1,2]. Lastly, we used the cdfquantreg package of R for QRM [14]. In addition, using the functions provided in the R base module and the pscl package, we carried out performance evaluation between the proposed model and the comparative models [32,34]. The elements of this matrix are the frequency values of the keywords occurring in the patent documents. This is structured data that can be used in CDF-based QRM. Also, we determined the keyword of blockchain for the dependent variable and used the other keywords for independent variables in this experiment. To select the patent keywords for blockchain technology, we considered the results of keyword extraction from previous research related to blockchain technology analysis [32]. Therefore, we determined one response variable (blockchain) and ten explanatory variables (access, authentication, bitcoin, cryptocurrency, databank, distributor, encash, ledger, network, and secretkey). We used the R project and package for our experiment [1,2,14,32–34]. Table 1 shows the summary statistics of the patent keywords.

Table 1. Summary statistics of blockchain patent keywords.

Keyword	Min	Median	Mean	Max
blockchain	0	2	4.4140	38
access	0	0	0.2606	11
authentication	0	0	0.4763	16
bitcoin	0	0	0.1701	20
cryptocurrency	0	0	0.2115	11
databank	0	0	1.7800	24
distributor	0	0	0.5626	14
encash	0	0	0.1058	4
ledger	0	0	0.8892	26
network	0	0	0.9332	16
secretkey	0	0	0.5144	10

In the results in Table 1, we found that the patent–keyword matrix data is very sparse and zero-inflated because most elements of the matrix are zeros. The median values of most keywords were also zero. Therefore, we have difficulty analyzing the patent keyword data using traditional data analysis methods. To overcome the problem, we proposed patent keyword analysis using CDF-based QRM in this paper. In the CDF-based QRM, the response variable must have real numbers between 0 and 1. So, we changed the values of blockchain keyword by the following normalization.

$$Blockchain_{normalization} = \frac{Blockchain - Min(Blockchain)}{Max(Blockchain) - Min(Blockchain)} \quad (14)$$

Using the Equation (14), the values of the response variable are changed to numerical values in the interval (0,1). The model of patent keyword analysis consists of one response variable of the keyword blockchain and ten explanatory variables of all keywords except blockchain as follows.

Y: blockchain
$X_1, X_2, \ldots, X_{10}$: access, authentication, bitcoin, cryptocurrency, databank, distributor, encash, ledger, network, secretkey

Using the indexes of (11), (12), and (13), we compared the performance between CDF-based QRM and LRM. Table 2 shows the results of model performance between the compared models according to loglikelihood, AIC, and BIC. In this paper, we compared our proposed QRM with LRM and the zero-inflated model in terms of explanatory and predictive power. Loglikelihood is an index that measures the explanatory power of the model, and AIC and BIC are indexes that compare the predictive power between models.

Table 2. Results of performance evaluation between comparative models.

Model	Loglikelihood			AIC			BIC		
	QRM	LRM	ZIP	QRM	LRM	ZIP	QRM	LRM	ZIP
access	1370.29	549.57	−3621.28	−2734.59	−1093.14	7250.55	−2719.36	−1077.91	7270.85
authentication	1370.81	548.26	−3627.43	−2735.61	−1090.51	7262.86	−2720.39	−1075.29	7270.85
bitcoin	1397.82	553.43	−3610.05	−2789.63	−1100.85	7228.10	−2774.41	−1085.63	7248.40
cryptocurrency	1402.22	557.10	−3598.48	−2812.43	−1108.21	7204.97	−2797.20	−1092.98	7225.27
databank	1383.69	561.47	−3599.97	−2761.38	−1116.94	7207.93	−2746.16	−1101.71	7228.23
distributor	1370.23	548.67	−3625.33	−2734.45	−1091.35	7258.66	−2719.23	−1075.13	7278.96
encash	1370.55	549.89	−3623.16	−2735.11	−1093.77	7254.32	−2719.88	−1078.55	7274.62
ledger	1390.69	573.09	−3528.22	−2775.38	−1140.18	7064.44	−2760.16	−1124.95	7084.74
network	1374.97	555.71	−3612.41	−2743.93	−1105.43	7232.82	−2728.70	−1090.20	7253.12
secretkey	1382.74	551.14	−3622.10	−2759.47	−1096.27	7252.20	−2744.25	−1081.05	7272.50
All keywords	1486.48	610.57	−3429.08	−2948.96	−7797.14	6902.16	−2888.06	−1136.24	7013.81

In Table 2, to compare the performance of CDF-based QRM and LRM, we built simple models consisting of one keyword each and a full model using all keywords. First, the loglikelihood result shows that the loglikelihoods of CDF-based QRM for all keywords are larger than those of LRM. This shows that the results of patent keyword analysis using CDF-based QRM are better than those of the LRM. Next, in the comparison results based on AIC, the AIC values of CDF-based QRM are smaller than those of LRM for both the model using all keywords as well as the model using each keyword. We illustrate that CDF-based QRM is superior to LRM from the AIC perspective. Lastly, we compared the BIC values between CDF-based QRM and LRM. In Table 2, we can see that the BIC values of CDF-based QRM are larger than those of LRM. This means that the model performance of CDF-based QRM is better than LRM. Therefore, we show the validity of our proposed approach to patent keyword analysis from the comparison results by loglikelihood, AIC, and BIC.

The last column of each index that evaluates the performance of the model presents the results of the analysis using the statistical zero-inflated model. In this paper, we used the zero-inflated Poisson (ZIP) model as a statistical zero-inflated model [10,11]. This model uses the Poisson distribution as the probability function of the statistical zero-inflated model. The following shows the ZIP model [10,11].

$$\mathrm{P}(\mathrm{X}=\mathrm{x}) = \begin{cases} \pi + (1-\pi)e^{-\lambda}, \mathrm{x}=0 \\ (1-\pi)\dfrac{e^{-\lambda}\lambda^{x}}{X!}, \mathrm{x}>0 \end{cases} \tag{15}$$

Equation (15) uses the probability function of the Poisson distribution as $f(x)$ in Equation (3). In Equation (15), the λ is the parameter of Poisson distribution. In all indexes of loglikelihood, AIC, and BIC, we confirmed that model performance of ZIP is inferior to that of QRM or LRM. This is because the proportion of zeros included in the patent–keyword matrix data exceeds half, as we confirmed in Table 1. Therefore, we could confirm that our QRM is superior to the LRM or ZIP models. Finally, we represent the estimated parameter and *p*-value of each keyword in Table 3.

Table 3. Estimated parameter and *p*-value of each keyword.

Keyword	Estimated Parameter	*p*-Value
access	0.4714	0.3091
authentication	−0.2966	0.3479
bitcoin	−3.0515	<0.0001
cryptocurrency	−3.7862	<0.0001
databank	0.4457	<0.0001
distributor	−0.5016	0.1500
encash	0.9106	0.3098
ledger	0.6810	<0.0001
network	0.7086	0.0007
secretkey	−2.6537	<0.0001

Depending on the keyword, we found that some keywords have a positive impact on blockchain while others have a negative impact. Additionally, through the result of *p*-value, we confirmed that the keywords bitcoin, cryptocurrency, databank, ledger, network, and secretkey have a statistically significant impact on blockchain technology because the *p*-values of these keywords are less than 0.05 at the 95% confidence level. We can apply the results in Table 3 to various technology management areas such as R&D planning. From the result of Table 3, we constructed a technology diagram of blockchain in Figure 3.

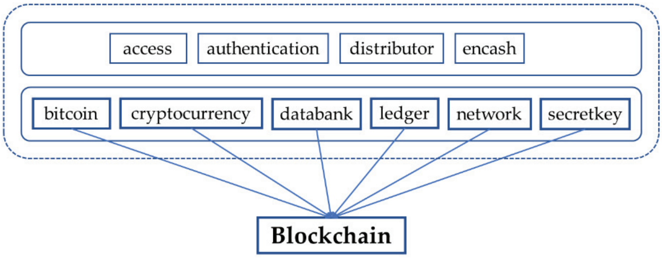


Figure 3. Blockchain technology diagram.

Among the 10 keywords related to blockchain, we can see that the keywords of bitcoin, cryptocurrency, databank, ledger, network, and secretkey have a statistically significant effect on blockchain. Therefore, we can see that technologies based on these keywords are primarily necessary for the development of blockchain technology. We expect that these results will contribute to R&D planning for blockchain technology development in countries and companies.

5. Discussion

From the result of Table 1, we found that the patent keyword data related to blockchain technology exhibit the zero-inflated problem. So, we used the proposed method to solve the problem. From the results in Table 2, we confirmed that the performance of QRM is better than those of the LRM and ZIP. Therefore, we estimated the model parameters and their p-values using the QRM in Table 3. Lastly, using the results in Table 3, we constructed a technology diagram of blockchain in Figure 3. From the results in Figure 3, we found that the sub-technologies based on the keywords of bitcoin, cryptocurrency, databank, ledger, network, and secretkey have a significant influence on the development of blockchain technology. In this paper, we applied the proposed method to analyze patent keyword data related to blockchain technology. From our experimental results, we showed how our method could be applied to real technology domains. Although the practical technology domain we used is blockchain, we believe that our proposed method can be extended to other technology fields. Once the target technology is determined, patent keyword analysis can be performed according to each step of the method proposed in this paper. Through this, we can conduct patent analysis necessary for R&D planning, new product development, technology forecasting, and technology innovation required in technology management.

In addition, we derived the QRM based on CDF to analyze the patent keywords. The patent–keyword matrix, which is usually used for patent keyword analysis, contains a large number of zeros, making it difficult for us to use existing linear models. If there are too many zero values, the zero values dominate the model building, which reduces the model performance. To solve this problem of zero inflation, we studied and proposed the CDF-based QRM in this paper.

In our study, we tried to identify the relationship structure between technologies through the PKA. Figure 3 was the final result obtained from our study. The technology diagram of Figure 3 provides a list of keywords that are statistically significant for blockchain technology. Therefore, in order to effectively develop blockchain technology, we must pay attention to detailed technologies based on these keywords. However, the results in Figure 3 do not provide any predictive information about the future technology of blockchain. In order to continuously develop blockchain technology, we need to predict the technology of blockchain. In addition, predicting the next technology related to the target technology will also be very meaningful in understanding the technology. To this end, it would also be meaningful to study how to use machine learning methods to predict the next behavior of animals [35]. Just as past patterns of animal behavior can be analyzed to predict future behavior, past patterns of technological development can be modeled to predict future technologies.

6. Conclusions

This paper presents a statistical model in order to solve the zero-inflated problem. We collected patent documents related to blockchain technology and analyzed them using a statistical data analysis method. Blockchain technology is a data management technology with distributed secure applications in various domains such as the financial field of Bitcoin. This technology is based on decentralization, immutability, transparency, and security. In this process, we constructed a patent–keyword matrix using preprocessed data for statistical analysis. Each element of this matrix is a frequency value of a keyword's occurrence in a patent document. Because most of the elements in this matrix are zero, we had difficulty analyzing this matrix using statistical analysis methods including the zero-inflated model. Therefore, we proposed a method of PKA to overcome the zero-inflated problem in the preprocessed patent data. Compared to existing single models such as LRM, we considered an analysis model consisting of two sub models representing location and dispersion. In addition, we changed the value of the response variable to a (0,1) interval. This is the concept of the CDF-based QRM.

In our experiment, we compared the model performance of the CDF-based QRM with LRM and ZIP to show the improved performance of our model. We searched the patents related to blockchain technology. The analytical results provided by the CDF-based QRM, LRM, and ZIP were evaluated using loglikelihood, AIC, and BIC. We found that all experimental results of the CDF-based QRM were better than those of the LRM and ZIP. Therefore, we showed the validity of the CDF-based QRM for our PKA. In the CDF-based QRM, we normalized the scale of the response variable y to solve the zero-inflated problem and confirmed the improved performance of the proposed method.

In this paper, the proposed model was used to finally select technology keywords that have a statistically significant impact on blockchain technology. We had difficulty identifying technological relationships between patent keywords using our proposed model. However, understanding the interrelationship structure between the sub-technologies required for blockchain technology development is an important task in understanding this technology. This part represents the limitations of our study. To overcome the limitation of our proposed model, we considered social network analysis (SNA) and Bayesian learning. In our future works, we apply SNA to our CDF-based QRM to make a technology diagram representing the technological relations between the patent technology keywords. In addition, we will apply Bayesian learning to the CDF-based QRM. We call this Bayesian learning for QRM. In this model, we assume the prior distributions for the parameters of the QRM model. This learning model updates the model parameters using the given data. That is, we will be able to improve the QRM performance of explanatory and predictive power using the Bayesian learning process whenever new data are added. The prior distribution of the parameters is updated by combining it with the likelihood function of newly observed data to form the posterior distribution of the parameters. That is, the parameters become random variables with probability distribution functions rather than fixed values and can be effectively used in the analysis of a zero-inflated patent–keyword matrix. Our research is expected to contribute to various fields by improving understanding of technology and finding relationships between detailed technologies through our PKA.

Author Contributions: S.P. designed this research and collected the dataset for the experiment. S.J. analyzed the data to show the validity of this paper, wrote the paper, and performed all the research steps. In addition, all authors cooperated with each other in revising the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Feinerer, I.; Hornik, K. *Package 'tm' Version 0.7-12, Text Mining Package*; CRAN of R Project; R Foundation for Statistical Computing: Vienna, Austria, 2024.
2. Feinerer, I.; Hornik, K.; Meyer, D. Text mining infrastructure in R. *J. Stat. Softw.* **2008**, *25*, 1–54. [CrossRef]
3. Park, S.; Jun, S. Zero-Inflated Patent Data Analysis Using Compound Poisson Models. *Appl. Sci.* **2023**, *13*, 4505. [CrossRef]
4. Uhm, D.; Jun, S. Zero-Inflated Patent Data Analysis Using Generating Synthetic Samples. *Future Internet* **2022**, *14*, 211. [CrossRef]
5. Jun, S. Zero-Inflated Text Data Analysis using Generative Adversarial Networks and Statistical Modeling. *Computers* **2023**, *12*, 258. [CrossRef]
6. Park, S.; Jun, S. Sustainable Technology Analysis of Blockchain Using Generalized Additive Modeling. *Sustainability* **2020**, *12*, 10501. [CrossRef]
7. Wagh, Y.S.; Kamalja, K.K. Zero-inflated models and estimation in zero-inflated Poisson distribution. *Commun. Stat.-Simul. Comput.* **2018**, *47*, 2248–2265. [CrossRef]
8. Feng, C.X. A comparison of zero-inflated and hurdle models for modeling zero-inflated count data. *J. Stat. Distrib. Appl.* **2021**, *8*, 8. [CrossRef]
9. Cameron, A.C.; Trivedi, P.K. *Regression Analysis of Count Data*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2013.
10. Hilbe, J.M. *Negative Binomial Regression*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2011.
11. Hilbe, J.M. *Modeling Count Data*; Cambridge University Press: Cambridge, UK, 2014.
12. Hogg, R.V.; Mckean, J.W.; Craig, A.T. *Introduction to Mathematical Statistics*, 8th ed.; Pearson: Essex, UK, 2020.
13. Shou, Y.; Smithson, M. cdfquantreg: An R Package for CDF-Quantile Regression. *J. Stat. Softw.* **2019**, *88*, 1–30. [CrossRef]
14. Shou, Y.; Smithson, M. *Package 'cdfquantreg' Version 1.3.1-2, Quantile Regression for Random Variables on the Unit Interval Package*; CRAN of R Project; R Foundation for Statistical Computing: Vienna, Austria, 2023.
15. Ding, J.; Du, D.; Duan, D.; Xia, Q.; Zhang, Q. A network analysis of global competition in photovoltaic technologies: Evidence from patent data. *Appl. Energy* **2024**, *375*, 124010. [CrossRef]
16. Shi, R.; Chai, K.; Wang, H.; Guo, S.; Zhai, Y.; Huang, J.; Yang, S.; Li, J.; Zhou, J.; Qiao, C.; et al. Comparative effectiveness of five Chinese patent medicines for non-alcoholic fatty liver disease: A systematic review and Bayesian network meta-analysis. *Phytomedicine* **2024**, *135*, 156124. [CrossRef]
17. Teshome, M.B.; Podrecca, M.; Orzes, G. Technological trends in mountain logistics: A patent analysis. *Res. Transp. Bus. Manag.* **2024**, *57*, 101202. [CrossRef]
18. Elsen, M.; Tietze, F. Contributions from low- and middle-income countries to the development of climate change adaptation technologies: A patent analysis. *Technol. Forecast. Soc. Change* **2024**, *209*, 123660. [CrossRef]
19. Zhao, X.; Wu, W.; Wu, D. How does AI perform in industry chain? A patent claims analysis approach. *Technol. Soc.* **2024**, *79*, 102720. [CrossRef]
20. Patel, M.S.; Franceschelli, D.; Grossbach, A.; Zhang, J.K.; Mercier, P.A.; Mattei, T.A. Top 50 Spine Surgery Publications Most Cited by Patents: A Bibliometric Analysis Focused on Research Driving Innovation. *World Neurosurg.* **2024**, *191*, 234–244. [CrossRef]
21. Ovsyannikov, I.R.; Zhdaneev, O.V. Forecast of innovative activity in key areas of energy transition technologies based on analysis of patent activity. *Int. J. Hydrogen Energy* **2024**, *87*, 1261–1276. [CrossRef]
22. Bruce, P.; Bruce, A.; Gedeck, P. *Practical Statistics for Data Scientists*, 2nd ed.; O'Reilly Media: Sebastopol, CA, USA, 2020.
23. Theodoridis, S. *Machine Learning a Bayesian and Optimization Perspective*; Elsevier: London, UK, 2015.
24. Montgomery, D.C.; Peck, E.A.; Vining, G.G. *Introduction to Linear Regression Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2012.
25. Smithson, M.; Shou, Y. CDF-quantile distributions for modelling random variables on the unit interval. *Br. J. Math. Stat. Psychol.* **2017**, *70*, 412–438. [CrossRef]
26. Chafamo, D.; Shanmugam, V.; Tokcan, N. C-ziptf: Stable tensor factorization for zero-infated multi-dimensional genomics data. *BMC Bioinform.* **2024**, *25*, 323. [CrossRef]
27. Yirdaw, B.E.; Debusho, L.K.; Samuel, A. Application of longitudinal multilevel zero infated Poisson regression in modeling of infectious diseases among infants in Ethiopia. *BMC Infect. Dis.* **2024**, *24*, 927. [CrossRef]
28. Zhou, W.; Huang, D.; Liang, Q.; Huang, T.; Wang, X.; Pei, H.; Chen, S.; Liu, L.; Wei, Y.; Qin, L.; et al. Early warning and predicting of COVID-19 using zero-infated negative binomial regression model and negative binomial regression model. *BMC Infect. Dis.* **2024**, *24*, 1006. [CrossRef]
29. Ren, J.; Loughnan, R.; Xu, B.; Thompson, W.K.; Fan, C.C. Estimating the total variance explained by whole-brain imaging for zero-inflated outcomes. *Commun. Biol.* **2024**, *7*, 836. [CrossRef]
30. KIPRIS. Korea Intellectual Property Rights Information Service. Available online: www.kipris.or.kr (accessed on 1 July 2023).
31. USPTO. The United States Patent and Trademark Office. Available online: http://www.uspto.gov (accessed on 1 June 2023).
32. R Development Core Team. R: A Language and Environment for Statistical Computing Version 4.4.0, R Foundation for Statistical Computing. Available online: http://www.R-project.org (accessed on 1 February 2024).
33. Foundation for Open Access Statistics, Journal of Statistical Software. Available online: https://www.jstatsoft.org (accessed on 1 June 2024).

34. Jackman, S.; Tahk, A.; Zeileis, A.; Maimone, C.; Fearon, J.; Meers, Z. *Package 'pscl' Version 1.5.9*; Political Science Computational Laboratory; CRAN of R Project; R Foundation for Statistical Computing: Vienna, Austria, 2023.
35. Meyer, P.G.; Cherstvy, A.G.; Seckler, H.; Hering, R.; Blaum, N.; Jeltsch, F.; Metzler, R. Directedeness, correlations, and daily cycles in springbok motion: From data via stochastic models to movement prediction. *Phys. Rev. Res.* **2023**, *5*, 043129. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

ADYOLOv5-Face: An Enhanced YOLO-Based Face Detector for Small Target Faces

Linrunjia Liu [1], Gaoshuai Wang [2] and Qiguang Miao [1,*]

[1] School of Computer Science and Technology, Xidian University, Xi'an 710071, China; liulinrunjia@xidian.edu.cn

[2] Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China; wanggaoshuai@sz.tsinghua.edu.cn

* Correspondence: qgmiao@xidian.edu.cn

Abstract: Benefiting from advancements in generic object detectors, significant progress has been achieved in the field of face detection. Among these algorithms, the You Only Look Once (YOLO) series plays an important role due to its low training computation cost. However, we have observed that face detectors based on lightweight YOLO models struggle with accurately detecting small faces. This is because they preserve more semantic information for large faces while compromising the detailed information for small faces. To address this issue, this study makes two contributions to enhance detection performance, particularly for small faces: (1) modifying the neck part of the architecture by integrating a Gather-and-Distribute mechanism instead of the traditional Feature Pyramid Network to tackle the information fusion challenges inherent in YOLO-based models; and (2) incorporating an additional detection head specifically designed for detecting small faces. To evaluate the performance of the proposed face detector, we introduce a new dataset named XD-Face for the face detection task. In the experimental section, the proposed model is trained using the Wider Face dataset and evaluated on both Wider Face and XD-face datasets. Experimental results demonstrate that the proposed face detector outperforms other excellent face detectors across all datasets involving small faces and achieved improvements of 1.1%, 1.09%, and 1.35% in the AP50 metric on the WiderFace validation dataset compared to the baseline YOLOv5s-based face detector.

Keywords: object detection; small face detection; YOLOv5; face dataset; classroom scenes; network structure design

Citation: Liu, L.; Wang, G.; Miao, Q. ADYOLOv5-Face: An Enhanced YOLO-Based Face Detector for Small Target Faces. *Electronics* **2024**, *13*, 4184. https://doi.org/10.3390/electronics13214184

Academic Editor: Manohar Das

Received: 13 September 2024
Revised: 12 October 2024
Accepted: 21 October 2024
Published: 25 October 2024

1. Introduction

Face detection aims to identify and locate faces within images or video streams, serving as a fundamental step in various face-related applications such as face verification, recognition, and expression analysis. With the advent of Convolutional Neural Network (CNN)-based object detectors, face detection has witnessed significant advancements in recent years. Evolving from two-stage approaches like Faster RCNN [1] and Mask RCNN [2] to one-stage methods such as SSD [3], RetinaNet [4] and YOLO [5], the object detection framework has facilitated the development of various face detectors based on these methods. Notable examples include RetinaFace [6], YOLOv5Face [7], RefineFace [8], and TinaFace [9]. Among these, YOLO series detectors stand out due to their low computational cost and high accuracy, making them widely adopted in industry applications.

However, despite the effectiveness of those deep learning-based models, existing face detectors exhibit a significant drawback: while they excel in recognizing large faces, their accuracy diminishes when it comes to detecting tiny faces. This disparity arises because large-scale targets necessitate a larger receptive field, causing the features of small targets to gradually fade on the deep feature map after multiple downsampling steps.

To address this issue, FaceBoxes [10] designed multi-scale anchors to enrich the receptive field and discretized anchors across different layers to handle faces of various sizes.

YOLOFaceV2 introduced a novel Receptive Field Enhancement module (RFE), providing a richer receptive field to capture both large and small-scale faces simultaneously. Reference [11] explored using attention mechanisms for facial feature extraction, preventing the misidentification of small-scale face features due to occlusion. However, these methods failed to fully utilize information between channels. They utilize a Feature Pyramid Network (FPN) in the neck architecture, which integrates information layer by layer, implying that features from different layers cannot be fused directly without passing through intermediate layers. Inspired by Gold-YOLO [12], we integrate a novel Gather-and-Distribute (GD) mechanism in the neck architecture, replacing FPN. This modification allows for the seamless transmission of information across layers, thereby enhancing feature fusion capabilities.

Other research methods improved small-face detection by deploying a large number of small anchors across the image. While these methods effectively increased the recall rate, they also exacerbated the issue of extreme class imbalance, leading to numerous false positives in detection results [13]. To address this problem, Lin et al. introduced Focal Loss, which dynamically adjusts the weight of hard samples [4]. Similarly, the Gradient Harmonizing Mechanism (GHM) [14] suppresses the gradients of easy samples to prioritize hard ones, while PISA [15] assigns weights to positive and negative samples based on different criteria. YOLOFaceV2 [16] proposed a Slide Weight Function (SWF) to adjust the weights of imbalanced samples, enabling adaptive learning of the threshold parameters for positive and negative samples. Building on previous work, we incorporate four detection heads instead of three, for detecting tiny, small, medium, and large faces. While the additional tiny head increases small anchors for detecting small objects, Normalized Wasserstein Distance (NWD) loss and Intersection over Union (IOU) loss are combined in this work to balance the detection of both large and small samples.

To evaluate the performance of our proposed method, we create a dataset specifically focusing on scenes with small faces, named the XD-face dataset. We choose 2802 images and label 102,250 faces with small size and occlusion in classroom scenes. We compare the recognition results of our proposed model on both the WiderFace and XD-face datasets with those of other state-of-the-art approaches. With our self-trained model, our method achieves mAP scores of 94.80%, 93.77%, and 84.37% on the easy, medium, and hard subsets of the Wider Face dataset, respectively, and 59.7% on the XD-face dataset.

The key contributions of this work are summarized as follows:

(1) Improvement of the neck structure of YOLO series models and utilization of the GD mechanism instead of the FPN to enhance information fusion capabilities;

(2) Addition of an extra detection head to improve the detection of tiny faces;

(3) Constructing a novel dataset for face detection, focusing specifically on classroom scenes with small faces.

2. Related Work

2.1. Two-Stage Face Detection Methods

In recent years, face detection methods based on convolutional neural networks can generally be divided into two-stage methods and single-stage methods. The core idea of two-stage face detection methods is to generate a limited number of candidate regions (Proposals) in the first stage, and then refine the candidate set in the second stage to generate the final results.

Faster R-CNN [1] is a typical two-stage object detection method. In the first stage, it utilizes a Region Proposal Network (RPN) to generate a series of candidate regions, followed by classification and regression predictions using candidate boxes in the second stage. Jiang et al. [17] first applied the Faster R-CNN model to face detection and achieved good detection results on two widely used face detection datasets. Sun et al. [18] improved upon Faster R-CNN by employing feature concatenation, hard sample mining, multi-scale training, model pre-training, and fine-tuning of key parameters to enhance face detection performance. Zhu et al. [19] subsequently proposed CMS-RCNN, which detects unconstrained faces by utilizing contextual information. Khan et al. [20] used multi-task cascaded convolutional neural networks and modified the layer density with increasing neuron count. It enriches feature information while increasing the amount of calculation. Nonetheless, early methods of face detection based on deep learning exhibited significant drawbacks, including intricate training processes, a susceptibility to local optima, protracted detection times, and suboptimal accuracy.

2.2. One-Stage Face Detection Methods

Due to the substantial computational time required by two-stage face detection methods, single-stage face detection methods were subsequently proposed. Single-stage face detection methods based on SSD [3] and YOLO [5] do not require generating candidate regions.

Compared to two-stage methods, single-stage methods offer faster detection speed and represent a type of end-to-end face detection method. The network structure of single-stage face detection methods can generally be divided into three parts: backbone network, neck network, and head network.

Backbone: The backbone network usually employs models such as VGG [21], ResNet [22], DenseNet [23,24], SENet [25], MobileNet [26], ShuffleNet [27], etc. BFBox [28] searches for backbone networks suitable for face space. SCRFD [29] introduces a computation redistribution method to redistribute computational resources between the backbone network, neck network, and head network of the model.

Neck: After the introduction of the FPN structure [30], it often serves as the basic neck network for face detection. FANet [31] creates a new hierarchical efficient FPN with rich semantics at all scales. BFBox [28] proposes an FPN attention module to jointly search the architecture of the backbone network and FPN.

To further enhance face features, some face detection methods choose to use different convolution kernels in each branch to enlarge the receptive field. SSH [32] adds larger convolution kernels to each detection module to incorporate contextual information. Wang et al. [33] conducted research on low-light face detection in unannotated scenarios by combining context learning with contrastive learning. DSFD [34] introduces a feature enhancement module to strengthen the original features, making them more recognizable and robust. RefineFace [8] constructs an RFE module to provide diversified perspectives for detecting faces with extreme poses. SmallHardFace [35] expands the receptive field using dilated convolution. YOLO5Face [7] introduces face keypoint loss into the network and improves the SPP module in YOLOv5 by using smaller convolution kernels instead of larger ones to make it more suitable for face detection.

Although these methods enhance the receptive field and enrich semantic information, providing a simple and effective means to improve multi-scale object detection performance, they may result in the loss of spatial information. Therefore, this paper proposes the use of a GD mechanism instead of FPN in the neck architecture to combine deep semantic information with shallow features.

Head: The head part of the network structure connects the loss function for optimizing model parameters through backpropagation. RetinaFace [6] introduced facial keypoint information into the face detection system, thereby augmenting the facial keypoint regression loss function. Fang et al. [36] introduced the hierarchical loss Triple Loss to optimize the face detector based on the feature pyramid network. Inspired by IoU-awareness [37], TinaFace [9] incorporated a regression branch into the head network and utilized cross-entropy loss to predict the IoU value between annotated bounding boxes and detection results. Although the introduction of different loss functions alleviates the issue of sample imbalance, it also introduces additional hyperparameters, making the model training more complex.

Considering some of the current advanced face detection algorithms are designed to be overly complex and have high computational costs, this paper proposes improvements based on the commonly used object detection model YOLOv5. To solve the information fusion problem of the traditional FPN structure, this paper integrates a GD mechanism.

For improving the detection accuracy of small faces, a tiny-face detection head is added in this paper.

3. Proposed Method

3.1. Overview of ADYOLOv5-Face

With the maturity of facial detection neural network technology, most one-stage facial detection neural models possess a three-stage structure consisting of backbone, neck, and detection networks. The backbone network is responsible for extracting high-dimensional feature information: locational information on high-resolution feature maps and semantic information on low-resolution feature maps. Then, the neck network fuses different features extracted by the backbone network. Finally, the head network usually contains three detection heads for detecting small, medium and large objects.

This paper aims to focus on the information fusion capability of the neck network and introduce the GD mechanism to resolve the issue of information loss caused by cross-layer feature integration in current models. In the head part, one more head for tiny object detection is added. As for the backbone part, the CSPDarknet53 was kept unchanged, identical to its structure in YOLOv5. The architecture of our ADYOLOv5-Face detector is depicted in Figure 1.

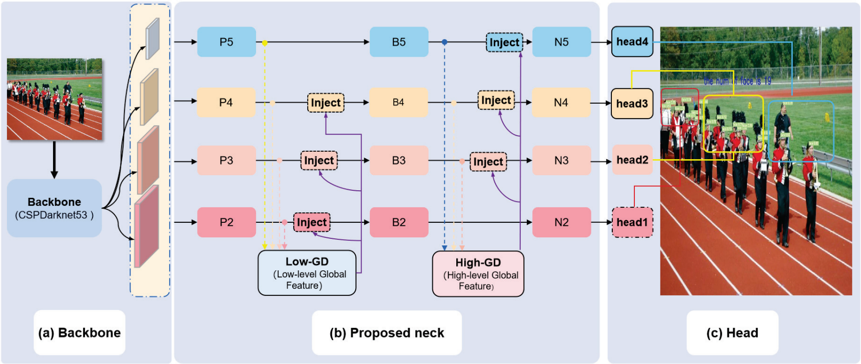


Figure 1. The architecture of the ADYOLOv5-Face: (**a**) CSPDarknet53 backbone as the original YOLOv5 version. (**b**) The neck uses a structure like the Gather-and-Distribute mechanism. (**c**) Four prediction heads use the feature maps from the neck. Head1 contains more detailed information that is added to detect tiny faces.

3.2. Architecture of the Neck Part

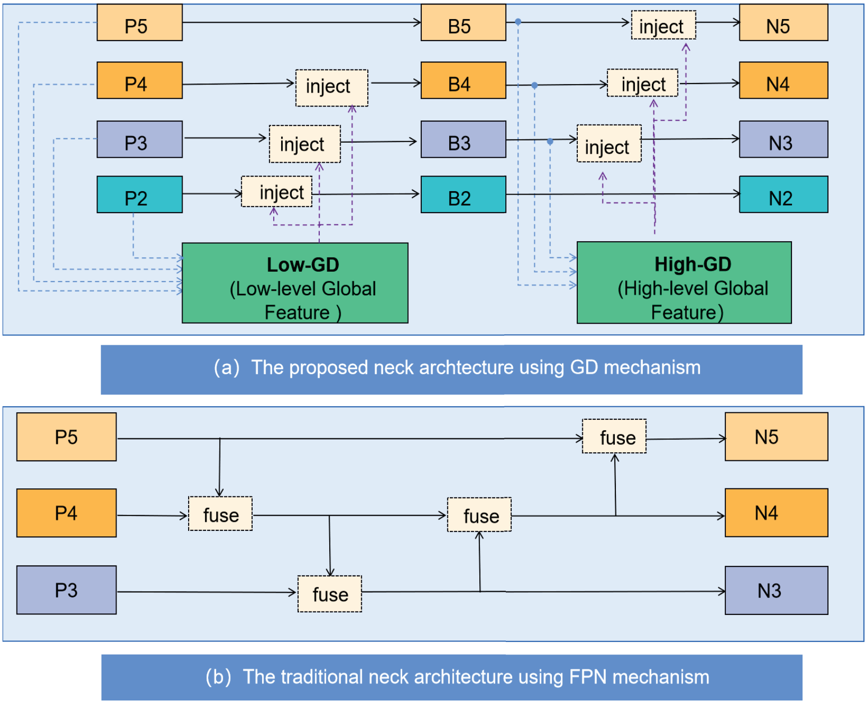
The backbone part is responsible for extracting high-dimensional feature information, while the neck part fuses different features extracted by the backbone network—namely, the positional information from high-resolution feature maps and the semantic information from low-resolution feature maps. As shown in Figure 2a, the proposed neck structure in this project is as follows:

Step 1: Concatenate the four layers of features, $P2$, $P3$, $P4$, and $P5$, extracted by the backbone network to obtain the low-level global feature F_{global_low}.

Step 2: Inject the low-level global feature F_{global_low} back into the local features $P2$, $P3$, and $P4$, achieving an initial fusion of global and local features, resulting in features $B2$, $B3$, and $B4$. $B2$ will be directly used as the final output, preserving the high-resolution positional information of the low-level features without loss.

Step 3: The low-level fused features $B2$, $B3$, and $B4$ are further concatenated to obtain a high-level global feature F_{global_high} with richer semantic information. To fully integrate the global features, a transformer module is proposed, combining attention mechanisms and convolution operations to better handle information from different positions in the image.

Step 4: Finally, $F2$ is injected back into the $B3$, $B4$, and $B5$ features, resulting in high-level fused features $N3$, $N4$, and $N5$, which carry rich semantic information and are fed into the prediction network as output layer features.



(a) The proposed neck archtecture using GD mechanism

(b) The traditional neck architecture using FPN mechanism

Figure 2. The comparison of the proposed neck part with the traditional neck part.

Compared to the traditional mid-level network structure shown in Figure 2b, the complex mid-level network proposed in this paper directly fuses global features with features from each layer during feature fusion, avoiding information loss that may occur during cross-layer transmission and thus enhancing small object detection performance. At the same time, the design of the global feature fusion module also considers the integration of new technologies with traditional methods, improving the model's feature fusion capability without significantly increasing latency, thereby enhancing the object detection performance. Lastly, since the focus of this project is on detecting small faces, the model strengthens the use of low-level information ($P2$) by adding a low-level output feature ($N2$) to preserve more detailed features.

3.3. Details of the Neck Part GD Mechanism

The GD mechanism consists of three parts: feature alignment module (FAM), information fusion module (IFM) and information injection module (IIM).

To better represent the features of varying sizes, the GD mechanism can be divided into low-stage and high-stage. The feature alignment module and information fusion module are different in the two GD mechanism branches, while the information injection module is the same.

- Low-stage Gather-and-Distribute mechanism

For the low-stage Gather-and-Distribute mechanism (Low-GD), the structure is shown in Figure 3. The details are provided as follows:

(1) Align the four features $P2, P3, P4$, and $P5$ extracted by the backbone network in a unified size using the average pooling operation (AvgPool) and Bilinear operation.

$$F_{align} = Low\_FAM([P2, P3, P4, P5]) \tag{1}$$

(2) Concatenate the four features, and employ RepBlock to obtain the low-level global feature.

$$F_{fuse} = RepBlock(F_{align}) \tag{2}$$

(3) Split the low-level global feature into F_{inj_B2}, F_{inj_B3} and F_{inj_B4} with different channel dimensions.

$$F_{global\_low} = F_{inj\_B2}, F_{inj\_B3}, F_{inj\_B4} = Split(F_{fuse}) \tag{3}$$

(4) Next is the information injection module. As shown in Figure 1, the three split features (F_{inj_Bi}) are injected into the local features $P2, P3$, and $P4$ to achieve the initial fusion of global and local features, resulting in features $B2, B3$, and $B4$. $B2$ will be directly output as the final feature, preserving the locational information of the low-level high-resolution features without loss. The details of the information injection module are shown in Figure 4, and the formula is as follows:

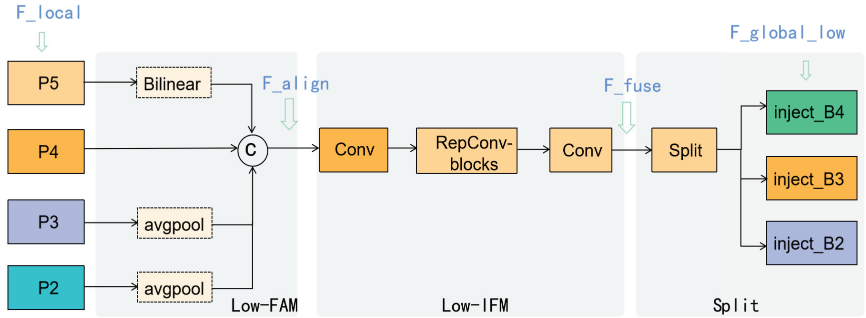


Figure 3. The structure of the Low-GD in the neck part. The notations in blue are aligned with the corresponding features in Equations (1)–(3).

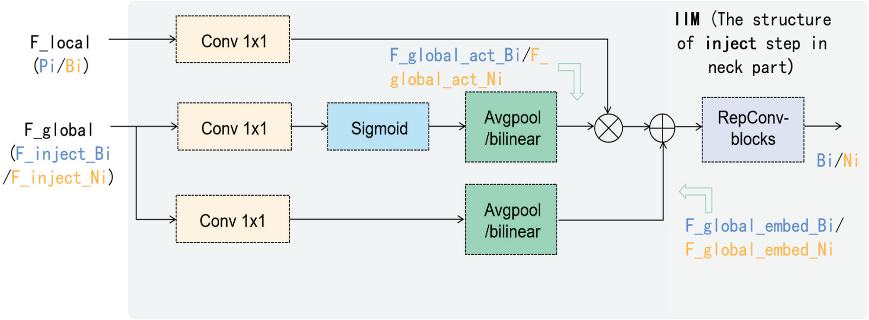


Figure 4. The structure of inject step in the neck part, which is named the information injection module (IIM). The notations in blue are aligned with the corresponding features in Equations (4)–(7), while the notations in yellow are aligned with the corresponding features in Equations (11)–(14).

$$F_{global\_act\_Bi} = resize(sigmoid(Conv(F_{inj\_Bi}))) \tag{4}$$

$$F_{global\_embed\_Bi} = resize(Conv(F_{inj\_Bi})) \tag{5}$$

$$F_{att\_fuse\_Bi} = Conv(Pi) * F_{global\_act\_Bi} + F_{global\_embed\_Bi} \tag{6}$$

$$Bi = RepBlock(F_{att\_fuse\_Bi}) \tag{7}$$

- High-stage gather-and-distribute mechanism

For the high-stage Gather-and-Gistribute mechanism (High-GD), the structure is shown in Figure 5. The low-order fused features $B3$, $B4$, and $B5$ will be further concatenated to obtain the high-order global feature with richer semantic information. In order to fully integrate global features, a transformer module instead of the RepConv-Block is proposed to better handle information from different positions in the image by combining attention mechanisms with convolution operations.

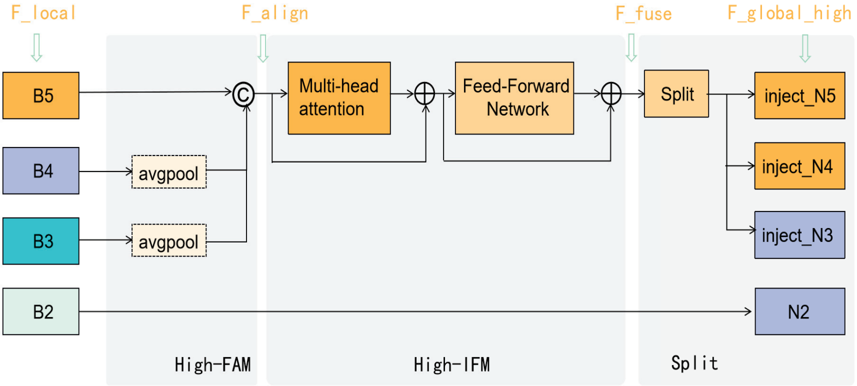


Figure 5. The structure of the High-GD in the neck part. The notations in blue are aligned with the corresponding features in Equations (8)–(10).

The details are provided as follows:

(1) Align the three features $B3$, $B4$, and $B5$ extracted by the backbone network in a unified size by AvgPool.

$$F_{align} = High\_FAM([B3, B4, B5]) \tag{8}$$

(2) Concatenate the three features and employ the transformer fusion module to obtain the high-level global feature.

$$F_{fuse} = Transformer(F_{align}) \tag{9}$$

(3) Split the high-level global feature into F_{inj_N3}, F_{inj_N4} and F_{inj_N5} with different channel dimensions.

$$F_{global\_high} = F_{inj\_N3}, F_{inj\_N4}, F_{inj\_N5} = Split(Conv1 \times 1(F_{fuse})) \tag{10}$$

(4) The information injection module here is the same as in the Low-GD branch, as shown in Figure 4. The three split features (F_{inj_Ni}) are injected into the features $B3$, $B4$, and $B5$ to obtain high-order fused information $N3$, $N4$, and $N5$ with rich semantic information, which are then fed into the detection network as output layer features. The formula is as follows:

$$F_{global\_act\_Ni} = resize(sigmoid(Conv(F_{inj\_Ni}))) \tag{11}$$

$$F_{global\_embed\_Ni} = resize(Conv(F_{inj\_Ni})) \tag{12}$$

$$F_{att\_fuse\_Ni} = Conv(Bi) * F_{global\_act\_Ni} + F_{global\_embed\_Ni} \tag{13}$$

$$Ni = RepBlock(F_{att\_fuse\_Ni}) \tag{14}$$

3.4. Prediction Head for Tiny Faces

After investigating the WiderFace dataset, we find that it contains many extremely small faces. Therefore, one more head is added to the proposed ADYOLOv5-Face for detecting tiny faces. As shown in Figure 1, head1 is generated from feature map $B2$, which contains more details of the tiny faces compared to the other three prediction heads. The performance of tiny face detection has large improvements after adding the tiny prediction head, even though the computation cost has also increased.

3.5. Loss

Since metrics based on IoU, such as IoU itself and its extensions, are highly sensitive to positional deviations of small objects, ref. [38] proposed a new evaluation method for small object detection based on the Wasserstein distance. Equation (15) calculates the Wasserstein distance between the predicted object and the ground truth object.

$$W_2^2(\mathcal{N}_a, \mathcal{N}_b) = \left\| \left(\left[cx_a, cy_a, \frac{w_a}{2}, \frac{h_a}{2} \right]^T, \left[cx_b, cy_b, \frac{w_b}{2} \cdot \frac{h_b}{2} \right]^T \right) \right\|_2^2 \tag{15}$$

where c is a constant specific to the dataset, $\mathcal{N}_a$ and $\mathcal{N}_b$ are Gaussian distributions characterized by bounding boxes $A = (cx_a, cy_a, w_a, h_a)$ and $B = (cx_b, cy_b, w_b, h_b)$, respectively.

However, $W_2^2(\mathcal{N}_a, \mathcal{N}_b)$ is a distance metric and cannot be directly used as a similarity measure (i.e., a value between 0 and 1 like IoU). Therefore, we normalize its exponential form and obtain a new metric called the Normalized Wasserstein Distance (NWD), as shown in Equation (16).

$$NWD(\mathcal{N}_a, \mathcal{N}_b) = exp\left(-\frac{\sqrt{W_2^2(\mathcal{N}_a, \mathcal{N}_b)}}{C} \right) \tag{16}$$

The loss function based on the NWD metric is expressed as follows:

$$\mathcal{L}(NWD) = 1 - NWD(\mathcal{N}_a, \mathcal{N}_b) \tag{17}$$

Therefore, this paper uses the NWD loss to compensate for the shortcomings of IoU loss in small object detection while retaining the IoU loss for detecting large objects.

4. Experiment Setup

4.1. Datasets

Proposed Dataset (XD-face): The XD-face dataset comprises face detection data extracted from nine classroom videos captured within our school environment, encompassing a wide array of face images varying in size, pose, and occlusion levels. As illustrated in Table 1, the dataset consists of 2802 images and a total of 102,250 labeled faces, each annotated with high accuracy (with data organized as 5 values per row: category, x_center, y_center, width, height), reflecting variations in illumination and face density. Due to the fixed camera positioning within the classroom setting, even in scenes with few individuals, the faces captured tend to be very small. The dataset meets the requirements in this paper for detecting small faces and is thus utilized to evaluate the proposed model, benchmarked against other YOLO-based methods. Figure 6 illustrates the visualization of the conditions and detection results in XD-face. Even though the faces in this dataset are very small and are occluded in most locations, they are detected by our detector.

The XD-face dataset, as a classroom scene dataset, holds potential for applications in face counting, face recognition, and domain adaptation.

Table 1. The contents of the proposed XD-face dataset encompass a rich array of facial images in different scenes.

Name	Images	Faces	Dense/Bright	Dense/Dark	Sparse/Bright	Sparse/Dark
A-208	142	6208	142	0	0	0
A-211	150	2001	100	50	0	0
B-301	320	25,108	160	160	0	0
B-403	300	11,775	150	150	0	0
B-418	315	17,351	165	150	0	0
B-507	335	21,478	185	150	0	0
B-509	300	2216	0	0	0	300
C-107	400	5373	0	0	400	0
C-225	540	10,740	300	0	240	0
Total	2802	102,250	1202	660	640	300

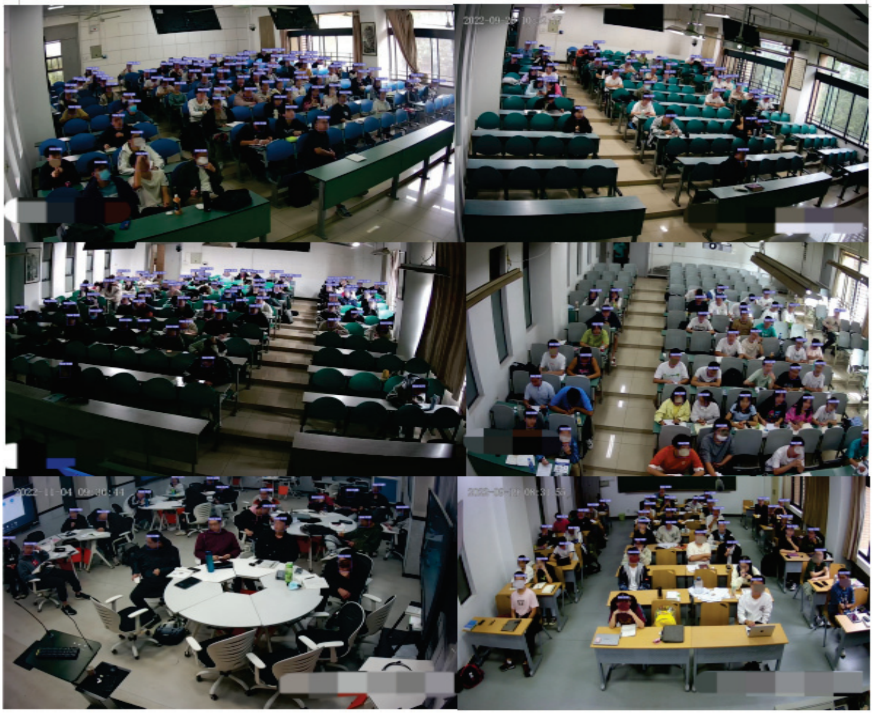


Figure 6. Visual comparisons of detecting conditions and results in our dataset: XD-face. XD-face contains 2802 images and labels 102,250 faces with small size and occlusion in several different classroom scenes.

Firstly, in terms of face counting, the diverse classroom scenes in the dataset can assist researchers in developing more accurate algorithms for identifying and counting the number of students in a class, which is crucial for educational management and classroom optimization.

In face recognition, the XD-face dataset can enhance the efficiency of classroom attendance and identity verification systems, such as automatically recognizing students and recording attendance. Additionally, teachers can use this technology to track student engagement and performance, providing data to support personalized teaching.

Regarding domain adaptation, the features of this dataset allow algorithms to adjust across different classroom environments, improving recognition under varying lighting conditions and angles. This is vital for ensuring system stability and reliability in remote or hybrid learning environments.

Overall, the XD-face dataset provides a rich foundation for the application of facial recognition technologies in education, contributing to improved teaching efficiency and student management.

Wider Face: As a widely recognized benchmark dataset utilized across different face detection methods, Wider Face comprises 61 scene classes, totaling 32,203 images and 393,703 labeled faces. Distinguished from the proposed XD-face dataset, Wider Face encompasses a diverse range of facial sizes, including small, medium, and large faces. The dataset categorizes faces into three levels based on their sizes: Easy, Medium, and Hard, with the Hard level representing the most challenging subset, ideal for evaluating the accuracy of proposed models. In this study, the training set, comprising 40% of the dataset, is utilized for training the proposed model, while the validation set, comprising 10% of the dataset, is employed for evaluation purposes. The evaluation metric employed for Wider Face is the mean average precision at an IOU threshold of 0.5 (mAP@.5).

4.2. Experimental Evaluation Metrics

This paper selects Average Precision (AP) [39] as the evaluation metric. The area under the PR curve represents the AP.

PR Curve: It reflects the relationship between precision and recall. The horizontal axis represents recall, and the vertical axis represents precision.

Confusion Matrix: A confusion matrix is a crucial tool for evaluating the performance of classification models, especially for binary and multi-class classification tasks. For binary classification problems, the confusion matrix typically takes a 2 × 2 form. The confusion matrix is defined as shown in Table 2:

Table 2. Confusion matrix for face detection model.

Confusion Matrix		**Actual**	
		Face	**Non-Face**
Predicted	Face	TP	FP
	Non-Face	FN	TN

- True Positive (TP): the number of instances correctly predicted as positive (face).
- False Negative (FN): the number of instances incorrectly predicted as negative (non-face) when they are actually positive (face).
- False Positive (FP): the number of instances incorrectly predicted as positive (face) when they are actually negative (non-face).
- True Negative (TN): the number of instances correctly predicted as negative (non-face).

Precision measures the proportion of true positive instances among the instances predicted as positive by the model, indicating the accuracy of the model's positive predictions. The formula for precision is given in Equation (18):

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

Recall measures the proportion of true positive instances among all actual positive instances, indicating the model's coverage ability. The formula for recall is given in Equation (19):

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

In face detection, a key metric for evaluating detection results is IoU, which measures the overlap between the predicted bounding box and the ground truth bounding box. It is an important standard for assessing model accuracy. The IoU is calculated using the formula shown in Equation (20):

$$IoU = \frac{A \cap B}{A \cup B} \tag{20}$$

Here, B is the predicted bounding box and G is the ground truth bounding box. If the IoU of the predicted box and the ground truth box is greater than a preset threshold (usually 0.5), the predicted box is considered a TP; otherwise, it is considered an FP. Precision and recall are then calculated using Equations (18) and (19).

AP is a summary measure of the PR curve, assessing the model's overall performance across different thresholds. AP is the numerical integration of the area under the PR curve, obtained by calculating the average precision across all recall levels. The higher the AP, the better the model performs across different recall levels.

AP50: IOU for the NMS is set to 0.5.

AP@50:5:95: The IOU values range from 0.5 to 0.95, with a step size of 0.5.

Since AP50 is a commonly used metric in object detection, we primarily assess the performance of our detector and other state-of-the-art face detectors using AP50. For the experiments on XD-face, however, we evaluate the performance of these detectors across four metrics—precision, recall, AP50, and AP50:5:95—in order to highlight the detection difficulty on the proposed dataset.

4.3. Ablation Experiment

We utilize the YOLOv5s model as our baseline and integrate the occlusion-aware repulsion loss, drawing inspiration from the work of [16]. The training process runs 250 epochs on the Wider Face training set with a batch size of 32. For optimization, we employ SGD with momentum. The initial learning rate is set to 1×10^{-2}, gradually decaying to 1×10^{-5}, with a weight decay of 5×10^{-3}. Following this configuration, the proposed ADYOLOv5-Face model implements modifications to the neck and head structures.

The detection accuracy (using AP50 as the metric) of the baseline model and the proposed model in the validation set of Wider Face is shown in Table 3. For the baseline model, the detection accuracy in easy, medium, are hard levels is 93.70%, 92.68% and 83.02%, respectively. When the GD mechanism is applied into the neck structure, the detection accuracy of easy and medium levels is increased by 1.46% and 0.74%, respectively, while the detection performance in the hard level is damaged by 1.66%. To improve the detection accuracy of small faces, a tiny face detection layer is added to enhance the representation ability of detail features. Therefore, the advanced model achieves 94.80%, 93.77% and 84.37% detection accuracy in easy, medium, are hard levels, respectively. Compared to the baseline model, the ADYOLOv5-Face model has 1.1%, 1.09% and 1.35% increments in easy, medium, and hard levels, respectively. In this way, the proposed model improves the detection ability for small target faces without sacrificing the recognition accuracy of large faces.

Table 3. Ablation study results of the AP50 metric on the Wider Face validation set.

Modification	Easy (%)	Medium (%)	Hard (%)	Params (M)	Flops (G)
baseline	93.70	92.68	83.02	7.063	16.4
+ neck	95.16	93.42	81.36	10.008	21.2
+neck +head	94.80	93.77	84.37	10.123	22.8

4.4. Contrast Experiment

4.4.1. Experiments on Wider Face

In this section, we compare our model with various state-of-the-art face detectors, as presented in Table 4, which includes detectors based on ResNet, YOLOv5, YOLOv7 [40], and YOLOv8 networks. The data presented in the table are obtained from the paper [7].

Table 4. Comparison of our ADYOLOv5-Face with the state-of-the-art face detectors on the Wider Face validation dataset (using AP50 as the evaluation metric).

Detector	Backbone	Easy (%)	Medium (%)	Hard (%)
DSFD (2019) [34]	ResNet152	94.29	91.47	71.39
RetinaFace (2020) [6]	ResNet50	94.92	91.90	64.17
HAMBox (2020) [41]	ResNet50	95.27	93.76	76.75
TinaFace (2020) [9]	ResNet50	95.61	94.25	81.43
SCRFD-2.5GF (2021) [29]	Basic ResNet	93.78	92.16	77.87
TinyYolov3 (2022) [42]	YOLOv3-tiny	95.26	89.2	77.9
YOLOv7-tiny-Face (2022) [7]	YOLOv7-tiny	94.7	92.6	82.1
YFaces-Tiny (2024) [43]	YOLOv7-tiny	94.07	92.36	83.06
YOLOv8n-face (2023) [7]	YOLOv8n	94.5	92.2	79
YOLOv5s-Face (2021) [7]	YOLOv5s	94.33	92.61	83.15
ADYOLOv5-Face (ours)	YOLOv5s	94.80	93.77	84.37

Upon evaluating the detection performances (using AP50 as the metric) for large-scale faces, it is observed that TinaFace achieves the highest detection accuracy in the easy (95.61%) and medium (94.25%) subsets on the Wider Face validation dataset, slightly outperforming our proposed model (94.80% and 93.77%, respectively). However, when evaluating the detection performance for small faces, our proposed model outperforms other face detectors. Specifically, our proposed model achieves an 84.3% detection accuracy in the hard-level subset, demonstrating a 20.2% improvement over RetinaFace and 1.22% over the second-best model, YOLOv5s-Face.

In comparison with YOLOv5s-Face, YOLOv7-tiny-Face, and YOLOv8n-face, which belong to the same YOLO algorithm and possess similar model size and computational requirements, our proposed model attains the highest accuracy across all three subsets. Notably, it demonstrates improvements of 1.22%, 2.27%, and 5.37%, respectively, in the hard-level subset. Figure 7 provides a clearer illustration of the superiority of the proposed method's performance on the hard set.

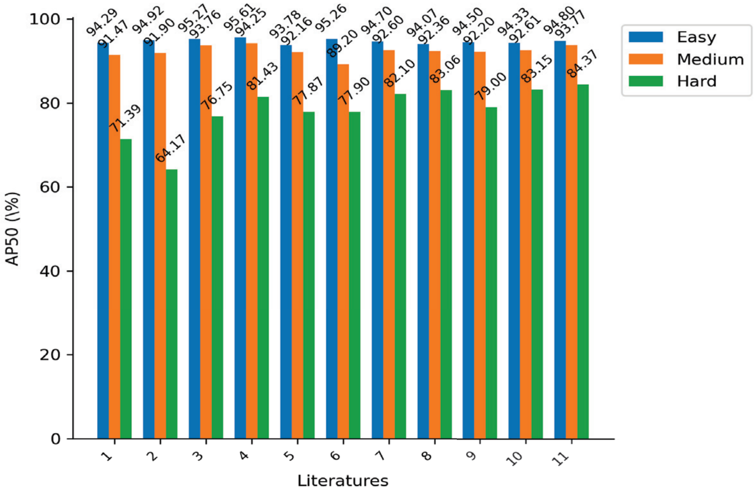


Figure 7. Detection results on the Wider Face validation dataset. Literature 1: DSFD (2019) [34], 2: RetinaFace (2020) [6], 3: HAMBox (2020) [41], 4: TinaFace (2020) [9], 5: SCRFD-2.5GF (2021) [29], 6: TinyYolov3 (2022) [42], 7:YOLOv7-tiny-Face (2022) [7], 8: YFaces-Tiny (2024) [43], 9: YOLOv8n-face (2023) [7], 10: YOLOv5s-Face (2021) [7], 11: ADYOLOv5-Face (ours).

4.4.2. Experiments on XD-Face

To comprehensively assess the ADYOLOv5-Face model's capability in detecting small target faces, we conducted evaluation experiments on the proposed XD-face dataset. YOLO-based face detection methods were chosen as comparison benchmarks, and YOLO series evaluation metrics (precision, recall, AP50 and AP50:5:95) were utilized to verify detection accuracy. The data presented in Table 5 were obtained by testing the trained models available on the official websites of these face detectors. As depicted in Table 5, the proposed ADYOLOv5-Face model exhibits superior performance in terms of precision and AP, whereas the YOLOv5s-Face model performs better in the recall metric. Figure 8 also provides a clearer illustration of the evaluation results on the AP metric.

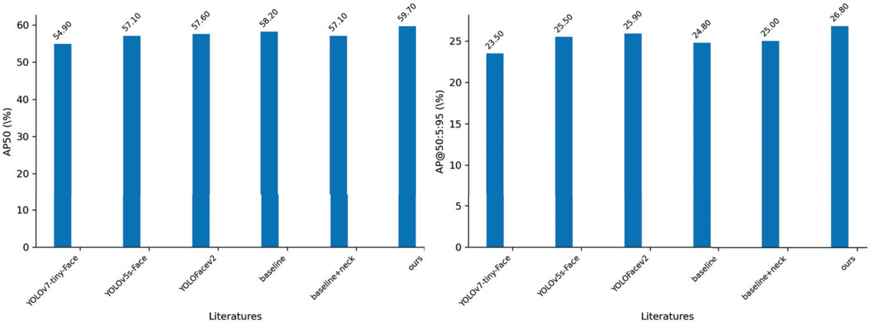


Figure 8. Detection results of the AP50 metric (**left**) and the AP@50:5:95 metric (**right**) on the XD-face dataset.

Given that the evaluation metric of AP50 (where IOU for the NMS is set to 0.5) is commonly used in object detection, we primarily analyze the performance of these face detectors based on AP50. Compared to the baseline method, the proposed model's performance increased by 1.5%, achieving a score of 59.7%. Furthermore, when compared to the other two YOLOv5-based face detectors (YOLOv5s-Face and YOLOfacev2), our proposed model demonstrates improvements of 2.6% and 2.1%, respectively. Additionally, our model outperforms the YOLOv7-tiny-based face detector by 4.8% in terms of the AP@50 metric.

Table 5. Comparison of our ADYOLOv5-Face with the YOLO-based face detectors on the XD-Face validation dataset under four metrics: precision, recall, AP50 and AP50:5:95.

Detector	Precision (%)	Recall (%)	AP50 (%)	AP@50:5:95 (%)
YOLOv7-tiny-Face	63.9	63.7	54.9	23.5
YOLOv5s-Face	46.8	74.6	57.1	25.5
YOLOFacev2	65.5	67.8	57.6	25.9
baseline	65.5	66.5	58.2	24.8
baseline+neck	65.7	66.7	57.1	25.0
ADYOLOv5-Face (ours)	66.5	70.2	59.7	26.8

In conclusion, when dealing with extremely small and occluded faces, our proposed model achieves superior detection accuracy compared to other state-of-the-art face detectors.

5. Conclusions

In this paper, we introduce ADYOLOv5-Face, which is based on the YOLOv5 object detector, and present a novel dataset (XD-face) for face detection tasks. We propose two variations of the YOLO structure, both of which demonstrate significant enhancements to face detection performance. Through evaluation experiments conducted on the Wider Face and XD-face datasets, we demonstrate that ADYOLOv5-Face can effectively compete with other state-of-the-art face detectors while utilizing lightweight base models. Our results indicate that ADYOLOv5-Face either closely matches or even surpasses existing face detectors on both the Wider Face and XD-face benchmarks. These findings underscore the effectiveness of the advanced YOLOv5 architecture in achieving state-of-the-art performance, particularly in the detection of small faces.

The proposed method shows significant effectiveness in handling dense facial data and has potential applications in face counting, face recognition, and domain adaptation. In terms of face counting, it can more accurately identify and count the number of students in a classroom, contributing to the development of intelligent education. For face recognition, improving face detection accuracy serves as a fundamental step towards reliable facial identification. In domain adaptation, the algorithm can self-adjust to different classroom environments, enhancing recognition performance under varying lighting conditions and angles. Overall, the proposed algorithm provides a solid foundation for the application of facial recognition technologies in the education field, facilitating improvements in teaching efficiency and student management.

Author Contributions: Writing—original draft, L.L.; Visualization, G.W.; Funding acquisition, Q.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was jointly funded by the New Teacher Innovation Fund of Xidian University under Grant No. XJSJ23035, and the 2024 Higher Education Scientific Research Planning Project of the Chinese Association of Higher Education under Grant No. 241jpfhZD01.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

YOLO	You Only Look Once
CNN	Convolutional Neural Network
RFE	Receptive Field Enhancement module
FPN	Feature Pyramid Network
GD mechanism	Gather-and-Distribute mechanism
GHM	Gradient Harmonizing Mechanism
SWF	Slide Weight Function (SWF)
NWD	Normalized Wasserstein Distance
IoU	Intersection over Union
RPN	Region Proposal Network
FAM	feature alignment module
IFM	information fusion module
IIM	information injection module
Low-GD	low-stage Gather-and-Distribute mechanism (Low-GD)
AvgPool	average pooling operation
High-GD	high-stage gather-and-distribute mechanism
AP	Average Precision
PR	Precision Recall
TP	True Positive
FN	False Negative
FP	False Positive
TN	True Negative

References

1. Ren, S.; He, K.; Girshick, R.B.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
2. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2961–2969.
3. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In *Computer Vision—ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Proceedings, Part I 14; Springer: Cham, Switzerland, 2016; pp. 21–37.
4. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
5. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
6. Deng, J.; Guo, J.; Ververas, E.; Kotsia, I.; Zafeiriou, S. Retinaface: Single-shot multi-level face localisation in the wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 5203–5212.
7. Qi, D.; Tan, W.; Yao, Q.; Liu, J. YOLO5Face: Why reinventing a face detector. In *Computer Vision—ECCV 2022 Workshops*; Springer: Cham, Switzerland, 2022; pp. 228–244.
8. Zhang, S.; Chi, C.; Lei, Z.; Li, S.Z. Refineface: Refinement neural network for high performance face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 4008–4020. [CrossRef] [PubMed]
9. Zhu, Y.; Cai, H.; Zhang, S.; Wang, C.; Xiong, Y. Tinaface: Strong but simple baseline for face detection. *arXiv* **2020**, arXiv:2011.13183.
10. Zhang, S.; Zhu, X.; Lei, Z.; Shi, H.; Wang, X.; Li, S.Z. Faceboxes: A CPU real-time face detector with high accuracy. In Proceedings of the 2017 IEEE International Joint Conference on Biometrics (IJCB), Denver, CO, USA, 1–4 October 2017; pp. 1–9.
11. Ju, L.; Kittler, J.; Rana, M.A.; Yang, W.; Feng, Z. Keep an eye on faces: Robust face detection with heatmap-Assisted spatial attention and scale-Aware layer attention. *Pattern Recognit.* **2023**, *140*, 109553. [CrossRef]
12. Wang, C.; He, W.; Nie, Y.; Guo, J.; Liu, C.; Wang, Y.; Han, K. Gold-YOLO: Efficient Object Detector via Gather-and-Distribute Mechanism. In Proceedings of the 37th International Conference on Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023.
13. Chi, C.; Zhang, S.; Xing, J.; Lei, Z.; Li, S.Z.; Zou, X. Selective refinement network for high performance face detection. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8231–8238.
14. Chen, M.; Ren, X.; Yan, Z. Real-time indoor object detection based on deep learning and gradient harmonizing mechanism. In Proceedings of the 2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS), Liuzhou, China, 20–22 November 2020; pp. 772–777.

15. Cao, Y.; Chen, K.; Loy, C.C.; Lin, D. Prime sample attention in object detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11583–11591.
16. Yu, Z.; Huang, H.; Chen, W.; Su, Y.; Liu, Y.; Wang, X. Yolo-facev2: A scale and occlusion aware face detector. *Pattern Recognit.* **2024**, *155*, 110714. [CrossRef]
17. Jiang, H.; Learned-Miller, E. Face detection with the faster R-CNN. In Proceedings of the 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), Washington, DC, USA, 30 May–3 June 2017; pp. 650–657.
18. Sun, X.; Wu, P.; Hoi, S.C. Face detection using deep learning: An improved faster RCNN approach. *Neurocomputing* **2018**, *299*, 42–50. [CrossRef]
19. Zhu, C.; Zheng, Y.; Luu, K.; Savvides, M. CMS-RCNN: Contextual multi-scale region-based cnn for unconstrained face detection. In *Deep Learning for Biometrics*; Springer: Cham, Switzerland, 2017; pp. 57–79.
20. Khan, S.S.; Sengupta, D.; Ghosh, A.; Chaudhuri, A. MTCNN++: A CNN-based face detection algorithm inspired by MTCNN. *Vis. Comput.* **2024**, *40*, 899–917. [CrossRef]
21. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
23. Huang, G.; Liu, Z.; Pleiss, G.; Van Der Maaten, L.; Weinberger, K.Q. Convolutional networks with dense connectivity. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *44*, 8704–8716. [CrossRef] [PubMed]
24. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
25. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
26. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
27. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
28. Liu, Y.; Tang, X. Bfbox: Searching face-appropriate backbone and feature pyramid network for face detector. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 13568–13577.
29. Guo, J.; Deng, J.; Lattas, A.; Zafeiriou, S. Sample and computation redistribution for efficient face detection. *arXiv* **2021**, arXiv:2105.04714.
30. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
31. Zhang, J.; Wu, X.; Hoi, S.C.; Zhu, J. Feature agglomeration networks for single stage face detection. *Neurocomputing* **2020**, *380*, 180–189. [CrossRef]
32. Najibi, M.; Samangouei, P.; Chellappa, R.; Davis, L.S. SSH: Single Stage Headless Face Detector. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
33. Wang, W.; Wang, X.; Yang, W.; Liu, J. Unsupervised face detection in the dark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 1250–1266. [CrossRef] [PubMed]
34. Li, J.; Wang, Y.; Wang, C.; Tai, Y.; Qian, J.; Yang, J.; Wang, C.; Li, J.; Huang, F. DSFD: Dual Shot Face Detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
35. Zhang, Z.; Shen, W.; Qiao, S.; Wang, Y.; Wang, B.; Yuille, A. Robust face detection via learning small faces on hard images. In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass, CO, USA, 1–5 March 2020; pp. 1361–1370.
36. Fang, Z.; Ren, J.; Marshall, S.; Zhao, H.; Wang, Z.; Huang, K.; Xiao, B. Triple loss for hard face detection. *Neurocomputing* **2020**, *398*, 20–30. [CrossRef]
37. Wu, S.; Li, X.; Wang, X. IoU-aware single-stage object detector for accurate localization. *Image Vis. Comput.* **2020**, *97*, 103911. [CrossRef]
38. Wang, J.; Xu, C.; Yang, W.; Yu, L. A normalized Gaussian Wasserstein distance for tiny object detection. *arXiv* **2021**, arXiv:2110.13389.
39. Everingham, M.; Gool, L.V.; Williams, C.K.I.; Winn, J.M.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
40. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
41. Liu, Y.; Tang, X.; Han, J.; Liu, J.; Rui, D.; Wu, X. Hambox: Delving into mining high-quality anchors on face detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 13043–13051.

42. Gao, J.; Yang, T. Face detection algorithm based on improved TinyYOLOv3 and attention mechanism. *Comput. Commun.* **2022**, *181*, 329–337. [CrossRef]
43. Sufian Chan, A.A.; Abdullah, M.; Mustam, S.M.; Poad, F.A.; Joret, A. Face Detection with YOLOv7: A Comparative Study of YOLO-Based Face Detection Models. In Proceedings of the 2024 International Conference on Green Energy, Computing and Sustainable Technology (GECOST), Miri Sarawak, Malaysia, 17–19 January 2024; pp. 105–109.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

Article

A Hybrid Approach to Modeling Heart Rate Response for Personalized Fitness Recommendations Using Wearable Data

Hyston Kayange, Jonghyeok Mun, Yohan Park, Jongsun Choi * and Jaeyoung Choi

School of Computer Science and Engineering, Soongsil University, Seoul 05978, Republic of Korea; hyston@soongsil.ac.kr (H.K.); jonghyeokmun@soongsil.ac.kr (J.M.); imjin3027@soongsil.ac.kr (Y.P.); choi@ssu.ac.kr (J.C.)

* Correspondence: jongsun.choi@ssu.ac.kr

Abstract: Heart rate (HR) is a key indicator of fitness and cardiovascular health, and accurate HR monitoring and prediction are essential for enhancing personalized fitness experiences. The rise of wearable technology has significantly improved the ability to track personal health, including HR metrics. Accurate modeling of HR response during workouts is crucial for providing effective fitness recommendations, which help users achieve their goals while maintaining safe workout intensities. Although several HR monitoring and prediction models have been developed for personalized fitness recommendations, many remain impractical for real-world applications, and the domain of personalization in fitness applications still lacks sufficient research and innovation. This paper presents a hybrid approach to modeling HR response to workout intensity for personalized fitness recommendations. The proposed approach integrates a physiological model using Dynamic Bayesian Networks (DBNs) to capture heart rate dynamics during workout sessions. DBNs, combined with Long Short-Term Memory (LSTM) networks, model the evolution of HR over time based on workout intensity and individual fitness characteristics. The DBN parameters are dynamically derived from flexible neural networks that account for each user's personalized health state, enabling the prediction of a full HR profile for each workout, while incorporating factors such as workout history and environmental factors. An adaptive feature selection module further enhances the model's performance by focusing on relevant data and ensuring responsiveness to new data. We validated the proposed approach on the FitRec dataset, and experimental results show that our model can accurately predict HR responses to workout intensity in future sessions, achieving an average mean absolute error of 5.2 BPM per workout—significantly improving upon existing models. In addition to HR prediction, the model provides real-time fitness personalized recommendations based on individual's observed workout intensity to an exercise. These findings demonstrate the model's effectiveness in delivering precise, user personalized heart response to exercise with potential applications in fitness apps for personalized training and health monitoring.

Keywords: personalization; fitness recommendations; dynamic Bayesian networks; wearable data; heart rate

Citation: Kayange, H.; Mun, J.; Park, Y.; Choi, J.; Choi, J. A Hybrid Approach to Modeling Heart Rate Response for Personalized Fitness Recommendations Using Wearable Data. *Electronics* **2024**, *13*, 3888. https://doi.org/10.3390/electronics13193888

Academic Editors: Chao Zhang, Wentao Li, Huiyan Zhang and Tao Zhan

Received: 3 September 2024
Revised: 29 September 2024
Accepted: 29 September 2024
Published: 30 September 2024

1. Introduction

It is well understood that physical activity provides major health benefits for the heart, body, and mind, playing a vital role in preventing and treating many chronic diseases, such as heart disease and diabetes [1]. This is why the World Health Organization (WHO), and other health organizations emphasize the importance of daily physical activity and have issued public health guidelines to prevent and manage chronic illnesses associated with physical inactivity [2].

The rapid growth and popularity of wearable technologies have significantly impacted how individuals engage with their health practices. Wearable devices and mobile health (mHealth) apps have empowered users to monitor and track their physiological data and daily exercise activities [3]. These technologies not only allow users to track their physical fitness but also serve as motivational tools, helping individuals tailor their fitness plans toward achieving results and fostering safer training environments. For instance, previous research has shown that data from wearables are a data-rich resource, which can provide valuable insights into an individual's overall health—crucial for developing personalized fitness recommendation systems [4–6].

Personalization plays an essential role in many modern applications, from enhancing user engagement to improving overall satisfaction. However, modeling an individual's fitness in fitness tracking apps for personalization presents a significant challenge, as user behavior and workout preferences can vary and evolve over time. Factors such as differences in fitness levels, workout plans, exercise intensity, heart rate zones, and environmental conditions highlight the need for robust modeling techniques that capture dynamic changes in user features and account for substantial individual variation.

Ensuring accuracy and relevance in personalized fitness applications is challenging. While personalization has been a core focus for fitness recommendations, and some studies have shown notable improvements, a significant gap remains in most algorithms' ability to adapt to dynamic changes in user data over time, which limits their practicality in real-world recommendation systems. Furthermore, transforming noisy wearable data into transparent and actionable insights for personalization adds another layer of complexity to this task [4].

Additionally, wearable data often lack critical implicit user behavior data, such as clicks, likes, and views, which are essential for understanding behavior patterns making recommendations. This gap presents another challenge in building effective personalized fitness models. Despite these challenges, personalization remains vital for effective fitness recommendations. Heart rate is a fundamental physiological feature that reflects an individual's response to workout intensity, serving as an indicator of both exercise intensity and overall health in daily workout routines [7]. Accurately monitoring and predicting heart rate response during exercise are crucial steps for developing effective, personalized recommendations algorithms. Wearables offer an innovative approach to monitoring heart rate and other significant metrics during various fitness activities, providing valuable data to understand individual health [8]. However, interpreting and leveraging these data for personalized fitness recommendations present several challenges:

1. Individual Variability: The response of heart rate to exercise varies greatly among individuals based on age, current fitness levels, genetic factors, and other factors including overall health conditions [9,10].
2. Temporal Dynamics: An individual's heart rate response may vary depending on their fitness level over time, so models must capture and adapt to these temporal changes [11,12].
3. Environmental Factors: External conditions like temperature, humidity, and altitude can affect the heart's response to exercise, making accurate prediction and personalization more challenging [4].
4. Data Quality: Despite advancements in wearable technology, data from these devices can still be noisy or inconsistent, especially during high-intensity activities or when the device fit is suboptimal.
5. Contextual Information: The heart rate metric alone may not provide a complete understanding of an exercise session. Integrating other contextual data, such as perceived exertion, workout type, and recovery status, is crucial for building a comprehensive fitness model [5].

To address these challenges and improve the accuracy of personalized fitness recommendations, we proposed a scalable algorithm, physiological Hybrid DBN model, which combines a physiological model with flexible machine learning techniques to learn an individual's personalized, multidimensional representation of fitness. The model describes heart rate dynamics during exercise using Dynamic Bayesian Networks (DBNs). We derived DBN parameters using neural networks that connect a personalized representation based on the user's workout history and immediate workout intensity, augmented with additional features through the use of LSTM networks. Our approach integrates wearable data measurements such as speed, elevation gain, distance covered, speed variability, and time elapsed to enhance fitness recommendation algorithms in health applications. This enables the delivery of personalized recommendations tailored to individual physiological responses to varying workout intensities.

Related literature explores the development of physiological models combined with machine learning techniques and other inference methods [4,13,14] for heart rate prediction. Our algorithm monitors heart rate evolution and estimates user-specific parameters over time, learning to map an individual's recent workout history to a personalized representation that predict heart rate response over time. We employed DBNs to address the limitations of existing algorithms in making long-term predictions, as DBNs are well suited for capturing the temporal relationships in how heart rate evolves over time in response to exercise intensity. Additionally, DBNs handle the inherited uncertainty in prediction by considering the probabilistic nature of HR changes alongside additional factors, such as environmental conditions and other external influences.

The Hybrid DBN model is further enhanced to predict personalized heart rates by adjusting user-specific parameters over time, allowing the model to reflect individual's unique physiological responses. We incorporated an adaptive feature selection module that tunes feature weights based on their relevance, considering that in real-world scenarios, user behavior and experiences vary and change. This ensures that the model remains responsive and adaptive in user behavior and exercise patterns.

The remainder of the paper is structured as follows: Section 2 provides a literature review of existing work related to physiological modeling, heart rate prediction, and fitness personalization. Section 3 presents materials and methods. Section 4 details our methodology, including the design of our model architecture. Section 5 presents the results of our experiments. Section 6 discusses the implications of our model for personalized fitness recommendations. Finally, Section 7 concludes the paper.

2. Related Work

This section reviews the existing literature on wearable data technologies, heart rate prediction models, the integration of physiological models with machine learning, the use of Dynamic Bayesian Networks (DBNs), and the development of recommender systems for personalized fitness recommendations:

Wearable Data Technologies: Wearable devices, such as fitness trackers and smartwatches, have empowered individuals to continuously monitor their health metrics, providing a rich data source that can be leveraged in machine learning models to uncover causal relationships between human health and device signals [15,16]. Ferguson [15] highlights how data mining from wearables can play a crucial role in personalized health interventions. Numerous studies have shown the value of wearable data in predicting health outcomes, ranging from clinical monitoring tools to fitness and activity planners [17–19]. These applications include machine learning models for cardiovascular fitness and disease surveillance [20–22]. Nazaret [4] discusses the utilization of wearable technology in monitoring heart rate (HR) and other physiological signals to model personalized heart rate responses to exercise and environmental factors. The study uses data from wearable devices, specifically Apple Watches, to capture detailed workout information such as step count, speed, elevation change, and environmental factors like temperature and humidity. These data are collected in real-time during various activities, allowing for a comprehensive analysis of an individual's HR dynamics in uncontrolled, real-world environments. Jiamo [5] discusses the utilizing wearable devices, such as smartwatches and fitness trackers, to collect extensive data for modeling heart rate and activity patterns. These devices continuously capture various types of data types, including heart rate, GPS location, altitude, and other contextual information like weather and user demographics [5]. This data can be used to create personalized fitness and exercise behavior models, specifically focusing on heart rate dynamics and prediction.

Heart Rate Prediction Models: Heart rate is a critical physiological parameter used to assess cardiovascular health, exercise intensity, and other physiological states. Several models have been developed to predict heart rate, ranging from simple regression-based models to more scalable algorithms. Oyeleye [23] conducted a review of heart rate prediction using various machine learning techniques, including Autoregressive Integrated Moving Average (ARIMA) and linear regression, as well as Long Short-Term Memory (LSTM) models. Their results highlighted the potential of heart rate prediction in various areas of health monitoring. Kuano Tao [24] utilized linear regression and artificial intelligence networks to estimate heart rate and assess cardiovascular fitness in middle-aged adults. Hybrid approaches have also been developed; for instance, Lin [25] proposed a new method for heart rate prediction using LSTM-BILSTM-Att, which improved accuracy compared to baseline model. Zetao [7] introduced a study using LSTM to predict heart rate during different activities as part of a fitness training optimization system. Additionally, scalable algorithms for personalized heart rate response to exercise and environmental factors using wearables data have been developed [4] with the aim is to monitor heart rate evolution in response to workout intensity.

Integration of Physiological Models with machine learning techniques: Numerous studies have focused on integrating physiological models with machine learning techniques, particularly on heart rate parameters for various health outcomes. These models aim to overcome the limitations of traditional methods and standalone machine learning models, offering more accurate heart rate predictions. Recent research has explored combining physiological models with flexible neural networks to leverage both physiological understanding and data-driven models, enhancing prediction accuracy and adaptability in personalized interventions. For example, Nazaret [4] developed a physiological model using Ordinary Differential Equations (ODE) combined with flexible neural networks to monitor heart rate evolution during workout sessions, enabling personalized heart rate profiling. Signorini [26] integrated machine learning techniques and physiological heart rate features for antepartum fetal monitoring, helping identify potential pathologies early in pregnancy management. Another model, proposed by [27], predicts fitness levels by focusing on physiological features such as heart rate, step count data, and total oxygen consumption. Milan [28] emphasized the importance of dynamic models that capture physiological parameters, such as heart rate, in response to exercise, aiming to better understand the relationship between physiological responses and exercise.

Dynamic Bayesian Networks (DBNs): Dynamic Bayesian Networks have emerged as powerful tools for modeling time-series data due to their ability to capture temporal dependencies and handle uncertainty in dynamic environments. DBNs are particularly well suited for modeling physiological data, such as heart rate (HR), where responses can be highly variable and influenced by numerous factors. While there are not many studies specifically applying DBNs to heart rate modeling, DBNs have shown their strength in healthcare. For instance, Fernando [29] proposed a framework that uses Bayesian networks to describe heart rate dynamics and estimate the autoregulation of the autonomic nervous system. Zhang [30] introduced a heart prediction model using Bayesian networks to overcome the limitations of short-term models, enabling multiple predictions to enhance heart rate prediction accuracy during running activities. Ladyzynski [31] demonstrated DBNs' capabilities in health monitoring and treatment of chronic lymphocytic leukemia, validating the use of DBNs in predicting health changes over time. Qi chen [32] applied DBNs to model complicated relationships among physiological variables across time slices for predicting physiological changes, organ dysfunction and mortality risk in critical trauma patients. Their results indicated that DBNs can serve as effective real-time tools for predict physiological changes. Marshall [33] introduced a dynamic Bayesian network to investigate the relationship between patient variables, cardiovascular disease and survival rates in heart disease patients. To our knowledge, limited research has applied DBNs as physiological inference methods to model heart rate in fitness environment for personalized fitness recommendations. In our work, we demonstrated the potential of DBNs in physiological modeling for fitness personalization.

Recurrent Neural Networks (RNNs): RNNs, particularly Long Short-Term Memory (LSTM) networks, have been widely used in recent years to analyze time-series data for sequential modeling tasks, such as speech recognition and machine translation [34]. In heart rate prediction, Oyeleye [22] investigated advanced deep learning models, including LSTM, for time-series modeling to overcome the limitations of traditional models such as ARIMA. Their work demonstrated improved results across multiple metrics compared to baseline models. Their work demonstrated improved results across multiple metrics compared to baseline models. Lin [25] proposed a method for predicting heart rate using LSTM-BILSTM-Att technique. Additionally, Jiamo [5] proposed FitRec, which utilizes LSTM networks to model heart rate by estimating heart rate profiles during ongoing exercise. Similarly, Xia Liu [35] proposed a multi-level deep learning approach using LSTM for heart rate prediction, which was later used to provide personalized recommendations.

Adaptive Feature Selection: In the realm of personalized recommender systems, Lin [35] highlights the importance of feature selection, shedding valuable insights on how each variable influences recommendation tasks. This analysis not only indicates the transparency of recommendation systems but also provides a deeper understanding of users' preferences. Building on this, AdaFS [36] introduces an adaptive feature selection framework designed to optimize recommendation performance by selecting the most predictive features. AdaFS employs a novel controller network that automatically adapts to the dynamic nature of recommendation environments, ensuring that relevant features are selected as the system evolves. Similarly, Kayange [37] proposed the ProAdaFS framework, a probabilistic and adaptive feature selection approach for recommendation systems. This framework accounts for the fact that features in user-item interactions may vary over time, continually updating recommendation models to adapt to changes in user behavior and new data. We applied a similar concept in our physiological hybrid DBN model, recognizing that user behavior and preferences in practical recommendation systems fluctuate over time. This adaptive module dynamically enhances our model by incorporating evolving user exercise patterns, thereby improving performance.

Personalized Fitness recommendations: Several studies have focused on personalized fitness recommendations, utilizing approaches ranging from recurrent neural networks to physiological models and machine learning techniques. Jiamo [5] proposed a personalized fitness recommendation system FitRec, which models sequential fitness data, such as heart rate sequences, during ongoing workout sessions. Later Xia Liu [35] introduced a multi-level deep learning approach for fitness recommendations, focusing on personalizing workout distance and speed sequences in response to user activity. Additionally, as mentioned earlier, Nazaret [4] proposed a physiological model to personalize heart response to a workout intensity. Their model can also be applied to personal activity planning and predicting future workouts.

Despite the progress in wearable technology and fitness personalization, there remains a research gap in ensuring the accuracy and relevance of personalized fitness recommendations. The field lacks sufficient innovation and comprehensive research. From our review of existing literature, we found that most models focus on short-term predictions, which makes it difficult to maintain accuracy in long-term fitness environments. Additionally, many models are static and fail to adapt to the dynamic nature of user data over time. Utilizing fitness data for recommendations is also challenging, as these data often lack behavioral indicators like clicks or likes, which are crucial in understanding user behavior in traditional recommendation models. To address these gaps, we proposed a hybrid physiological model using Dynamic Bayesian Networks (DBNs) to model the evolution of heart rate over time in response to exercise. This approach also incorporates LSTM networks and an adaptive feature selection module, allowing the system to dynamically adapt to new data. Our proposed method aims to overcome the limitations of existing models and provide more accurate, personalized fitness recommendations by combining physiological modeling with machine learning techniques.

3. Materials and Methods

3.1. Dataset

We used the FitRec dataset from [5], which is collected from Endomondo fitness tracking applications and is a collection of workout data from wearables. The dataset provides rich insights into various physical activities performed by users, including different types of sports workouts such as running, biking, and other fitness routines, along with physiological data like heart rate. The dataset is sourced as a JSON format containing multiple entries for different individual's workout sessions. Each entry contains timestamps, duration, distance covered, heart rate, speed, and GPS coordinates, making it an ideal dataset for studies focused on user behavior and predicting workout sessions for enhanced personalized fitness recommendations. The key variables used in this work, including heart rate, speed, distance, altitude and other derived variables are summarized in Table 1. For our study, we filtered the dataset to include only running workout sessions, with additional contextual information such as user ID, workout ID, and gender. A detailed summary of the dataset statistics, including the number of users and total workouts, is provided in Table 2. The JSON entries were parsed and converted into a Pandas DataFrame for efficient data manipulation and analysis.

Table 1. Description of workout data variables.

Variable	Unit
Measurement	
Heart rate	Beats per Minute (BPM)
Time Grid	Seconds
Distance	Meters
Speed (Horizontal)	Meters per Second (m/s)
Speed (Vertical)	Meters per Second (m/s)
Contextual	
UserID	
WorkoutID	
Sport	
Gender	Male (1), Female (0)
Altitude	Meters (m)
Longitude	Degrees
Latitude	Degrees
Derived	
Elevation Gain	Meters (m)
Average Speed	Kilometers per Hour (KMPH)
Speed Variability	Meters per Second (m/s)
Max Heart Rate	Beats per Minute (BPM)
Total Distance	Meters (m)

Table 2. Statics of the FitRec dataset on running sport.

Statistics	Value
Total number of users	665 (590 male, 65 female)
Total number of workouts	38,323
Average workout speed	3.7 km/h
Workout frequency per User	68
Average workout duration	48.54 Min

3.2. Data Processing

3.2.1. Temporal Data Alignment

We converted the timestamps into Python datetime objects to easily calculate the duration of each individual's workout session, from start to finish. We retained workouts with durations between 10 min and 2 h 20 min, removing sessions that were either too short or excessively long. This filtering step ensures that our results reflect typical real-world workout sessions.

3.2.2. Interpolation and Grid Alignment

We applied a uniform grid at a 10-s interval for each workout to standardize the temporal resolution, resulting in a heart rate time series, $\hat{HR} \in \mathbb{R}^d$ [4] and workout intensity time series, $I \in \mathbb{R}^d$, for each session. GPS coordinates, altitude, and heart rate were interpolated to ensure it aligns with the applied time grid. This approach minimizes the issue arising from irregular sampling intervals, ensuring a continuous representation of each workout's dynamics.

3.2.3. Feature Engineering

To process the distance covered, we used the Haversine formula [37], which calculates the distance between two data points based on their latitude and longitude. For speed, we derived horizontal and vertical speeds from the interpolated distance and altitude, respectively. Additionally, we engineered features such as elevation gain, average speed, speed variability, maximum heart rate, and gender encodings to enrich the model inputs for better performance. We also applied normalization and standardization, particularly for heart rate, to ensure that the input data are consistent for model training, enhancing both the model's performance and stability. Specifically, heart rate values were normalized by subtracting the mean heart rate and dividing by the average standard deviation, thereby centering the data and scaling them for uniformity across users.

3.2.4. Personalized Heart Rate Model

Following the approach from [4], we developed a personalized heart rate model by assigning each individual personalized parameters (A, B, including a drive function f and $HR_{\min}$, $HR_{\max}$), These parameters help capture the dynamic response to exercise. As mentioned by [4], learning each individual's parameter health representation is computationally expensive. To address this, we assumed that a low-dimensional latent vector $z \in \mathbb{R}^{\boldsymbol{di}}$ can represent an individual's health status at a given time, where di is the dimensionality of the latent space.

4. Proposed Model

Our physiological DBN model integrates LSTM and flexible neural networks to predict personalized heart rate responses to exercise intensity during workout sessions. The architecture of our model consists of several key components, as illustrated in Figure 1.

4.1. Dynamic Bayesian Network

Dynamic Bayesian Networks (DBNs) are an extension of Bayesian networks, designed to handle time-dependent data and capture the relationships between different features over time. This makes DBNs well suited for modeling sequences of variables where temporal dependencies are key. Moreover, DBNs are particularly effective in modeling time-series data due to their ability to capture both temporal dynamics and probabilistic relationships among variables. In this research, we applied DBNs to model heart rate changes in response to varying exercise intensities over time. Figure 2 provides a graphical representation of DBNs, where z_t represents the individual health status at time t and E_t represent the observed heart rate. DBNs allow the model to capture temporal correlations and non-linear relationships in heart rate variations, which is essential given the noisy nature of wearable data. Additionally, DBNs enable the model to adapt its predictions based on changes in user behavior. Our model development focused on two core components of DBNs.

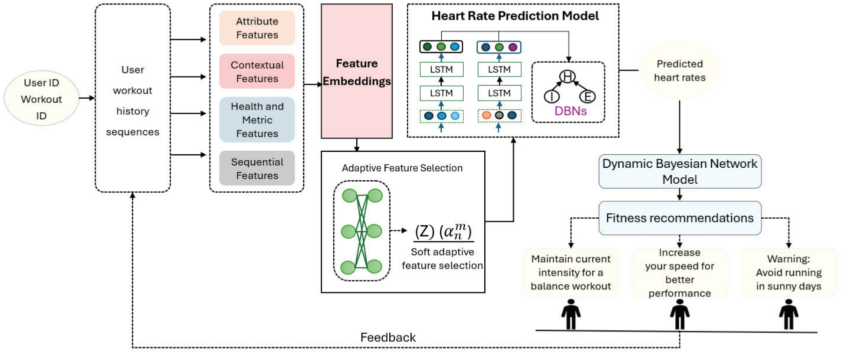

User ID
Workout
ID
User
workout
history
sequences
Attribute Features
Contextual Features
Health and Metric Features
Sequential Features
Feature Embeddings
Heart Rate Prediction Model
LSTM
LSTM
LSTM
LSTM
H
I
E
DBNs
Adaptive Feature Selection
(Z) (α_n^m)
Soft adaptive feature selection
Predicted heart rates
Dynamic Bayesian Network Model
Fitness recommendations
Maintain current intensity for a balance workout
Increase your speed for better performance
Warning: Avoid running in sunny days
Feedback

Figure 1. Model architecture of the proposed approach for personalized heart rate prediction to enhance fitness recommendations.

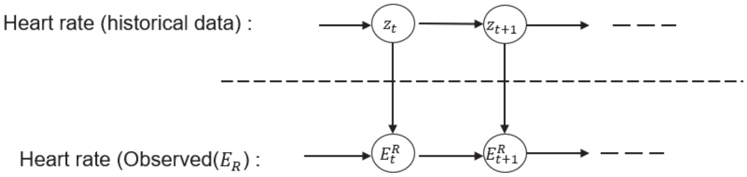


Figure 2. A graphical representation of DBNs where z_t is an individual health status at time t and E is the observed heart rate.

State Transition: The state transition probability describes how the system evolves from one state to another over time. In the context of heart rate prediction, it captures how heart rate changes in response to variations in exercise intensity and duration. Figure 3 illustrates a graphical representation of our DBN model for heart rate prediction based on exercise workout intensity, where y_t represents variables influencing heart rate and x_t represents heart rate frequencies at time t. In our study, the state transition is modeled using the following equation:

$$P(x_t \mid x_{t-1}, S_t) = (x_t \mid f_{\text{trans}}(x_{t-1}, S_t), \sigma^2), \tag{1}$$

where x_t is the heart rate at time step t, x_{t-1} is the heart rate at the previous time step $t-1$, and S_t is the state of influencing factors at time t, such as exercise intensity, speed, distance, elevation gain, and environmental conditions like temperature. The function f_{trans} (x_{t-1}, S_t) is a transition function that models the expected heart rate at time t, based on the previous heart rate x_{t-1} and the influencing factors, and S_t. σ^2 represents the variance of the transition noise, indicating uncertainty in the prediction. We assume that x_t follows a normal distribution with mean μ = f_{trans} (x_{t-1}, S_t), and variance σ^2.

Equation (1) expresses that the heart rate at the current time step is normally distributed around a mean value f_{trans} (x_{t-1}, S_t), which is a function of the previous heart rate and the current influencing factors. The noise term σ^2 accounts for uncertainties in heart rate transition. Incorporating state transition in our model captures temporal dependencies in heart rate dynamics, where the heart rate depends on both the previous state and contextual factors. This approach is crucial for monitoring user behavior changes over time.

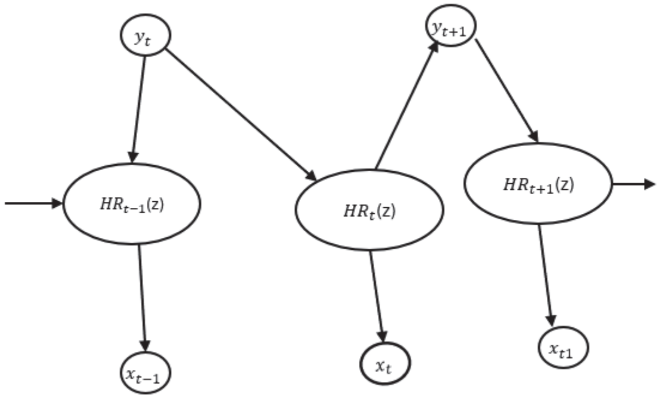


Figure 3. Graphical representation of the DBN model for heart rate prediction HR_t based on exercise workout intensity, y_t are variables influencing HR at state t and x_t are heart rate frequencies at time t.

Emission Modeling: The emission equation models how the latent physiological state relates to the observed heart rate at each time step. The emission equation is modeled as follows:

$$Y_t = f_{\text{em}}(X_t) + \epsilon_t,\ \epsilon_t \sim N(0, \sigma^2), \tag{2}$$

where Y_t is the observed heart rate at time t, X_t is the latent state at time t. The function $f_{em}(X_t)$ is the function that estimates the observed heart rate from the latent state, and ϵ_t represents prediction error, which follows a normal distribution with mean 0 and variance σ^2. The function $f_{em}(X_t)$ maps the latent physiological state to the actual observed heart rate, capturing complex non-linear relationships.

Combining Equations (1) and (2), we can model the evolution of heart rate during workout sessions using our DBN model. The model is formulated as follows:

$$HR_{t+1} = HR_t(z) + f(HR_t(z), S_t(z)) + \varepsilon_t. \tag{3}$$

HR_{t+1} is the predicted heart rate at time t + 1, and $HR_t(z)$ is the current heart rate at time t where z is the individual's health status. The function $f(HR_t(z)$, $S_t(z)$ estimates the change in heart rate based on individual's current health state, considering both the current heart rate $HR_t(z)$ and influencing factors $S_t(z)$, such as exercise intensity, speed, and environmental conditions. Lastly, ε_t is the noise accounting prediction error, representing random fluctuations.

4.2. Long Short-Term Memory (LSTM) Networks

LSTMs are a type of recurrent neural network (RNN) designed to model time-series data and capture long-range dependencies. Unlike traditional RNNs, LSTMs use memory cells to retain information over extended periods, enabling them to learn patterns and relationships in sequences that may not have been visible before. In our model, we used LSTM networks to process users' workout histories and create personalized health representations through flexible neural networks. These representations are critical for informing the parameters of the Dynamic Bayesian Network (DBN) model, enhancing its ability to make accurate personalized predictions about heart rate responses.

We introduced an LSTM encoder that considers historical workout data, including features such as past heart rate data, exercise intensity, duration, speed, elevation gain, and other relevant physiological and contextual information. The LSTM processes this sequence data and learns a latent representation, z, which captures the user's unique physiological characteristics and exercise patterns over time. This latent representation allows the model to personalize its predictions based on the user's specific physiological state and workout history.

The LSTM network consists of a series of gates, as depicted in Figure 4: input, forget, and output gates, which regulate the flow of information through the memory cell. In our model, for a given step t, the LSTM calculates h_t, the output of the LSTM, which is subsequently fed into the DBN model. The h_t is computed as follows:

$$f_t = \sigma\Big(W_f \cdot [h_{t-1}, x_t] + b_f\Big), \tag{4}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{5}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{6}$$

$$\widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \tag{7}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t,\ h_t = o_t \cdot \tanh(C_t), \tag{8}$$

where f_t, i_t, o_t represent the forget gate, input, and output gates, respectively, C_t is the cell state at time step t, and $\widetilde{C}_t$ is the candidate cell state, h_t is the hidden state output at time step t, σ denotes the sigmoid activation function, tanh denotes the hyperbolic tangent activation function, and W and b are weight matrices and bias vectors, respectively.

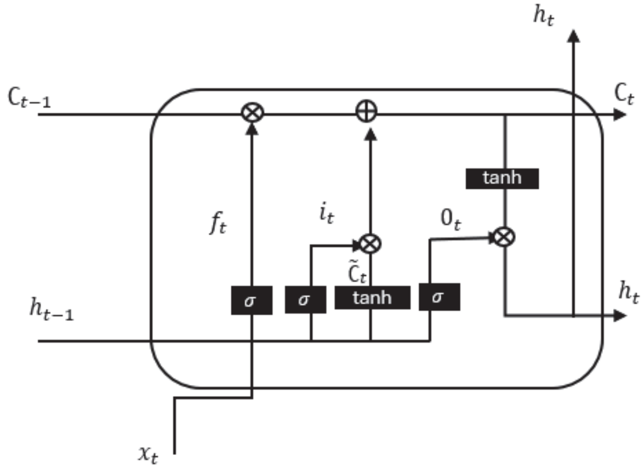


Figure 4. Structure of an LSTM cell.

4.3. Personalized Heart Rate Prediction

To accurately predict heart rate over time t during a workout session, our physiological model uses the following equation:

$$HR(t) = HR_{\min} + (HR_{\max} - HR_{\min}) \cdot \Big(1 - e^{-A(z) - B(z) \cdot I(t)}\Big), \tag{9}$$

where $HR(t)$ is the predicted heart rate at time t, $HR_{\min}$ is the resting heart rate of the individual before a workout session, and $HR_{\max}$ is the maximum heart rate reached during an exercise session. $A(z)$ is the decay constant that controls how quickly the heart rate approaches its maximum, personalized through latent vector z. $B(z)$ is the intensity coefficient that modulates the heart rate response based on exercise intensity I(t), also personalized through latent vector z, which represent the exercise intensity at time t.

This equation captures the dynamic heart rate behavior as it responds to exercise. It combines physiological aspects (resting and maximum heart rate) and exercise-specific factors (intensity and duration) to provide a comprehensive prediction model that adapts dynamically to exercise intensity through the decay function. The term $e^{-A(z)-B(z) \cdot I(t)}$

models the physiological response to exercise, where the rate of change in heart rate is controlled by both the duration and intensity of an exercise, modulated by personalized parameters. This formulation allows the model to account for physiological constraints and dynamic responses to exercise over time. It provides a transparent prediction that adapts to each individual's unique characteristics and workout history, improving the precision of fitness recommendations.

4.4. Adaptive Feature Selection

We employed an adaptive feature selection module as an attention mechanism within the LSTM encoder, which assigns weights to different features based on their relevance at time t. This module illustrated in Figure 5, takes the latent vector z and assigns a weight of α_n^m where n is the n^{th} feature and m is the m^{th} data point in the input latent vector z. To achieve this, we designed a multi-layer perceptron (MLP) network controller that dynamically assigns weights to the latent vector z based on each feature's predictive power. This enables the model to prioritize the most relevant features for heart rate prediction, ensuring more accurate and personalized recommendations.

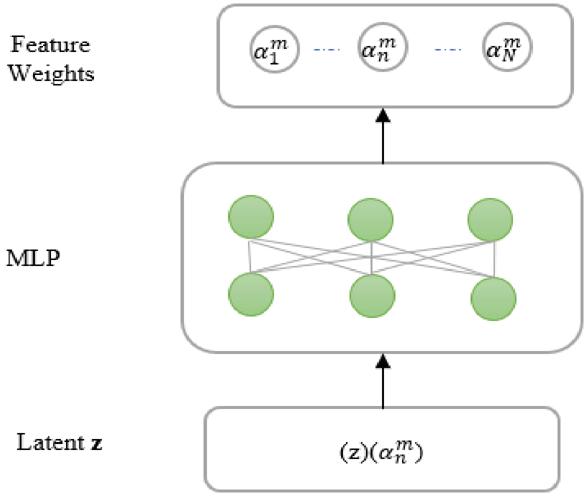


Figure 5. Adaptive feature selection module.

5. Experiments

This section focuses on evaluating the proposed model's performance in predicting heart rate (HR) responses and its effectiveness in providing personalized fitness recommendations. We will outline the experimental setup, describe the evaluation metrics used, and present the results of our experiments, highlighting the model's accuracy and adaptability.

5.1. Experimental Setup

We conducted our experiments using the FitRec dataset, which contains 38,323 logged workout sessions of running workouts across 665 individuals. The dataset includes various workout types such as running, cycling, and walking, alongside wearable data (e.g., heart rate), speed/pace, elevation gain, and GPS coordinates. To focus on the most relevant information, we filtered the dataset to include only workouts lasting between 10 min and 2 h and 20 min. Several additional features were derived from the raw data including speed variability, calculated as the standard deviation of speed, elevation gain computed as the sum of positive altitude changes, normalized heart rate. We created a time grid for each workout session to facilitate interpolation and alignment of features.

For training and validation, the dataset was split into two subsets: 80% for training and 20% for validation, ensuring that workouts from the same user were not split across subsets. This split was necessary to evaluate the model's generalization capability to unseen data.

DBN parameters were configured to model the state transitions and emission probabilities, capturing the dynamics of heart rate response during exercise. The LSTM encoder was set up with two layers, a hidden dimension of 128, and a dropout rate of 0.2 to prevent overfitting. We also utilized flexible neural networks, specifically the PersonalizedScalerNN [4], to dynamically adjust the DBN parameters based on the latent representations generated by the LSTM encoder.

For the training procedure, we used the Adam optimizer with a learning rate of 1×10^{-3}. The model was trained by minimizing the mean squared error (MSE) between the predicted and actual heart rate values. To prevent exploding gradients, gradient clipping was employed

5.2. Evaluation Metrics

To evaluate the performance of our model in both heart rate prediction and personalized fitness recommendation, we used the following metrics:

1. Mean Absolute Error (MAE): This metric measures the average absolute difference between predicted and actual heart rate values. It provides a straightforward measure of prediction accuracy by calculating the magnitude of errors.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{10}$$

2. Root Mean Squared Error [RMSE]: measures the square root of the average squared differences between actual and predicted values. It gives higher weight to larger errors, making it sensitive to large deviations.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{11}$$

where n is the total number of data points, y_i is the actual heart rate data and, $\hat{y}_i$ is the predicted heart rate data.

5.3. Results and Analysis

5.3.1. Heart Rate Prediction Performance

The proposed hybrid DBN model achieved an average Mean Absolute Error (MAE) of 5.2 BPM and a Root Mean Square Error (RMSE) of 8.1 BPM across all validation workouts. These results demonstrate the model's ability to accurately predict heart rate responses to exercise intensity during workout sessions, indicating its effectiveness in capturing dynamic physiological changes.

Figure 6 illustrates the model's performance in predicting heart rate during workout sessions. The x-axis represents the time from the start of the workout session, while the y-axis shows the instantaneous heart rate in beats per minute (BPM). The solid blue line indicates the actual heart rate measured during the workout, and the dashed orange line represents the predicted heart rates generated by the model. The shaded region around the predicted values represents uncertainty bands of ±5 BPM, accounting for variability in heart rate measurements.

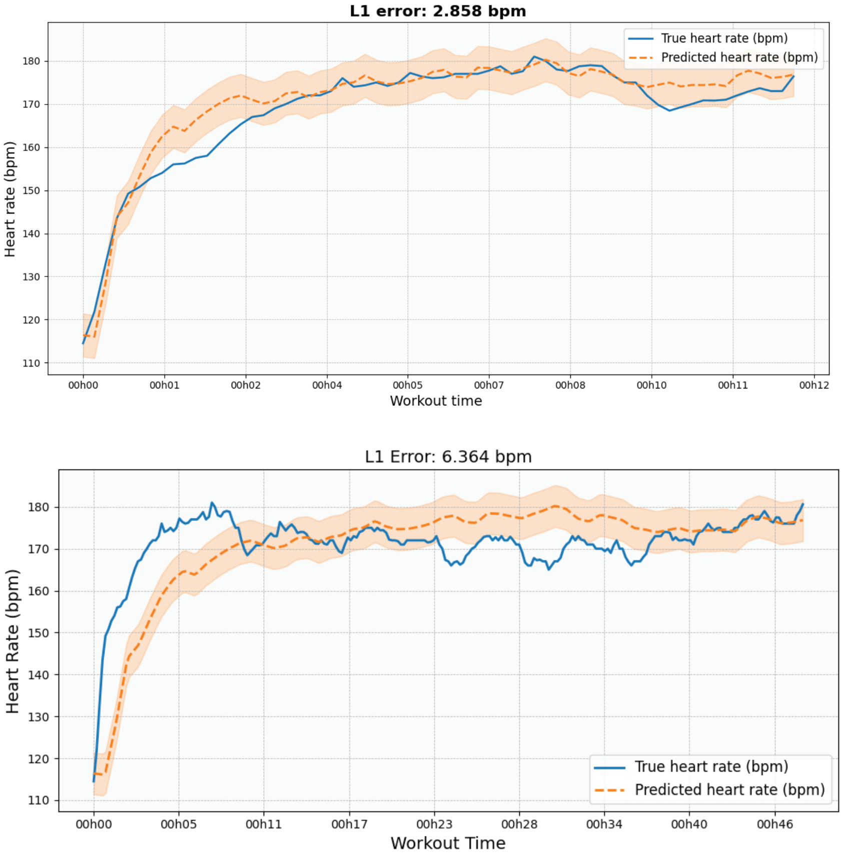


Figure 6. True vs. predicted heart rates for the specific session.

5.3.2. Performance Analysis

We demonstrated visualizations of the predicted heart rate versus actual heart rate for a variety of workout sessions, as a measure for the model proposed in this paper.

Figure 7 shows the difference between the predicted and actual heart rates across all workout sessions, plotted against the average of those heart rates. The plot has a mean bias of 0.42 BPM, indicating that, on average, the model slightly overestimates heart rates. The limits of agreement, set at ± 1.96 times the standard deviation from the mean bias, range from -15.53 BPM to 16.38 BPM. These relatively tight limits suggest that most predictions are close to the actual heart rates, highlighting the model's accuracy across various heart rate ranges. The points on the plot are evenly distributed around the mean bias line, with no clear patterns or systematic errors, indicating that the errors are random and not biased toward any particular heart rate level. The narrow range of the limits further supports the notion that our DBN model provides reliable heart rate predictions with minimal error. Such consistent performance is critical in real-world applications, where accurate heart rate monitoring is essential for making personalized fitness recommendations.

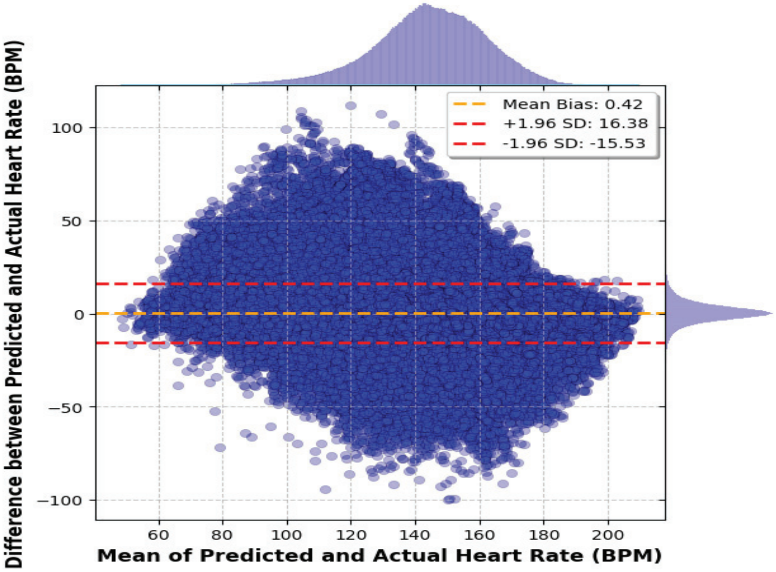


Figure 7. The Bland–Altman plot.

For further insights, Figure 8 demonstrates the model's performance on true vs predicted heart rates for a specific session.

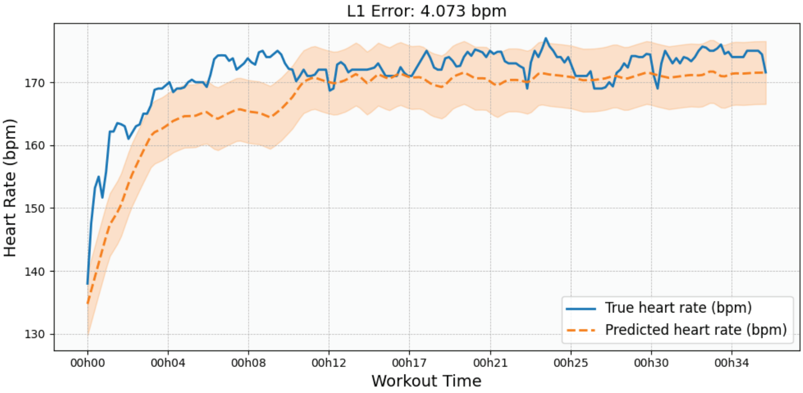


Figure 8. True vs. predicted heart rates for the specific session.

Figure 9 illustrates the model's ability to predict heart rate evolution across different individuals during workout sessions. The solid blue lines represent the actual heart rate evolution, while the dashed orange lines depict the model's predictions. Across all individuals, the model closely follows the heart rate trends, demonstrating its capacity to generalize well across varying users and workout intensities. The model accurately predicts heart rate responses to different exercise intensities, capturing rapid increases during high-intensity intervals and smooth declines during recovery periods. For most individuals, the predicted heart rate closely aligns with the actual heart rate, particularly during steady-state phases of the workout. However, minor discrepancies can be seen during rapid heart rate fluctuations, which may be attributed to physiological differences between individuals or noise in the sensor data. Overall, the model's performance across multiple individuals highlights its robustness in handling varying user profiles and workout conditions, making it effective for personalized heart rate prediction and health tracking applications, as well as for delivering personalized fitness recommendations in real-world scenario.

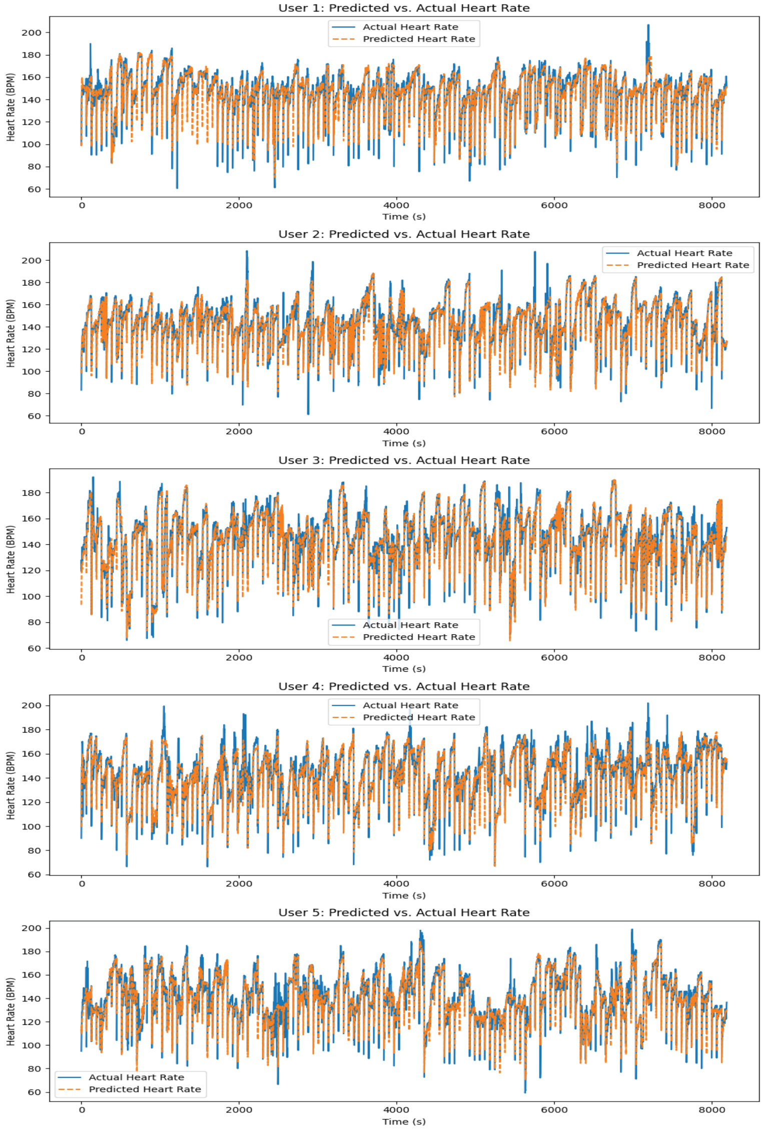


Figure 9. Model performance on actual and predicted heart rate evolution for different individuals.

Table 3 presents a comparison of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) between the proposed hybrid DBN model and two state-of-the-art models, hybrid ODE [4] and FitRec [5]. All models were evaluated on the same dataset for consistency. The Hybrid DBN model achieves a significantly lower MAE of 5.2 BPM and RMSE of 8.1 BPM, demonstrating its effectiveness in capturing the complex dynamics of heart rate response during exercise.

Table 3. DBN Model comparison against state-the-of art.

Model	MAE (BPM)	RMSE (BPM)
Hybrid ODE Model	6.1	-
FitRec (U/S/C)	7.0	17.1
Hybrid DBN Model	5.2	8.1

This improved performance highlights the Hybrid DBN model's ability to model personalized heart rate predictions more accurately, making it a reliable tool for understanding user behavior and providing personalized fitness recommendations. Compared to the FitRec model, which has an MAE of 7.0 BPM and an RMSE of 17.1 BPM, the Hybrid DBN model offers a substantial improvement in both error metrics.

5.3.3. Ablation Study

The ablation study in Table 4 illustrates the contributions of different components to the overall performance of the Hybrid DBN model. The table shows that removing the LSTM encoder increases the MAE, highlighting the critical role of the individual's health status representation (i.e., past workout data) in ensuring precise predictions.

Table 4. Ablation study on components contribution.

Model	MAE (BPM)
Hybrid DBN Model	5.2
Without LSTM Encoder	21.3
Without Adaptive Feature Selection	12.5
Without Personalized Parameters	11.7

Furthermore, omitting the adaptive feature selection module results in a significant decline in model performance, emphasizing the module's function in dynamically selecting the most relevant features. Lastly, the elimination of personalized parameters also negatively affects prediction accuracy, demonstrating the importance of personalization for adapting the model to individual users.

6. Personalized Recommendations

In addition to predicting heart rate responses to exercise, our model demonstrates the ability to offer personalized fitness recommendations tailored to individual behavior patterns. In this section, we explore recommendation tasks that showcase the model's practical utility in real-world fitness environments:

1. Workout Optimization Recommendation: Based on a user's predicted heart rate profile and specific workout goals, we suggest adjusting workout parameters, such as speed or intensity, to optimize performance and achieve desired outcomes. For instance, if a user aims to maintain a particular heart rate zone (e.g., fat-burning or aerobic zone), our model can dynamically adjust recommendations during the workout, advising users to increase or decrease their pace accordingly. This functionality is particularly beneficial for users who aim to meet specific heart rate targets or improve cardiovascular endurance. Additionally, the system can suggest variations in exercise routines to prevent fitness plateaus, recommending different workout intensities or durations based on historical performance data.
   Scenari: A user who frequently runs on a flat course might receive a suggestion to try a hilly route that provides a similar heart rate response due to its variability in incline. This approach helps users diversify their workouts, potentially enhancing overall fitness by exposing them to different physical demands while keeping the exercise routine engaging.
2. Real-Time Workout Guidance: Our model serves as a real-time personal trainer, predicting the short-term fluctuations in heart rate based on the current user pace and

environmental conditions. If the model predicts that the person's heart rate will go above some threshold (e.g., 85% of maximum heart rate), the system can proactively alert the user to adjust their pace, ensuring a safer and more effective workout. It helps avoid overexertion, making exercise much safer and far more effective, especially for novices or people with specific health concerns.
Scenario: During a high-intensity interval training (HIIT) session, the system monitors the user's heart rate in real-time. If the heart rate approaches a critical level, the system suggests a brief recovery period or a reduction in intensity to avoid excessive strain.

3. Workout route and Activity Suggestion: The model can recommend specific workout routes or activities that align with the user's fitness goals and preferences by analyzing historical workout data and heart rate responses. This is especially beneficial for users who wish to explore new routes or maintain their regular exercise routine while traveling. The recommendation system can identify routes that match a user's preferred heart rate profile or suggest new routes that offer similar physiological challenges, such as maintaining a steady heart rate or achieving specific exertion levels.
Scenario: A user who frequently runs on a flat course might receive a suggestion to try a hilly route that provides a similar heart rate response due to its variability in incline. This approach helps users diversify their workouts, potentially enhancing overall fitness by exposing them to different physical demands while keeping the exercise routine engaging.

7. Conclusions

In this study, we developed a novel hybrid physiological model using Dynamic Bayesian Networks (DBNs). The integration of DBNs enables robust modeling of heart rate dynamics over time, capturing both the physiological aspects of heart rate response and the influence of external factors, such as workout intensity and environmental conditions. The incorporation of LSTM networks further enhances the model's ability to learn from sequential data, providing a personalized health representation that adapts to each user's specific physiological state and exercise patterns.

However, there are some limitations to our study. The model was trained and validated on the FitRec dataset, which may not fully represent the diversity of real-world users, and the data are skewed toward male individuals. Testing our model with data from a broader range of sources, such as those collected from Apple Fitness or Samsung Health platforms, could offer a more comprehensive evaluation of its performance across different devices and user demographics. Access to such data would also allow us to better explore the model's generalization and adaptability.

Despite these limitations, our results demonstrate that the proposed model not only predicts heart rate with improved accuracy but also provides valuable insights into individual fitness levels and responses to exercise. This capability can be applied to developing personalized fitness recommendations, offering real-time guidance based on predicted heart rate zones and exercise intensity.

Future work will focus on extending the model to other types of exercises and further improving its interpretability by incorporating more extensive data from wearables. Additionally, exploring the model's potential to predict long-term health outcomes and integrating it into comprehensive health monitoring systems could lead to more personalized and effective fitness and health recommendations.

Overall, this work represents a significant step forward in leveraging machine learning and physiological modeling to provide personalized fitness insights, demonstrating the potential of hybrid models in health and fitness applications.

Author Contributions: Conceptualization, H.K.; methodology, H.K., Y.P. and J.M.; software, H.K.; validation, J.C. (Jongsun Choi) and J.C. (Jaeyoung Choi); formal analysis, H.K.; investigation, H.K.; resources, J.C. (Jongsun Choi); data curation, H.K.; writing—original draft preparation, H.K.; writing—review and editing, J.M, J.C. (Jongsun Choi), J.C. (Jaeyoung Choi), Y.P. and H.K.; visualization, H.K.; supervision, J.M. and J.C. (Jongsun Choi); project administration, J.C. (Jongsun Choi) and J.C. (Jaeyoung Choi); funding acquisition, J.C. (Jongsun Choi) and J.C. (Jaeyoung Choi). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022-0-00218).

Data Availability Statement: Restrictions apply to the availability of these data. The data supporting this work were obtained from the FitRec dataset, originally created by Jiamo Ni and collaborators. Access to the dataset can be requested from the authors through https://github.com/nijianmo/fit-rec. The code used for the model implementation and analysis is available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Anderson, E.; Durstine, J.L. Physical activity, exercise, and chronic diseases. A brief review. *Sports Med.* **2019**, *1*, 3–10. [CrossRef] [PubMed]
2. Rippe, J.M. Lifestyle Medicine. The Health Promoting Power of Daily Habits and Practices. *Am. J. Lifestyle Med.* **2018**, *12*, 499–512. [CrossRef] [PubMed]
3. Shei, R.J.; Holder, I.G.; Oumsang, A.S.; Paris, B.A.; Paris, H.L. Wearable activity trackers–advanced technology or advanced marketing? *Eur. J. Appl. Physiol.* **2022**, *122*, 1975–1990. [CrossRef] [PubMed]
4. Nazaret, A.; Tonekaboni, A.S.; Darnell, G.; Ren, S.Y.; Sapiro, G.; Miller, A.C. Modeling personalized heart rate response to exercise and environmental factors with wearables data. *NPJ Digit. Med.* **2023**, *6*, 207. [CrossRef] [PubMed]
5. Jiamo, N.; Muhlstein, L.; McAuley, J. Modeling heart rate and activity data for personalized fitness recommendation. In Proceedings of the World Wide Web Conference, WWW 2019, San Francisco, CA, USA, 13–17 May 2019; pp. 1343–1353.
6. Zahedani, A.D.; McLaughlin, T.; Veluvali, A.; Aghaeepour, N.; Hosseinian, A.; Agarwal, S.; Ruan, J.; Tripathi, S.; Woodward, M.; Hashemi, N.; et al. Digital health application integrating wearable data and behavioral patterns improves metabolic health. *NPJ Digit. Med.* **2024**, *7*, 9. [CrossRef]
7. Zhu, Z.; Li, H.; Xiao, J.; Xu, W.; Huang, M.C. A fitness training optimization system based on heart rate prediction under different activities. *Methods* **2022**, *205*, 89–96. [CrossRef]
8. Martín-Escudero, P.; Cabanas, A.M.; Dotor-Castilla, M.L.; Galindo-Canales, M.; Miguel-Tobal, F.; Fernández-Pérez, C.; Fuentes-Ferrer, M.; Giannetti, R. Are Activity Wrist-Worn Devices Accurate for Determining Heart Rate during Intense Exercise? *Bioengineering* **2023**, *10*, 254. [CrossRef]
9. Whipple, M.O.; Schorr, E.N.; Talley, K.M.C.; Lindquist, R.; Bronas, U.G.; Treat-Jacobson, D. Variability in individual response to aerobic exercise interventions among older adults. *J. Aging Phys.* **2018**, *26*, 655–670. [CrossRef]
10. Brydges, C.R.; Liu-Ambrose, T.; Bielak, A.A.M. Using intraindividual variability as an indicator of cognitive improvement in a physical exercise intervention of older women with mild cognitive impairment. *Neuropsychology* **2020**, *34*, 825–834. [CrossRef]
11. Jelinek, H.F.; Karmakar, C.; Kiviniemi, A.M.; Hautala, A.J.; Tulppo, M.P.; Mäkikallio, T.H.; Huikuri, H.V.; Khandoker, A.H.; Palaniswami, M. Temporal dynamics of the circadian heart rate following low and high volume exercise training in sedentary male subjects. *Eur. J. Appl. Physiol.* **2015**, *115*, 2069–2080. [CrossRef]
12. Stephenson, M.D.; Thompson, A.G.; Merrigan, J.J.; Stone, J.D.; Hagen, J.A. Applying Heart Rate Variability to Monitor Health and Performance in Tactical Personnel: A Narrative Review. *Int. J. Environ. Res. Public Health* **2021**, *18*, 8143. [CrossRef] [PubMed]
13. Mazzoleni, M.J.; Battaglini, C.L.; Martin, K.J.; Coffman, E.M.; Ekaidat, J.A.; Wood, W.A.; Mann, B.P. A dynamical systems approach for the submaximal prediction of maximum heart rate and maximal oxygen uptake. *Sports Eng.* **2018**, *21*, 31–41. [CrossRef]
14. Mazzoleni, M.J.; Battaglini, C.L.; Martin, K.J.; Coffman, E.M.; Mann, B.P. Modeling and predicting heart rate dynamics across a broad range of transient exercise intensities during cycling. *Sports Eng.* **2016**, *19*, 117–127. [CrossRef]
15. Ferguson, T.; Olds, T.; Curtis, R.; Blake, H.; Crozier, A.J.; Dankiw, K.; Dumuid, D.; Kasai, D.; O'Connor, E.; Virgara, R.; et al. Effectiveness of wearable activity trackers to increase physical activity and improve health: A systematic review of systematic reviews and meta-analyses. *Lancet Digit. Health* **2022**, *4*, e615–e626. [CrossRef] [PubMed]
16. Banaee, H.; Ahmed, M.U.; Loutfi, A. Data mining for wearable sensors in health monitoring systems: A review of recent trends and challenges. *Sensors* **2013**, *13*, 17472–17500. [CrossRef]
17. Tang, M.S.S.; Moore, K.; McGavigan, A.; Clark, R.A.; Ganesan, A.N. Effectiveness of wearable trackers on physical activity in healthy adults: Systematic review and meta-analysis of randomized controlled trials. *JMIR mHealth uHealth* **2022**, *8*, e18868. [CrossRef]

18. Greco, G.; Poli, L.; Clemente, F.M.; Fischetti, F.; Cataldi, S. The Effectiveness of New Digital Technologies in Increasing Physical Activity Levels and Promoting Active and Healthy Ageing: A Narrative Review. *Health Soc. Care Community* **2023**, *2023*, 2803620. [CrossRef]
19. Ates, H.C.; Yetisen, A.K.; Güder, F.; Dincer, C. Wearable devices for the detection of COVID-19. *Nat. Electron.* **2021**, *4*, 13–14. [CrossRef]
20. Hasasneh, A.; Hijazi, H.; Talib, M.A.; Afadar, Y.; Nassif, A.B.; Nasir, Q. Wearable Devices and Explainable Unsupervised Learning for COVID-19 Detection and Monitoring. *Diagnostics* **2023**, *13*, 3071. [CrossRef]
21. Natarajan, A.; Su, H.W.; Heneghan, C.; Blunt, L.; O'Connor, C.; Niehaus, L. Measurement of respiratory rate using wearable devices and applications to COVID-19 detection. *NPJ Digit. Med.* **2021**, *4*, 136. [CrossRef]
22. Oyeleye, M.; Chen, T.; Titarenko, S.; Antoniou, G. A Predictive Analysis of Heart Rates Using Machine Learning Techniques. *Int. J. Environ. Res. Public Health* **2022**, *19*, 2417. [CrossRef] [PubMed]
23. Tao, K.; Li, J.; Shan, W.; Yan, H.; Lu, Y. Estimation of Heart Rate Using Regression Models and Artificial Neural Network in Middle-Aged Adults. *Front. Physiol.* **2021**, *12*, 742754. [CrossRef] [PubMed]
24. Lin, H.; Zhang, S.; Li, Q.; Li, Y.; Li, J.; Yang, Y. A new method for heart rate prediction based on LSTM-BiLSTM-Att. *Measurement* **2023**, *207*, 112384. [CrossRef]
25. Signorini, M.G.; Pini, N.; Malovini, A.; Bellazzi, R.; Magenes, G. Integrating machine learning techniques and physiology based heart rate features for antepartum fetal monitoring. *Comput. Methods Programs Biomed.* **2022**, *185*, 105015. [CrossRef] [PubMed]
26. Neshitov, A.; Tyapochkin, K.; Kovaleva, M.; Dreneva, A.; Surkova, E.; Smorodnikova, E.; Pravdin, P. Estimation of cardiorespiratory fitness using heart rate and step count data. *Sci. Rep.* **2023**, *13*, 15808. [CrossRef]
27. Stork, M.; Novak, J.; Zeman, V. Dynamic models of some physiological parameters in response to exercise. In Proceedings of the International Conference on Applied Electronics 2019, Pilsen, Czech Republic, 10–11 September 2019; Volume 22, pp. 122–129.
28. Rosas, F.E.; Candia-Rivera, D.; Luppi, A.I.; Guo, Y.; Mediano, P.A.M. Bayesian at heart: Towards autonomic outflow estimation via generative state-space modelling of heart rate dynamics. *Comput. Biol. Med.* **2024**, *170*, 107857. [CrossRef]
29. Zhang, H.; Wen, B.; Liu, J. The Prediction of Heart Rate during Running Using Bayesian Combined Predictor. In Proceedings of the 14th International Wireless Communications and Mobile Computing Conference, IWCMC 2018, Limassol, Cyprus, 25–29 June 2018.
30. Ladyzynski, P.; Molik, M.; Foltynski, P. Dynamic Bayesian networks for prediction of health status and treatment effect in patients with chronic lymphocytic leukemia. *Sci. Rep.* **2022**, *12*, 1811. [CrossRef]
31. Chen, Q.; Tang, B.; Song, J.; Jiang, Y.; Zhao, X.; Ruan, Y.; Zhao, F.; Wu, G.; Chen, T.; He, J. Dynamic Bayesian network for predicting physiological changes, organ dysfunctions and mortality risk in critical trauma patients. *BMC Med. Inform. Decis. Mak.* **2022**, *22*, 119. [CrossRef]
32. Marshall, A.H.; Hill, L.A.; Kee, F. Continuous Dynamic Bayesian networks for predicting survival of ischaemic heart disease patients. In Proceedings of the IEEE Symposium on Computer-Based Medical Systems, Perth, PA, USA, 12–15 October 2010; pp. 178–183.
33. Malhotra, R.; Singh, P. Recent advances in deep learning models: A systematic literature review. *Multimed. Tools Appl.* **2023**, *82*, 44977–45060. [CrossRef]
34. Liu, X.; Gao, B.; Suleiman, B.; You, H.; Ma, Z.; Liu, Y.; Anaissi, A. Privacy-Preserving Personalized Fitness Recommender System P3FitRec: A Multi-level Deep Learning Approach. *ACM Trans. Knowl. Discov. Data* **2023**, *17*, 1–24.
35. Lin, W.; Zhao, X.; Wang, Y.; Xu, T.; Wu, X. AdaFS: Adaptive Feature Selection in Deep Recommender System. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, Washington, DC, USA, 14–18 August 2022; pp. 3309–3317.
36. Kayange, H.; Mun, J.; Park, Y.; Choi, J.; Choi, J. ProAdaFS: Probabilistic and Adaptive Feature Selection in Deep Recommendation Systems. In Proceedings of the 38th International Conference on Information Networking ICOIN, Ho Chi Minh City, Vietnam, 17–19 January 2024; pp. 756–761.
37. Robusto, C.C. The Cosine-Haversine Formula. *Am. Math. Mon.* **1957**, *64*, 38–40. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

Article

Bayesian Modeling of Travel Times on the Example of Food Delivery: Part 2—Model Creation and Handling Uncertainty

Jan Pomykacz, Justyna Gibas and Jerzy Baranowski *

Department of Automatic Control & Robotics, AGH University of Kraków, 30-059 Kraków, Poland; janpomykacz@student.agh.edu.pl (J.P.); justynagibas@student.agh.edu.pl (J.G.)
* Correspondence: jb@agh.edu.pl

Abstract: The e-commerce sector is in a constant state of growth and evolution, particularly within its subdomain of online food delivery. As such, ensuring customer satisfaction is critical for companies working in this field. One way to achieve this is by providing an accurate delivery time estimation. While companies can track couriers via GPS, they often lack real-time data on traffic and road conditions, complicating delivery time predictions. To address this, a range of statistical and machine learning techniques are employed, including neural networks and specialized expert systems, with different degrees of success. One issue with neural networks and machine learning models is their heavy dependence on vast, high-quality data. To mitigate this issue, we propose two Bayesian generalized linear models to predict the time of delivery. Utilizing a linear combination of predictor variables, we generate a practical range of outputs with the Hamiltonian Monte Carlo sampling method. These models offer a balance of generality and adaptability, allowing for tuning with expert knowledge. They were compared with the PSIS-LOO criteria and WAIC. The results show that both models accurately estimated delivery times from the dataset while maintaining numerical stability. A model with more predictor variables proved to be more accurate.

Keywords: online food delivery (OFD); delivery time estimation; Bayesian inference; generalized linear models

Citation: Pomykacz, J.; Gibas, J.; Baranowski, J. Bayesian Modeling of Travel Times on the Example of Food Delivery: Part 2—Model Creation and Handling Uncertainty. *Electronics* **2024**, *13*, 3418. https://doi.org/10.3390/electronics13173418

Academic Editors: Chao Zhang, Wentao Li, Huiyan Zhang and Tao Zhan

Received: 19 June 2024
Revised: 15 August 2024
Accepted: 16 August 2024
Published: 28 August 2024

1. Introduction

The e-commerce sector is in a constant state of growth and evolution, particularly within its subdomain of online food delivery (OFD) [1,2]. Recent market forecasts indicate a steady rise in revenue for companies offering such services. With numerous players in the market, ensuring customer satisfaction is paramount for a company's survival. Customers increasingly demand user-friendly applications that simplify the ordering process with just a few taps while also providing features such as delivery time estimates and communication channels with couriers [3]. However, estimating delivery times accurately without managing the uncertainty associated with real-time events and decision making may be suboptimal. This is indicated by recent studies, which either focus on this [4,5], account for this [6], or indicate this in future works [7].

The existing research in this field is broad. Some works focus on static origin–destination time-travel prediction [4]. Others create commercial-grade solutions capable of handling real-time data [6]. Recent studies have started to focus on more complex problems, e.g., the restaurant-meal-delivery problem, which is characterized by a fleet of delivery vehicles that serve dynamic customer requests over the course of a day [5]. A more detailed description is provided in Section 2.

Machine learning models are common in the task of time-travel prediction. Among these methods, neural networks, including Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory Networks (LSTMs), are prominently utilized [8].

Although Bayesian statistics and inference have gained increasing popularity, there remains a notable scarcity of articles addressing their application in delivery time prediction. In their case study, Abdi et al. list less than ten methods based on either a Naive Bayes classificator, Bayesian Network, or Bayesian graphical model while examining around a hundred articles [8]. The method boasts several advantages: it offers straightforward, interpretable models; the capacity to adapt and improve with new data; and provides a measure of uncertainty for each prediction. However, it also presents challenges, notably its computational demands and the potential for poor model performance due to incorrect assumptions.

This article introduces two Bayesian models designed for predicting food-delivery times. Utilizing a linear combination of predictor variables, we generate a practical range of outputs. These models offer a balance of generality and adaptability, allowing for tuning with expert knowledge. This ensures flexibility and stability in various contexts. To assess their performance and identify any potential drawbacks, we compared the models using the PSIS-LOO criteria and WAIC.

The main contributions of this paper are as follows: (1) To the best of our knowledge, this is the first application of Bayesian inference to online food-delivery-time prediction. (2) By specifying models as linear combinations of predictors, we achieve high interpretability, which aids in identifying the primary factors influencing delivery time. (3) Our results indicate that Bayesian inference holds promise for further exploration in this context, as it can lead to promising results.

The remainder of this paper is structured as follows: In the next section, relevant studies in the extant literature are reviewed and discussed. Section 3.1 provides a short introduction to the concept of Bayesian statistics. Section 3.2 refers to the numerical computation methods utilized in our work. Section 3.3 reveals the data source and contains reference to part 1 of this article, where preprocessing is described. Section 3.4 focuses on the model definition, prior distribution selection, and prior predictive checks. In Section 4, we present and explain the results of our work as posterior predictive checks, where Section 4.1 focuses on Model 1 and Section 4.2 focuses on Model 2. In Section 4.3, we compare models to see how they fare against each other. Section 5 discusses the limitations of our models. Finally, Section 6 summarizes the conclusions drawn from this study.

2. Literature Review

Food-delivery-time estimation can be perceived in the category of Estimated Time of Arrival (ETA). In our work, it will also include meal-preparation time, but the rest focuses solely on the travel time between origin and destination. Overall, there are two common approaches to ETA: route-focused and origin–destination-focused.

The route-based approach focuses on segmenting routes and estimating the travel time of each segment. Lee et al. implemented a real-time expert system that takes present and historical data and produces travel-time-prediction rules via data-mining techniques. Also, they implement a dynamic weight combination governed by meta-rules, which allows for a real-time road events response to enhance the prediction's precision [9]. Li et al. proposed a deep generative model—DeepGTT. It is a probabilistic model designed to generate a travel time distribution, from which travel time as well as the uncertainty about it can be inferred [10]. Asghari et al. presented algorithms for computing the probability distribution of travel times for each link of a given route. It differs from other works as the authors mention that elsewhere, probabilistic link travel times are given a priori. This, and the work mentioned beforehand, are one of the few works that focus on distributions rather than strict numbers [11]. Wang et al. proposed a model for estimating the travel time of any path consisting of connected road segments based on current and historical GPS records, as well as map sources. Due to data sparsity (not every road will be traveled by a vehicle with GPS) and the trade-offs associated with multiple ways of connecting road fragments to form a route, the problem as a whole was not solved [12]. Wang et al. formulate ETA as a spatial–temporal problem. They adapted different neural networks, as well as proposed the authorial Wide-Deep-Recurrent model and trained them on floating-car data. The solution showed promising results and was deployed for Didi Chuxing's vehicle-for-hire company [13]. Han et al. propose an incremental ETA learning framework to address issues of the scalability and robustness of real-world large-scale ETA scenarios. The framework works as an incremental travel-time predictor that is updated on newly generated traffic data. The authors also include a historical traffic knowledge consolidation module to reuse historical data and an adversarial training module to mitigate and resist traffic noise perturbations caused by low-quality data. The model was employed at Didi Chuxing's company, substantially improving the prediction accuracy [6].

The origin–destination methods refrain from estimating routes, stating that it is time consuming and potentially erroneous and gives a worse result than OD methods. Zhu et al. predict the Order Fulfillment Cycle Time (OFCT), which is the time between placing an order and receiving the meal. Their approach consists of identifying key factors behind the OFCT and capturing them within multiple features from diverse data sources, and then feeding them to the DNN created for this task. It is worth noting that their approach is specifically tailored to food delivery, which aligns with the common goal outlined in our article [14]. Li et al. proposed the MURAT model with the goal of predicting travel time given the origin and destination location, as well as the departure time. They also present a multi-task learning framework to integrate prior historical data into the training process to boost performance [15]. Wu, C. H. et al. examined a classical machine learning algorithm, which is support vector regression. Their findings show the feasibility of such a method for travel-time prediction [16]. Wang et al. leverage the increasing availability of travel data. Their approach is to use large historical datasets to accurately predict the travel time between the origin and destination without computing the route. The shown solution outperformed the services of Bing Maps and Baidu Maps at the time [17]. Lin et al. propose a framework called the Origin–Destination Travel Time Oracle to estimate travel time given the origin–destination pair and departure time. It uses historical trajectories alongside the OD pair to infer image-based Pixelated Trajectories. Based on the inferred trajectory, a Masked Vision Transformer is capable of estimating travel time. The results outperform most of the other solutions highlighted in the paper [18]. Zhou et al. examine the ETA problem in the context of e-commerce platforms. They introduce the Inductive Graph Transformer. Unlike other graph transformer architectures, it trains the transformer as a regression function that captures both information from raw features as well as dense embeddings encoded by a graph neural network. The graph neural network is also simplified to allow the solution to be applied to large-scale industrial scenarios. The results show performance improvement with metrics such as the mean absolute error, mean absolute percentage error, and mean absolute relative error compared to other models [19]. Zhang et al. propose the Graph-Structure-Learning-Based Quantile Regression model for ETA in e-commerce. According to the authors' knowledge, this is the first application of graph structure learning in this field and suggests that most of the other work utilizing fixed graph structures may be suboptimal. For the ETA, they design multi-objective quantile regression loss capable of finding a Pareto solution to the problem. The authors also propose fast sampling-based methods to reduce the computational complexity and enable the solution to be used for large-scale graphs. The results are shown to outperform baseline models [20].

Recently, new work was introduced into this field, building on existing ETA solutions to tackle problems of higher complexity. Ulmer et al. consider the restaurant-meal-delivery problem, which regard the optimization of the fleet of delivery vehicles serving dynamic customer requests throughout the day. The present anticipatory customer-assignment policy is used to handle the uncertainty of an unknown meal-preparation time as well as unknown customer localization. The policy is based on a time buffer and postponing to reduce making decisions that would result in delivery delays. Based on data from the city of Iowa, the authors show results that outperform other restaurant-delivery policies [5]. Hildebrandt and Ulmer combined ETA and the restaurant-meal-delivery problem. They proposed an offline method, which maps a set of features to expected arrival times using gradient-boosted decision tree. The results show that it has a better performance in comparison to planning on means, which is a sum of the expected times of each action on route. The second proposed model is called offline–online, with real-time predictions in mind. It uses a pretrained DNN to approximate the exact route of delivery in a full online-simulation scenario. The authors show that this approach achieves a near full-optimal online-simulation accuracy with a fraction of the computational time [21]. Xue et al. focus on minimizing the cost of the restaurant-delivery problem with an uncertain cooking time and travel time and give insight into the influence of those uncertainties on food-platform preference. They propose a scenario-based chance-constrained programming model to capture the variability of cooking and travel times and develop an island harmony search algorithm to generate high-quality solutions. The results show that both uncertainties are critical for the restaurant-delivery problem [4]. Gao et al. combine the ETA problem with the estimation of the delivery route. While problems are closely related when it comes to food delivery, they are often examined separately. The authors propose a deep network named FDNET, consisting of route- and time-prediction modules. The route-prediction module is used to determine the next localization that a courier will visit in a multi-delivery scenario. The time-prediction module estimates the travel time between two adjacent locations based on the drivers and spatiotemporal features. Offline experiments show promising results compared to the frequently used machine learning models [7].

3. Materials and Methods

3.1. Bayesian Inference

For a better understanding of problem formulation and the proposed solution, a short introduction to Bayesian inference is in order. It is a method of statistical inference, in which we fit a predefined probability model to a set of data and evaluate the outcomes with regard to the observed parameters of the model and unobserved quantities, like predictions for new data points [22]. It is performed with the use of Bayes' rule, shown in Equation (1):

$$p(\theta \mid y) = \frac{p(\theta, y)}{p(y)} = \frac{p(\theta)p(y \mid \theta)}{p(y)}, \tag{1}$$

or rewritten in an unnormalized version:

$$p(\theta \mid y) \propto p(\theta)p(y \mid \theta). \tag{2}$$

This tells us the relation between *theta*, which is an unobservable vector of variables of interest, and y, which is a vector of observed variables. The left-hand side of the equation is called the posterior distribution, while the right-hand side is a product of prior distribution and the likelihood function. We define the prior predictive distribution as

$$p(y) = \int p(\theta)p(y \mid \theta)\, d\theta, \tag{3}$$

and the posterior predictive distribution as

$$p(\tilde{y} \mid y) = \int p(\tilde{y} \mid \theta)p(\theta \mid y)\, d\theta. \tag{4}$$

The prior predictive distribution is not conditional on the previous observation y of the process and refers to observed data, while the posterior predictive distribution is conditional on y and predicts potential future observations $\tilde{y}$ [22].

Bayesian statistics is widely utilized in the behavioral and social sciences, largely due to the increasing availability of user-friendly software and comprehensive tutorials tailored for scientists in these fields. It is primarily employed for theory development and estimation. This approach is particularly well-suited for these disciplines because meaningful priors can be derived from extensive literature, and informative priors are valuable for modeling complex behaviors and working with small sample sizes, both of which are common in the social sciences. Bayesian methods are also used in ecological modeling due to their ability to handle complex, high-dimensional, and spatiotemporal models, as well as imperfect or incomplete data. These models often involve computationally expensive likelihoods. Bayesian techniques, such as data augmentation, can fit these models more effectively without requiring oversimplification, which may be necessary in a frequentist framework. Applications of Bayesian statistics in ecology span various scales, from individual organisms to entire ecosystems, and include tasks like understanding population dynamics, modeling spatial patterns, studying population genetics, estimating abundance, and assessing conservation efforts [23].

3.2. Stan Programming

The models were created in Stan. It is a programming language written in C++ and used for statistical inference. It provides a concise way of defining Bayesian models as simple scripts, yet allows for the efficient computation of Markov Chain Monte Carlo methods, which are essential parts of Bayesian inference [24].

The algorithm used in Stan sampling is Hamiltonian Monte Carlo. It is a Markov Chain Monte Carlo (MCMC) method, which uses derivatives of the density function being sampled to generate efficient transitions spanning the posterior distribution. The goal of the sampler is to draw from density $p(\theta \mid y)$, where θ is a vector of parameters and y is a data sample. HMC introduces momentum variables ρ and draws from the joint density:

$$p(\rho, \theta) = p(\rho \mid \theta) p(\theta), \tag{5}$$

$$\rho \sim MultiNormal(0, M), \tag{6}$$

where M is a Euclidean metric.

The joint density $p(\rho, \theta)$ defines a Hamiltonian:

$$\begin{aligned} H(\rho, \theta) &= -logp(\rho, \theta) \\ &= -logp(\rho \mid \theta) - logp(\theta) \\ &= T(\rho \mid \theta) + V(\theta) \end{aligned} \tag{7}$$

where $T(\rho \mid \theta)$ and $V(\theta)$ are called kinetic and potential energy, respectively.

Transitions are generated in two steps. First, a value of momentum is generated independently of the current parameters. Then, the joint system of current parameters and new momentum is defined as Hamilton's equations:

$$\frac{\partial \theta}{\partial t} = \frac{\partial H}{\partial \rho} = \frac{\partial T}{\partial \rho} \tag{8}$$

$$\frac{\partial \rho}{\partial t} = -\frac{\partial H}{\partial \theta} = -\frac{\partial T}{\partial \theta} - \frac{\partial V}{\partial \theta} \tag{9}$$

Since the momentum density is independent of the parameters' density $p(\rho \mid \theta) = p(\rho)$, the term $-\frac{\partial T}{\partial \theta}$ is zero, canceling the first term of the second equation.

Stan's implementation of HMC uses the Leapfrog integrator, as it provides stability for Hamiltonian systems of equations. It starts with sampling new momentum independently of parameters or a previous momentum value. In discrete time steps, denoted as ϵ, it half-step updates momentum and full-step updates parameters:

$$\rho \leftarrow \rho - \frac{\epsilon}{2} \frac{\partial V}{\partial \theta} \tag{10}$$

$$\theta \leftarrow \theta + \epsilon M^{-1} \rho \tag{11}$$

$$\rho \leftarrow \rho - \frac{\epsilon}{2}\frac{\partial V}{\partial \theta} \tag{12}$$

After applying L leapfrog steps, a total of $L\epsilon$ time is simulated. The resulting state of the simulation is denoted as (ρ^*, θ^*). Lastly, the proposal (ρ^*, θ^*) generated by the transition from (ρ, θ) has a probability of being accepted defined as

$$min(1, exp(H(\rho, \theta) - H(\rho^*, \theta^*))) \tag{13}$$

If the proposal is not accepted, the previous parameter value θ is utilized in the next iteration [24].

Stan is able to automatically optimize ϵ to match an acceptance-rate target, able to estimate M based on warmup sample iterations, and able to dynamically adapt L on the fly during sampling (and during warmup). This helps to mitigate the risks associated with divergence caused by improper algorithm-parameter selection [24].

3.3. Data

The data used for inference come from Kaggle [25]. Data preprocessing was performed in two steps. The first one was generating the shortest route between an origin and destination pair. It was necessary as the geographical coordinates presented in the raw data were unsuitable to create a meaningful probability distribution for our models. It was performed using OSRM API. The second step was data cleaning and analysis. We decided to remove coordinates outside of India's geographical boundaries, which was the country where the data originated from. We computed the meal-preparation time as the difference between the time when the order was picked up by the courier and the time when the order was made in the restaurant. Similar to coordinates, timestamps would provide difficulties when trying to associate them with distributions. We used z-score standardization for numerical variables. It was necessary to do so, as our models use an exponential function on a linear combination of predictors, and if the latter were too large, it caused numerical problems with computation. Finally, we mapped categorical variables to numerical indices, which would be used to associate the category with its corresponding distribution. In-depth preprocessing was described in part 1 of this article [26]. The chosen features are presented in Table 1. There were 45,593 raw data samples. After processing, we ended with 34,920, which will be further denoted as N. Histograms of the data are presented in Figure 1 [26].

Table 1. Features computed from dataset.

Model Variable	Data Type	Description	Obtained From
distance	Vector of floats	Standardized [1] route distances.	Computed via OSRM API [26].
traffic_level	Array of integers	Mapping categorical traffic level to number (1—jam, 2—high, 3—medium, 4—low).	Provided in raw dataset.
meal_preparation_time	Vector of floats	Standardized [1] meal-preparation times.	Difference between order date and pickup by the courier, both of which were in raw data.
delivery_person_rating	Vector of floats	Standardized [1] rating of delivery person.	Provided in raw dataset.
number_of_deliveries	Array of integers	Number of deliveries.	Provided in raw dataset.

[1] Standardization was conducted in preprocessing step [26].

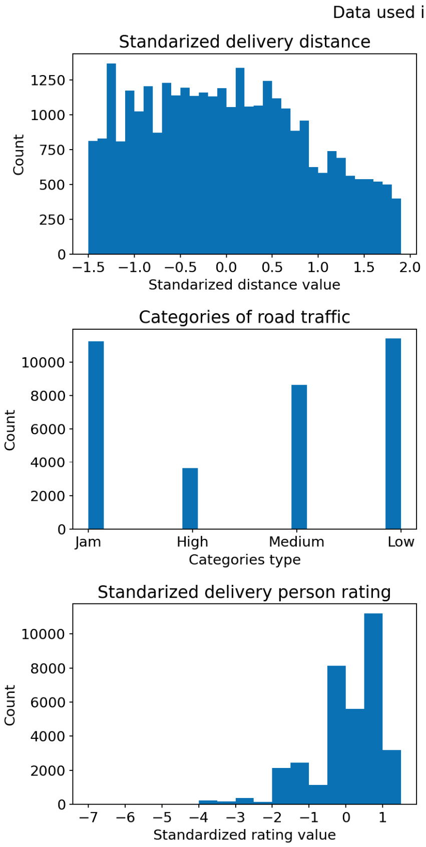


Figure 1. Histograms of data used in inference. Standardization was computed as z-score. X-axis represents value of the predictor and Y-axis is their count for predefined bins. (**top-left**) Standardized distance, which is z-score of distance data received from OSRM API. Raw distances were limited to 30 km. (**top-right**) Standardized meal-preparation time, which is z-score of meal-preparation time. Meal-preparation time was calculated as difference between time the order was received and the time when courier picked up delivery. (**center-left**) Categories of road traffic, which are raw categorical data describing traffic conditions during each delivery. It can be one of four states: low, medium, high, and jam. (**center-right**) Distinct deliveries count, which describes number of deliveries that courier had to make during his trip. (**botom-left**) Standardized delivery-person rating, which is z-score of the delivery-person rating. Original data had rating in range of 2.5 and 5.0 with 0.1 quantization.

3.4. Models

Both models are generalized linear models. We defined the linear predictor as $\eta = X\beta$, where X denotes the vector of features described in Table 1 and B is the vector of coefficients. Each coefficient's distribution is described in the appropriate model section. Both vectors are size Nx1 [22].

We then used the logarithmic link function to transform the linear predictor's domain to positive real numbers. It was one of the possible options, but nevertheless necessary, as both models are defined by the inverse gamma function. This way, we obtained the explanatory variable μ_i, representing the mean of the outcome variable [22]. We defined the prior distribution for the standard deviation of our models, denoted as σ, to be an exponential distribution with a rate parameter equal to 0.5.

Lastly, we defined likelihood as an inverse gamma distribution with parameter shape (α) and scale (β), computed from μ and σ in such a way that the resulting distribution had a mean and standard deviation of μ and σ, respectively. The reasoning behind this particular distribution was to model the skewness of the data effectively. Also, time has to be strictly positive and continuous, which the inverse gamma also provides. The variables are defined in Table 2, and the predictors are defined in Table 1.

Table 2. Models' variables.

Model Variable	Explanation
$delivery\_time_i$	Posterior distribution of delivery time, defined by inverse gamma distribution.
α_i	Shape parameter of inverse gamma function, computed from mean and standard deviation of this distribution.
β_i	Scale parameter of inverse gamma function, computed from mean and standard deviation of this distribution.
σ_i	Prior distribution of standard deviation of model, defined by exponential distribution.
μ_i	Generalized linear model with logarithmic link function. Represents mean of the delivery times as a function of selected predictors.
$mean_i$	Prior distribution of the average meal-delivery scenario. Defined by normal distribution. Acts as an intercept to linear model, since after z-score standardization, the most average case of delivery would yield exp(0), which would invalidate model.
$distance\_coeff_i$	Prior distribution of the distance's linear coefficient. Defined by normal distribution. Used to represent influence of this predictor on the model output.
$meal\_prep\_coeff_i$	Prior distribution of the meal-preparation time's linear coefficient. Defined by normal distribution. Used to represent influence of this predictor on the model output.
$traffic\_level\_coeff_i[j]$ [1]	Prior distributions of the traffic level's linear coefficient. Defined by normal distributions. Used to represent influence of these predictors on the model output.
$person\_rating\_coeff_i$	Prior distribution of the delivery person's rating's linear coefficient. Defined by normal distribution. Used to represent influence of this predictor on the model output.
$deliveries\_number\_coeff_i[j]$ [1]	Prior distributions of the delivery number's linear coefficient. Defined by normal distributions. Used to represent influence of these predictors on the model output.

[1] Here, predictor acts as an index variable.

While the models themselves were defined in Stan, the experiments were conducted via CmdStanPy, which is one of Python's interfaces for it [24]. For each sample from the dataset, the entire inference process was performed. It consisted of 1000 warmup iterations and 1000 regular iterations. The warmup iterations were discarded. Each equation described in the models' sections below was computed according to formulas (denoted by the equal operator), while sampling was performed with HMC (see Section 3.2, (denoted by the tilde operator).

3.4.1. Model 1

The first model is defined as follows:

$$delivery\_time_i \sim InverseGamma(\alpha_i, \beta_i) \tag{14}$$

$$\alpha_i = \frac{\mu_i^2}{\sigma_i^2} + 2 \tag{15}$$

$$\beta_i = \frac{\mu_i^3}{\sigma_i^2} + \mu_i \tag{16}$$

$$\sigma_i \sim Exponential(0.5) \tag{17}$$

$$\begin{aligned}\mu_i = \exp(&distance\_coeff_i \cdot distance_i + traffic\_level\_coeff[traffic\_level_i] + \\ &+ meal\_prep\_coeff_i \cdot meal\_preparation\_time_i + mean_i)\end{aligned} \tag{18}$$

$$mean_i \sim N(3, 0.1) \tag{19}$$

$$distance\_coeff_i \sim Normal(0, 0.3) \tag{20}$$

$$meal\_prep\_coeff_i \sim Normal(0, 0.3) \tag{21}$$

$$traffic\_level\_coeff[1] \sim Normal(0, 0.3) \tag{22}$$

$$traffic\_level\_coeff[2] \sim Normal(0, 0.3) \tag{23}$$

$$traffic\_level\_coeff[3] \sim Normal(0, 0.3) \tag{24}$$

$$traffic\_level\_coeff[4] \sim Normal(0, 0.3) \tag{25}$$

3.4.2. Model 2

The second model is an extension of the first model by two predictors: the number of deliveries and standardized delivery-person rating. As such, we only present the changes necessary to create Model 2 out of Model 1:

$$\begin{aligned}\mu_i = \exp(&distance\_coeff_i \cdot distance_i + traffic\_level\_coeff[traffic\_level_i] + \\ &+ meal\_prep\_coeff_i \cdot meal\_preparation\_time_i + \\ &+ deliveries\_number\_coeff[number\_of\_deliveries_i] + \\ &+ person\_rating\_coeff_i \cdot delivery\_person\_rating_i + mean_i)\end{aligned} \tag{26}$$

$$person\_rating\_coeff_i \sim Normal(0, 0.3) \tag{27}$$

$$deliveries\_number\_coeff[1] \sim Normal(0, 0.3) \tag{28}$$

$$deliveries\_number\_coeff[2] \sim Normal(0, 0.3) \tag{29}$$

$$deliveries\_number\_coeff[3] \sim Normal(0, 0.3) \tag{30}$$

$$deliveries\_number\_coeff[4] \sim Normal(0, 0.3) \tag{31}$$

3.4.3. Priors and Prior Predictive Checks

We decided to use unbounded weakly informative priors for all parameters for two main reasons. First, we lack expert knowledge on the influence of each feature. Second, the abundance of data reduces the influence of priors on the final distribution as more data points are added.

We chose a normal distribution with a mean of 0 and a standard deviation of 0.3 for our parameters. This distribution provides a value range of approximately -1 to 1, with most values likely clustering around 0. This choice reflects our initial assumption that each parameter is not highly influential while still allowing for a small probability that they could be significantly influential. The standard deviation of 0.3 was selected to accommodate the exponential distribution coming from the inverse link function, as larger values from the linear combination could numerically destabilize the model.

The exception to this is the intercept parameter $mean_i$, for which we chose a strong prior: $N(3, 0.1)$. It results in the base mean delivery time around 15–30 min. We opted for a strong prior here to ensure that our model accurately reflects the most average delivery time. Given that our features are standardized with a mean of 0, the linear combination for the most average case would be close to zero, leading to an unrealistic delivery time in the posterior distribution. The $mean_i$ prior helps anchor the model, providing a trusted average time for an average case.

For both models, prior predictive checks gave good results, i.e., the observed data were included within the simulated data range, and no outright impossible values were generated from either of the models. They can be observed in Figures 2–7.

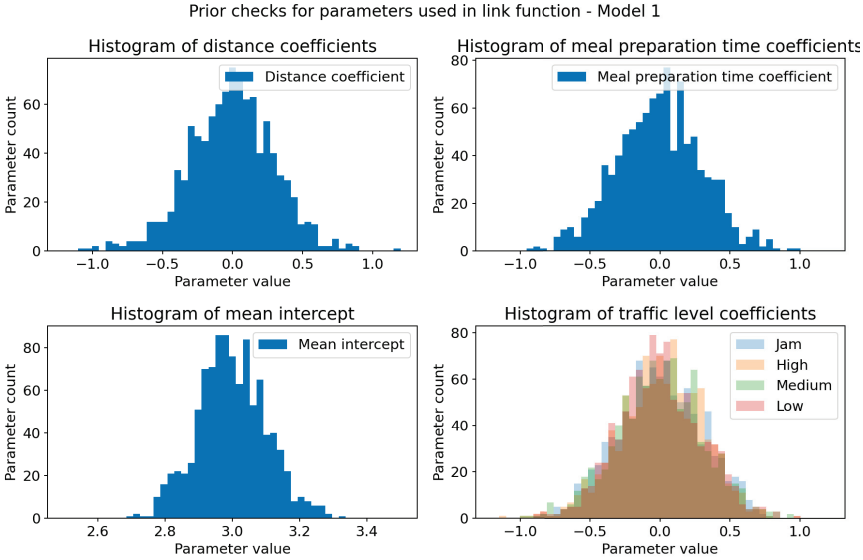


Figure 2. Sampling check for prior distributions of Model 1's link-function parameters (parameters with _coeff suffix). X-axis represents coefficient values and Y-axis represents sample count. Each of the coefficients follows its distribution, which is necessary for prior check to be successful. (**top-left**) Prior distribution of distance coefficient, defined as $Normal(0, 0.3)$. (**top-right**) Prior distribution of meal-preparation-time coefficient, defined as $Normal(0, 0.3)$. (**bottom-left**) Prior distribution of *mean* parameter, defined as $Normal(3, 0.1)$. *mean* parameter represents our belief of what mean delivery time should be in case all other parameters are 0. (**bottom-right**) Joint plot of prior distributions of traffic-level coefficients, all defined as $Normal(0, 0.3)$.

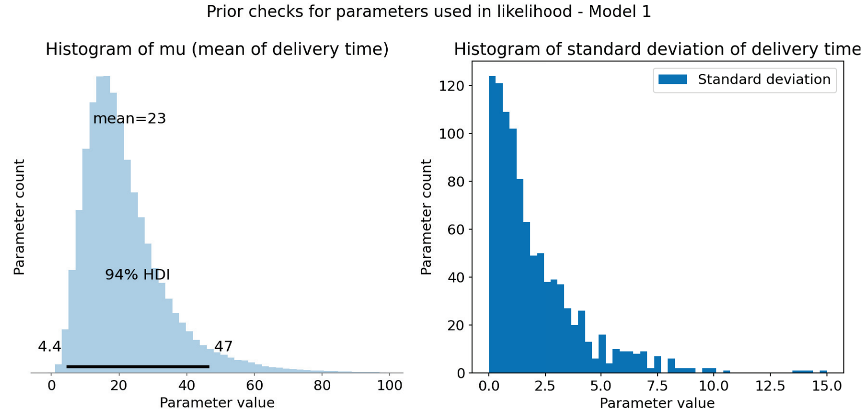


Figure 3. Computation and sample check of Model 1's likelihood parameters. X-axis is time in min and Y-axis represents sample count. (**left**) Computed μ represents mean delivery time for each sample. HDI 94% is represented as black bar at the bottom of the plot and tells us that 94% of shown mean times fall in range of 4.4 to 47 min. Mean of this distribution (at the top of the plot) is 23 min, which is reasonable value. (**right**) Prior distribution of standard deviation of the model, defined as $Exponential(0.5)$.

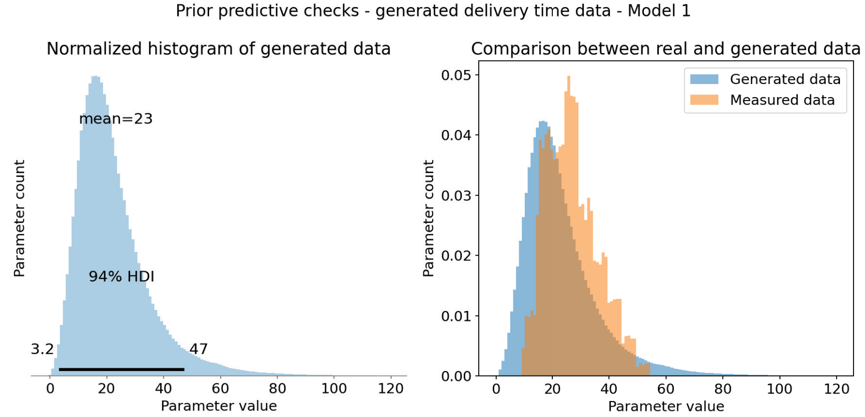


Figure 4. Prior predictive checks—Model 1. (**left**) HDI 94% is represented as black bar at the bottom of the plot and tells us that 94% of shown mean times fall in range of 3.2 to 47 min, which is broad range. Mean of this distribution (at the top of the plot) is 23 min, which is reasonable value. (**right**) Real and simulated data overlay. Both are normalized so that integral of the graph is 1. It was necessary for comparison. Measured data are included within generated data, which means that all observations are possible within prior model. This means that prior checks are successful.

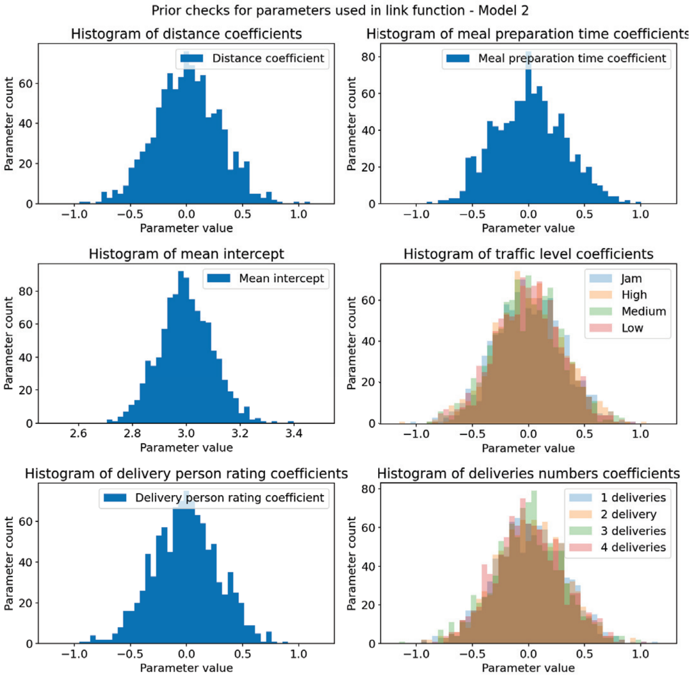


Figure 5. Sampling check for prior distributions of Model 2's link-function parameters (parameters with _coeff suffix). X-axis represents coefficient values and Y-axis represents sample count. Each of the coefficients follows its distribution, which is necessary for prior check to be successful. (**top-left**) Prior distribution of distance coefficient, defined as $Normal(0, 0.3)$. (**top-right**) Prior distribution of meal-preparation-time coefficient, defined as $Normal(0, 0.3)$. (**center-left**) Prior distribution of *mean* parameter, defined as $Normal(3, 0.1)$. *mean* parameter represents our belief of what mean delivery time should be in case all other parameters are 0. (**center-right**) Joint plot of prior distributions of traffic-level coefficients, all defined as $Normal(0, 0.3)$. (**bottom-left**) Prior distribution of delivery-person-rating coefficient, defined as $Normal(0, 0.3)$. (**bottom-right**) Joint plot of prior distributions of deliveries-number coefficients, all defined as $Normal(0, 0.3)$.

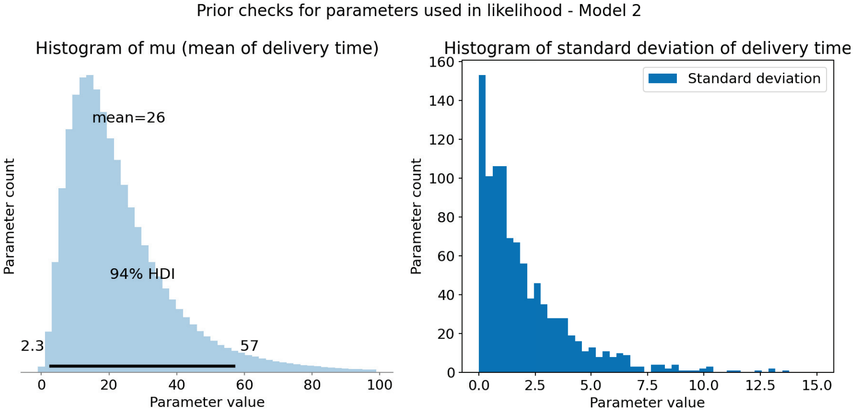


Figure 6. Computation and sample check of Model 2's likelihood parameters. X-axis is time in min and Y-axis is sample count. (**left**) Computed μ represents mean delivery time for each sample. HDI

94% is represented as black bar at the bottom of the plot and tells us that 94% of shown mean times fall in range of 2.3 to 57 min. Mean of this distribution (at the top of the plot) is 26 min, more than for Model 1, but still within reasonable range. (**right**) Prior distribution of standard deviation of the model, defined as $Exponential(0.5)$.

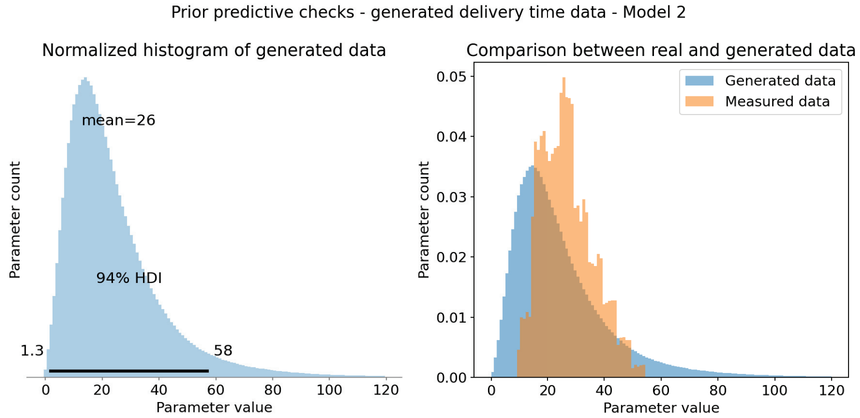


Figure 7. Prior predictive checks—Model 2. (**left**) HDI 94% is represented as black bar at the bottom of the plot and tells us that 94% of shown mean times fall in range of 1.3 to 58 min. It is very broad, improbable range, but for prior checks it is sufficient. Mean of this distribution (at the top of the plot) is 26 min, which is reasonable value. (**right**) Real and simulated data overlay. Both are normalized so that integral of the graph is 1. It was necessary for comparison. Measured data are included within generated data, which means that all observations are possible within prior model. This means that prior checks are successful.

4. Results

In this section, we present the posterior distributions of our models. Each one was trained on full data. There were 1000 warmup and 1000 sampling iterations performed on four parallel chains. The selected algorithm was Hamiltonian Monte Carlo with the engine No-U-Turn Sampler.. For computation Stan 2.34 was used.

Posterior predictive checks were performed by simulating new data from the posterior distribution obtained during model training. These simulations were used to verify if the simulated data resembled the original data, with histograms chosen for comparison purposes [23]. The link-function parameters were subjectively assessed for their numerical influence, considering the z-score standardization of predictors. This standardization implies that the average delivery time corresponds to a predictor value of zero. Narrow distributions indicate a near-constant predictor effect, while wide distributions suggest uncertainty in the predictor's impact. Positive values reflect a direct relationship, and negative values indicate an inverse relationship between the predictor and the outcome. The parameters in the likelihood function were evaluated for their plausibility in real-life scenarios. Since the parameters of our models are coefficients of a linear function, rather than the priors for the data distributions themselves, we believe that the subjective interpretation of them is justified.

4.1. Posterior Predictive Checks for Model 1

Model 1 gave decent results. All of the observed data fall within the samples from the posterior distribution, and visual overlap is quite high. The posterior distribution exhibits a long tail, which is the drawback of using the inverse gamma function. The data are represented in Figure 8.

The model coefficients for the distance and meal-preparation time ended with very narrow distributions, albeit positive ones, indicating that they impact the output variable.

The mean intercept parameter ended with a mean closer to 3.1, which is also closer to the mean of the dataset ($e^{3.1} \approx 22.18$ while the mean of the dataset is ≈27.05). The traffic-level coefficient represents a trend in which low traffic contributes to faster delivery times, and as the traffic level increases, the delivery times become longer. This interpretation is viable as it is not a multiplicand but a sum component, so negative values will result in a smaller mean and positive values in a larger mean. There is almost no distinction between the influence of high traffic and jams. The data are represented in Figure 9.

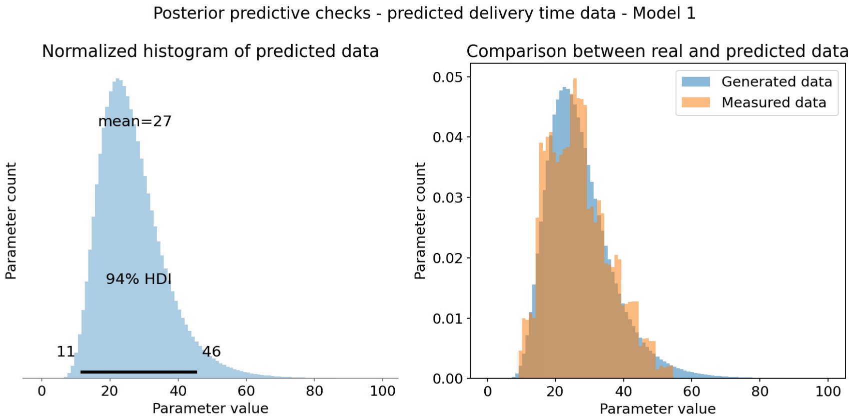


Figure 8. Posterior predictive checks—Model 1. (**left**) HDI 94% is represented as black bar at the bottom of the plot and tells us that 94% of shown mean times fall in range of 11 to 46 min. It is broad range, but realistic nevertheless. Mean of this distribution (at the top of the plot) is 27 min, which is reasonable value. It follows inverse gamma distribution as defined. (**right**) Real and simulated data overlay. Both are normalized so that integral of the graph is 1. It was necessary for comparison. Measured data have high overlap with sampled data from posterior distribution.

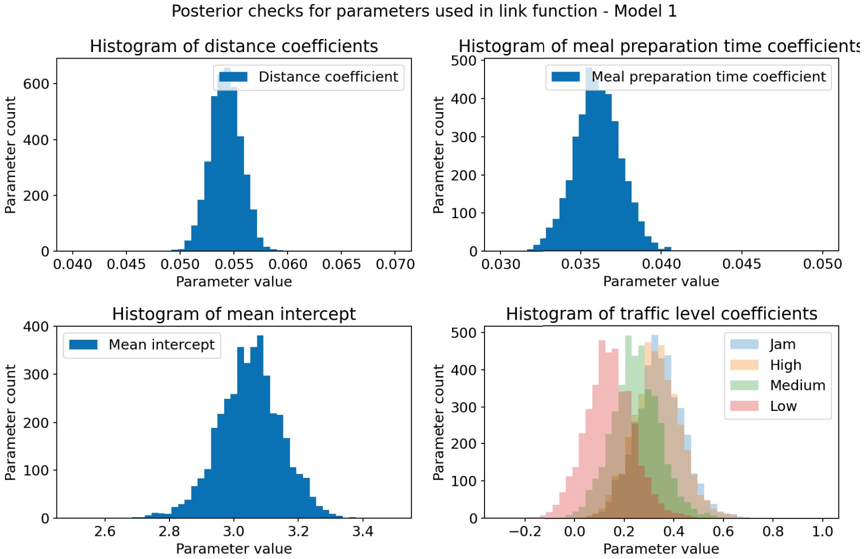


Figure 9. Sampling check for posterior distributions of Model 1's link-function parameters. X-axis represents coefficient values and Y-axis represents sample count. (**top-left**) Posterior distribution of

distance coefficient. It is much narrower than prior distribution, but still follows normal distribution. Positive value indicates that it has impact on the output variable. (**top-right**) Posterior distribution of meal-preparation-time coefficient. Conclusions are the same as for the distance coefficient. (**bottom-left**) Posterior distribution of *mean* parameter. It has mean closer to 3.1, which more likely represents mean of the dataset. (**bottom-right**) Joint plot of posterior distributions of traffic-level coefficients. The bigger the traffic level, the more impact it has on the outcome variable. Jams and high levels have the same impact.

The linear model of mean delivery times μ_i results in probable values with respect to the dataset. The standard deviation completely changed its distribution and now follows a normal distribution centered around 9.5. It is quite close to the std of the dataset, which is ≈8.99. The data are represented in Figure 10.

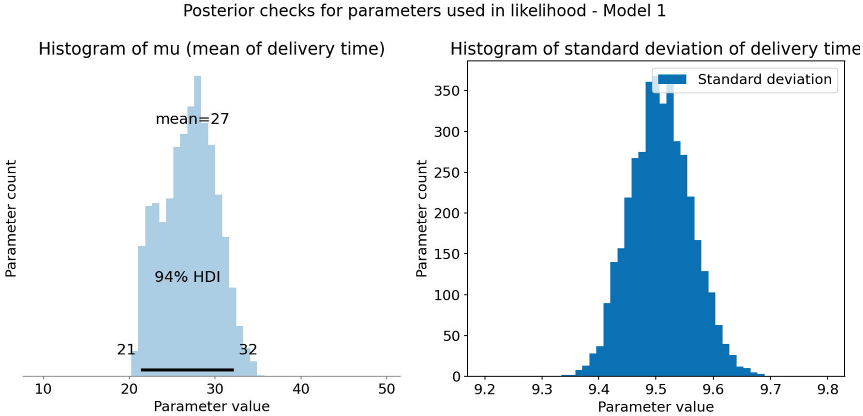


Figure 10. Computation and sample check of Model 1's likelihood parameters. X-axis represents coefficient values and Y-axis represents sample count. (**left**) Computed μ represents mean delivery time for each sample. HDI 94% is represented as black bar at the bottom of the plot and tells us that 94% of shown mean times fall in range of 21 to 32 min. Those are much more realistic values than the ones from prior distribution. Mean of this distribution (at the top of the plot) is 27 min, a reasonable value. (**right**) Posterior distribution of standard deviation of the model; it no longer resembles prior, and now it follows normal distribution with mean ≈9.5.

4.2. Posterior Predictive Checks for Model 2

Model 2 gave visually better results. All of the observed data fall within the samples from the posterior distribution, as with Model 1. The tail is shorter than in Model 1. The data are represented in Figure 11. Since there is not much difference between the posterior distributions for the shared features of both models, we will only comment on new features, distinct to Model 2, as well as on likelihood-related parameters.

The delivery-person rating follows a narrow normal distribution centered around −0.085. Since it is negative and has a small std, we can reason that the delivery-person rating is inversely related to delivery time. This is expected as couriers with higher scores are more likely to deliver food faster. The number of deliveries follows exactly the same trend as traffic-level coefficients, but numerically is more important as the values range is greater. The data are represented in Figure 12.

The linear model of the mean delivery times μ_i has a much longer tail with regard to Model 1, which results in a different 94% HDI. The standard deviation has a similar distribution to Model 1, although its mean value is smaller, around 7.85. The data are represented in Figures 11 and 13.

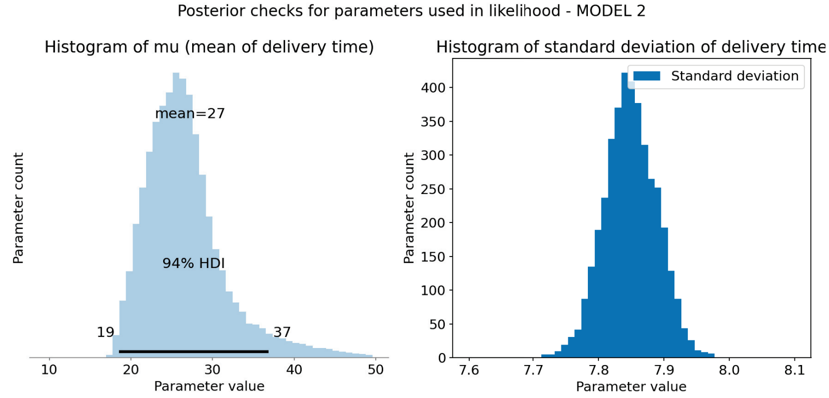


Figure 11. Computation and sample check of Model 2's likelihood parameters. X-axis represents coefficient values and Y-axis represents sample count. (**left**) Computed μ represents mean delivery time for each sample. HDI 94% is represented as black bar at the bottom of the plot and tells us that 94% of shown mean times fall in range of 19 to 37 min. Those are much more realistic values than the ones from prior distribution. Mean of this distribution (at the top of the plot) is 27 min, a reasonable value. (**right**) Posterior distribution of standard deviation of the model; it no longer resembles prior, and now it follows normal distribution with mean ≈7.85.

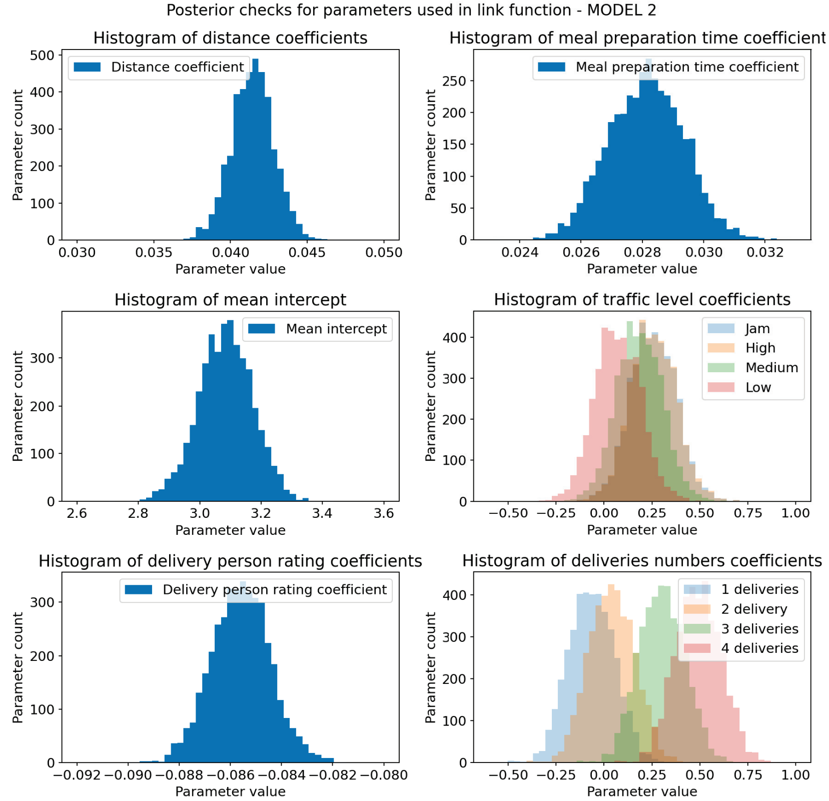


Figure 12. Sampling check for posterior distributions of Model 2's link-function parameters. X-axis represents coefficient values and Y-axis represents sample count. (**top-left**) Posterior distribution of

distance coefficient. It is much narrower than prior distribution, but still follows normal distribution. Positive value indicates that it has impact on the output variable. (**top-right**) Posterior distribution of meal-preparation-time coefficient. Conclusions are the same as for the distance coefficient. (**center-left**) Posterior distribution of *mean* parameter. It has mean closer to 3.1, which more likely represents mean of the dataset. (**center-right**) Joint plot of posterior distributions of the traffic-level coefficients. The bigger the traffic level, the more impact it has on the outcome variable. Jams and high levels have the same impact. (**bottom-left**) Posterior distribution of the delivery-person-rating coefficient. It is much narrower than prior distribution, but still follows normal distribution. Negative values indicate inverse relationship between delivery time and rating; the bigger the courier rating, the faster delivery will be made. (**bottom-right**) Joint plot of posterior distributions of deliveries-number coefficients. The more deliveries, the more impact it has on the outcome variable. This is the same trend as for the traffic level, but greater range translates to greater impact.

4.3. Model Comparison

The models were compared using the WAIC and PSIS-LOO criteria using ArviZ library for an exploratory analysis of Bayesian Models [27].

The WAIC (Widely Applicable or Watanabe–Akaike Information Criterion) is a statistical measure used to estimate the out-of-sample predictive accuracy of a model. It does this by evaluating the within-sample predictive accuracy and making necessary adjustments. The WAIC calculates the log pointwise posterior predictive density (LPPD) and includes a correction for the effective number of parameters to account for overfitting. This correction is performed by subtracting the sum of the posterior variances of the log predictive densities for each data point [28].

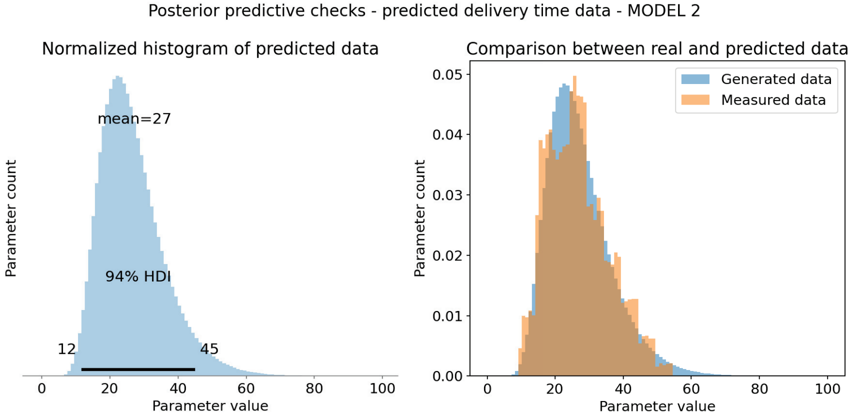


Figure 13. Posterior predictive checks—Model 2. (**left**) HDI 94% is represented as black bar at the bottom of the plot and tells us that 94% of shown mean times fall in range of 12 to 45 min. It is slightly narrower than Model 1 range, but realistic nevertheless. Mean of this distribution (at the top of the plot) is 27 min, which is reasonable value. It follows inverse gamma distribution as defined. (**right**) Real and simulated data overlay. Both are normalized so that integral of the graph is 1. It was necessary for comparison. Measured data have high overlap with sampled data from posterior distribution. Generated data have shorter tail than Model 1, which is desirable.

Model 2 has a higher ELPD score (denoted as waic), which indicates its better within-sample fit. The WAIC also correctly states that it has a higher number of effective parameters (p_waic). The weight parameter clearly states that Model 2 has nearly one probability within the given data. It is slightly more uncertain than Model 1, which is indicated by the SE (standard error) parameter, but when compared to differences in the WAIC score and size of the dataset, it is not overly large. Overall, the WAIC clearly evaluated Model 2 as superior. It is presented in the Table 3 and Figure 14.

Table 3. Comparison results with WAIC.

Model	Rank	waic	p_waic	d_waic	Weight	SE	dSE
2	0	−117,249.732164	14.307179	0.000000	0.997408	145.013769	0.000000
1	1	−122,442.256308	9.314501	5192.524145	0.002592	137.412164	88.510163

Parameters description [29]: Model: indicates whether it is Model 1 or Model 2. Rank: rank of the models, with 0 indicating the best model. waic: ELPD score, where higher values suggest better out-of-sample predictive fit. p_waic: estimated effective number of model parameters. d_waic: difference between ELPD scores, relative to the best model. Weight: relative weight of each model, interpreted as the probability of the model given the data. SE: standard error of the ELPD estimate. dSE: standard error of the difference in ELPD between each model and the top-ranked model.

The PSIS-LOO (Pareto Smoothed Importance Sampling using Leave-One-Out validation) method is used to calculate the out-of-sample predictive fit by summing the log leave-one-out predictive densities. These densities are evaluated using importance ratios (IS-LOO). However, the importance ratios can exhibit high or infinite variance, leading to instability in the estimates. To mitigate this issue, a generalized Pareto distribution is fitted to the largest 20% of the importance ratios [28].

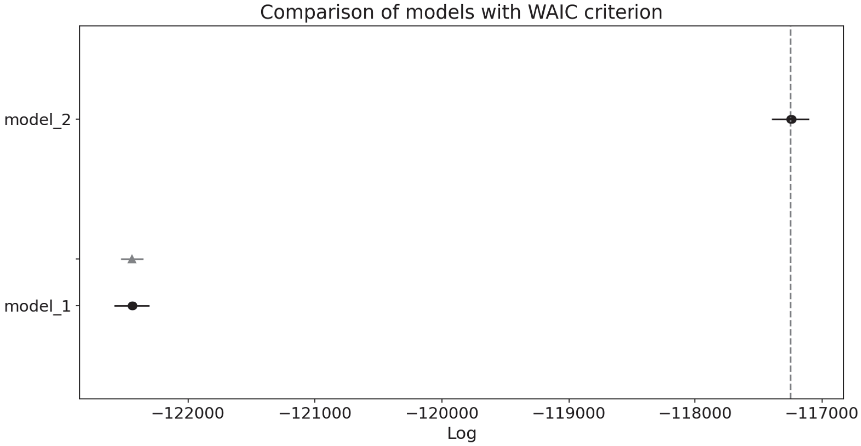


Figure 14. Comparison plot for WAIC. Black dots indicate ELPD of each model with their standard error (black lines). Grey triangle represents standard error of difference in ELPD between Model 1 and top-ranked Model 2. Plot indicates that Model 2 performs better with a dashed line.

The PSIS-LOO evaluation provides near identical results as the WAIC, and the same conclusions as above can be drawn. It is presented in the Table 4 and Figure 15. An alternative approach for comparison is of course statistical significance testing, using, for instance, Kolmogorov–Smirnov-based tests (see, for example, [30]). This approach is very popular in applications that rely on frequentist statistics. Our work, however, is based on a Bayesian paradigm, where classical statistical significance or hypothesis testing is not as interpretable; this is why we decided to not include them in this paper.

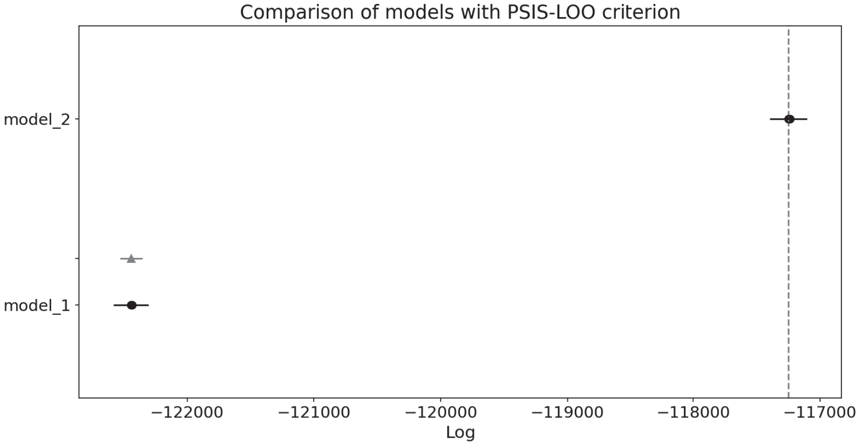


Figure 15. Comparison plot for PSIS-LOO criterion. Black dots indicate ELPD of each model with their standard error (black lines). Grey triangle represents standard error of difference in ELPD between Model 1 and top-ranked Model 2. Plot indicates that Model 2 performs better with a dashed line.

Table 4. Comparison result with PSIS-LOO criterion.

Model	Rank	loo	p_loo	d_loo	Weight	SE	dSE
2	0	−117,249.732784	14.307799	0.000000	0.997408	145.013813	0.000000
1	1	−122,442.256329	9.314522	5192.523545	0.002592	137.412164	88.510227

Parameters description [29]: Model: indicates whether it is Model 1 or Model 2. Rank: rank of the models, with 0 indicating the best model. loo: ELPD score, where higher values suggest better out-of-sample predictive fit. p_loo: estimated effective number of model parameters. d_loo: difference between ELPD scores, relative to the best model. Weight: relative weight of each model, interpreted as the probability of the model given the data. SE: standard error of the ELPD estimate. dSE: standard error of the difference in ELPD between each model and the top-ranked model.

5. Discussion

One of the potential limitations is scaling models to a larger dataset. MCMC methods are computationally expensive and time consuming. For the presented data, it took around 7 h on an Intel Core i7-11370H 3.30 GHz chip to run the inference. It also produced approx. 3.7 GB of data per model.

It is important to recognize that the methodology of Bayesian inference itself has inherent limitations. Bayesian inference is optimal when the assumed model is correct. However, since models are never perfect and only approximate reality, every model introduces implicit limitations. Additionally, models are influenced by the subjective choice of priors. Very diffuse or uniform priors can lead to overcertainty in estimates, while strong priors that do not accurately represent the true probability distribution of the data can result in poor model generalization. Furthermore, all models require subjective interpretation and decisions by researchers, which, if not properly justified, can be a significant shortcoming [23].

6. Conclusions

In this paper, we explored the application of Bayesian inference for predicting food-delivery times, a novel approach not previously employed for this specific task to the best of our knowledge. Our results indicate significant potential in this methodology, particularly with Model 2, which, as an extension of Model 1, demonstrated a superior performance.

A major advantage of our approach is its ability to capture model uncertainty and provide interpretability, as well as to assess the impact of predictors, thereby offering insights into areas of improvement for food-delivery companies. We hope that our findings empower future exploration of Bayesian methods. For the field of time-travel estimation, this work could serve as a baseline for researchers to build upon and refine. In other areas, we aim to demonstrate an accessible approach and encourage scientists to experiment with it.

Future research should aim to test these models on data from more reputable sources. This endeavor may prove challenging, as food-delivery data are often proprietary and not publicly accessible. Additionally, it is crucial to validate the models with out-of-sample datasets to ensure robustness. Expert input on prior selection should also be considered. Finally, the business applications of these findings merit consideration, both for historical data analysis and real-time implementation. Further work could involve developing additional models with diverse datasets to explore predictor relationships. This deeper understanding could potentially enhance feature selection in machine learning models, which are commonly used for the ETA task.

Author Contributions: Conceptualization, J.P., J.G., and J.B.; methodology, J.P., J.G., and J.B.; software, J.P. and J.G.; validation, J.P. and J.B.; formal analysis, J.P.; investigation, J.P.; resources, J.B.; data curation, J.P. and J.G.; writing—original draft preparation, J.P.; writing—review and editing, J.P. and J.B.; visualization, J.P.; supervision, J.B.; project administration, J.B.; funding acquisition, J.B. All authors have read and agreed to the published version of the manuscript.

Funding: The first author's work was supported by AGH's Research University Excellence Initiative under the project "Research Education Track". The third author's work was partially realized in the scope of a project titled "Process Fault Prediction and Detection". This project was financed by The National Science Centre on the base of decision no. UMO-2021/41/B/ST7/03851. Part of this work was funded by AGH's Research University Excellence Initiative under the project "DUDU—Diagnostyka Uszkodzeń i Degradacji Urządzeń".

Data Availability Statement: All code prepared as part of this project is available in the repository https://github.com/JohnnyBeet/Food-delivery-time-prediction/tree/modelling. (accessed on 25 August 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

OFD	Online food delivery
GPS	Global Positioning System
PSIS-LOO	Pareto Smoothed Importance Sampling using Leave-One-Out validation
WAIC	Watanabe–Akaike Information Criterion
DNN	Deep Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
ETA	Estimated Time of Arrival
OFCT	Order Fulfillment Cycle Time
OD	Origin–destination
OSRM	Open-Source Routing Machine
HDI	Highest-Density Interval
LPPD	Log pointwise posterior predictive density
ELPD	Expected Log pointwise Predictive Density
SE	Standard error
dSE	Standard error of the difference in ELPD between each model
MCMC	Markov Chain Monte Carlo
HMC	Hamiltonian Monte Carlo

References

1. Statista. Online Food Delivery—Worldwide. 2024. Available online: https://www.statista.com/outlook/emo/online-food-delivery/worldwide (accessed on 4 May 2024).
2. IMARC Group. India Online Food Delivery Market Report. 2023. Available online: https://www.imarcgroup.com/india-online-food-delivery-market (accessed on 4 May 2024).
3. Alalwan, A.A. Mobile food ordering apps: An empirical study of the factors affecting customer e-satisfaction and continued intention to reuse. *Int. J. Inf. Manag.* **2020**, *36*, 28–44. [CrossRef]
4. Xue, G.; Wang, Z.; Wang, Y. The restaurant delivery problem with uncertain cooking time and travel time. *Comput. Ind. Eng.* **2024**, *190*, 110039. [CrossRef]
5. Ulmer, M.W.; Thomas, B.W.; Campbell, A.M.; Woyak, N. The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. *Transp. Sci.* **2021**, *55*, 75–100. [CrossRef]
6. Han, J.; Liu, H.; Liu, S.; Chen, X.; Tan, N.; Chai, H.; Xiong, H. iETA: A Robust and Scalable Incremental Learning Framework for Time-of-Arrival Estimation. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, 6–10 August 2023; pp. 4100–4111.
7. Gao, C.; Zhang, F.; Wu, G.; Hu, Q.; Ru, Q.; Hao, J.; He, R.; Sun, Z. A deep learning method for route and time prediction in food delivery service. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Singapore, 14–18 August 2021; pp. 2879–2889.
8. Abdi, A.; Amrit, C. A review of travel and arrival-time prediction methods on road networks: Classification, challenges and opportunities. *PeerJ Comput. Sci.* **2021**, *37*, e689. [CrossRef] [PubMed]
9. Lee, W.H.; Tseng, S.S.; Tsai, S.H. A knowledge based real-time travel time prediction system for urban network. *Expert Syst. Appl.* **2009**, *36*, 4239–4247. [CrossRef]
10. Li, X.; Cong, G.; Sun, A.; Cheng, Y. Learning travel time distributions with deep generative model. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 1017–1027.
11. Asghari, M.; Emrich, T.; Demiryurek, U.; Shahabi, C. Probabilistic estimation of link travel times in dynamic road networks. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2015, Bellevue, WA, USA, 3–6 November 2015; pp. 1–10.
12. Wang, Y.; Zheng, Y.; Xue, Y. Travel time estimation of a path using sparse trajectories. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 25–34.
13. Wang, Z.; Fu, K.; Ye, J. Learning to estimate the travel time. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2018; pp. 858–866.
14. Zhu, L.; Yu, W.; Zhou, K.; Wang, X.; Feng, W.; Wang, P.; Chen, N.; Lee, P. Order fulfillment cycle time estimation for on-demand food delivery. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, 6–10 July 2020; pp. 2571–2580.
15. Li, Y.; Fu, K.; Wang, Z.; Shahabi, C.; Ye, J.; Liu, Y. Multi-task representation learning for travel time estimation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, San Francisco, CA, USA, 26–29 August 2018; pp. 1695–1704.
16. Wu, C.H.; Ho, J.M.; Lee, D.T. Travel-time prediction with support vector regression. *IEEE Trans. Intell. Transp. Syst.* **2004**, *5*, 276–281. [CrossRef]
17. Wang, H.; Tang, X.; Kuo, Y.H.; Kifer, D.; Li, Z. A simple baseline for travel time estimation using large-scale trip data. *ACM Trans. Intell. Syst. Technol. TIST* **2019**, *10*, 1–22. [CrossRef]
18. Lin, Y.; Wan, H.; Hu, J.; Guo, S.; Yang, B.; Lin, Y.; Jensen, C.S. Origin-destination travel time oracle for map-based services. *Proc. ACM Manag. Data* **2023**, *1*, 1–27. [CrossRef]
19. Zhou, X.; Wang, J.; Liu, Y.; Wu, X.; Shen, Z.; Leung, C. Inductive graph transformer for delivery time estimation. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, New York, NY, USA, 27 February 2023; pp. 679–687.
20. Zhang, L.; Zhou, X.; Zeng, Z.; Cao, Y.; Xu, Y.; Wang, M.; Wu, X.; Liu, Y.; Cui, L.; Shen, Z. Delivery time prediction using large-scale graph structure learning based on quantile regression. In Proceedings of the 2023 IEEE 39th International Conference on Data Engineering (ICDE), Anaheim, CA, USA, 3–7 April 2023; pp. 3403–3416.
21. Hildebrandt, F.D.; Ulmer, M.W. Supervised learning for arrival time estimations in restaurant meal delivery. *Transp. Sci.* **2022**, *56*, 1058–1084. [CrossRef]
22. Gelman, A.; Carlin, J.B.; Stern, H.S.; Dunson, D.B.; Vehtari, A.; Rubin, D.B. *Bayesian Data Analysis*, 3rd ed.; Chapman and Hall/CRC: New York, USA, 2013;
23. van de Schoot, R.; Depaoli, S.; King, R.; Kramer, B.; Märtens, K.; Tadesse, M.G.; Vannucci, M.; Gelman, A.; Veen, D.; Willemsen, J.; et al. Bayesian statistics and modelling. *Nat. Rev. Methods Prim.* **2021**, *1*, 1. [CrossRef]
24. Team, S.D. Stan Modeling Language Users Guide and Reference Manual, 2.34. Available online: https://mc-stan.org/docs/reference-manual/mcmc.html. (accessed on 21 July 2024).
25. Food Delivery Dataset. 2023. Available online: https://www.kaggle.com/datasets/gauravmalik26/food-delivery-dataset (accessed on 13 May 2024).
26. Gibas, J.; Pomykacz, J.; Baranowski, J. Bayesian modelling of travel times on the example of food delivery: Part 1—Spatial data analysis and processing. *Electronics* **2024**, *accepted*.

27. ArviZ. API Reference. 2024. Available online: https://python.arviz.org/en/stable/api/index.html (accessed on 20 May 2024).
28. Vehtari, A.; Gelman, A.; Gabry, J. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Stat. Comput.* **2017**, *27*, 1413–1432. [CrossRef]
29. ArviZ. Arviz.Compare. 2024. Available online: https://python.arviz.org/en/stable/api/generated/arviz.compare.html (accessed on 20 May 2024).
30. Hassani, H.; Silva, E.S. A Kolmogorov-Smirnov Based Test for Comparing the Predictive Accuracy of Two Sets of Forecasts. *Econometrics* **2015**, *3*, 590–609. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

Article

Bayesian Modeling of Travel Times on the Example of Food Delivery: Part 1—Spatial Data Analysis and Processing

Justyna Gibas, Jan Pomykacz and Jerzy Baranowski *

Department of Automatic Control & Robotics, AGH University of Krakow, 30-059 Krakow, Poland; justynagibas@student.agh.edu.pl (J.G.); janpomykacz@student.agh.edu.pl (J.P.)
* Correspondence: jb@agh.edu.pl

Abstract: Online food delivery services are rapidly growing in popularity, making customer satisfaction critical for company success in a competitive market. Accurate delivery time predictions are key to ensuring high customer satisfaction. While various methods for travel time estimation exist, effective data analysis and processing are often overlooked. This paper addresses this gap by leveraging spatial data analysis and preprocessing techniques to enhance the data quality used in Bayesian models for predicting food delivery times. We utilized the OSRM API to generate routes that accurately reflect real-world conditions. Next, we visualized these routes using various techniques to identify and examine suspicious results. Our analysis of route distribution identified two groups of outliers, leading us to establish an appropriate boundary for maximum route distance to be used in future Bayesian modeling. A total 3% of the data were classified as outliers, and 15% of the samples contained invalid data. The spatial analysis revealed that these outliers were primarily deliveries to the outskirts or beyond the city limits. Spatial analysis shows that the Indian OFD market has similar trends to the Chinese and English markets and is concentrated in densely populated areas. By refining the data quality through these methods, we aim to improve the accuracy of delivery time predictions, ultimately enhancing customer satisfaction.

Keywords: food delivery services; travel time estimation; spatial analysis; data preprocessing; Bayesian modeling

Citation: Gibas, J.; Pomykacz, J.; Baranowski, J. Bayesian Modeling of Travel Times on the Example of Food Delivery: Part 1—Spatial Data Analysis and Processing. *Electronics* **2024**, *13*, 3387. https://doi.org/10.3390/electronics13173387

Academic Editors: Chao Zhang, Wentao Li, Huiyan Zhang and Tao Zhan

Received: 20 June 2024
Revised: 16 August 2024
Accepted: 17 August 2024
Published: 26 August 2024

1. Introduction

The e-commerce sector is in a constant state of growth and evolution, particularly within its subdomain of online food delivery (OFD) [1,2]. Recent market forecasts indicate a steady rise in revenue for companies offering such services. With numerous players in the market, ensuring customer satisfaction is paramount for a company's survival. Customers increasingly demand user-friendly applications that simplify the ordering process with just a few taps, while also providing features such as delivery time estimates and communication channels with couriers [3]. However, estimating delivery times accurately without real-time data presents a significant challenge. While companies can track courier positions via GPS, they lack access to real-time information on factors such as traffic, accidents, and roadworks.

Before any model, algorithm, or computational technique can be applied, the initial step involves finding and preparing data. Effective data analysis and processing play a pivotal role in determining the performance and outcomes of models, given that many techniques rely on clean and comprehensive data. Nonetheless, real-world data seldom meet these ideal conditions and are often challenging to interpret. Even though Bayesian models, discussed further in Part 2 of this article [4], are less affected by noisy data and outliers, they still require data processing and reasonable assumptions.

Visualization of data is equally significant. It offers valuable insights, particularly when combined with expertise in the relevant field. For instance, plotting geospatial data on maps can reveal outliers, patterns, or anomalies. Proper visualization is also essential to draw appropriate conclusions and evaluate model outputs [5].

This article leverages spatial data analysis and standard data preprocessing techniques to enhance the data quality utilized in Bayesian models aimed at predicting food delivery times. By examining the spatial distribution of online food deliveries, we identify and rectify faulty data points. It also gives us insight into the Indian food delivery market and customer behavior, which, to the best of our knowledge, has not yet been explored. Drawing from existing literature, we identify the most crucial features for our models. Additionally, we investigate the relationships between variables in our dataset to uncover any patterns that could impact the behavior of our models.

The contributions of this paper can be summarized as follows: (1) By leveraging the OSRM API, we effectively transformed raw data into meaningful predictors, which are essential for the implementation of Bayesian models. (2) Standardizing the data proved to be a crucial step for numerical stability in the models without compromising their interpretability. (3) Visualization provided valuable insights into outliers, enabling us to justifiably exclude them from the training data. (4) Route analysis provides insight into the Indian food delivery market and the ability to compare with other countries (mainly England and China).

The remainder of this paper is organized as follows: In the next section, the relevant studies in the extant literature are reviewed and discussed. Section 3 provide overview of the used data and describe data preprocessing and visualization methods in detail. In Section 4, we perform in-depth analysis of the processing results and spatial analysis of routes. Finally, in Section 5, we make a conclusion and describe possible future work.

2. Literature Review

2.1. Distance and Travel Time Estimation

Typically, OFD platforms acquire customers' GPS coordinates when they open the app to place an order [6]. Then, their location is used for restaurant recommendations and to specify the delivery destination. Subsequently, customer GPS coordinates can be used to predict travel time with an origin–destination based approach. Although it allows for estimating time without knowing the exact route and distance, a significant issue in this particular method is that it requires an exact departure time, which is usually unknown. Moreover, most of the research using origin–destination (OD)-based methods have focused on other applications such as taxi trip duration [7].

Our problem demands that we provide customers with accurate waiting times without relying on knowledge of the precise preparation time [8] or the courier's departure. Moreover, variables used in Bayesian models need to be obtainable from the probabilistic distributions. To reduce complexity, we transformed raw timestamps into meal preparation times, aligning the predictor variable's units with the output variable (minutes). Meal preparation times likely follow well-defined distributions (e.g., normal, gamma, uniform), whereas event timestamps may exhibit complex distributions with peaks corresponding to breakfast, lunch, or dinner order times. We believe that this decision increases the explainability of the models. As for the geographic coordinates, we could try representing them as probability distributions. However, a challenge would likely arise when interpreting the outcomes of our models, since coordinates alone convey very little information about possible routes or the time required to travel them. Instead, we could choose to transform the coordinates into approximate shortest route distances. This approach would provide a clear dependency: the longer the route, the longer it takes to travel. Similarly to transforming timestamps into meal preparation time, we believe that transforming geographical coordinates into route distances increases the explainability and interpretability of our models.

Joshi et al. articulated a compelling argument advocating for the consideration of road distance over geometric distances like the Haversine distance for accurate delivery time estimation. They contend that relying solely on geometric distances can result in oversimplified representations, potentially leading to unrealistic data inputs into predictive models. Their approach does not entail route generation but rather involves mapping GPS coordinates onto a road network map of the analyzed area. They mapped the city network to the graph and assumed that the weight of each edge is he average travel time across all delivery vehicles in the corresponding road. Their study was focused on effective batching of orders and assignment of orders to vehicles [9]. Ji et al. adopted a method for estimating travel time that utilizes GPS trajectories aligned with specific road network segments. They used GPS trajectories of food carriers and removed stay points. Then, they mapped GPS trajectories to road segments and obtained the travel time of carriers on each road segment (which was included in their dataset). Their research, however, is tailored towards optimizing the grouping of tasks related to efficient delivery operations [10]. Ulmer et al. assumed that all drivers follow a mobile navigation device to determine the best paths. They approximated road distance by multiplying the Euclidean distance by a factor of 1.4. It is demonstrated that this approach closely approximates the relationship between Euclidean and street distances. They chose such a method as they were attempting to dynamically control a fleet of drivers and had so many potential paths that they exceeded the limitations of commercial mapping services [11]

Alternatively, some researchers have turned to popular routing tools like Google Maps API, Baidu Maps API, and the TOM-TOM API [12–17]. Yet, the utilization of these services incurs exorbitant expenses for OFD companies, often surpassing a staggering one million dollars annually. It should be noted that these costs are particularly high for large companies, where the number of orders is the primary factor influencing the price of the service. Companies attempt to reduce these costs by caching a set of routes or utilizing historical delivery data [18]. As a cost-effective solution, exploration into open-source routing services like OSRM (Open-Source Routing Machine) has been initiated [19].

2.2. Spatial Analysis

Spatial analysis related to food outlets and OFD platforms has also grown in popularity, as researchers are more interested in factors that influence customer behavior. Most of the spatial approaches are concentrated on the distribution of food outlets, food accessibility, and its impact on diet and health. Most frequently used methodologies in spatial analysis are statistics and GIS [20].

Another objective of spatial analysis in the food delivery context is to explore the factors influencing the utilization of these services, particularly in relation to built infrastructure. Typically, densely populated areas are examined [15,21] because online food outlets tend to be more prevalent in urban regions [22]. Recent studies have examined the distribution of different types of food outlet (e.g., fine dining, fast food), which can affect the usage of OFD platforms [23]. Regardless of the research area, the periphery has the most limited access to food outlets and OFD platforms cannot improve this [15,22–24]. Even in studies that are more focused on food delivery optimization, spatial analysis of the restaurants and customers' distribution play crucial roles [25].

Spatial analysis of deliveries or journeys is not as common and typically focuses on exploring the connections between distinct regions [10,26]. In the realm of region segmentation, researchers often employ two primary methods: the grid-based approach and the road network-based approach. The grid-based method is particularly useful for visualizing smaller areas, such as those in Wang et al.'s article [26]. The road network-based approach is claimed to be more informative; yet, in densely populated urban areas, this approach may yield excessively small regions, necessitating a merging step prior to further analysis [10].

3. Materials and Methods

3.1. Data

In this study, we employed the Food Delivery Dataset, which is presently accessible on Kaggle [27]. Initially, the dataset was made available by HackerEarth for their machine learning competition. It encompasses more than 45,000 deliveries spanning 21 cities across India. The data span a three-month period, encompassing February through April of 2023. The dataset includes the location of restaurants and delivery destinations as well as other nonspatial information such as weather, traffic conditions, and time and date the order was placed. Table 1 provides a comprehensive list of all variables under consideration, along with their respective meanings.

Table 1. Data available in the dataset and their description.

Variable name	Meaning
Restaurant_latitude	Latitude of the restaurant
Restaurant_longitude	Longitude of the restaurant
Delivery_location_latitude	Latitude of the delivery destination
Delivery_location_longitude	Longitude of the delivery destination
Road_traffic_density	Road traffic intensity (Low, Medium, High or Jam)
Weatherconditions	Current weather conditions (e.g., Sunny, Stormy, Fog)
multiple_deliveries	Quantity of simultaneous deliveries (number 0–4)
Delivery_person_Ratings	Average rating of the courier
Delivery_person_Age	Age of the courier
Order_Date	Date of placing the order
Time_Orderd	Time of placing the order
Time_Order_picked	Time of picking up by courier
Time_taken	Delivery time in minutes

3.2. Routes Generation

While the exact trajectory of the courier remains uncertain and unpredictable, approximating the route is essential for our purposes. This will not only ensure an appropriate level of reality reproduction, but will also allow to detect incorrect data. Routing engines are not able to generate routes between geographical coordinates that cannot be connected by a road network, thus implicitly detecting improper origin–destination pairs. To achieve this, we have opted to leverage the OSRM API [28], an open-source routing engine. By constructing tailored queries containing GPS coordinates for both the starting and ending points of the route, we can utilize the API to generate optimal route suggestions [29]. Furthermore, customization options such as transportation mode (e.g., car) and route type (e.g., shortest route) allow for further refinement of our queries. After receiving the routing request, the OSRM API passes it on to the OSRM routing engine, which employs the OpenStreetMap (OSM) [30] data to produce the optimal route according to the specified parameters. OpenStreetMap is an open geographic database that is updated and maintained by a community. It can be treated as an alternative to Google Maps. The routing engine considers multiple factors, including road types, speed limits, and turn restrictions, to guarantee an accurate and efficient route. The output of the routing request contains information about the route, including the route geometry, distance, and estimated travel time. If there is an issue processing the request, the API will provide an error code (e.g., "InvalidUrl") indicating the reason behind the failure. In our case, there were three types of possible errors: "NoSegment", "TooBig", and "TooManyRequests". The first error indicates that one of the supplied input coordinates could not snap to the street segment. The second scenario applies if the request size violates one of the service size restrictions. The last expected error appears when the server is overloaded.

3.3. Preprocessing

Preparing data as needed by the model is an essential part of delivery time estimation. Figure 1 shows the preprocessing steps, which are described in detail in the following.

Data cleaning is initial part of preprocessing. It ensures that missing values will not lead to poor results and wrong conclusions. In our approach, we decided to remove rows with missing data and rows where restaurant or destination GPS coordinates are outside the geographical boundaries of India. Approximately 15% of the samples were removed in this process. In the case of the removed geographical coordinates, they were logically incorrect (points in countries other than India, points in the ocean), which in later steps would lead to a huge number of errors when using the OSRM API.

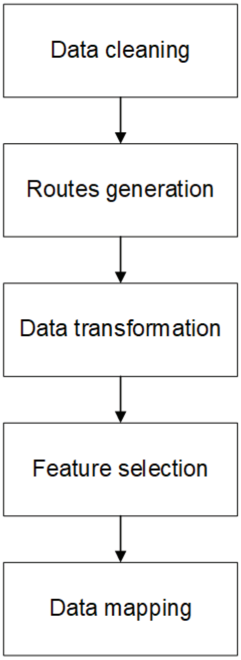


Figure 1. Preprocessing steps include the following: deletion of incomplete or out-of-India samples, generating routes via OSRM API, converting the date and time into a sustainable format, selecting predictors with the greatest information value, and mapping data to format appropriate for Bayesian model.

Routes generation is performed using OSRM API, which was described in Section 3.2. The created OSRM client uses asynchronous HTTP to find the shortest route between restaurant and delivery destination. If response is successful, distance and route geometry are saved; otherwise, the unsuccessful request and its reason are logged for future analysis. Out of the suspected error types, we only encountered “TooManyRequests”. It was most likely caused by too many people trying to use the server at once. To overcome this issue, we logged those failed routes and fetched them again when the service was more responsive. As for the other two types of errors, it would be difficult to correct them; so, the data related to such problematic routes would be excluded from modeling. Distance obtained from routing API is integrated with rest of dataset as an additional column.

Data transformation techniques are used to convert data into a sustainable format. Date and time are especially difficult to analyze due to various factors. To overcome this, meal preparation time is calculated based on Time_Order and Time_Order_picked. One variable represents time of placing the order and another time of picking up by courier. Their difference will determine the meal preparation time. This will not correspond to an exact time, as the prepared dish may be waiting for the courier to arrive. This will enable the model to be utilized in cases where the restaurant furnishes an estimated preparation duration for the order, or where the statistics of preparation time can be derived from historical data [8].

Feature selection is one of the ways to cope with dimensionality. The goal is to remove irrelevant and redundant features, which may include accidental correlations in models and reduce their generalization abilities. Feature selection also decreases the risk of overfitting and reduces the search space, making the learning process faster and less memory consuming [31].

Feature scaling is a critical step in constructing effective models as it helps mitigate bias stemming from variations in the ranges and magnitudes of data. Among the most widely used techniques are normalization and standardization. We have chosen to implement standardization for selected features with a continuous distribution. This decision ensures that operations and results of the model will be more straightforward to interpret.

Mapping categorical data is essential to use them as input for a model. It is crucial to note that, unlike in machine learning models, for Bayesian models created using the Stan library [32], data must be mapped starting from 1 upwards (as vectors are indexed from 1).

3.4. Visualization

As mentioned previously, visualization is also an important part of preprocessing. Plotting the data can reveal outliers or anomalies that cannot be easily identified in other ways. In this research, we will perform visual analysis of the generated routes followed by analysis of distributions of input data of the models.

The main focus of the route analysis is spatial visualization. We will explore various visualization methods, including heatmaps, interactive maps, and road network graph maps.

For the route analysis, each of the 21 cities is treated separately, allowing for a more detailed examination. The allocation of routes to each city was accomplished through clustering. We utilized the KMeans algorithm provided by scikit-learn [33], with clustering based on the locations of restaurants and delivery destinations.

To implement interactive maps, we used Folium [34], which is a wrapper for the Leaflet.js library. Folium allows for the creation of interactive Leaflet maps and supports a wide range of overlay formats, such as images, videos, and GeoJSON, enabling the embedding of multiple layers. For our maps, we used OSM as the base layer and added routes based on the geometry returned by the OSRM API. Additionally, routes from each city were plotted on separate layers for clarity. Each city has been added as a separate layer that can be activated on the map. The main advantage of this approach is the ability to analyze small areas of the city without compromising image quality. Moreover, routes are embedded on the OSM, which gives the opportunity to check infrastructure near the starting and ending points of the route. This may also reveal anomalies, e.g., if the staring point of the route is not near the restaurant.

Heatmaps are particularly useful for analyzing large datasets or densely located points. Our goal is to depict road usage with an intensity map similar to Navarro's approach [35]. Our approach was to divide each city into a grid of squares of selected length. Each grid would be represented by a pixel on an image. If there was a route point in such a grid, then grid value was incremented by one, thus changing color of the pixel and effectively creating a heatmap. However, the raw route geometry data are unevenly distributed, as illustrated in Figure 2. Points are clustered near intersections and turns, whereas on straight sections, the routes are sparsely spaced. To achieve a meaningful scale, we need to interpolate points on these straight sections. We implement linear interpolation for each route segment where the distance between consecutive points exceeds a selected threshold (e.g., 5 m). Linear interpolation is a popular method; however, it has high error when the distance between interpolated points is too large. This is particularly noticeable in the curved road segment [36].

Other widely used methods include cubic interpolation, neighbor-based interpolation (also known as distance-based interpolation), and spherical linear interpolation. Cubic interpolation uses the current value and gradient vector to estimate intermediate points. This method has very low error rates but requires significantly more computational time, often up to ten times longer [37]. Neighbor-based interpolation determines the interpolated value by considering the surrounding points, calculating a weighted average of the nearby observations or using the nearest observation values. Spherical linear interpolation is particularly useful over long distances where the curvature of the Earth must be taken into account. In our case, we try to interpolate data on straight road segments, and the distance between two consecutive coordinates is relatively small, usually smaller than 100 m. We find that linear interpolation is sufficiently precise for this task.

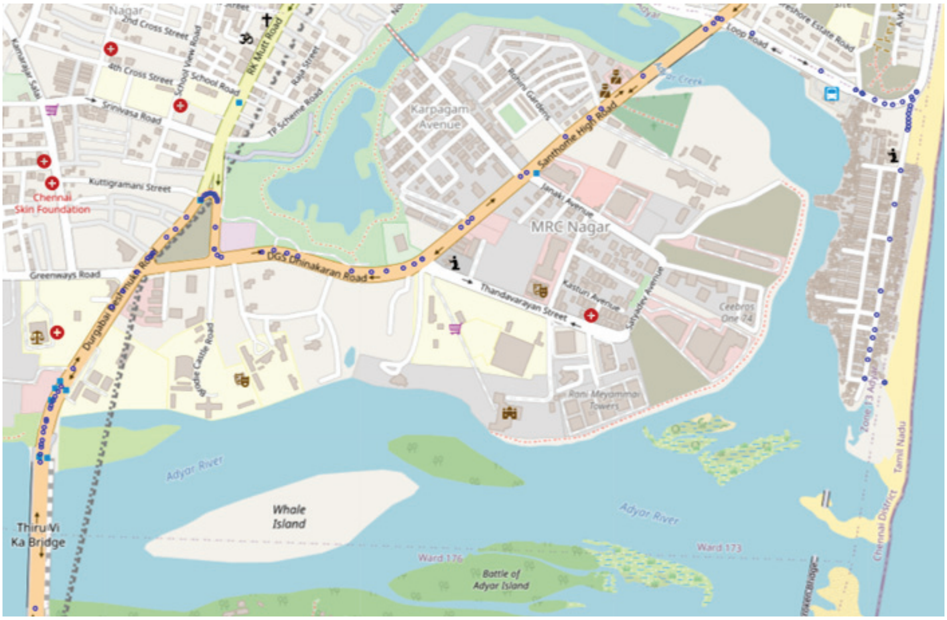


Figure 2. Raw distribution of sample route geometry. GPS coordinates are clustered near intersections and turns, whereas on straight sections, they are sparsely spaced. Considering raw distribution of route geometry data for some visualization method interpolation is necessary.

Another method we considered was visualizing routes on road network graphs. This required generating a graph representing the road network. We utilized the OSMnx library [38], which provides tools to model, analyze, and visualize street networks and other geospatial features from OSM. The generated graph, along with the routes obtained from the OSRM API, needed to be converted into a common format and then plotted as a high-resolution image. In contrast to interactive maps, differences in road segments' intensity will be more visible. Additionally, there will be no unnecessary elements like different icon types.

In order to minimize computation time and ensure the appropriate level of map detail, we recommend generating a graph only for the city area where the analyzed routes occur. In our case, the maximum and minimum values of longitude and latitude that occurred on routes in a given city were selected as the limit values. No information about any route will be lost, but four points of all routes are located exactly on the border of the image. This will not influence analysis of routes distribution.

3.5. Models Overview

For better understanding of the importance of preprocessing steps and conclusions, a short overview of the proposed Bayesian models is in order. Bayesian inference is a method of statistical inference, in which we fit a predefined probability model to a set of data and evaluate outcomes with regards to observed parameters of the model and unobserved quantities, like predictions for new data points [39].

Both models are generalized linear models. We have defined linear predictor as $\eta = X\beta$, where X denotes vector of features and β is vector of coefficients. Both vectors are size N × 1 [39]. We then used a logarithmic link function to transform the linear predictor's domain to positive real numbers. It is necessary step, as both models are defined by inverse gamma function. This particular distribution effectively models skewness of the data and provides strictly positive continuous outputs. An in-depth description of models and explanation of Bayesian inference is presented in Part 2 of this article [4].

4. Results

4.1. Preprocessing Results

Preprocessing was conducted as outlined in Section 3.3. During data transformation, we discovered instances where deliveries, from the moment the courier picked up the order to its delivery, were recorded with zero or negative times. We detected them by comparing the calculated meal preparation time and total delivery time. Consequently, additional data cleaning was necessary, as data used in modeling need to be obtainable from the distributions used. Samples with such travel times were excluded from modeling. Negative travel times can destabilize models and make it impossible to obtain reasonable results. A possible cause for these anomalies is that not all of the data used in calculations were initially recorded and missing values were added manually to the system.

To estimate delivery times, companies and researchers commonly utilize a variety of features. These include spatial features (such as the location of the restaurant and destination, and the city road map), cooking time features, order features (like the number of items ordered and the date and time), and courier features (such as workload) [8,9,40–42]. Our first model utilizes the distance between the restaurant and delivery destination, meal preparation time, and traffic density. These features are used, respectively, in 42%, 9%, and 12% of the researches [43]. The second model is extended by courier features: the number of simultaneous deliveries and courier rating, which are also crucial in predicting food delivery times [8]. In our dataset, traffic density refers to road traffic intensity; however, the exact measurement methodology is not provided. A detailed analysis of the input data will be presented in Section 4.1.2.

One of the most critical steps in our preprocessing flow was obtaining real world distances. This allowed us to construct a well-interpreted model. Utilization of standardization assured the numerical stability of our models, which would otherwise be difficult to achieve. Last but not least, careful consideration of the features used in modeling was equally important. Our second model, which took into account two additional variables, required over two hours more to perform sampling.

4.1.1. Routes Generation

We used OSRM API to generate approximate delivery routes based on demo server. The parameters and their values in our requests are as follows: service—route, version—v1, profile—driving, overview—false, geometries—geojson, and steps—true. GeoJSON is a standardized format for encoding geographic data structures. The steps parameter is used to return information related to each part of the route. All approximate distances were successfully generated, with no incorrect routes identified between the restaurant and delivery destination. The histogram of the obtained distances is shown in the accompanying Figure 3, and their statistics are presented in Table 2. The histogram and statistics reveal that the distances in our dataset are significantly larger than those in other studies. In research conducted by Wang and He, the 95th percentile for distance is 6 km [15], whereas in our data, it is approximately 28.5 km. The shape of the distance distribution in their study is similar to ours, except for two peaks in our distribution. However, they analyzed customer behavior only in Shenzhen, which is a metropolis with a unique population composition and many subcenters. It has a very different dynamic compared to the Indian cities included in this study.

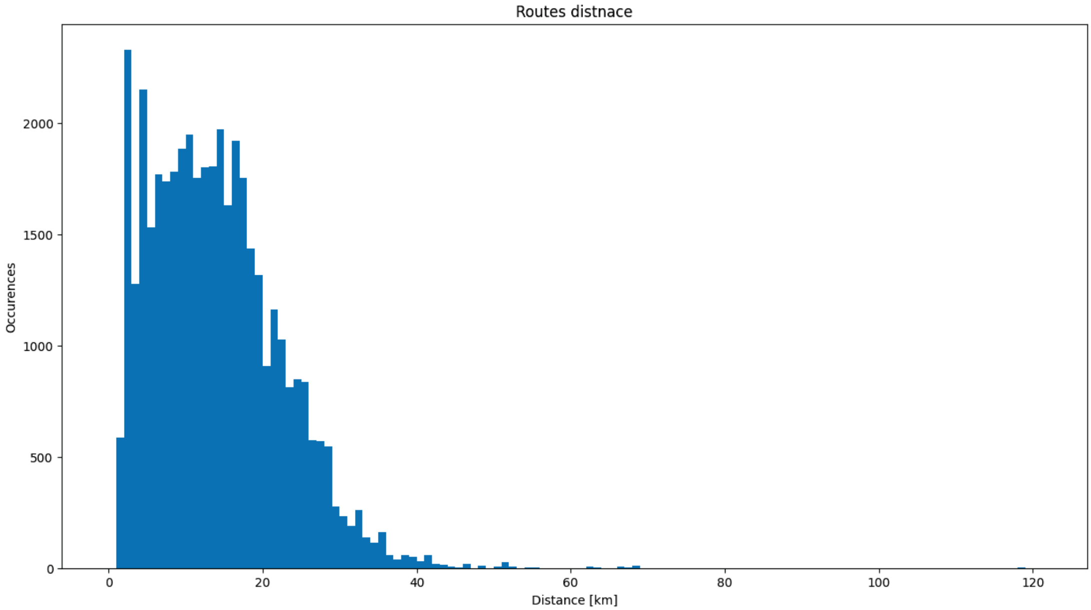


Figure 3. Histogram of total distance of generated routes; each bin represents 1 km interval. Distribution of route distances is skewed; however, in Bayesian modeling, such distributions are not problematic. Models can handle various data distributions including joint distributions. Most of the deliveries have a distance under 25 km. Two peaks in the distribution correspond to distances of 2–3 km and 4–5 km. Previous research shows that most orders are carried out within a distance of 2–3 km [15]; however, it analyzes only one city. Our data include deliveries from cities that vary in size and number of inhabitants; therefore, the first peak may correspond to high-density cities and the second to smaller ones. Moreover, two outlying groups can be identified (around 65 km and 120 km).

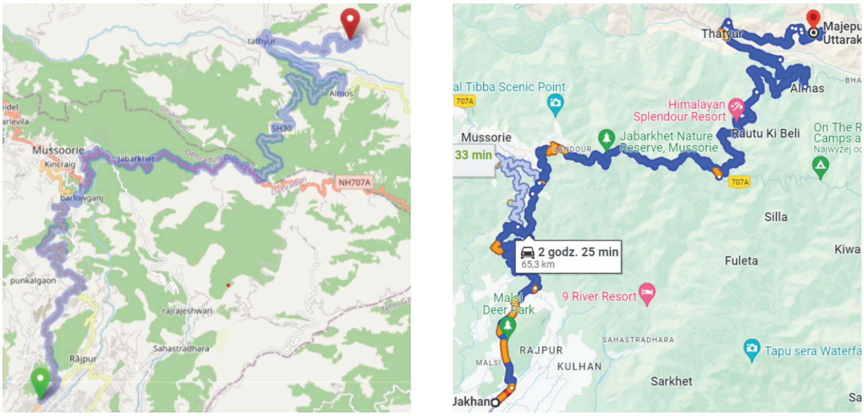
Table 2. Statistics of routes generated using the OSRM API.

Total number of routes	41522
Average route distance	13.99 km
Standard deviation of route distance	8.42 km
Minimum route distance	1.49 km
95th percentile	28.53 km
99th percentile	36.42 km
Maximum route distance	121.89 km
Distance intervals with the highest number of routes	2–3 km (2331) 4–5 km (2153)

The number of deliveries significantly drops when the distance exceeds 25 km. However, there are routes with distances that are considerably longer. The histogram indicates two distinct groups: one around 65 km and another around 120 km. Both of these groups fall above the 99th percentile, classifying them as outliers. To understand the reasons behind these outliers, we conducted a thorough investigation and compared their routes with other routing engines—particularly Google Maps, which employs different mapping techniques. We use websites dedicated to individual services (https://www.google.pl/maps, https://www.openstreetmap.org (accessed on 16 August 2024)) and check each pair of geographical coordinates that are in the group of outliers.

First, we checked if both engines returned routes with similar distances and paths. This way, we excluded incorrect mappings from the OSRM engine or erroneous point mapping in OSM. Next, we analyzed the surroundings of the starting and ending points. We verified whether there was actually a restaurant near the restaurant locations and whether the delivery location was in the vicinity of buildings. We identified unrealistic delivery locations; we defined these as locations where delivery is highly improbable, such as points on highways and bridges or points far from any buildings. In the future, the analysis of the surroundings could be automated using OSM, which provides the ability to check if selected infrastructure objects (e.g., shops and restaurants) are within a chosen radius of a point.

All routes with road distances around 65 km are concentrated in the Dehradun area. Dehradun is situated in a valley at the foot of the Himalayas, resulting in a limited number of roads leading to the surrounding areas. A comparison of the selected route determined by the OSRM API and Google Maps is shown in Figure 4. Both engines visually identified the same route; however, Google Maps indicates that the route is approximately 2 km longer. This discrepancy may stem from differences in point positioning on various maps and the distinct distance calculation algorithms used. While the exact route lengths also vary among engines utilizing only OSM data, these differences are negligible over shorter distances.



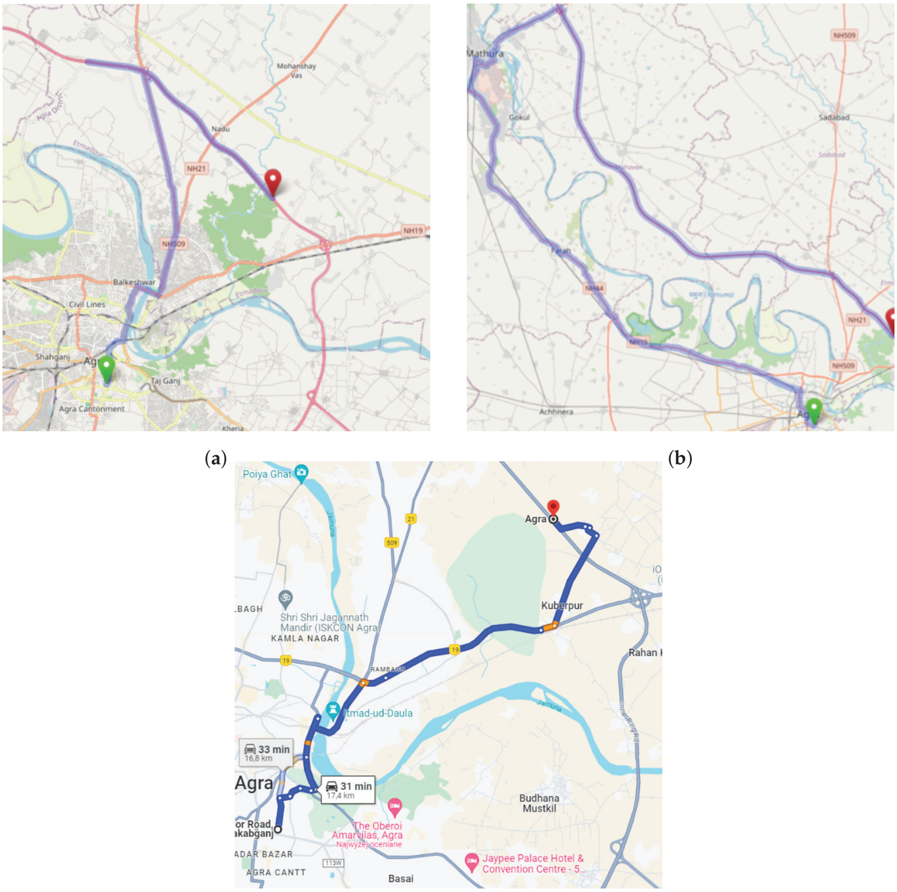
(**a**) (**b**)

Figure 4. Comparison of routes determined using different routing engines. (**a**) Route obtained using OSRM on OSM. (**b**) Route obtained using Google Maps. Both services generate approximately the same route (difference in distance compared to the length of the entire route is negligible).

The second problematic group of routes is found in the city of Agra. Analyzing this case is particularly challenging because online routing engines display varying results depending on the day of access. Additionally, Google Maps often selects different destinations for the same location—sometimes directing to an expressway and other times to a parallel street. The OSM data used by OSRM indicate that the given location is situated in one of the lanes on the nearby Yamuna Expressway. This suggests that the routes allowing direct U-turns on this type of road are incorrectly designated. A comparison of different routes for the same locations is illustrated in Figure 5.

Discrepancies between routes generated using OSRM API and Google Maps may be caused by differences in road network representation. Both services use graphs; however, one is a commercial tool and the other is open source. Therefore, there may be slight differences in the positions of features on these maps. There are also limitations to location accuracy. We have noticed that entered coordinates change their values once routing engines start processing. The differences are negligible but may affect the results obtained. Additionally, maps may add some noise to location data due to localization privacy policies.

The discrepancies in the routing engine do not impact the relevance of this paper. Although using different routing engines might produce varying routes, resulting in different distributions in our models, the models proposed in Part 2 of this article focus on the applicability of Bayesian modeling for delivery time prediction. Consequently, the variation in the distribution of one predictor would not affect the study's findings.



(**a**) (**b**)

(**c**)

Figure 5. Comparison of routes results. (**a**) Route obtained using OSRM on OSM, which involves direct U-turn on Yamuna Expressway (obtained 18 May 2024). (**b**) Route obtained using OSRM on OSM (obtained 16 April 2024). (**c**) Route obtained using Google Maps that maps the location on road near Yamuna Expressway (obtained 18 May 2024). The lack of repeatability in the obtained routes may result from the lack of a direct point corresponding to the coordinate from the dataset.

To ensure our models are trained on meaningful data, we decided to filter out the outliers. We set a 30 km upper limit for deliveries, which encompasses over 96% of the data available after preprocessing. Orders that take up to 45 min are usually accepted [25], although too long a delivery time significantly reduces the freshness and quality of the meal. Therefore, we assume that deliveries over 30 km cannot meet this requirement. Standardization was performed after applying this distance filter to prevent excessive data spread caused by biases in the mean and standard deviation.

4.1.2. Input Data Analysis

Following the described preprocessing and outlier filtering, we ended up with nearly 35,000 data samples. The data prepared for Bayesian models are presented in Figure 6.

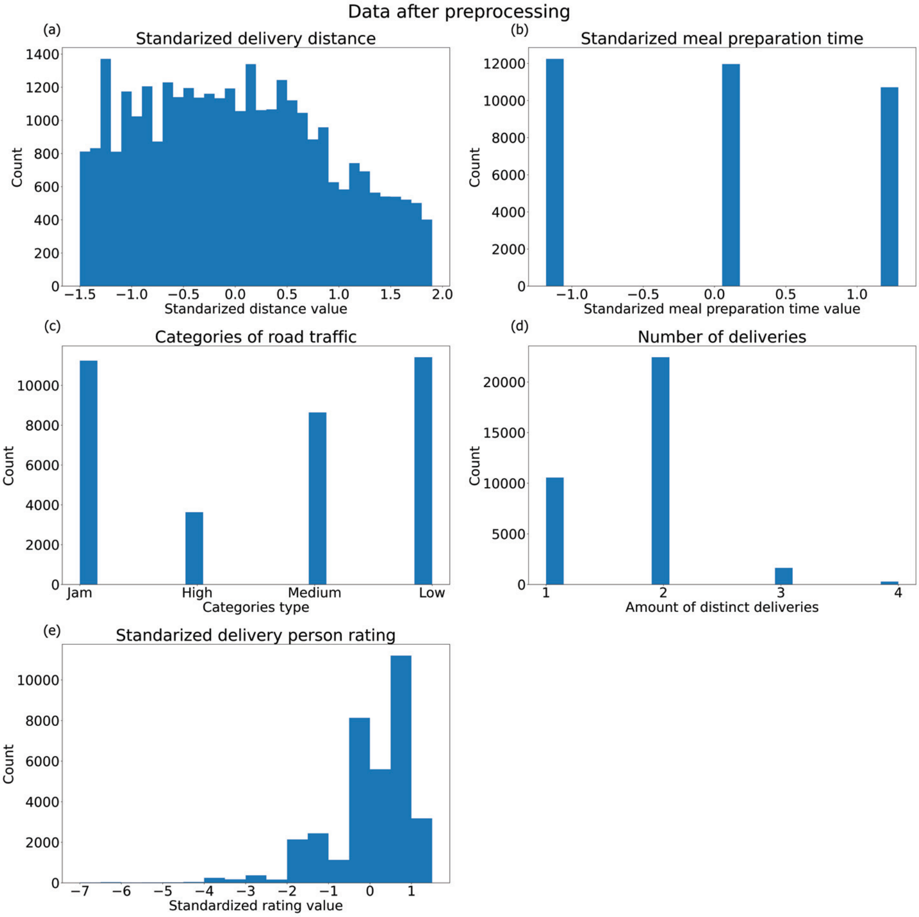


Figure 6. Histograms of data used in inference. (**a**) Standardized distance, bins defined as <−1.5;2> with steps of 0.1. (**b**) Standardized meal preparation time, 20 bins equally spaced, automatically defined by program. (**c**) Categories of road traffic, from highest to lowest. (**d**) Distinct deliveries count. (**e**) Standardized delivery person rating, bins defined as <−7;2> with steps of 0.5.

The basic distance statistics remained relatively stable after filtration. The average distance decreased to 13.33 km, while the standard deviation reduced to 7.17 km. This outcome was expected, given that the filtered values constituted only about 3% of the dataset eligible for model utilization.

The distribution of order preparation time presents an intriguing puzzle. Initially perceived as a continuous variable, the values are distributed almost evenly across 5, 10, and 15 min intervals. This phenomenon could stem from the provision of approximate order and pick-up times rather than precise values. Additionally, calculated meal preparation time may not reflect to exact cooking time. On the other hand, time required to prepare the food is related to the restaurant type. Expected times for fast, fast casual, and gourmet restaurants are 5, 10, and 15 min [11]. This will allow the type of restaurant to be indirectly taken into account. However, in case of higher cooking times, it may lead to a significant increase in the expected delivery time. Nevertheless, there are no anomalies in the dataset because the meal preparation time is assumed to be between 5 and 15 min [25].

The prevalence of high ratings among couriers has led to an inflated average of 4.6 in the standardization process. Furthermore, the remarkably small standard deviation of 0.32 translates to substantially reduced standardized values for couriers with lower ratings. Notably, no courier has a rating below 2.5.

The categorical variables "traffic density" and "number of deliveries" appear to align with expectations. Couriers typically handle no more than two deliveries concurrently. Interestingly, deliveries occur with equal frequency during rush hours and periods of very low car traffic. However, deliveries in moderate traffic conditions are relatively less common.

4.2. Spatial Analysis

The aim of the spatial analysis was to analyze the frequency of use of road segments. Orders are not evenly distributed among all 21 cities. The analysis included all those located in India, and a route for them could be determined (including data considered as outliers). Jaipur is the clear leader in terms of the number of orders, with over 3400 deliveries located there. Eight cities have very similar values at the level of 3150 deliveries. These include the largest cities in India such as Bangalore and Mumbai. Cities in which the previously calculated outliers occurred have a significantly lower number of deliveries in their area. A chart showing the number of orders for each city is shown in Figure 7. Discrepancies in orders among different cities are expected, as previous studies conducted in China and England have shown that densely populated areas have higher OFD platform usage [15,22]. Limited representations of smaller cities may lead to inaccurate delivery time estimation in these cities. On the other hand, it may improve the generalizability of the model.

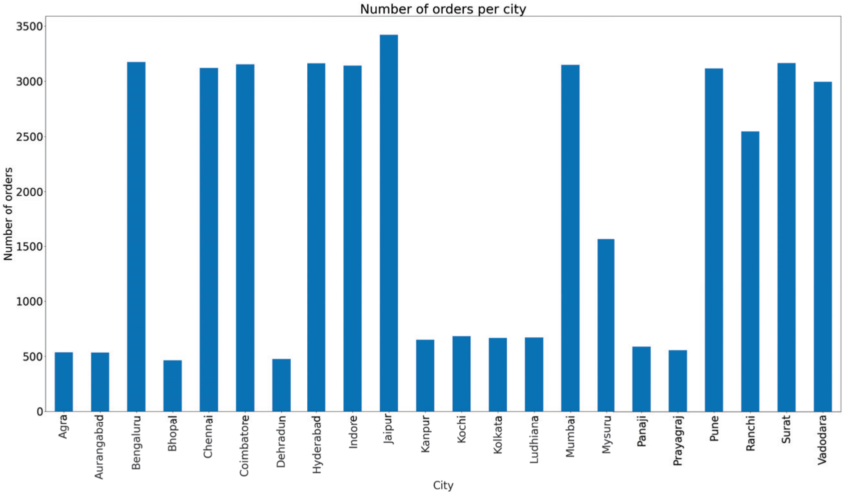
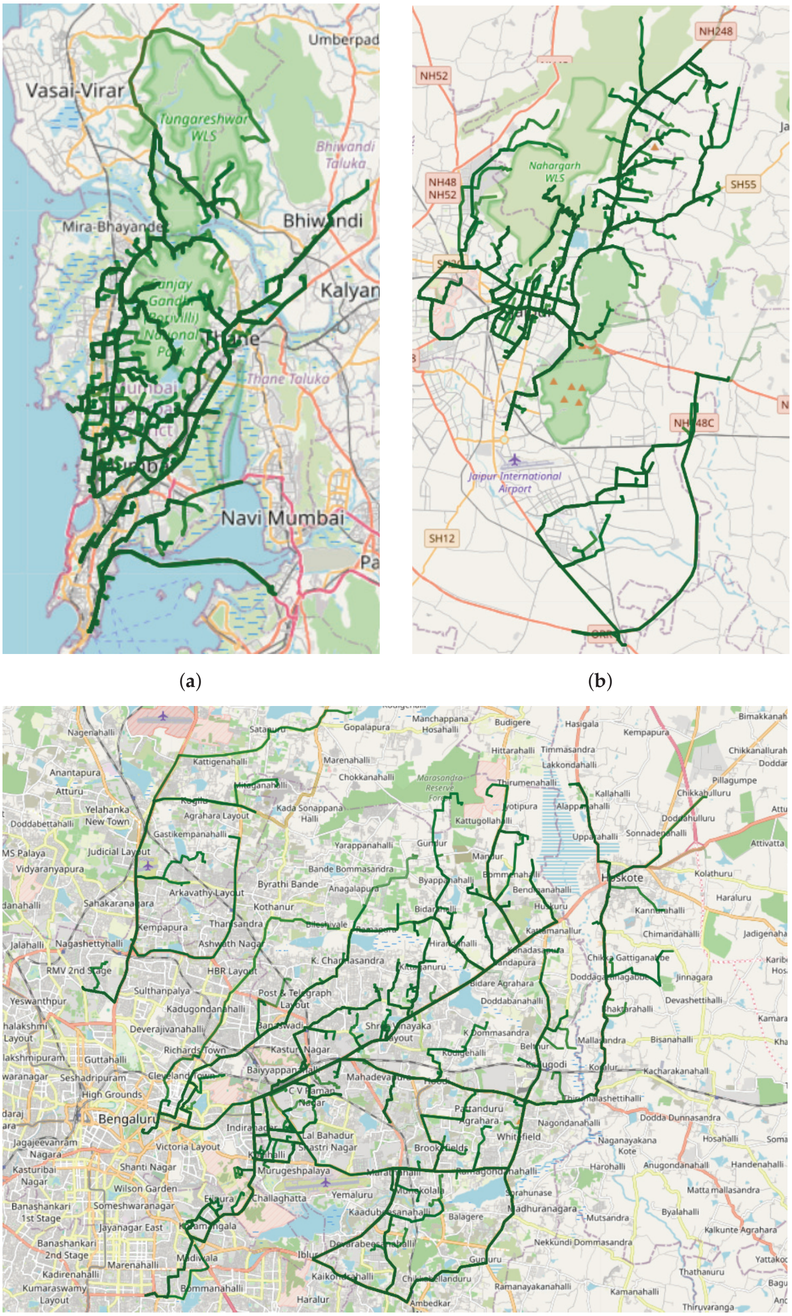


Figure 7. Number of orders in each city. Larger Indian cities have a higher number of orders while smaller cities there have even 6 times fewer orders. This confirms that ordering food online is a typical urban phenomenon.

Interactive maps created using Folium present completed routes on a road map with OSM. The intensity of a given road fragment depends on how often it occurs on routes. Additionally, for readability and easier analysis, the routes from each city have been added as a separate layer that can be activated. Sample visualization results are shown in Figure 8. The main advantage of this approach is the ability to analyze small areas of the city without compromising image quality. It help us to identify outliers that were related to unreal delivery locations, e.g., deliveries to places without nearby buildings.



(**a**)

(**b**)

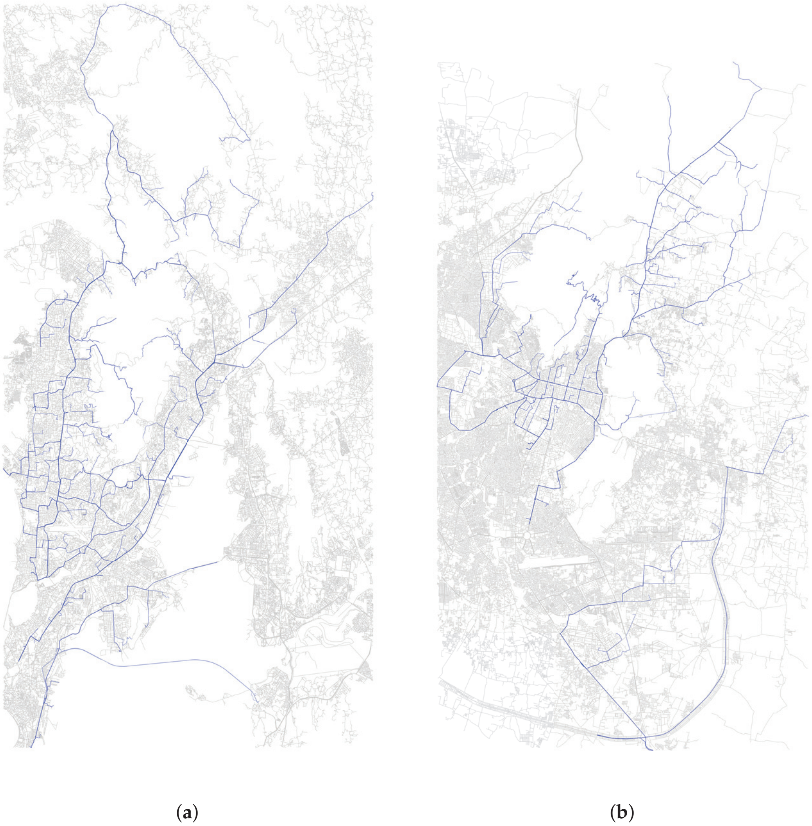
(**c**)

Figure 8. Route visualizations using Folium for (**a**) Mumbai, (**b**) Jaipur, and (**c**) Bangalore. Sections of roads that were heavily trafficked with deliveries have a more intense color, while sections that have been traveled once or twice are much less visible.

It can be noticed that the roads in city centers are definitely the most frequently traveled. The size of the marker is not related to the results because it is automatically adjusted to the scale of the map. The closer the area, the thinner the drawn routes. Routes that have been traveled once or twice are barely visible. Such an example is noticeable at the left edge of the Jaipur image.

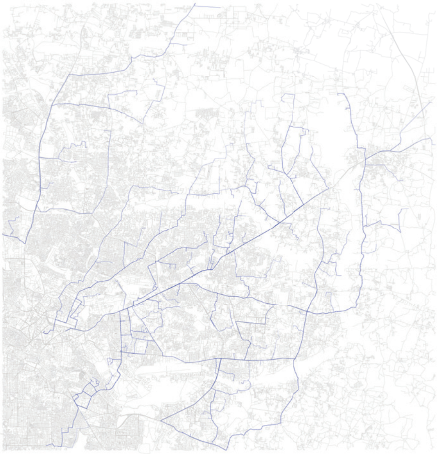
The visualization result using a city graph for similar cities is shown in Figure 9. Similarly to the previous maps, the intensity of a road fragment depends on its frequency in the routes. In general, in large-scale images generated using OSMnx, differences in saturation levels are not so easily noticeable. Due to the dense network of streets in the centers where deliveries are concentrated, all routes appear to be seldom traveled. However, after a small zoom, they are much more visible than on maps generated using Folium.

The very visible difference in the intensity of bridges on the Mumbai map made using different methods interesting, so we decided to analyze it more closely. As it turned out, there was another road running in the immediate vicinity of the bridge, which also had a route that—at a sufficiently large distance—was displayed directly on top of itself, disturbing the results. Moreover, we were dealing with a route that made no sense because it ended in the middle of a bridge (Google maps even projected the point onto the waters of the bay) and it was filtered as an outlier (it was about 40 km long).



(**a**) (**b**)

Figure 9. *Cont.*



(**c**)

Figure 9. Route visualizations of routes on a city street graph made in OSMnx for (**a**) Mumbai, (**b**) Jaipur, and (**c**) Bangalore. Sections of roads that were heavily trafficked with deliveries have a more intense color, while sections that were traveled once or twice are much less visible.

Despite utilizing linear interpolation to increase the number of points on routes, we did not achieve satisfactory results with our heatmap visualizations. We were unable to determine a suitable grid length. When we selected a grid length of 25 m, the resulting images had a resolution that was too high, appearing uniform to the human eye and requiring significant zoom to see the intensity of corresponding road segment. Conversely, a grid length of 1 km allowed for the perception of routes across the entire city but was excessively broad. This broadness caused nearby routes to blend together and thus fail to represent the intensity of the distinct routes' segments.

5. Discussion

This research underscores the importance of data preprocessing and spatial analysis in the context of online food delivery services. By integrating routing information from the OSRM API, we have demonstrated a method for identifying and eliminating outliers in delivery data, thus enhancing the accuracy of subsequent predictive models.

Based on the route distribution in Figure 3 and direct analysis of the two groups of outliers in Figures 4 and 5, we classify 3% of the data as outliers. This also indicates that outliers are predominantly deliveries to more distant locations, whereas deliveries within city limits exhibit fewer outliers. Outliers located in the city boundaries most often refer to unreal destination points (e.g., points on highways or bridges). One of them is located on the map of Mumbai (bridge) shown in Figure 8. Additionally, our spatial analysis shows that the Indian OFD market has similar trends to the Chinese [15] and English [22,24] markets. The distribution of orders among cities presented in Figure 7 confirms that use of this type of platform is much more popular in densely populated areas.

Our approach has some limitations. First, the study area is restricted to India. Consequently, the results may not be generalizable on an international scale. To confirm the generalizability of our findings, we would need to gather and test similar datasets from various global regions. Consistent performance across these diverse datasets would validate our models' robustness. Second, we mainly used one map provider, OSM, and one routing engine, OSRM; therefore, the results may differ from those obtained using other tools. Third, the data used cover a period of three months; however, the number of deliveries in cities is relatively low. In this dataset, the average number of deliveries in Jaipur is under 40 per day. This does not accurately reflect the actual workload of OFD companies, which is likely to be much larger, and some methods may not perform as expected in the case of large amounts of data. The most noticeable problem may be the low responsiveness of Folium's interactive maps, which makes analysis much more difficult. In the case of Bayesian models, a large amount of data will increase the model fitting time and require more computational power.

Additionally, the use of the OSRM API involves several limitations, especially when considering the usage of a shared server. The number of requests per minute is limited and common to all users. Being an open-source project, OSRM does not offer any quality guarantees and, in some regions, the data may be sparse or outdated. These issues can affect all open-source routing engines. Moreover, there are certain geographical areas where access to external maps or GPS services is restricted. Those limitations could impose difficulties in using OSRM in different parts of the world.

In the second part of this article, we will delve into the application of Bayesian models to the preprocessed dataset, examining their efficacy in predicting delivery times and exploring potential improvements to the modeling approach. Our dataset does not accurately reflect the actual workload of OFD companies; therefore, it is highly recommended for future research to evaluate used methods on a larger dataset. The analysis of the Indian OFD market is based solely on deliveries; further research may take into account other elements such as social and cultural factors. Additionally, future studies could consider refinement of the preprocessing steps, such as advanced handling of missing values and more sophisticated outlier detection methods.

Author Contributions: Conceptualization, J.G., J.P. and J.B.; methodology, J.G., J.P. and J.B.; software, J.G. and J.P.; validation, J.G. and J.B.; formal analysis, J.G.; investigation, J.G.; resources, J.B.; data curation, J.G. and J.P.; writing—original draft preparation, J.G.; writing—review and editing, J.G. and J.B.; visualization, J.G. and J.P.; supervision, J.B.; project administration, J.B.; funding acquisition, J.B. All authors have read and agreed to the published version of the manuscript.

Funding: The third author's work was partially realized in the scope of a project titled "Process Fault Prediction and Detection". The project was financed by The National Science Centre on the base of decision no. UMO-2021/41/B/ST7/03851. Part of work was funded by AGH's Research University Excellence Initiative under project "DUDU—Diagnostyka Uszkodzeń i Degradacji Urządzeń".

Data Availability Statement: All code prepared as part of this project is available in the following repository: https://github.com/JohnnyBeet/Food-delivery-time-prediction/tree/preprocessing (accessed on 16 August 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

OFD	Online food delivery
GPS	Global Positioning System
OD	Origin–destination
OSRM	Open-Source Routing Machine
API	Application Programming Interface
OSM	OpenStreetMap

References

1. Statista. Online Food Delivery—Worldwide. 2024. Available online: https://www.statista.com/outlook/emo/online-food-delivery/worldwide (accessed on 4 May 2024).
2. IMARC Group. India Online Food Delivery Market Report. 2023. Available online: https://www.imarcgroup.com/india-online-food-delivery-market (accessed on 4 May 2024).
3. Alalwan, A.A. Mobile food ordering apps: An empirical study of the factors affecting customer e-satisfaction and continued intention to reuse. *Int. J. Inf. Manag.* **2020**, *50*, 28–44. [CrossRef]
4. Pomykacz, J.; Gibas, J.; Baranowski, J. Bayesian modelling of travel times on the example of food delivery: Part 2—Model creation and handling uncertainty. *Preprints* **2024**, 2024061443.
5. Unwin, A. Why is data visualization important? what is important in data visualization? *Harv. Data Sci. Rev.* **2020**, *2*, 1.

6. Li, B.; Chen, L.; Xiong, D.; Chen, S.; He, R.; Sun, Z.; Lim, S.; Jiang, H. Simultaneous detection of multiple areas-of-interest using geospatial data from an online food delivery platform (industrial paper). In Proceedings of the 30th International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 1–4 November 2022; pp. 1–10.
7. de Araujo, A.C.; Etemad, A. End-to-end prediction of parcel delivery time with deep learning for smart-city applications. *IEEE Internet Things J.* **2021**, *8*, 17043–17056. [CrossRef]
8. Zhu, L.; Yu, W.; Zhou, K.; Wang, X.; Feng, W.; Wang, P.; Chen, N.; Lee, P. Order fulfillment cycle time estimation for on-demand food delivery. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, 6–10 July 2020; pp. 2571–2580.
9. Joshi, M.; Singh, A.; Ranu, S.; Bagchi, A.; Karia, P.; Kala, P. Batching and matching for food delivery in dynamic road networks. In Proceedings of the 2021 IEEE 37th international conference on data engineering (ICDE), Chania, Greece, 19–22 April 2021; pp. 2099–2104.
10. Ji, S.; Zheng, Y.; Wang, Z.; Li, T. Alleviating users' pain of waiting: Effective task grouping for online-to-offline food delivery services. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 773–783.
11. Ulmer, M.W.; Thomas, B.W.; Campbell, A.M.; Woyak, N. The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. *Transp. Sci.* **2021**, *55*, 75–100. [CrossRef]
12. Garus, A.; Christidis, P.; Mourtzouchou, A.; Duboz, L.; Ciuffo, B. Unravelling the last-mile conundrum: A comparative study of autonomous delivery robots, delivery bicycles, and light commercial vehicles in 14 varied European landscapes. *Sustain. Cities Soc.* **2024**, *108*, 105490. [CrossRef]
13. Malhotra, I.; Chandra, P.; Majumdar, S.K. Route Optimization Application using Server-Client Architecture and Google APIs. In Proceedings of the 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 13–15 March 2019; pp. 210–214.
14. Paithane, P.; Wagh, S.J.; Kakarwal, S. Optimization of route distance using k-NN algorithm for on-demand food delivery. *Syst. Res. Inf. Technol.* **2023**, 85–101. [CrossRef]
15. Wang, Z.; He, S.Y. Impacts of food accessibility and built environment on on-demand food delivery usage. *Transp. Res. Part D Transp. Environ.* **2021**, *100*, 103017. [CrossRef]
16. Abahussein, S.; Ye, D.; Zhu, C.; Cheng, Z.; Siddique, U.; Shen, S. Multi-Agent Reinforcement Learning for Online Food Delivery with Location Privacy Preservation. *Information* **2023**, *14*, 597. [CrossRef]
17. Muñoz-Villamizar, A.; Solano-Charris, E.L.; Reyes-Rubiano, L.; Faulin, J. Measuring Disruptions in Last-Mile Delivery Operations. *Logistics* **2021**, *5*, 17. [CrossRef]
18. Yu, X.; Li, X.Y.; Zhao, J.; Shen, G.; Freris, N.M.; Zhang, L. Antigone: Accurate navigation path caching in dynamic road networks leveraging route apis. In Proceedings of the IEEE INFOCOM 2022-IEEE Conference on Computer Communications, Online, 2–5 May 2022; pp. 1599–1608.
19. Fu, J.; Bhatti, H.J.; Eek, M. Optimization of Freight Charging Infrastructure Placement Using Multiday Travel Data. In Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Bizkaia, Spain, 24–28 September 2023; pp. 1576–1582.
20. Vonthron, S.; Perrin, C.; Soulard, C.T. Foodscape: A scoping review and a research agenda for food security-related studies. *PLoS ONE* **2020**, *15*, e0233218. [CrossRef] [PubMed]
21. Hu, X.; Zhang, G.; Shi, Y.; Yu, P. How Information and Communications Technology Affects the Micro-Location Choices of Stores on On-Demand Food Delivery Platforms: Evidence from Xinjiekou's Central Business District in Nanjing. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 44. [CrossRef]
22. Keeble, M.; Adams, J.; Bishop, T.R.; Burgoine, T. Socioeconomic inequalities in food outlet access through an online food delivery service in England: A cross-sectional descriptive analysis. *Appl. Geogr.* **2021**, *133*, 102498. [CrossRef]
23. Maulidi, C.; Dwicaksono, A.; Aritenang, A.F.; Winarso, H. Food service spatial pattern after the emergence of online retail. *J. Infrastruct. Policy Dev.* **2024**, *8*, 3005. [CrossRef]
24. Janatabadi, F.; Newing, A.; Ermagun, A. Social and spatial inequalities of contemporary food deserts: A compound of store and online access to food in the United Kingdom. *Appl. Geogr.* **2024**, *163*, 103184. [CrossRef]
25. Jahanshahi, H.; Bozanta, A.; Cevik, M.; Kavuk, E.M.; Tosun, A.; Sonuc, S.B.; Kosucu, B.; Başar, A. A deep reinforcement learning approach for the meal delivery problem. *Knowl.-Based Syst.* **2022**, *243*, 108489. [CrossRef]
26. Wang, L.; Fu, H.; Wu, S.; Liu, Q.; Tan, X.; Huang, F.; Zhang, M.; Wu, W. CAMLO: Cross-Attentive Multi-View Network for Long-Term Origin-Destination Flow Prediction. In Proceedings of the 2024 SIAM International Conference on Data Mining (SDM), Atlanta, GA, USA, 21–25 October 2024; pp. 454–462.
27. Food Delivery Dataset. 2023. Available online: https://www.kaggle.com/datasets/gauravmalik26/food-delivery-dataset (accessed on 13 May 2024).
28. Open Source Routing Machine. 2024. Available online: https://project-osrm.org/ (accessed on 13 May 2024).
29. Open Source Routing Machine API. 2024. Available online: https://project-osrm.org/docs/v5.5.1/api/#route-service (accessed on 13 May 2024).
30. OpenStreetMap. 2024. Available online: https://www.openstreetmap.org/ (accessed on 13 May 2024).
31. García, S.; Ramírez-Gallego, S.; Luengo, J.; Benítez, J.M.; Herrera, F. Big data preprocessing: Methods and prospects. *Big Data Anal.* **2016**, *1*, 1–22. [CrossRef]

32. Stan. 2024. Available online: https://mc-stan.org/ (accessed on 14 May 2024).
33. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
34. Folium. 2024. Available online: https://python-visualization.github.io/folium/latest/ (accessed on 14 May 2024).
35. Navarro, D. How to Visualise a Billion Rows of Data in R with Apache Arrow. 2022. Available online: https://blog.djnavarro.net/posts/2022-08-23_visualising-a-billion-rows (accessed on 15 May 2024).
36. Hoteit, S.; Secci, S.; Sobolevsky, S.; Ratti, C.; Pujolle, G. Estimating human trajectories and hotspots through mobile phone data. *Comput. Netw.* **2014**, *64*, 296–307. [CrossRef]
37. Carfora, M.F. Interpolation on spherical geodesic grids: A comparative study. *J. Comput. Appl. Math.* **2007**, *210*, 99–105. [CrossRef]
38. Boeing, G. Modeling and Analyzing Urban Networks and Amenities with OSMnx. 2024. Working Paper. Available online: https://geoffboeing.com/publications/osmnx-paper/ (accessed on 5 June 2024).
39. Gelman, A.; Carlin, J.B.; Stern, H.S.; Dunson, D.B.; Vehtari, A.; Rubin, D.B. *Bayesian Data Analysis*, 3rd ed.; Chapman and Hall/CRC: New York, NY, USA, 2013.
40. The Swiggy Delivery Challenge. 2018. Available online: https://bytes.swiggy.com/the-swiggy-delivery-challenge-part-one-6a2abb4f82f6 (accessed on 18 May 2024).
41. DeepETA: How Uber Predicts Arrival Times Using Deep Learning. 2022. Available online: https://www.uber.com/en-PL/blog/deepeta-how-uber-predicts-arrival-times/ (accessed on 18 May 2024).
    accessed on 18.05.2024.
42. Predicting Time to Cook, Arrive, and Deliver at Uber Eats. 2019. Available online: https://www.infoq.com/articles/uber-eats-time-predictions/ (accessed on 18 May 2024).
43. Abdi, A.; Amrit, C. A review of travel and arrival-time prediction methods on road networks: Classification, challenges and opportunities. *PeerJ Comput. Sci.* **2021**, *7*, e689. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Neuromarketing and Big Data Analysis of Banking Firms' Website Interfaces and Performance

Nikolaos T. Giannakopoulos [1,*], Damianos P. Sakas [1] and Stavros P. Migkos [2]

[1] BICTEVAC Laboratory—Business Information and Communication Technologies in Value Chains Laboratory, Department of Agribusiness and Supply Chain Management, School of Applied Economics and Social Sciences, Agricultural University of Athens, 118 55 Athens, Greece

[2] Department of Accounting & Finance, School of Economic Sciences, University of Western Macedonia, 501 00 Kozani, Greece; aff01225@uowm.gr

* Correspondence: n.giannakopoulos@aua.gr

Abstract: In today's competitive digital landscape, banking firms must leverage qualitative and quantitative analysis to enhance their website interfaces, ensuring they meet user needs and expectations. By combining detailed user feedback with data-driven insights, banks can create more intuitive and engaging online experiences, ultimately driving customer satisfaction and loyalty. Thus, the need for website customer behavior analysis to evaluate its interface is critical. This study focused on the five biggest banking firms and collected big data from their websites. Statistical analysis was followed to validate findings and ensure the reliability of the results. At the same time, agent-based modeling (ABM) and System Dynamics (SD) were utilized to simulate user behavior, thereby allowing for the prediction of responses to interface changes and the optimization of their website, and to obtain a comprehensive understanding of user behavior, thereby enabling banking firms to create more intuitive and user-friendly website interfaces. This interdisciplinary approach found that various website analytical metrics, such as organic and paid traffic costs, referral domains, and email sources, tend to impact banking firms' purchase conversion, display ads, organic traffic, and bounce rate. Moreover, these insights into banking firms' website visibility, combined with the behavioral data of the neuromarketing study, indicate specific areas for their website interface and performance improvement.

Keywords: neuromarketing; banking firms; strategy; website interface; big data; digital marketing analytics; DSS

Citation: Giannakopoulos, N.T.; Sakas, D.P.; Migkos, S.P. Neuromarketing and Big Data Analysis of Banking Firms' Website Interfaces and Performance. *Electronics* **2024**, *13*, 3256. https://doi.org/10.3390/electronics13163256

Academic Editors: Wentao Li, Huiyan Zhang, Tao Zhan and Chao Zhang

Received: 24 July 2024
Revised: 8 August 2024
Accepted: 14 August 2024
Published: 16 August 2024

1. Introduction

Advancements in communication and information technology have equipped consumers with an array of online channels to gather, exchange, and share information about services, products, and personal brand experiences [1]. The onset of the digital age has caused a substantial paradigm shift across industries, compelling organizations to rethink their strategies and adopt digital transformation [2]. This shift has been especially profound in the banking sector, revolutionizing traditional practices and ushering in a new era focused on customer centricity and innovation [3].

Highly engaged customers tend to shop more, recommend more, and demonstrate stronger loyalty. A critical aspect of customer engagement strategies is maintaining a high-quality customer experience [4]. Interactive social media activities, leveraging information and communication technology, can enhance customer engagement and support the customer experience [5]. Consumer interaction and engagement are vital for marketers to sustain long-term customer relationships. Digital marketing creates participation opportunities that build trust, goodwill, and commitment between individuals and brands [6]. Lee et al. [7] found that incorporating humor and emotion can significantly boost consumer engagement in the banking sector. Fans of brand pages can view posts and interact by liking, sharing, and commenting [8], resulting in high levels of online activity due to the interactivity of the content.

Operational elements on a webpage are crucial in web design. Each element is considered an object in the program script, with unique methods and attributes controlling its behavior [7]. To gain consumer trust and build a high-quality brand, some construction companies actively seek quality inspection certifications to prove the quality of their projects. Faghih et al. [9] noted that the user interface (UI) is the point of interaction between the user and computer software. The success of a software application heavily depends on user interface design (UID), which significantly impacts ease of use and learning time. UI should prioritize maximizing usability and user experience [10].

User experience (UX) reflects the satisfaction of users, measuring their impressions when using or planning to use an application [11]. Additionally, UX and UI are connected to user emotions and satisfaction during software interaction, regardless of functional or non-functional requirements [11]. The user interface (UI), which communicates system functionality, usability, and satisfaction, is often closely linked with UX [12].

The implications of integrating website customer big data and neuromarketing for banking firm websites are profound. Leveraging big data analytics allows banks to gain deep insights into customer behaviors, preferences, and interactions on their websites [13]. By analyzing patterns and trends, banks can tailor their website content, design, and functionalities to better meet customer needs and enhance user experience. Neuromarketing techniques, such as eye-tracking and brain imaging, provide additional layers of understanding by revealing subconscious responses to various website elements [14]. This combined approach can lead to the more effective personalization of services, improved customer satisfaction, and increased engagement. For example, banks can optimize website layouts to highlight key information and offers, use data-driven insights to craft compelling messaging, and design intuitive navigation paths that align with natural user behaviors [15]. Ultimately, the integration of customer big data and neuromarketing enables banking firms to create more efficient, user-centric websites that can drive higher conversion rates, foster customer loyalty, and maintain a competitive edge in the digital landscape.

Therefore, our paper aims to study the website behavior of banking firms' customers, through big data and neuromarketing analyses, and discern potential factors that impact their website interface and performance. From the above analysis, the main research questions (RQs) for the present study are presented below:

RQ1: Does the analytical customer behavior of banking firm websites impact their visibility and engagement?

RQ2: Does the customer behavior of banking firm websites enhance their interface?

These research questions address critical aspects of the interplay between customer behavior on banking firm websites and the resulting impact on visibility, engagement, and interface enhancement. RQ1 explores whether analytical insights into customer behavior directly influence the visibility and engagement levels of banking websites. This question is motivated by the growing importance of user-centric design and personalized experiences in the digital banking landscape, suggesting that a deep understanding of customer interactions can lead to more effective marketing strategies and higher user retention. RQ2 investigates the extent to which customer behavior data informs and improves the website interface, highlighting the role of iterative design based on real user feedback. This research fills a gap in the existing literature by integrating behavioral analytics with interface design, an area often overlooked in traditional banking studies, which tend to focus more on security and functionality rather than user experience. Addressing these questions can provide banking firms with actionable insights to refine their digital strategies, ultimately leading to more engaging and user-friendly websites that cater to the evolving needs and preferences of their customers. This research not only contributes to academic discourse by bridging behavioral analytics and interface design but also offers practical implications for enhancing digital engagement in the banking sector.

The stages of this paper are as follows: the second part outlines the theoretical frameworks and research hypotheses; the third part details the methodology; the fourth part presents the study's results, including statistical and modeling analyses; and the fifth part covers the discussion and conclusions. This structure facilitates a comprehensive analysis of how banking firms can leverage marketing analytics and customer website behavior to improve their webpage interfaces.

2. Literature Background

2.1. Banking Firms, Digital Marketing, and User Engagement

As banks navigate an ever-evolving landscape, integrating social media applications has become a crucial factor in this sector [16]. The digital transformation of banks involves a wide range of changes, from adopting advanced technologies to restructuring internal processes and reimagining customer experiences [17]. This transformation is fundamentally driven by the need to adapt to changing customer expectations, stay ahead of disruptive market forces, and seize the opportunities presented by the digital revolution [18]. Essentially, a key assumption of our model is that for a bank to engage customers effectively, it must achieve a high-efficiency grade in traditional CRM. This means that banks need to satisfy their customers, earn their trust, and ensure that their customers feel and act with loyalty under any circumstances [19].

The strategic use of digital marketing focuses on building personalized relationships with consumers [20]. Banks leverage the value of social media to provide direct and real-time marketing, thereby enabling them to offer customized responses to clients regardless of geographic location. CRM, as a digital marketing strategy, has been discerned as an important factor in improving customer engagement [21], especially in the banking sector. Banks leverage social media platforms for personnel selection, crowdsourcing, and promoting their corporate values [21]. Additionally, they focus on gathering customer data and enhancing financial education [22]. However, there is a marked difference in the importance placed on building image and reputation compared to marketing and business development.

Trust in service providers and economic stability significantly enhance customer emotional, cognitive, and behavioral engagement with banking firms [23]. To adapt successfully, banks must prioritize organizational culture, customer engagement, financial innovation, and proactive responses to fintech disruptions [24]. The implication of video and affiliate marketing strategies, in the wider aspect of digital marketing, tends to increase the engagement of banking firms with their brand [25,26].

The primary factors influencing customer experience in digital banking include service quality, functional quality, perceived value, employee–customer engagement, perceived usability, and perceived risk [27]. There is a strong connection between customer experience, satisfaction, and loyalty, which in turn impact financial performance [27]. Digital tools have enabled companies to better target their markets by tracking customer preferences, offering more personalized solutions, and facilitating value co-creation in the financial services sector [28]. The findings indicate that key website attributes such as visualization, interactivity, aesthetics, customization, ease of use, and telepresence positively influence customer experience. Additionally, the results highlight positive relationships between customer experience, customer trust, and customer retention [29]. Therefore, more light should be shed on the role of website customer behavioral data on website interface and the performance of firms in the banking sector.

2.2. Metrics and KPIs of Friendly Website User Interface (UI)

Lestari et al. [30] found that responsive web design effectively maintains the user experience by ensuring content readability and enhancing the enjoyment of using websites. It also reduces the need for excessive scrolling when reading content. Almeida and Monteiro [31] noted that creating user experiences requires a range of multidisciplinary skills, including a knowledge of tools, processes, and business intricacies. Walsh et al. [32]

highlighted that responsive web design is a modern approach allowing developers to create webpages that offer a consistent user experience across different device sizes.

Usability is defined as the ease with which users can interact with an interface [33]. According to ISO standard 9241–11 [34], usability is "the extent to which a system, product, or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use". Efficiency, as per this standard, refers to "resources used for the results achieved". Additionally, customizing a menu can reduce user input time on application interfaces [35].

Kim and Cho [36] describe user experience (UX) as the overall experience related to a user's perceptions and thoughts when interacting with a system, product, content, or service. User interface (UI), on the other hand, involves the visual elements and commands used to operate the system, input data, and use content [37]. Von Saucken et al. [38] stated that UX enhances UI by incorporating emotional aspects. A banking firm's website with a user-friendly interface is closely linked to its performance [9]. Therefore, some of the website performance metrics related to the indication of a website's friendliness are its increased website traffic through various sources (visibility), page views per visit, conversion rates (i.e., purchase conversion), and decreased bounce rates [39].

2.3. Neuromarketing and Big Data Analysis Implications on Website Interface and Performance

Neuromarketing proposes that emotional and rational thinking coexist and are interdependent [40]. Emotions capture a subject's attention, allowing the rational brain to engage with the presented situation. Gabriel et al. [41] demonstrated that affective neuroscience, when applied to marketing, accurately predicts customer reactions to products. Neuromarketing, which integrates neuroscience and marketing, aims to understand customers' impulses, feelings, and emotions, thereby influencing purchase decisions and facilitating interaction between consumers and companies [42].

Neuromarketing combines neuroimaging with marketing science to better understand consumer behavior and brand loyalty, offering a wider perspective on marketing science [43]. It has advanced over the last 5 years, with EEG and physiological response measuring techniques being preferred over fMRI for consumer response prediction and classification [44]. It has also become popular due to its potential to provide hidden consumer experience insights and faster marketing methods, but its affordability remains uncertain [45]. Neuromarketing and big data analytics offer strategic consumer engagement by integrating neuroscience, biometrics, multimedia technology, marketing strategy, and big data management [46]. Moreover, it can enhance people's motivation and learning performance in online classrooms using visual material, virtual boards, and class activities [47].

Berčík et al. [48] found that monitoring visual attention indicates the need for larger text fonts and copywriting adjustments to reduce text volume, reorganize content, or replace complex texts with animations and infographics. These modifications primarily enhance the user interface (UI) and overall user experience (UX). EEG data analysis can accurately predict consumer decision-making and distinguish between "Like" and "Dislike" preferences in advertisements, with frontal and centro-parietal locations being the most discriminative channels [49].

On the other hand, quantitative analyses and cross-country comparisons can provide a broader understanding of strategies, outcomes, diverse contexts, market dynamics, and organizational factors within the banking sector [24]. Uygun et al. [50] stated that the utilization of big data analysis from websites could lead to enhanced webpage usability (for users/visitors) and overall website performance. Moreover, Li and Zhang [51] highlighted the role of website big data analysis in improving the performance of enterprises' e-commerce platforms and their webpage display. Big data analysis based on social media platforms has been discerned as an important factor in enhancing the engagement of decentralized finance firm customers and their websites' performance [52].

Therefore, from the aforementioned literature review, a gap has been spotted that concerns the implication of website users' quantitative and qualitative data on a website's interface and performance. Moreover, this gap is extended to the specific utilization of website customer big data (as a method of quantitative analysis) and neuromarketing applications (as a form of qualitative analysis), as well as to these analyses' impact on the banking sector.

2.4. Hypotheses Development

Following the settlement of this study's RQs, the authors moved to the deployment of four research hypotheses to further analyze the aims of this paper. The four hypotheses (H1–H4) mainly refer to banking firms' website analytical consumer behavioral data to answer RQ1. RQ2 can be answered by utilizing an eye-tracking and heatmap analysis tool.

The first research hypothesis (H1) posits that there is a strong connection between banking firms' digital marketing analytics and their customers' purchasing conversion [53]. This hypothesis is explored through the first research question (RQ1), which investigates whether the analytical understanding of customer behavior on banking firm websites affects their visibility and engagement [54]. The underlying premise is that by effectively analyzing customer interactions and behaviors on their digital platforms, banking firms can enhance their visibility and engagement metrics. This, in turn, is expected to drive higher conversion rates as firms tailor their marketing strategies to meet customer needs and preferences more accurately. Therefore, if this hypothesis is confirmed, it would suggest that robust digital marketing analytics are a critical component for banking firms in optimizing their online presence and achieving better customer conversion outcomes.

Hypothesis 1 (H1). *Banking firms' digital marketing analytics are strongly connected with their customers' purchasing conversion.*

Hypothesis 2 (H2) suggests that the extent to which banking firms utilize display ads on their webpages is heavily influenced by their digital marketing analytics [55]. This hypothesis aligns with the first research question (RQ1), which examines whether the analytical insights into customer behavior on banking firm websites impact their visibility and engagement. The connection implied is that digital marketing analytics provide critical data on customer preferences, behaviors, and interactions, thereby enabling firms to strategically deploy display ads to maximize visibility and engagement [56]. By leveraging analytics, banking firms can optimize the placement, frequency, and content of display ads to better attract and engage customers, thereby enhancing their online presence and potentially improving conversion rates. Thus, if H2 holds true, it underscores the importance of integrating comprehensive digital marketing analytics into advertising strategies to effectively target and retain customers.

Hypothesis 2 (H2). *The amount of display ads that banking firms use on their webpages is strongly dependent on their digital marketing analytics.*

Hypothesis 3 (H3) posits that a banking firm's organic visibility is negatively impacted by its digital marketing analytics [57]. This hypothesis stems from the first research question (RQ1), which investigates whether the analytical insights into customer behavior on banking firm websites affect their visibility and engagement. The premise here is that an over-reliance on digital marketing analytics might lead to strategies that prioritize short-term gains through paid or targeted advertising at the expense of long-term organic visibility. For example, banking firms might focus more on personalized ads and sponsored content driven by analytics, potentially neglecting best SEO practices and organic content development [58]. If H3 is validated, it would suggest that while digital marketing analytics can boost engagement through targeted efforts, they might inadvertently reduce the firm's visibility in organic search results, highlighting the need for a balanced approach that integrates both analytics-driven strategies and organic growth initiatives.

Hypothesis 3 (H3). *A banking firm's organic visibility is negatively impacted by its digital marketing analytics.*

Hypothesis 4 (H4) posits that the digital marketing analytics of banking firms tend to increase the bounce rate of their websites [59]. This hypothesis is examined through the lens of the first research question (RQ1), which explores whether the analytical understanding of customer behavior on banking firm websites influences their visibility and engagement. This hypothesis suggests that while digital marketing analytics aim to optimize user experience and engagement, they may inadvertently lead to higher bounce rates if the insights are not properly implemented. For instance, over-targeting or irrelevant content driven by misinterpreted analytics might cause users to leave the site quickly. If H4 is supported, it indicates that despite the potential benefits of analytics in crafting tailored marketing strategies, there is a risk of increased bounce rates if the data are not effectively utilized to enhance user experience [60]. This underscores the importance of not only gathering and analyzing data but also translating them into meaningful, user-friendly website improvements.

Hypothesis 4 (H4). *The digital marketing analytics of banking firms tend to increase the bounce rate of their website.*

3. Materials and Methods

3.1. Methodological Concept

To explore the study's research, both quantitative and qualitative analyses were employed to investigate the link between banking firms' website customer data and their webpage interface enhancement. This study utilized a four-stage methodological framework to achieve this. Within this framework, valuable insights into customer behavior on banking firm websites were gathered, which helped establish a framework to discern digital marketing strategies that can enhance the efficiency of their website interface.

- The research started with the collection of data on website customers and digital marketing activities from banking firm websites. A website's user behavioral data (pages per visit, bounce rate, time on site, etc.) were sourced from the website platform Semrush [61], which enables the extraction of big data from corporate webpages.
- The next step involved statistical analysis using methods such as descriptive statistics, correlation, and linear regression. By analyzing the coefficients obtained, researchers can determine the impact of banking firms' website customer data on their digital marketing and interface performance metrics, including purchase conversion, display ads, organic traffic, and bounce rate.
- After statistical analysis, a hybrid model (HM) incorporating agent-based models (ABMs) and System Dynamics (SD) was used for the simulation. The software AnyLogic (version 8.9.1) [62] was employed to create a hybrid model that simulates the relationships between the study's dependent and independent variables over 360 days. This model aims to represent the dynamic interaction between banking firms' website interface metrics and key metrics of their digital marketing strategies.
- The final stage included a neuromarketing approach to gain deeper insights from 26 participants who viewed the websites of the selected banking firms. They were instructed to search and observe, in 20 s, the selected banking firm websites and their provided financial products and services. Eye-tracking and heatmap analysis were conducted using the SeeSo Web Analysis platform (Eyedid SDK) [63]. This method seeks to extract additional information about the onsite activity and engagement of the participants from the qualitative methodological concept.

3.2. Fuzzy Cognitive Mapping (FCM) Framework

In this section, the authors illustrate the relationships among the study's variables using Fuzzy Cognitive Mapping (FCM). This method is highlighted for its effectiveness in demonstrating these connections. A clearer understanding is achieved by presenting the relationships between the variables, particularly regarding the link between banking firms' website interface efficiency variables and their customer behavioral analytical data. The authors employed MentalModeler [64] to develop a conceptual model of the paper's variables, as shown in Figure 1. This FCM model helps extract key insights from the relationships between variables. FCM effectively represents the static relationships and interconnections of the model's variables [65]. Additionally, FCM has been successfully applied in solving various decision-making problems across different fields [66].

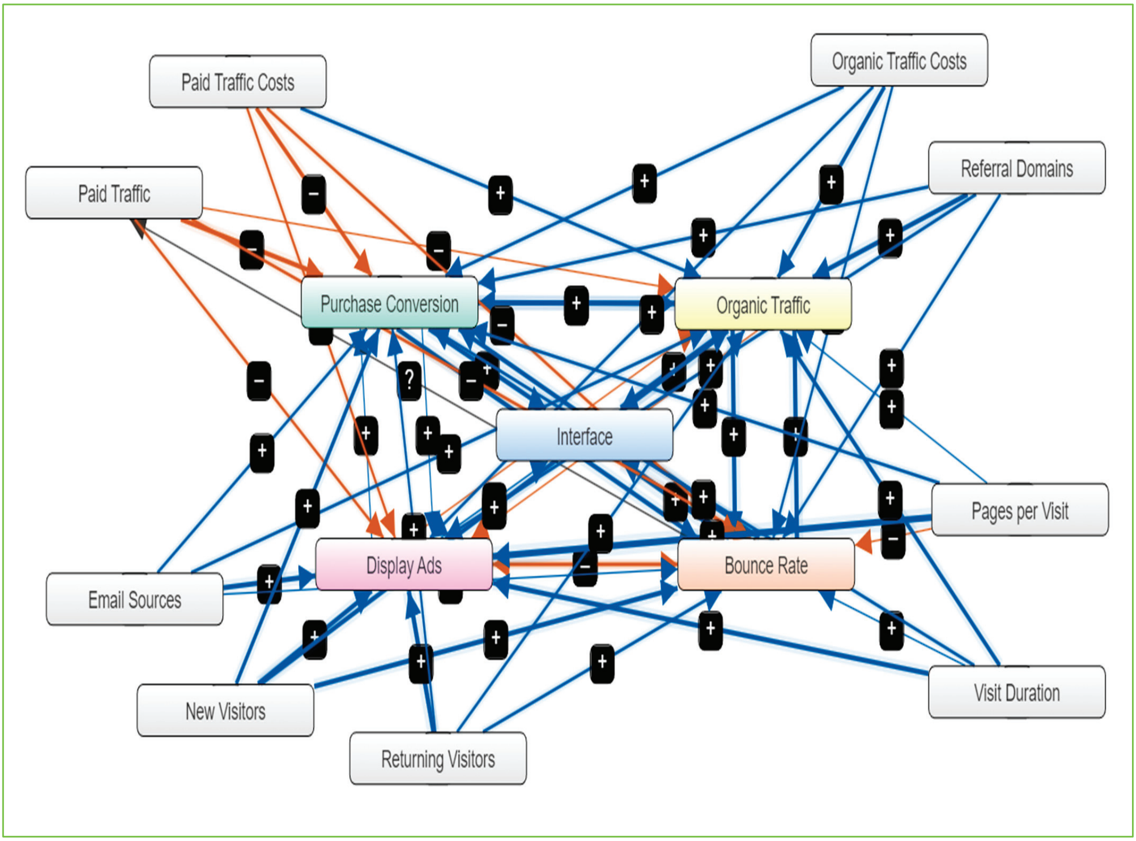


Figure 1. FCM Conceptual Framework of Banking Firm Websites. Blue and red arrows signify positive and negative correlations between variables, respectively. The symbols "+" and "–" represent the positive and negative per-centage changes, respectively.

3.3. Research Sample

The development of the present study was based on the exploitation of big data analytics from the websites of the sample firms, as well as the qualitative data obtained from the 26 participants who observed these webpages. As referred to in the Methodological Concept Section, this study focuses on analyzing banking firms' website interfaces, and thus, the 5 biggest and most established bank institutes were selected, based on their market capitalization (as of January 2024) [67]. Therefore, the biggest banking firms included in this research are (a) JPMorgan Chase, (b) the Bank of America, (c) the Industrial and Commercial Bank of China Limited, (d) Wells Fargo, and (e) the Agricultural Bank of China. The gathered data consist of various website analytical data originating from the visitors' and customers' onsite behaviors, while the collection period ran from 1 August 2023 to

29 February 2024. The Decision Support System (DSS) utilized to extract these data was the Semrush platform [61].

Concerning the qualitative part of the study, the authors selected 26 participants who were related to banking services provided through websites to perform the neuromarketing test. The authors selected 26 participants who were well aware of the financial services in the banking sector, and they were instructed to observe, for 1 min, the main webpage of the above banking firms. This test aimed to examine whether banking firm website customer behaviors can provide valuable insights into the performance and interface of the webpage by indicating potential areas of focus (on the webpage) or parts that do not create any engagement with the observer/visitor. For this test, the SeeSo Web Analysis platform (Eyedid SDK) [63] was utilized, and the combined heatmaps and gaze data were compiled into consolidated figures, as illustrated in Section 4.3.

4. Results

4.1. Statistical Analysis

A crucial part of the study's research is the extraction of the variables' relationships; thus, the authors began by performing a descriptive statistical analysis (Table 1). The statistical measures of mean, max, min, std. deviation, skewness, and kurtosis were selected. The latter two measures (skewness and kurtosis) are some of the variables' normality indicators when their values are between −2, and 2. Then, the variables' correlations, based on Pearson's statistic, were produced to explore the variables' connection, as shown in Table 2 below.

Table 1. Descriptive statistics of the five banking firms during the past six months.

	Mean	Min	Max	Std. Deviation	Skewness	Kurtosis
Organic Traffic	9,868,004.17	9,486,121.00	10,700,067.60	351,366.56	1.342	1.651
Organic Keywords	987,820.46	889,059.20	1,193,079.60	76,418.52	1.592	1.851
Organic Traffic Costs	37,155,781.98	28,929,891.40	44,660,727.20	5,822,486.64	−0.188	−1.627
Paid Traffic	337,898.57	232,588.80	487,373.40	66,696.66	0.396	1.333
Paid Keywords	6510.47	1815.20	9700.60	2624.74	−0.757	−0.580
Paid Traffic Costs	1,514,463.27	992,316.60	2,491,839.60	406,005.96	0.998	1.667
Email Sources	184,876.14	0.00	720,314.00	300,170.77	1.379	0.219
Display Ads	4199.57	0.00	20,892.00	7636.02	1.982	1.927
Purchase Conversion	7.71	7.00	8.00	0.49	−1.230	−0.840
Referral Domains	51,181.91	49,694.40	52,457.40	794.22	−0.360	−0.317
Visit Duration	519.40	368.00	737.00	128.25	0.658	−0.174
Bounce Rate	0.45	0.42	0.49	0.02	0.606	−1.361
Pages per Visit	3.43	2.00	5.00	0.97	0.277	0.042
New Visitors	15,149,188.40	14,150,098.00	16,212,804.00	801,388.14	0.025	−1.625
Returning Visitors	47,056,175.89	44,705,979.00	51,410,725.00	2,301,015.96	1.103	1.599

N = 180 observation days for the five selected banking firms.

Table 2. Correlation analysis matrix.

	Organic Traffic	Organic Traffic Costs	Paid Keywords	Paid Traffic Costs	Email Sources	Display Ads	Purchase Conversion	Referral Domains	Visit Duration	Bounce Rate	Pages per Visit	New Visitors	Return Visitors
Organic Traffic	1	0.604 *	0.264	0.037	0.174	−0.013	0.619	0.545	0.529	0.905**	0.068	0.796*	0.469
Organic Traffic Costs	0.604 *	1	0.037	0.000	0.607	0.413	0.206	0.830 **	0.124	0.242	0.657	0.489	0.628
Paid Traffic	−0.122	−0.052	0.533	0.889 **	−0.220	−0.304	−0.521	0.249	−0.705	−0.298	−0.022	−0.587	−0.539
Paid Traffic Costs	0.037	0.000	0.379	1	−0.371	−0.315	−0.547	0.241	−0.549	−0.193	−0.070	−0.458	−0.524
Email Sources	0.174	0.607	−0.257	−0.371	1	0.590	0.344	0.424	0.145	0.002	0.709	0.356	0.698
Display Ads	−0.013	0.413	−0.456	−0.315	0.590	1	0.160	0.299	0.635	−0.316	0.843 *	0.554	0.857 *
Purchase Conversion	0.619	0.206	−0.555	−0.547	0.344	0.160	1	0.175	0.224	0.600	0.300	0.539	0.485
Referral Domains	0.545	0.830 **	0.249	0.241	0.424	0.299	0.175	1	−0.223	0.179	0.737 *	0.269	0.394
Visit Duration	0.529	0.124	−0.748	−0.549	0.145	0.635	0.224	−0.223	1	0.163	0.309	0.804 *	0.717
Bounce Rate	0.905 **	0.242	−0.542	−0.193	0.002	−0.316	0.600	0.179	0.163	1	−0.051	0.581	0.192
Pages per Visit	0.068	0.657	−0.410	−0.070	0.709	0.843 *	0.300	0.737 *	0.309	−0.051	1	0.558	0.830 *
New Visitors	0.796 *	0.489	−0.904 **	−0.458	0.356	0.554	0.539	0.269	0.804 *	0.581	0.558	1	0.856 *
Returning Visitors	0.469	0.628	−0.773 *	−0.524	0.698	0.857 *	0.485	0.394	0.717	0.192	0.830 *	0.856 *	1

* and ** indicate statistical significance at the 95% and 99% levels, respectively.

In this stage, the simple linear regressions (SLRs) of the dependent variables of the study (purchase conversion, display ads, organic traffic, and bounce rate) were developed to estimate the impact of the independent variables (organic keywords, organic traffic costs, paid keywords, paid traffic costs, email sources, visit duration, pages per visit, and new and returning visitors) of banking firms' website visitor data. The first SLR model (Table 3) with purchase conversion as a dependent variable was verified overall, with a *p*-value < a = 0.01 level of significance and an R^2 = 1.000. The independent variables with the most significant impact were organic traffic costs, paid traffic costs, referral domains, and email sources (*p*-values < a = 0.01 level of significance). For every 1% of the increase in organic traffic costs, paid traffic costs, referral domains, and email sources, banking firms' purchase conversions vary by −167.0%, −136.9%, 169.6%, and 16.7%, respectively.

Table 3. Impact of banking firms' marketing analytics on their website purchase conversion.

Variables	Standardized Coefficient	R^2	F	*p*-Value
Organic Traffic Costs	−1.670	1.000	-	0.000 **
Paid Traffic Costs	−1.369			0.000 **
Referral Domains	1.696			0.000 **
Email Sources	0.167			0.000 **

** Indicates statistical significance at the 99% level.

In Table 4, where the SLR model of banking firms' display ads is presented, this is verified overall with a *p*-value < a = 0.01 level of significance and an R^2 = 1.000. The independent variables with the most significant impact (*p*-values < a = 0.01 level of significance) on display ads were the paid traffic costs, referral domains, and email sources. When paid traffic costs, referral domains, and email sources increase by 1%, banking firms' organic traffic variates by 19.8%, −6.5%, and −13.5%, respectively. Moving on to the SLR model of banking firms' organic traffic (Table 5), this regression is verified overall with a *p*-value < a = 0.01 level of significance and an R^2 = 1.000. The independent variables with the most significant impact (*p*-values < a = 0.01 level of significance) on organic traffic were also the paid traffic costs, referral domains, email sources, and display ads. For every 1% increase in paid traffic costs, referral domains, and email sources, organic traffic varied by −2.4%, −31.9%, and 4.1%, respectively.

Table 4. Impact of banking firms' marketing analytics on their website display ads.

Variables	Standardized Coefficient	R^2	F	*p*-Value
Paid Traffic Costs	0.198	1.000	-	0.000 **
Referral Domains	−0.065			0.000 **
Email Sources	−0.135			0.000 **

** Indicates statistical significance at the 99% level.

Table 5. Impact of banking firms' marketing analytics on their website organic traffic.

Variables	Standardized Coefficient	R^2	F	*p*-Value
Paid Traffic Costs	−0.024	1.000	-	0.000 **
Referral Domains	−0.319			0.000 **
Email Sources	0.041			0.000 **

** Indicates statistical significance at the 99% level.

Finally, in Table 6, the SLR model of banking firms' bounce rate is presented. This model was also verified overall with a *p*-value < a = 0.01 level of significance and an R^2 = 1.000. The independent variables with the most significant impact (*p*-values < a = 0.01 level of significance) on bounce rate were the same as for the purchase conversion model (organic traffic costs, paid traffic costs, referral domains, and email sources). For every 1% increase in organic traffic costs, paid traffic costs, referral domains, and email sources, the bounce rate varied by 104.5%, 2.5%, 33.4%, and −4.3%, respectively.

Table 6. Impact of banking firms' marketing analytics on their website bounce rate.

Variables	Standardized Coefficient	R^2	F	*p*-Value
Paid Traffic Costs	0.025			0.000 **
Referral Domains	0.334			0.000 **
Email Sources	−0.043			0.000 **

** Indicates statistical significance at the 95% level.

4.2. Simulation Model

To further study the connection between key website performance metrics and the behavior of banking firms' digital customers, the utilization of a hybrid model (HM) was discerned. This model extends to the agent-based modeling (ABM) and the System Dynamics (SD). The use of ABM and SD models to investigate social and ecological issues and improve decision-making has been explored by Nugroho and Uehara [68]. McGarraghy et al. [69] applied these models to assess the impact of policies on decision-making in food value chains. Similarly, Wang et al. [70] utilized ABM and SD analyses to study the reduction in carbon dioxide emissions from urban transportation. Additionally, Nguyen et al. [71] employed a hybrid conceptual model combining ABM and SD to examine the control of COVID-19 spread in care homes.

The execution of the hybrid model simulation, which refers to a 360-day model time, starts from the statechart of potential banking customers (Figure 2). Then, based on the statistical results from the collected data, the agents move either to the new visitor statechart or the returning visitor one. The bounce rate statechart leads the visitors/agents back to the first one or to either the organic or paid traffic statechart, based on their means of entering the banking firm website the first time (organic or paid search/keywords). From there, the remaining agents move to the display ads statechart or head back to the initial statechart (potential banking firms' customers). Finally, these agents move to the purchase conversion statechart, after this has been affected by the banking firms' display advertising. Throughout each of the 10,000 agents mobilized, the values of the dynamic variables of email sources, referral domains, and organic and paid costs are calculated using the normal distribution of the sample's variables. During the simulation process, the behavioral data of customers (including bounce rate, pages per visit, and time spent on site) are calculated for each of the 10,000 agents, using the normal distribution. The main commands and the Java route are outlined in Table A1 (Appendix A).

Through the development of the hybrid model simulation, the course of the banking firms' digital marketing performance metrics (organic traffic, purchase conversion, bounce rate, and display ads) is presented, across the trajectory of their website visitor/agent behavioral metrics. From the simulation, as seen in Figure 3, the following variables relationships arose: (a) banking firms' purchase conversion is positively impacted by email sources, paid costs, and referral domains but negatively affected by organic costs; (b) website bounce rate is positively impacted by email sources, paid costs, and referral domains but negatively affected by organic costs; (c) organic traffic is positively impacted by organic costs, email sources, and referral domains but negatively affected by paid costs; and (d) display ads are positively impacted by organic costs, paid costs, and referral domains but negatively affected by email sources.

4.3. Neuromarketing Applications

After having analyzed the variation in the study's variables through the hybrid modeling process, the need for a differentiated method arises. Since quantitative analysis offers valuable insights into customer behavior by identifying patterns, trends, and correlations, it alone is often deemed insufficient for a thorough understanding. It may not sufficiently explore the motivations, emotions, or underlying reasons behind customer actions. Qualitative methods are better equipped to delve into these aspects. To achieve a more comprehensive and actionable comprehension of customer behavior, many researchers and marketers advocate combining both quantitative and qualitative approaches.

This fusion enables a more nuanced interpretation of data, a deeper grasp of the customer's viewpoint, and a more effective means of addressing the complexities of human behavior.

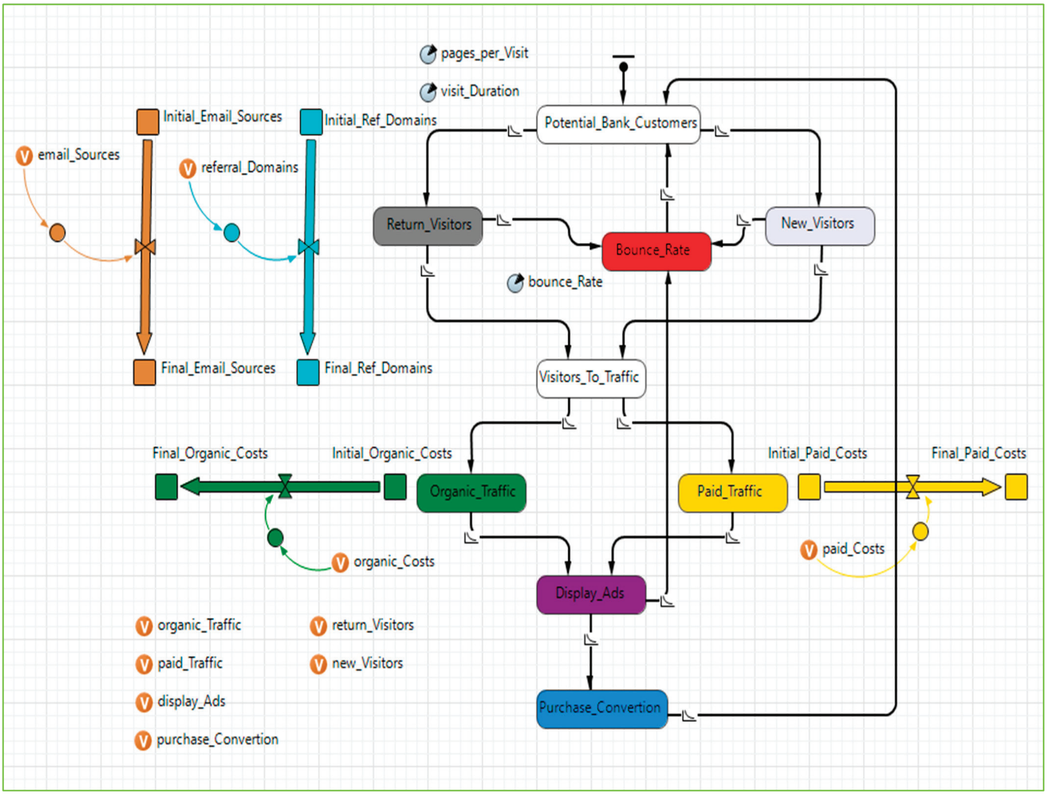


Figure 2. Hybrid Model (HM) Development.

Much neuromarketing research has been deployed to examine the implications of customer behavior on various products and services, with remarkable outcomes. Ezquerra et al. [72] studied student emotions, engagement, and attention in science activities using Facial Emotion Recognition (FER) tools by iMotions; meanwhile, Chen et al. [73] capitalized on both Galvanic Skin Response (GSR) and VR products to examine pupil responses to analyze the emotional and attentional activity of humans. Moreover, to extract valuable insights regarding people's arithmetic and memory evaluation, Muke et al. [74] used iMotions biometric platform and eye-tracking equipment. This same eye-tracking tool was utilized by Amiri et al. [75] to gather gazes and facial expressions of clients to assess the feedback on and evaluations of the purchased goods and services.

Therefore, by utilizing these neuromarketing tools (eye-tracking, heatmaps, and scan paths) of the SeeSo Web Analysis platform (Eyedid SDK) [63], several insights regarding banking firms' website interfaces and their visitor/customer behavior arise. From Figure 4, it can be discerned that banking firm websites with longer webpages, which need more time for visitors to scroll and observe all the data, had less continuous fixations and shorter fixation times (during the 20 s of observation), than those with a shorter webpage (and a longer fixation time). Moreover, the websites with a longer webpage had a greater number of fixations (fixations count) and a greater number of gazes (gazes count) on average. From the heatmap analysis of the banking firm websites (Figure 5), we can see that the participants, in all the included websites, intensely observed (increased heatmap intensity) their menus, the information that refers to their financial products/services (loans, savings, cards, etc.), their logos (brands), while also observing and interacting with their display ads. Finally, from the scan-path analysis (Figure 6), we can discern that the results of the heatmap analysis are also confirmed from the participants' scan path, and on average, their path began with observing the banking firm's brand (logo) followed by their menu options, information regarding their financial products and services, and their display ads, and then attention faded through the last parts of the webpage.

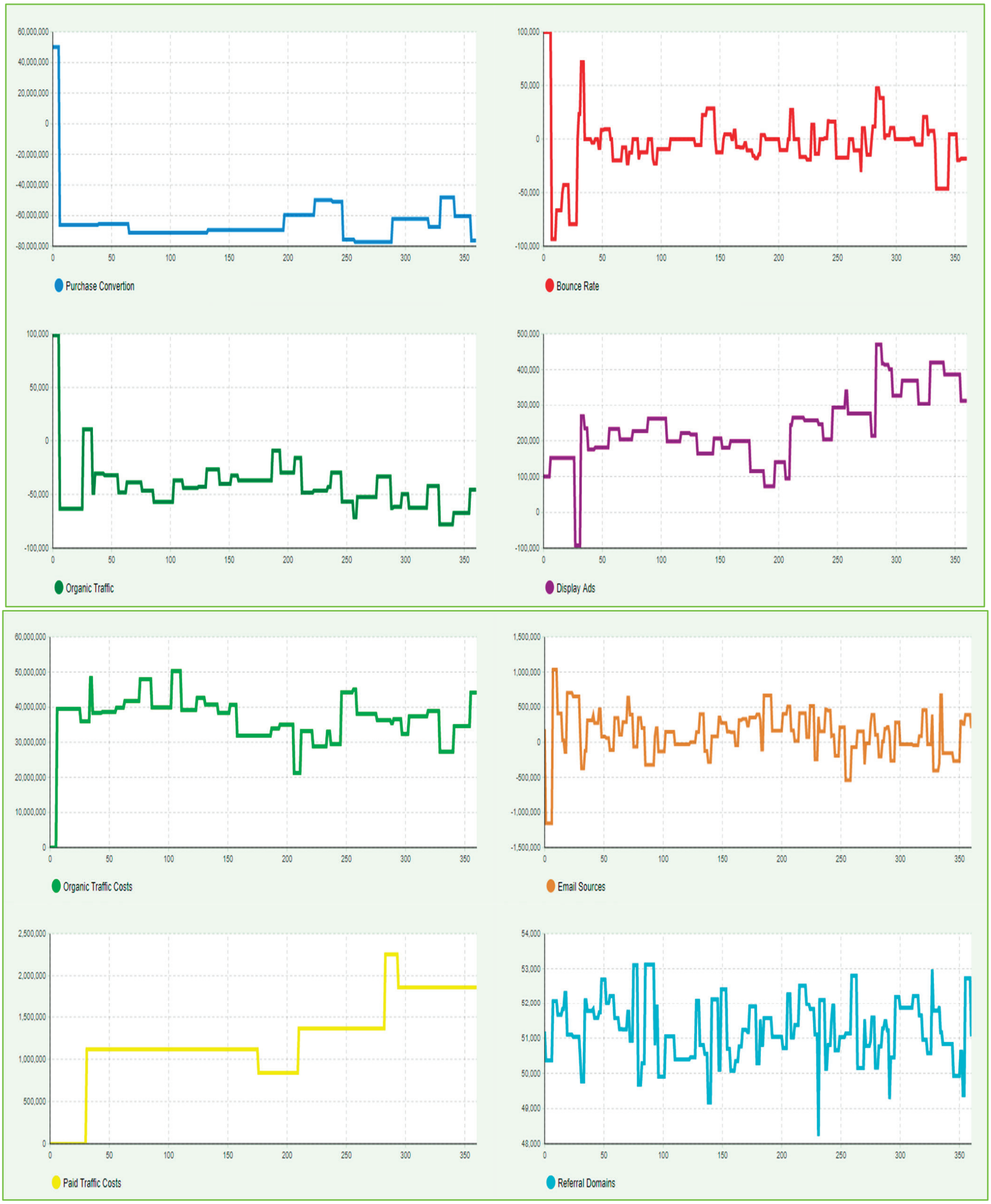
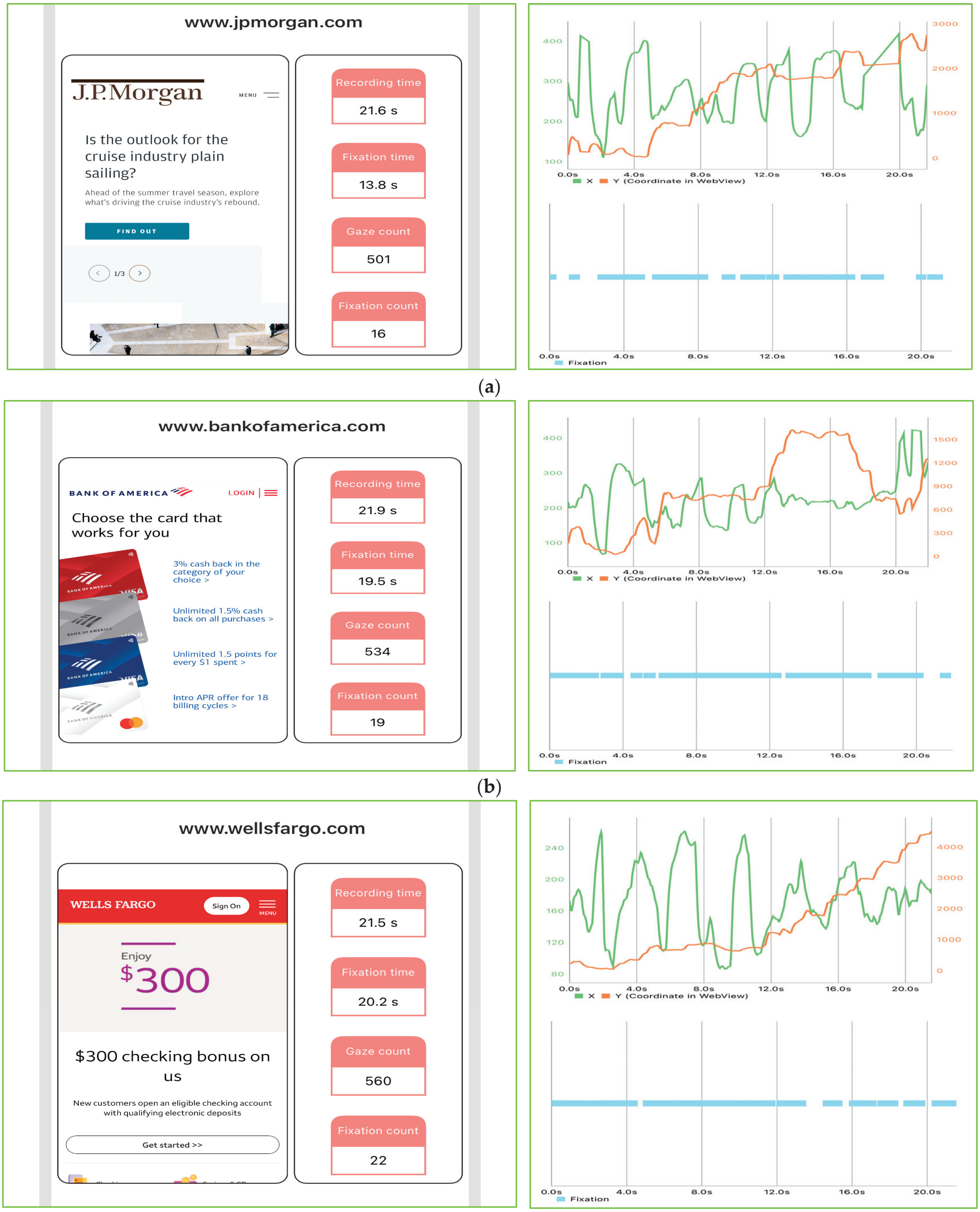


Figure 3. Simulation outputs of the hybrid model simulation in a period of 360 days.



www.jpmorgan.com

J.P.Morgan MENU

Is the outlook for the cruise industry plain sailing?

Ahead of the summer travel season, explore what's driving the cruise industry's rebound.

FIND OUT

1/3

Recording time	21.6 s
Fixation time	13.8 s
Gaze count	501
Fixation count	16

(**a**)

www.bankofamerica.com

BANK OF AMERICA LOGIN

Choose the card that works for you

3% cash back in the category of your choice >

Unlimited 1.5% cash back on all purchases >

Unlimited 1.5 points for every $1 spent >

Intro APR offer for 18 billing cycles >

Recording time	21.9 s
Fixation time	19.5 s
Gaze count	534
Fixation count	19

(**b**)

www.wellsfargo.com

WELLS FARGO Sign On MENU

Enjoy $300

$300 checking bonus on us

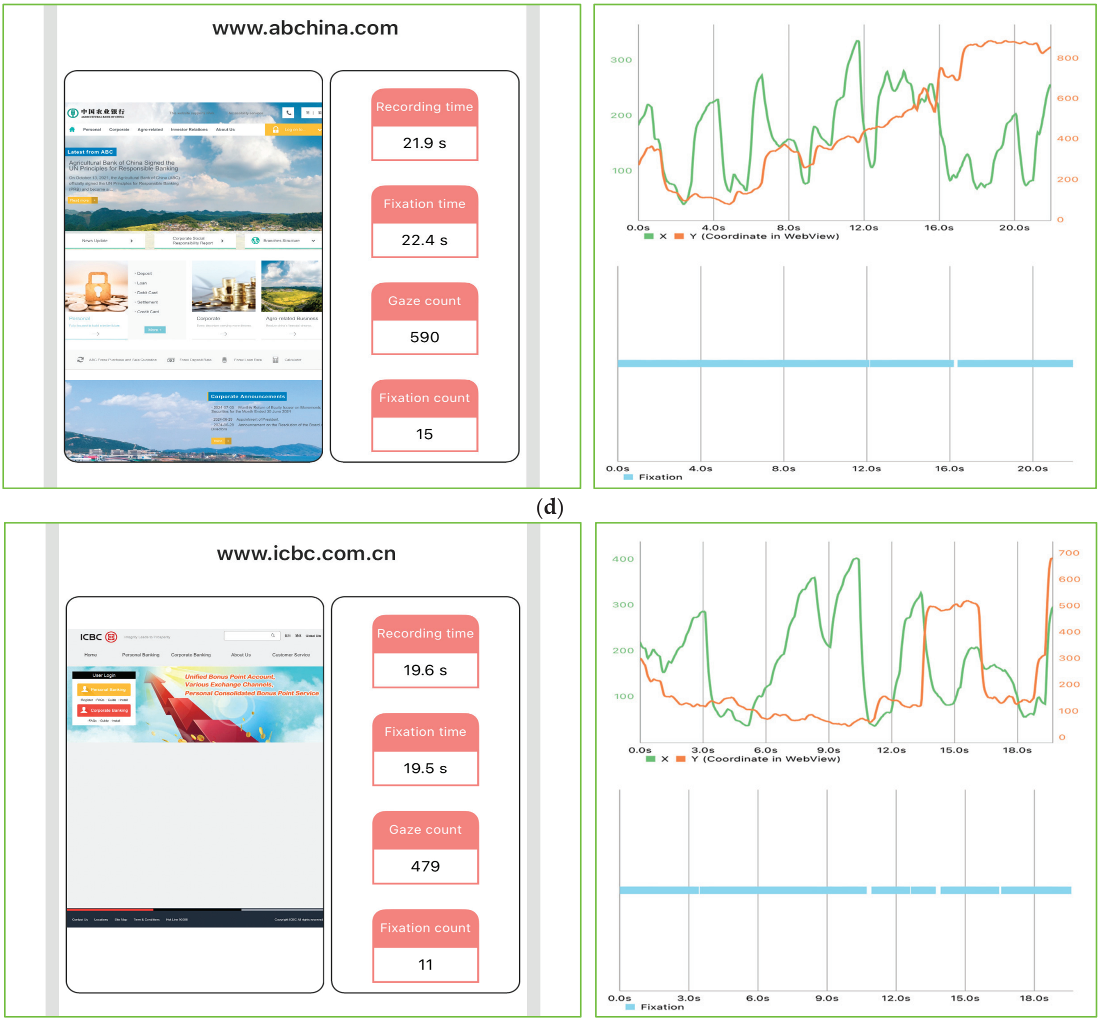
New customers open an eligible checking account with qualifying electronic deposits

Get started >>

Recording time	21.5 s
Fixation time	20.2 s
Gaze count	560
Fixation count	22

(**c**)

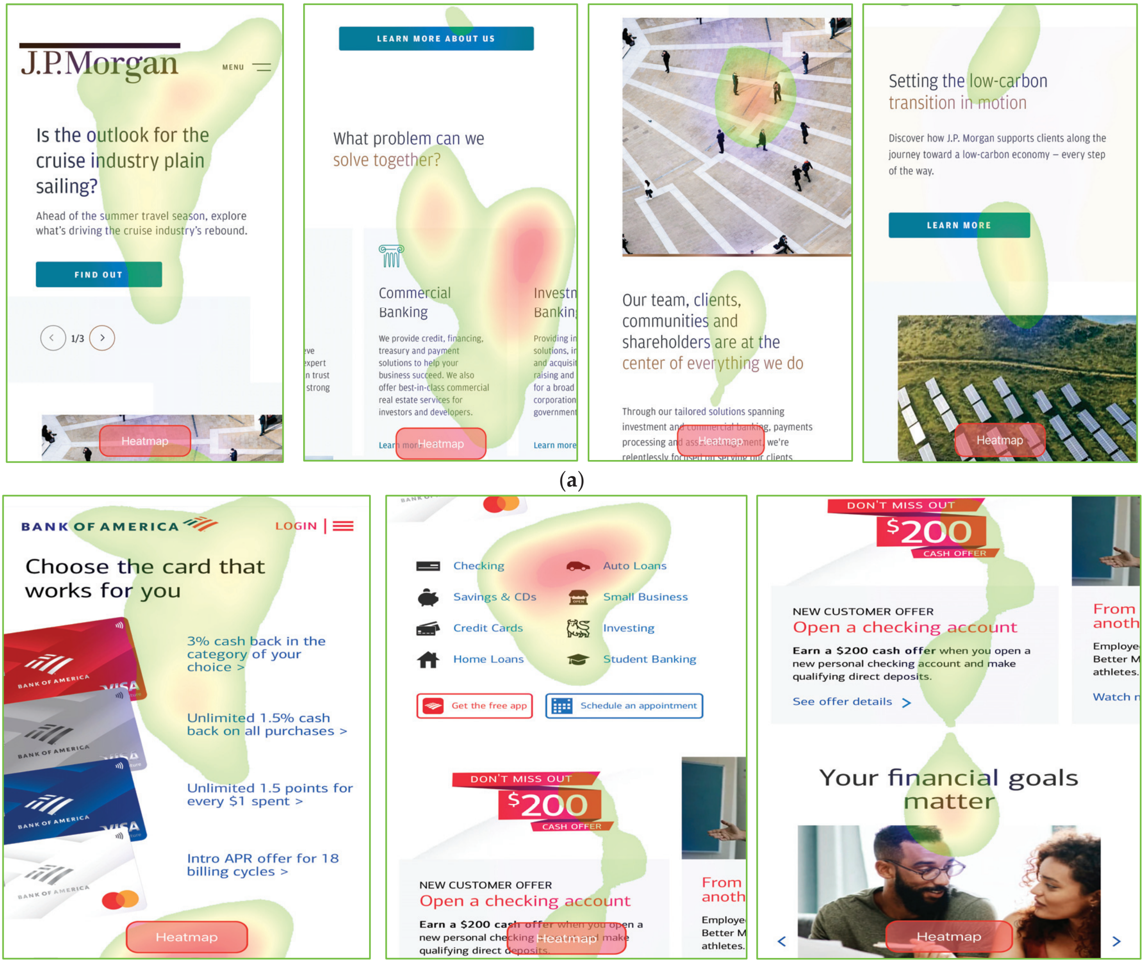
Figure 4. *Cont.*



(**d**)

(**e**)

Figure 4. Gaze and fixation analysis of banking firm websites. (**a**) JP Morgan, (**b**) Bank of America, (**c**) Wells Fargo, (**d**) Agricultural Bank of China, and (**e**) Industrial and Commercial Bank of China Limited.



J.P.Morgan

MENU

Is the outlook for the cruise industry plain sailing?

Ahead of the summer travel season, explore what's driving the cruise industry's rebound.

FIND OUT

1/3

Heatmap

LEARN MORE ABOUT US

What problem can we solve together?

Commercial Banking

We provide credit, financing, treasury and payment solutions to help your business succeed. We also offer best-in-class commercial real estate services for investors and developers.

Learn more

Heatmap

Investn Bankin

Providing in solutions, in and acquisit raising and for a broad corporation government

Learn more

Our team, clients, communities and shareholders are at the center of everything we do

Through our tailored solutions spanning investment and commercial banking, payments processing and asset management, we're relentlessly focused on serving our clients

Heatmap

Setting the low-carbon transition in motion

Discover how J.P. Morgan supports clients along the journey toward a low-carbon economy – every step of the way.

LEARN MORE

Heatmap

(**a**)

BANK OF AMERICA

LOGIN

Choose the card that works for you

3% cash back in the category of your choice >

Unlimited 1.5% cash back on all purchases >

Unlimited 1.5 points for every $1 spent >

Intro APR offer for 18 billing cycles >

Heatmap

Checking

Auto Loans

Savings & CDs

Small Business

Credit Cards

Investing

Home Loans

Student Banking

Get the free app

Schedule an appointment

DON'T MISS OUT

$200

CASH OFFER

NEW CUSTOMER OFFER

Open a checking account

Earn a $200 cash offer when you open a new personal checking account and make qualifying direct deposits.

Heatmap

From anoth

Employe Better M athletes.

See offer details >

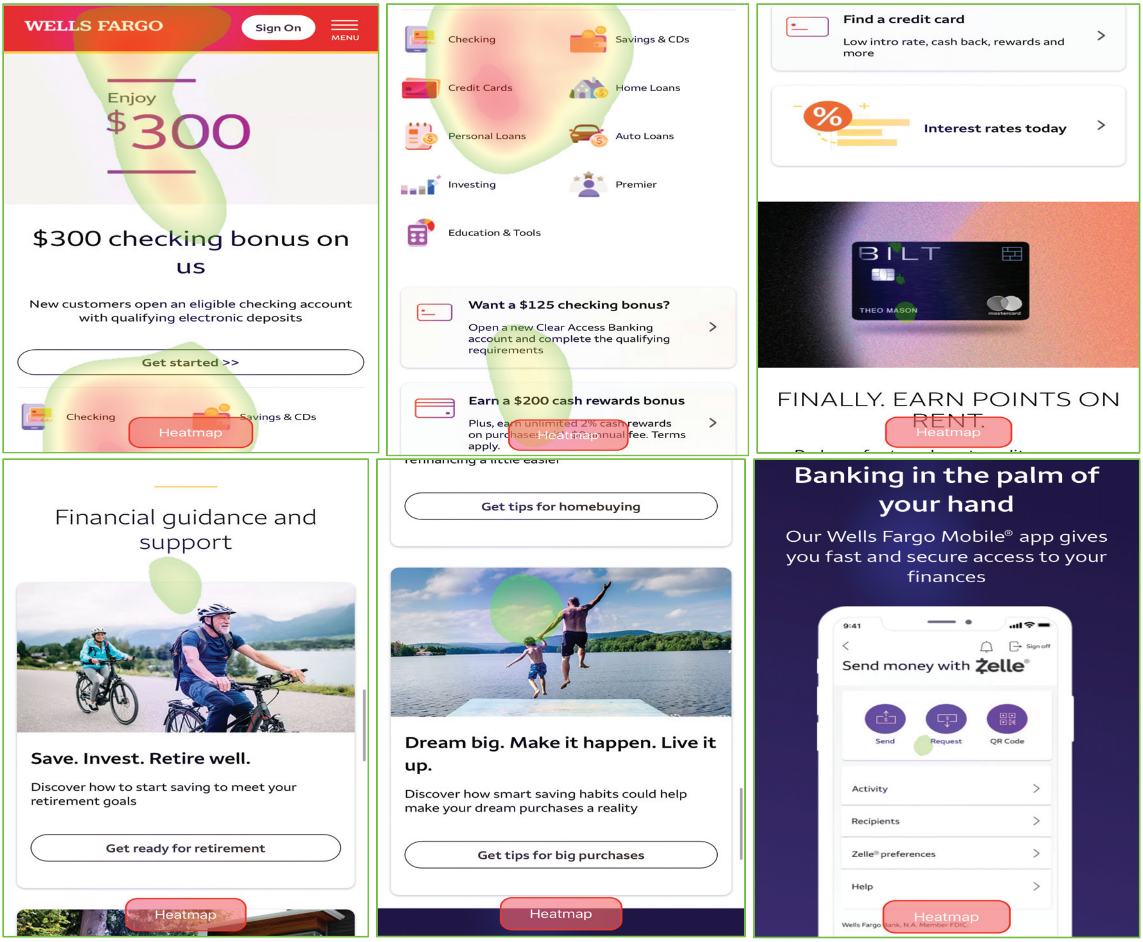
Watch n

Your financial goals matter

Heatmap

(**b**)

Figure 5. *Cont.*



WELLS FARGO Sign On MENU

Enjoy $300

$300 checking bonus on us

New customers open an eligible checking account with qualifying electronic deposits

Get started >>

Checking Savings & CDs

Heatmap

Checking Savings & CDs

Credit Cards Home Loans

Personal Loans Auto Loans

Investing Premier

Education & Tools

Want a $125 checking bonus?

Open a new Clear Access Banking account and complete the qualifying requirements

Earn a $200 cash rewards bonus

Plus, earn unlimited 2% cash rewards on purchases and no annual fee. Terms apply.

Heatmap

Find a credit card

Low intro rate, cash back, rewards and more

Interest rates today

BILT THEO MASON

FINALLY. EARN POINTS ON RENT.

Heatmap

Financial guidance and support

Save. Invest. Retire well.

Discover how to start saving to meet your retirement goals

Get ready for retirement

Heatmap

refinancing a little easier

Get tips for homebuying

Dream big. Make it happen. Live it up.

Discover how smart saving habits could help make your dream purchases a reality

Get tips for big purchases

Heatmap

Banking in the palm of your hand

Our Wells Fargo Mobile® app gives you fast and secure access to your finances

9:41

Send money with Zelle®

Send Request QR Code

Activity

Recipients

Zelle® preferences

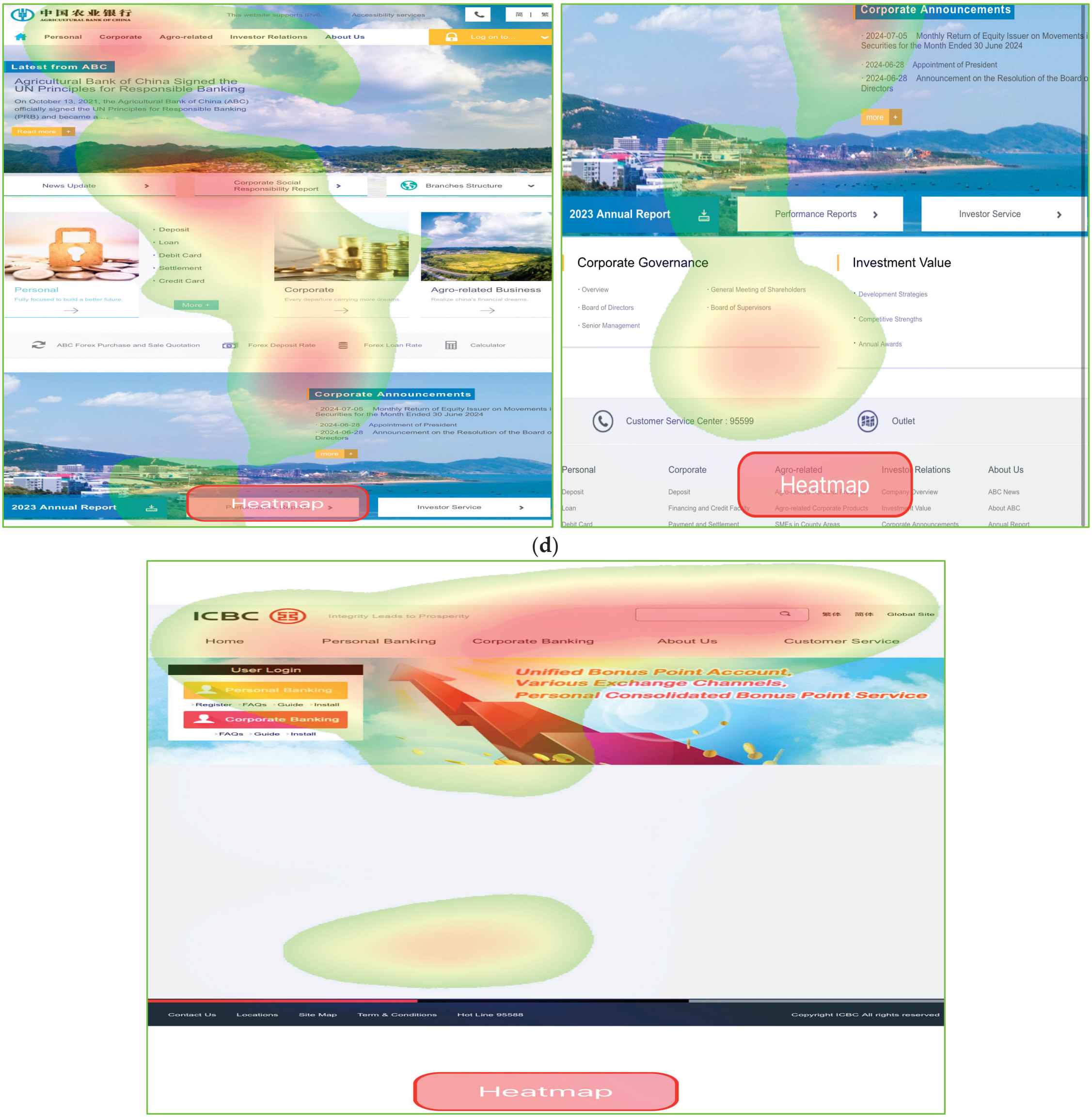
Help

Wells Fargo Bank, N.A. Member FDIC

Heatmap

(c)

Figure 5. *Cont.*



Agricultural Bank of China Signed the UN Principles for Responsible Banking

On October 13, 2021, the Agricultural Bank of China (ABC) officially signed the UN Principles for Responsible Banking (PRB) and became a ...

2024-07-05 Monthly Return of Equity Issuer on Movements in Securities for the Month Ended 30 June 2024

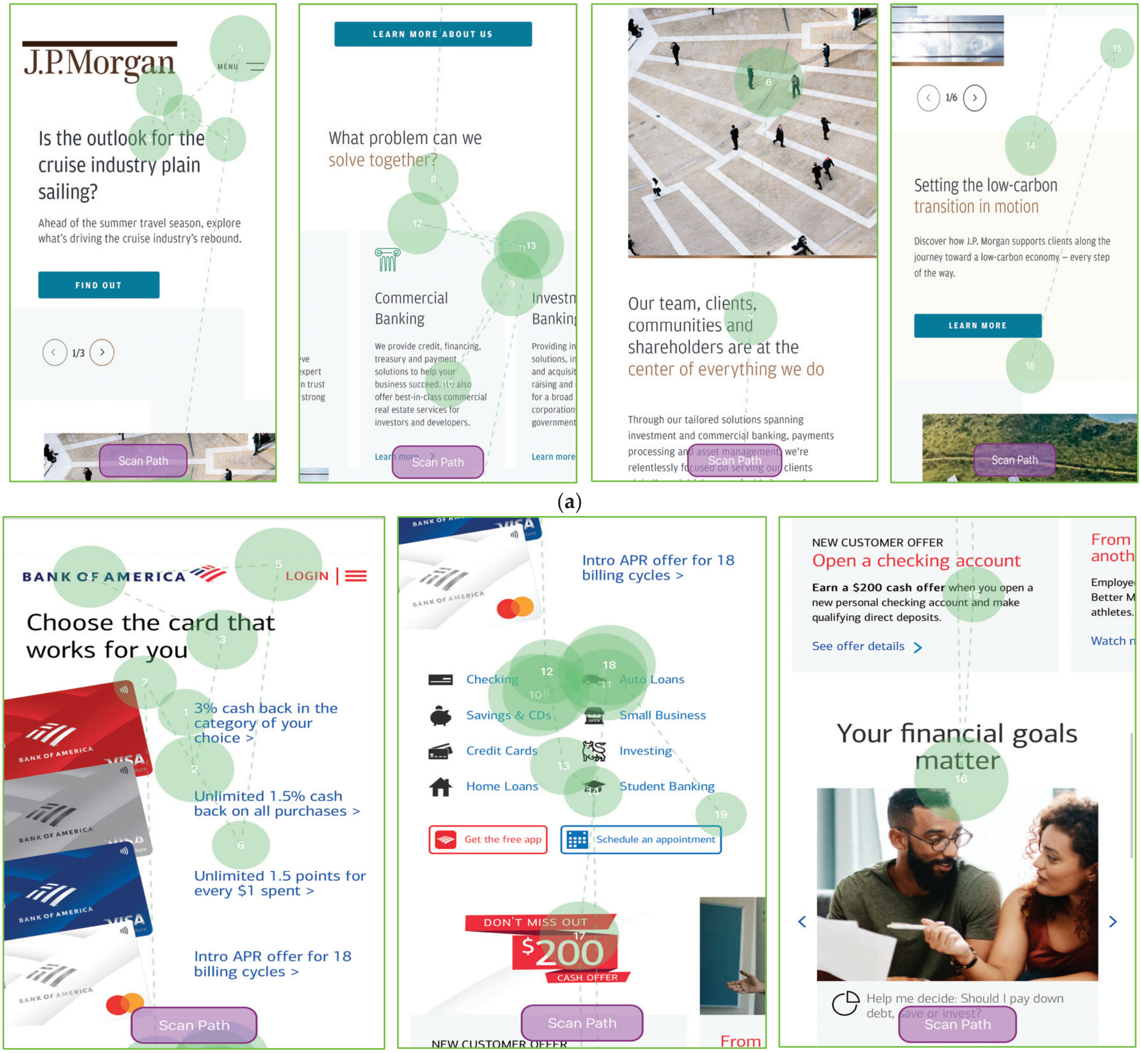
2024-06-28 Appointment of President

2024-06-28 Announcement on the Resolution of the Board of Directors

(**d**)

(**e**)

Figure 5. Heatmap analysis of banking firm websites. (**a**) JP Morgan, (**b**) Bank of America, (**c**) Wells Fargo, (**d**) Agricultural Bank of China, and (**e**) Industrial and Commercial Bank of China Limited. The intensity of the color (redder) indicates increased fixations and engagement of the participants (greener color shows reduced fixations/engagement).



J.P.Morgan

Is the outlook for the cruise industry plain sailing?

Ahead of the summer travel season, explore what's driving the cruise industry's rebound.

What problem can we solve together?

Commercial Banking

We provide credit, financing, treasury and payment solutions to help your business succeed. We also offer best-in-class commercial real estate services for investors and developers.

Learn more

Investm Bankin

Providing in solutions, in and acquisit raising and for a broad corporation government

Learn more

Our team, clients, communities and shareholders are at the center of everything we do

Through our tailored solutions spanning investment and commercial banking, payments processing and asset management, we're relentlessly focused on serving our clients

Setting the low-carbon transition in motion

Discover how J.P. Morgan supports clients along the journey toward a low-carbon economy – every step of the way.

(**a**)

BANK OF AMERICA

Choose the card that works for you

3% cash back in the category of your choice >

Unlimited 1.5% cash back on all purchases >

Unlimited 1.5 points for every $1 spent >

Intro APR offer for 18 billing cycles >

Intro APR offer for 18 billing cycles >

Checking
Auto Loans
Savings & CDs
Small Business
Credit Cards
Investing
Home Loans
Student Banking

DON'T MISS OUT $200 CASH OFFER

NEW CUSTOMER OFFER

From

NEW CUSTOMER OFFER

Open a checking account

Earn a $200 cash offer when you open a new personal checking account and make qualifying direct deposits.

See offer details >

From anoth

Employe Better M athletes.

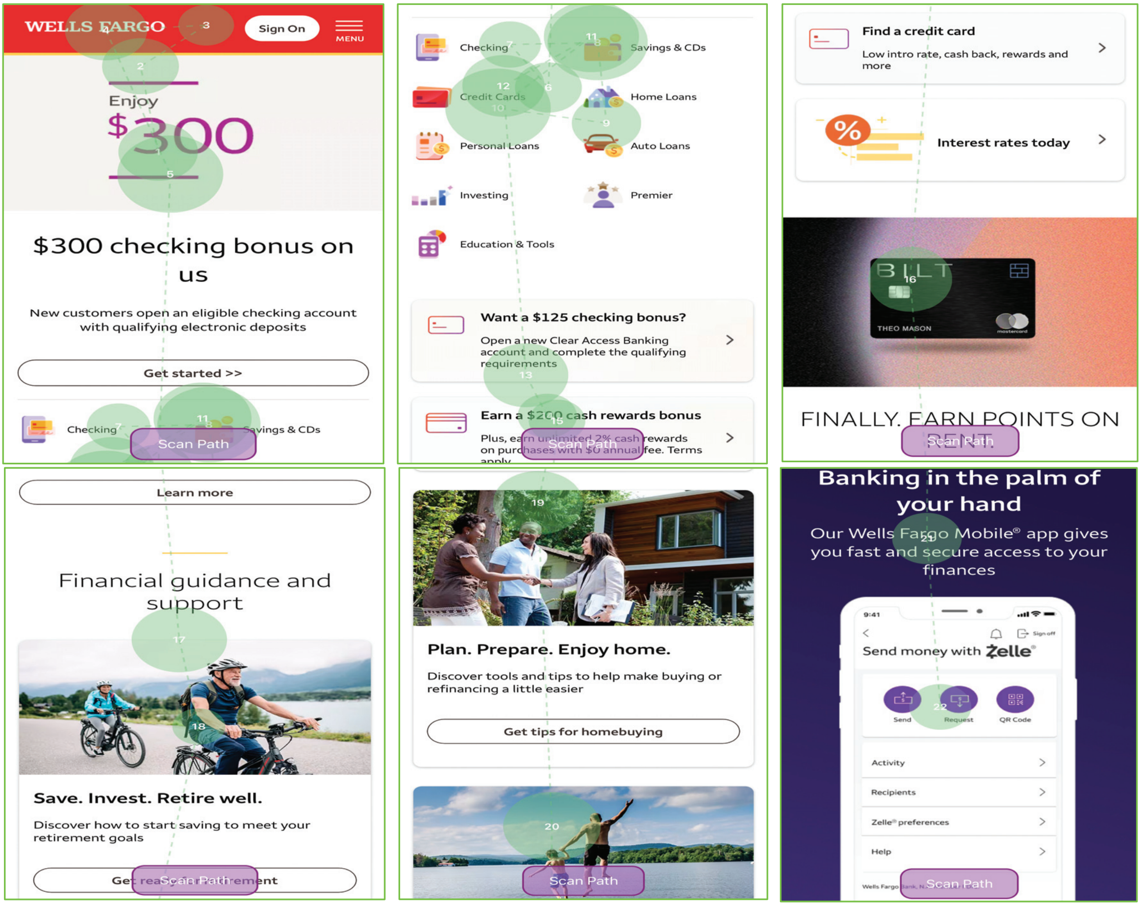
Watch n

Your financial goals matter

Help me decide: Should I pay down debt, save or invest?

(**b**)

Figure 6. *Cont.*



$300 checking bonus on us

New customers open an eligible checking account with qualifying electronic deposits

Want a $125 checking bonus?

Open a new Clear Access Banking account and complete the qualifying requirements

Earn a $200 cash rewards bonus

Plus, earn unlimited 2% cash rewards on purchases with $0 annual fee. Terms apply.

Low intro rate, cash back, rewards and more

FINALLY. EARN POINTS ON RENT.

Financial guidance and support

Save. Invest. Retire well.

Discover how to start saving to meet your retirement goals

Get ready for retirement

Plan. Prepare. Enjoy home.

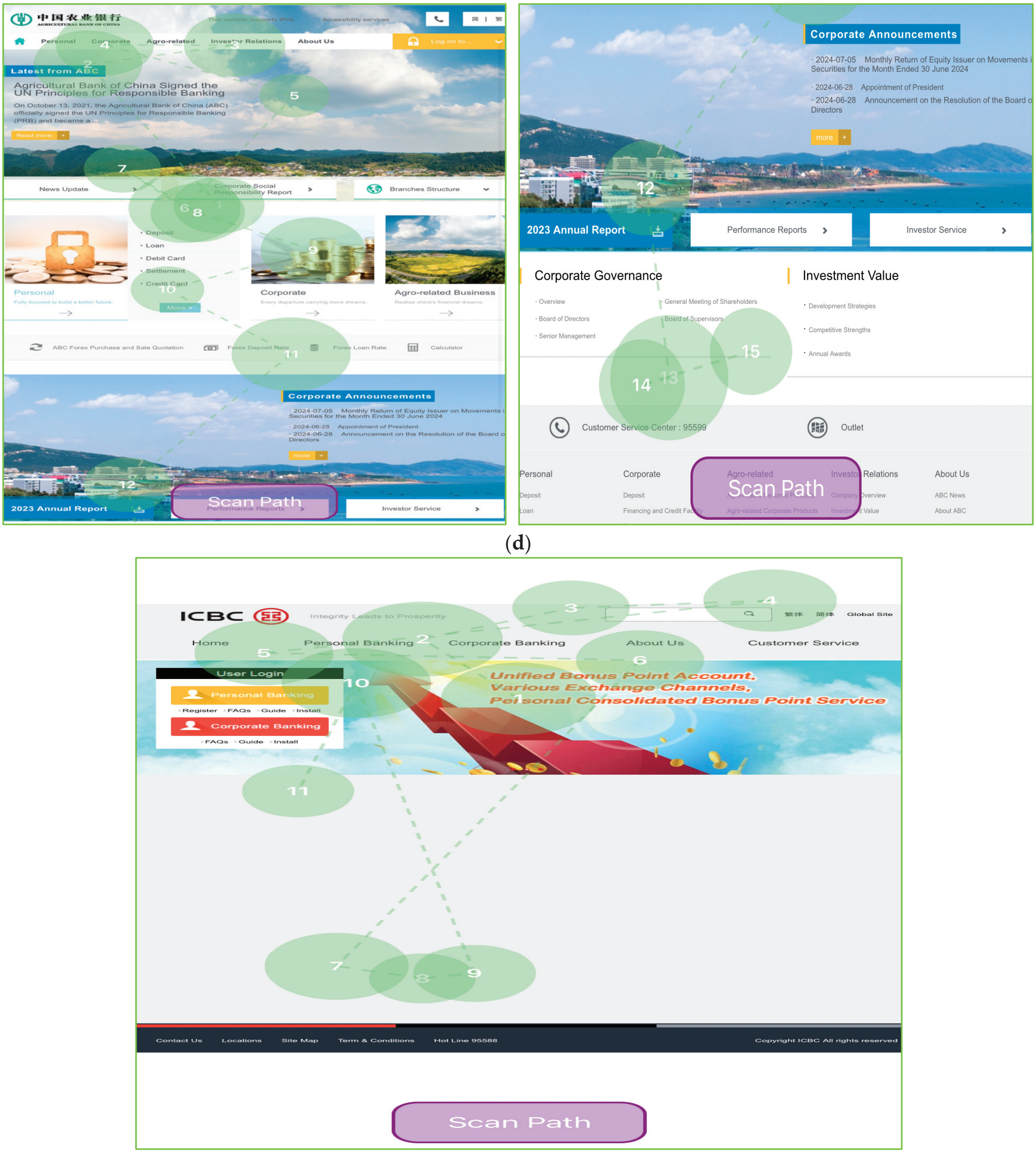
Discover tools and tips to help make buying or refinancing a little easier

Banking in the palm of your hand

Our Wells Fargo Mobile® app gives you fast and secure access to your finances

(**c**)

Figure 6. *Cont.*



(**d**)

(**e**)

Figure 6. Scan-path analysis of banking firm websites. (**a**) JP Morgan, (**b**) Bank of America, (**c**) Wells Fargo, (**d**) Agricultural Bank of China, and (**e**) Industrial and Commercial Bank of China Limited. The numbers indicate the sequence of the participants observation, while the size of the circles shows the amount of observation time spent in each spot (the bigger the circle the more time is spent).

5. Discussion

The purpose of this study has been to examine the impact of website visitors' behaviors on banking firms' website interfaces and the firms' overall digital marketing performance.

To achieve this objective, the variables of purchase conversion, bounce rate, organic traffic, and display ads as website interfaces and digital marketing performance metrics. The selected website behavioral metrics were the email source traffic, organic and paid costs, and referral domains. Statistical analyses (correlation and SLR models), hybrid model simulations (ABM and SD), and neuromarketing tests (eye-tracking) were deployed to extract the required insights.

From the produced simple linear regression models (SLR) executed in Section 4.1, it was discerned that all of the applied models were found to be verified overall (p-values < a = 0.01 level of significance). Therefore, hypotheses H1 to H4 were verified, meaning that the digital marketing analytics of banking firms significantly impact their website customers' purchase conversion, as well as the amount of display ads, organic visibility, and bounce rates on their websites. More specifically, the digital marketing analytics that were found to have a strong effect on the dependent variables (purchase conversion, display ads, organic visibility, and bounce rate) were the website organic traffic costs, paid traffic costs, referral domains, and email sources. It was discerned that purchase conversion was negatively connected with banking firms' website organic traffic costs and paid traffic costs while positively connected with their referral domains and email sources. The amount of referral domains was also negatively connected with banking firms' paid traffic costs and referral domains and positively connected with the bounce rate. Paid traffic costs were found to be positively connected with display ads and bounce rates and negatively connected with organic traffic. Email sources appear to negatively impact banking firms' website display ads and bounce rates and positively impact their organic traffic.

Regarding the simulation of the hybrid (ABM and SD) model, its outputs verify the results of SLR models and, therefore, the research hypotheses H1–H4. More specifically, through the 360-day simulation and the usage of 10,000 agents/website visitors, banking firms' email source traffic increases purchase conversion, bounce rate, and organic traffic [76] and decreases their display ads while the number of referral domains increases all of the above (conversion rate, bounce rate, organic traffic, and display ads) [77]. Organic traffic costs appear to decrease banking firms' purchase conversion and bounce rate and to increase organic traffic and display ads while paid traffic costs increase conversion rate, bounce rate, and display ads and decrease organic traffic [78].

Our research outputs mainly align with present studies in the field of digital banking and customer behavior and engagement. Mbama and Ezepue [27] identified service quality, functional quality, perceived value, employee–customer engagement, perceived usability, and perceived risk as the primary factors influencing customer experience in digital banking. They found a significant relationship between customer experience, satisfaction, loyalty, and financial performance. According to Islam et al. [29], key website attributes such as interactivity, aesthetics, customization, ease of use, and telepresence positively impact customer engagement. Their findings also indicate that customer engagement is positively associated with customer trust and retention. Hari et al. [79] found that interactivity, particularly through chatbots, enhances customer brand engagement, which in turn boosts satisfaction with the brand experience and increases customer intentions to use the brand.

When it comes to the neuromarketing model, the 26 participants who are related to banking firms' services/products showed valuable insights. The application of neuromarketing tools, including eye-tracking, heatmaps, and scan-path analysis, provided critical insights into how visitors interact with banking firm websites, offering substantial implications for interface and performance enhancement. The study revealed that websites with longer pages, requiring more time for visitors to scroll, resulted in fewer continuous fixations and shorter fixation times compared to shorter pages, which held visitor attention longer. Despite this, longer pages had a higher number of total fixations and gazes, suggesting a more fragmented but extensive interaction with the content. Heatmap analysis demonstrated that participants consistently focused on menus, product information, logos, and display ads, indicating these elements are crucial for user engagement. Scan-path analysis confirmed these findings, showing that users typically begin their navigation with the brand logo, followed by menu options, product information, and display ads, with attention diminishing towards the end of the page. These insights directly address the second research question (RQ2), by highlighting that customer behavior can indeed enhance the website interface. By understanding these behavioral patterns, banking firms can optimize their website design to improve user engagement, satisfaction, and overall performance, confirming that customer behavior analysis is vital for enhancing website interfaces.

6. Conclusions

6.1. Theoretical, Practical, and Managerial Implications

In this part of the study, the main theoretical and practical implications for the research findings are presented and analyzed. This paper examined the implication of big data analysis (website analytic metrics) and neuromarketing models in improving banking firms' website interfaces and performance. The results of the research extend to customer engagement and digital marketing analytic metrics since website visitor behavior is depicted by specific onsite behavioral KPIs. Moreover, the utilization of both quantitative (big data) and qualitative (neuromarketing test) analyses tends to provide a strong approach to the research by ensuring the transparency and reproducibility of the results and attempting to cover a major part of the banking firms' website customer behavior.

The integration of neuromarketing and big data analysis offers significant theoretical implications for enhancing banking firms' website interfaces and overall performance. The application of simple linear regression models (SLR) demonstrates that digital marketing analytics profoundly impact customer behaviors, such as the purchase of conversion rates, the display of ad interactions, organic visibility, and bounce rates. Specifically, factors like organic traffic costs, paid traffic costs, referral domains, and email sources were identified as critical determinants [80]. These findings underscore the need for a theoretical framework that encompasses both traditional marketing metrics and cognitive–behavioral insights from neuromarketing to better understand and predict consumer behavior on banking websites.

From a practical standpoint, the insights derived from neuromarketing tools like eye-tracking, heatmaps, and scan-path analysis can significantly improve website design and functionality. For instance, the study revealed that websites with shorter pages had longer fixation times, indicating that concise and focused content is more engaging for users. Additionally, heatmap analysis highlighted areas of intense user interest, such as menus, product information, and brand logos. By strategically placing critical information and interactive elements in these high-engagement zones, banking firms can enhance user experience, reduce bounce rates, and potentially increase conversion rates [81]. These practical adjustments, informed by empirical data, can lead to more effective and user-friendly website designs.

Neuromarketing insights also offer valuable strategies for boosting customer engagement. The study's findings suggest that email source traffic and referral domains play pivotal roles in increasing purchase conversion and organic traffic [82]. By leveraging these channels more effectively, banking firms can create more personalized and targeted marketing campaigns that resonate with their audience. Moreover, the use of neuromarketing tools to track eye movements and fixation patterns can help identify which elements of a webpage capture the most attention. This information can be used to optimize content placement and design features, ensuring that users engage more deeply with the site and its offerings, thus fostering greater customer loyalty and trust.

The integration of neuromarketing and big data analytics presents substantial managerial implications for banking firms aiming to enhance their website interfaces and overall digital performance. The study's findings suggest that strategically utilizing digital marketing analytics, such as organic and paid traffic costs, referral domains, and email sources, can significantly influence key metrics like purchase conversion rates, display ad interactions, organic visibility, and bounce rates. By applying insights from neuromarketing tools, such as eye-tracking and heatmap analysis, banks can optimize website design to maintain user attention, strategically place critical information in high-engagement zones, and create more personalized marketing campaigns. This dual approach not only enhances user experience and reduces bounce rates but also increases customer loyalty and trust, ultimately driving higher conversion rates and improving overall performance.

Our study's findings are aligned with present research in the field of banking firms' website performance and interface enhancement. As referenced by Müller et al. [83], a firm's performance in technology-intensive sectors is strongly connected to the utilization of big data analysis. By analyzing big data from a website's user activity, banking firms can have multiple benefits, such as stock market prediction, etc. [84]. Furthermore, big data can also assist the digital marketing efforts of firms and increase their knowledge of the market [85]; meanwhile, Ravi and Kamaruddin [86] stated that big data could aid in solving multiple banking firms' problems to enhance their overall performance.

For the utilization of neuromarketing applications, this study is aligned with other relevant research in the field. More specifically, Berčík et al. [48] noted that the implications of neuromarketing models tend to reveal detailed information about customers' behavior that can be used in digital marketing management and communication promotion. In accordance with Tichindelean et al.'s [87] study, we showed that the design of the webpages of banking firms has a great impact on webpage usability, customer engagement, and, thus, the website interface. Customers' behaviors are linked to their reaction to advertising processes [88] and show ways of enhancing their engagement. Since neuromarketing tools help marketers understand their customer's behavior [89], banking firms' customers were found to have a greater engagement with the brand if they were familiar with their services/products [90].

6.2. Future Work and Limitations

Future research should consider expanding the scope beyond YouTube metrics to include a broader range of digital marketing channels and qualitative methods. Longitudinal studies and experimental designs could provide deeper insights into the long-term effects of specific marketing strategies and the impact of demographic factors on user engagement. Additionally, exploring the synergies between different digital marketing activities, such as social media marketing and influencer partnerships, could offer a more holistic understanding of digital marketing effectiveness. By continuously refining methodologies and utilizing new tools and technologies, researchers and practitioners can better navigate the evolving landscape of digital marketing in the DeFi sector and beyond.

The study's focus on specific YouTube metrics and web analytics may limit the generalizability of its findings to other contexts or platforms within the DeFi ecosystem. Variations in platform features, user demographics, and market dynamics could affect the relevance of the insights. Additionally, the reliance on quantitative methods might overlook qualitative aspects of digital marketing effectiveness that cannot be captured solely through numerical data. The exclusive focus on YouTube metrics may ignore the impact of other digital marketing channels or strategies on the performance of DeFi platforms. Moreover, the study's methodology might not fully account for the continuous improvements in Google's algorithms, which could influence website and YouTube rankings over time.

Author Contributions: Conceptualization, N.T.G., D.P.S. and S.P.M.; methodology, N.T.G., D.P.S. and S.P.M.; software, N.T.G., D.P.S. and S.P.M.; validation, N.T.G., D.P.S. and S.P.M.; formal analysis, N.T.G., D.P.S. and S.P.M.; investigation, N.T.G., D.P.S. and S.P.M.; resources, N.T.G., D.P.S. and S.P.M.; data curation, N.T.G., D.P.S. and S.P.M.; writing—original draft preparation, N.T.G., D.P.S. and S.P.M.; writing—review and editing, N.T.G., D.P.S. and S.P.M.; visualization, N.T.G., D.P.S. and S.P.M.; supervision, N.T.G., D.P.S. and S.P.M.; project administration, N.T.G., D.P.S. and S.P.M.; funding acquisition, N.T.G., D.P.S. and S.P.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Java code for banking firms' modeling simulation.

Java Code of AnyLogic Simulation

```
@AnyLogicInternalCodegenAPI
   private void enterState(statechart_state self, boolean_destination) {
     switch( self ) {
       case Potential_Bank_Customers:
           logToDBEnterState(statechart, self);
         // (Simple state (not composite))
         statechart.setActiveState_xjal(Potential_Bank_Customers);
         transition1.start();
         transition2.start();
         return;
       case Return_Visitors:
           logToDBEnterState(statechart, self);
         // (Simple state (not composite))
         statechart.setActiveState_xjal(Return_Visitors);
         {
return_Visitors++;

pages_per_Visit = normal(0.97, 3.43);

visit_Duration = normal(128.25/60, 519.40/60);

referral_Domains = normal(794.22, 51,181.91);

email_Sources = normal(300,170.77, 184,876.14)
;}
         transition3.start();
         transition5.start();
         return;
       case Bounce_Rate:
           logToDBEnterState(statechart, self);
         // (Simple state (not composite))
         statechart.setActiveState_xjal(Bounce_Rate);
         {
bounce_Rate = organic_Traffic*(1.045) + paid_Costs*(0.025) + referral_Domains*(0.334) +
email_Sources*(−0.043)
;}
         transition.start();
         return;
       case Visitors_To_Traffic:
           logToDBEnterState(statechart, self);
         // (Simple state (not composite))
         statechart.setActiveState_xjal(Visitors_To_Traffic);
         transition7.start();
         transition8.start();
         return;
       case Organic_Traffic:
           logToDBEnterState(statechart, self);
         // (Simple state (not composite))
         statechart.setActiveState_xjal(Organic_Traffic);
         {
organic_Costs = normal(5,822,486.64, 37,155,781.98);

organic_Traffic = paid_Costs*(−0.024) + referral_Domains*(−0.319) + email_Sources*(0.041)
;}
```

Table A1. *Cont.*

Java Code of AnyLogic Simulation

```
        transition13.start();
        return;
      case Display_Ads:
          logToDBEnterState(statechart, self);
        // (Simple state (not composite))
        statechart.setActiveState_xjal(Display_Ads);
        {
display_Ads = paid_Costs*(0.198) + referral_Domains*(−0.065) + email_Sources*(−0.135)
;}
        transition10.start();
        transition11.start();
        return;
      case Purchase_Convertion:
          logToDBEnterState(statechart, self);
        // (Simple state (not composite))
        statechart.setActiveState_xjal(Purchase_Convertion);
        {
purchase_Convertion = organic_Costs*(−1.670) + paid_Costs*(−1.369) + referral_Domains*(1.696)
+ email_Sources*(0.167)
;}
        transition9.start();
        return;
      case Paid_Traffic:
          logToDBEnterState(statechart, self);
        // (Simple state (not composite))
        statechart.setActiveState_xjal(Paid_Traffic);
        {
paid_Costs = normal(406,005.96, 1,514,463.27);

paid_Traffic = normal(666.9666, 3378.9857)
;}
        transition14.start();
        return;
      case New_Visitors:
          logToDBEnterState(statechart, self);
        // (Simple state (not composite))
        statechart.setActiveState_xjal(New_Visitors);
        {
new_Visitors++;

pages_per_Visit = normal(0.97, 3.43);

visit_Duration = normal(128.25/60, 519.40/60);

referral_Domains = normal(794.22, 51,181.91);

email_Sources = normal(300,170.77, 184,876.14)
;}
        transition4.start();
        transition6.start();
        return;
      default:
        return;
    }
  }
```

References

1. Hennig-Thurau, T.; Malthouse, E.C.; Friege, C.; Gensler, S.; Lobschat, L.; Rangaswamy, A.; Skiera, B. The impact of new media on customer relationships. *J. Serv. Res.* **2010**, *13*, 311–330. [CrossRef]
2. Broby, D. Financial technology and the future of banking. *Financ. Innov.* **2021**, *7*, 1–19. [CrossRef]
3. Ding, Q.; He, W. Digital transformation, monetary policy and risk-taking of banks. *Financ. Res. Lett.* **2023**, *55*, 103986. [CrossRef]
4. Shukla, S. Analyzing customer engagement through e-CRM: The role of relationship marketing in the era of digital banking in Varanasi banks. *J. Commer. Econ. Comput. Sci.* **2021**, *7*, 57–65.
5. Hendriyani, C.; Raharja, S.J. Analysis building customer engagement through eCRM in the era of digital banking in Indonesia. *Int. J. Econ. Policy Emerg. Econ.* **2018**, *11*, 479–486.
6. Vivek, S.D.; Beatty, S.E.; Morgan, R.M. Customer engagement: Exploring customer relationships beyond purchase. *J. Mark. Theory Pract.* **2012**, *20*, 122–146. [CrossRef]

7. Lee, D.; Hosanagar, K.; Nair, H.S. Advertising content and consumer engagement on social media: Evidence from Facebook. *Manag. Sci.* **2018**, *64*, 5105–5131. [CrossRef]
8. Lin, K.-Y.; Lu, H.-P. Why people use social networking sites: An empirical study integrating network externalities and motivation theory. *Comput. Hum. Behav.* **2011**, *27*, 1152–1161. [CrossRef]
9. Lee, M.; Wang, Y.R.; Huang, C.F. Design and development of a friendly user interface for building construction traceability system. *Microsyst. Technol.* **2021**, *27*, 1773–1785. [CrossRef]
10. Faghih, B.; Azadehfar, M.; Katebi, S. User interface design for E-learning software. *Int. J. Soft Comput. Softw. Eng.* **2014**, *3*, 786–794. [CrossRef]
11. Cheng, S.; Yang, Y.; Xiu, L.; Yu, G. Effects of prior experience on the user experience of news aggregation app's features—Evidence from a behavioral experiment. *Int. J. Hum.-Comput. Interact.* **2022**, *39*, 1271–1279. [CrossRef]
12. Nielsen, J.; Norman, D. *The Definition of User Experience (UX)*; Nielsen Norman Group N N/g.: Fremont, CA, USA, 2018. Available online: https://www.nngroup.com/articles/definition-user-experience/ (accessed on 20 June 2024).
13. He, W.; Hung, J.-L.; Liu, L. Impact of big data analytics on banking: A case study. *J. Enterp. Inf. Manag.* **2023**, *36*, 459–479. [CrossRef]
14. Kalaganis, F.P.; Georgiadis, K.; Oikonomou, V.P.; Laskaris, N.A.; Nikolopoulos, S.; Kompatsiaris, I. Unlocking the Subconscious Consumer Bias: A Survey on the Past, Present, and Future of Hybrid EEG Schemes in Neuromarketing. *Front. Neuroergonomics* **2021**, *2*, 672982. [CrossRef]
15. Walker, P.R. How Does Website Design in the e-Banking Sector Affect Customer Attitudes and Behaviour? Ph.D. Thesis, University of Northumbria, Newcastle upon Tyne, UK, 2021. Available online: https://nrl.northumbria.ac.uk/id/eprint/5849/7/walker.philip_phd_(VOLUME_1of2).pdf (accessed on 12 June 2024).
16. Manser Payne, E.H.; Peltier, J.; Barger, V.A. Enhancing the value co-creation process: Artificial intelligence and mobile banking service platforms. *J. Res. Interact. Mark.* **2021**, *15*, 68–85. [CrossRef]
17. Diener, F.; Špacek, M. Digital transformation in banking: A managerial perspective on barriers to change. *Sustainability* **2021**, *13*, 2032. [CrossRef]
18. Khattak, M.A.; Ali, M.; Azmi, W.; Rizvi, S.A.R. Digital transformation, diversification and stability: What do we know about banks? *Econ. Anal. Policy* **2023**, *78*, 122–132. [CrossRef]
19. Giannakis-Bompolis, C.; Boutsouki, C. Customer Relationship Management in the Era of Social Web and Social Customer: An Investigation of Customer Engagement in the Greek Retail Banking Sector. *Procedia Soc. Behav. Sci.* **2014**, *148*, 67–78. [CrossRef]
20. Mogaji, E. Redefining banks in the digital era: A typology of banks and their research, managerial and policy implications. *Int. J. Bank Mark.* **2023**, *41*, 1899–1918. [CrossRef]
21. Salvi, A.; Petruzzella, F.; Raimo, N.; Vitolla, F. Transparency in the digitalization choices and the cost of equity capital. *Qual. Res. Financ. Mark.* **2023**, *15*, 630–646. [CrossRef]
22. Carmona, J.; Cruz, C. Banks' social media goals and strategies. *J. Bus. Res.* **2018**, *91*, 31–41. [CrossRef]
23. Kosiba, J.P.; Boateng, H.; Okoe, A.F.; Hinson, R. Trust and customer engagement in the banking sector in Ghana. *Serv. Ind. J.* **2018**, *40*, 960–973. [CrossRef]
24. Del Sarto, N.; Bocchialini, E.; Gai, L.; Ielasi, F. Digital banking: How social media is shaping the game. *Qual. Res. Financ. Mark.* **2024**. *ahead of print*. [CrossRef]
25. Sakas, D.P.; Giannakopoulos, N.T.; Trivellas, P. Exploring affiliate marketing's impact on customers' brand engagement and vulnerability in the online banking service sector. *Int. J. Bank Mark.* **2023**, *42*, 1282–1312. [CrossRef]
26. Sakas, D.P.; Giannakopoulos, N.T.; Terzi, M.C.; Kamperos, I.D.G.; Kanellos, N. What is the connection between Fintechs' video marketing and their vulnerable customers' brand engagement during crises? *Int. J. Bank Mark.* **2023**, *42*, 1313–1347. [CrossRef]
27. Mbama, C.I.; Ezepue, P.O. Digital banking, customer experience and bank financial performance: UK customers' perceptions. *Int. J. Bank Mark.* **2018**, *36*, 230–255. [CrossRef]
28. Khandelwal, R.; Kapoor, D. The Use of Digital Tools for Customer Engagement in the Financial Services Sector. In *Revolutionizing Customer-Centric Banking through ICT*; IGI Global: Hershey, PA, USA, 2024; pp. 29–55.
29. Islam, J.U.; Shahid, S.; Rasool, A.; Rahman, Z.; Khan, I.; Rather, R.A. Impact of website attributes on customer engagement in banking: A solicitation of stimulus-organism-response theory. *Int. J. Bank Mark.* **2020**, *38*, 1279–1303. [CrossRef]
30. Lestari, D.M.; Hardianto, D.; Hidayanto, A.N. Analysis of user experience quality on responsive web design from its informative perspective. *Int. J. Softw. Eng. Appl.* **2014**, *8*, 53–62. [CrossRef]
31. Almeida, F.; Monteiro, J. Approaches and principles for UX web experiences: A case study approach. *Int. J. Inf. Technol. Web Eng.* **2017**, *12*, 49–65. [CrossRef]
32. Walsh, T.A.; Kapfhammer, G.M.; McMinn, P. Automated layout failure detection for responsive web pages without an explicit oracle. In Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis, Santa Barbara, CA, USA, 10–14 July 2017. [CrossRef]
33. Rogers, Y.; Sharp, H.; Preece, J. *Interaction Design: Beyond Human-Computer Interaction*, 6th ed.; John Wiley & Sons Ltd.: New York, NY, USA, 2023.
34. *ISO9241-11*; Ergonomics of Human-System Interaction–Part 11: Usability for Definition and Concept. ISO: Geneva, Switzerland, 2018.

35. Hussain, I.; Khan, I.A.; Jadoon, W.; Jadoon, R.N.; Khan, A.N.; Shafi, M. Touch or click friendly: Towards adaptive user interfaces for complex applications. *PLoS ONE* **2024**, *19*, e0297056. [CrossRef] [PubMed]
36. Kim, S.; Cho, D. Technology Trends for UX/UI of Smart Contents. *Korea Contents Assoc. Rev.* **2016**, *14*, 29–33. [CrossRef]
37. Joo, H.S. A Study on UI/UX and Understanding of Computer Major Students. *Int. J. Adv. Smart Converg.* **2017**, *6*, 26–32.
38. Von Saucken, C.; Michailidou, I.; Lindemann, U. How to Design Experiences: Macro UX versus Micro UX Approach. *Lect. Notes Comuter Sci.* **2013**, *8015*, 130–139.
39. Instatus. Our Comprehensive List of Website Performance Metrics to Monitor. 2024. Available online: https://instatus.com/blog/website-performance-metrics (accessed on 20 June 2024).
40. Levrini, G.R.; Jeffman dos Santos, M. The influence of Price on purchase intentions: Comparative study between cognitive, sensory, and neurophysiological experiments. *Behav. Sci.* **2021**, *11*, 16. [CrossRef] [PubMed]
41. Gabriel, D.; Merat, E.; Jeudy, A.; Cambos, S.; Chabin, T.; Giustiniani, J.; Haffen, E. Emotional effects induced by the application of a cosmetic product: A real-time electrophysiological evaluation. *Appl. Sci.* **2021**, *11*, 4766. [CrossRef]
42. Filipović, F.; Baljak, L.; Naumović, T.; Labus, A.; Bogdanović, Z. Developing a web application for recognizing emotions in neuromarketing. In *Marketing and Smart Technologies*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 297–308.
43. Lee, N.; Broderick, A.J.; Chamberlain, L. What is 'neuromarketing'? A discussion and agenda for future research. *Int. J. Psychophysiol.* **2007**, *63*, 199–204. [CrossRef] [PubMed]
44. Rawnaque, F.; Rahman, K.; Anwar, S.; Vaidyanathan, R.; Chau, T.; Sarker, F.; Mamun, K. Technological advancements and opportunities in Neuromarketing: A systematic review. *Brain Inform.* **2020**, *7*, 10. [CrossRef] [PubMed]
45. Ariely, D.; Berns, G. Neuromarketing: The hope and hype of neuroimaging in business. *Nat. Rev. Neurosci.* **2010**, *11*, 284–292. [CrossRef]
46. Sousa, J. *Neuromarketing and Big Data Analytics for Strategic Consumer Engagement: Emerging Research and Opportunities*; IGI Global: Hershey, PA, USA, 2017. [CrossRef]
47. Šola, H.M.; Qureshi, F.H.; Khawaja, S. Exploring the Untapped Potential of Neuromarketing in Online Learning: Implications and Challenges for the Higher Education Sector in Europe. *Behav. Sci.* **2024**, *14*, 80. [CrossRef]
48. Berčík, J.; Neomániová, K.; Gálová, J. Using neuromarketing to understand user experience with the website (UX) and interface (UI) of a selected company. In *The Poprad Economic and Management Forum 2021, Conference Proceedings from International Scientific Conference, Poprad, Slovak Republic, 14 October 2021*; Madzík, P., Janošková, M., Eds.; VERBUM: Ružomberok, Slovakia, 2021; pp. 246–254.
49. Golnar-Nik, P.; Farashi, S.; Safari, M. The application of EEG power for the prediction and interpretation of consumer decision-making: A neuromarketing study. *Physiol. Behav.* **2019**, *207*, 90–98. [CrossRef]
50. Uygun, Y.; Oguz, R.F.; Olmezogullari, E.; Aktas, M.S. On the Large-scale Graph Data Processing for User Interface Testing in Big Data Science Projects. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 2049–2056. [CrossRef]
51. Li, L.; Zhang, J. Research and Analysis of an Enterprise E-Commerce Marketing System under the Big Data Environment. *J. Organ. End User Comput.* **2021**, *33*, 1–19. [CrossRef]
52. Sakas, D.P.; Giannakopoulos, N.T.; Terzi, M.C.; Kanellos, N.; Liontakis, A. Digital Transformation Management of Supply Chain Firms Based on Big Data from DeFi Social Media Profiles. *Electronics* **2023**, *12*, 4219. [CrossRef]
53. Bala, M.; Verma, D. A Critical Review of Digital Marketing. *Int. J. Manag. IT Eng.* **2018**, *8*, 321–339. Available online: https://ssrn.com/abstract=3545505 (accessed on 20 July 2024).
54. Pongpaew, W.; Speece, M.; Tiangsoongnern, L. Social presence and customer brand engagement on Facebook brand pages. *J. Prod. Brand Manag.* **2017**, *26*, 262–281. [CrossRef]
55. Chaffey, D.; Ellis-Chadwick, F. *Digital Marketing*; Pearson: London, UK, 2019.
56. Dodson, I. *The Art of Digital Marketing: The Definitive Guide to Creating Strategic, Targeted, and Measurable Online Campaigns*; John Wiley & Sons: New York, NY, USA, 2016.
57. Chawla, Y.; Chodak, G. Social media marketing for businesses: Organic promotions of web-links on Facebook. *J. Bus. Res.* **2021**, *135*, 49–65. [CrossRef]
58. McIlwain, C.D. Algorithmic Discrimination: A Framework and Approach to Auditing & Measuring the Impact of Race-Targeted Digital Advertising. *PolicyLink Rep.* **2023**, 1–50. [CrossRef]
59. Mladenović, D.; Rajapakse, A.; Kožuljević, N.; Shukla, Y. Search engine optimization (SEO) for digital marketers: Exploring determinants of online search visibility for blood bank service. *Online Inf. Rev.* **2023**, *47*, 661–679. [CrossRef]
60. Wedel, M.; Kannan, P.K. Marketing analytics for data-rich environments. *J. Mark.* **2016**, *80*, 97–121. [CrossRef]
61. Semrush. 2024. Available online: https://www.semrush.com/ (accessed on 12 April 2024).
62. Anylogic. 2024. Available online: https://www.anylogic.com/ (accessed on 12 April 2024).
63. SeeSo Web Analysis (Eyedid SDK). 2024. Available online: https://sdk.eyedid.ai/ (accessed on 20 April 2024).
64. MentalModeler. 2024. Available online: https://dev.mentalmodeler.com/ (accessed on 10 April 2024).
65. Migkos, S.P.; Sakas, D.P.; Giannakopoulos, N.T.; Konteos, G.; Metsiou, A. Analyzing Greece 2010 Memorandum's Impact on Macroeconomic and Financial Figures through FCM. *Economies* **2022**, *10*, 178. [CrossRef]
66. Mpelogianni, V.; Groumpos, P.P. Re-approaching fuzzy cognitive maps to increase the knowledge of a system. *AI Soc.* **2018**, *33*, 175–188. [CrossRef]

67. Forbes India. The 10 Largest Banks in the World in 2024. 2024. Available online: https://www.forbesindia.com/article/explainers/the-10-largest-banks-in-the-world/86967/1 (accessed on 6 January 2024).
68. Nugroho, S.; Uehara, T. Systematic Review of Agent-Based and System Dynamics Models for Social-Ecological System Case Studies. *Systems* **2023**, *11*, 530. [CrossRef]
69. McGarraghy, S.; Olafsdottir, G.; Kazakov, R.; Huber, É.; Loveluck, W.; Gudbrandsdottir, I.Y.; Čechura, L.; Esposito, G.; Samoggia, A.; Aubert, P.-M.; et al. Conceptual System Dynamics and Agent-Based Modelling Simulation of Interorganisational Fairness in Food Value Chains: Research Agenda and Case Studies. *Agriculture* **2022**, *12*, 280. [CrossRef]
70. Wang, H.; Shi, W.; He, W.; Xue, H.; Zeng, W. Simulation of urban transport carbon dioxide emission reduction environment economic policy in China: An integrated approach using agent-based modelling and system dynamics. *J. Clean. Prod.* **2023**, *392*, 136221. [CrossRef]
71. Nguyen, L.K.N.; Howick, S.; Megiddo, I. A framework for conceptualising hybrid system dynamics and agent-based simulation model. *Eur. J. Oper. Res.* **2024**, *315*, 1153–1166. [CrossRef]
72. Ezquerra, A.; Agen, F.; Bogdan Toma, R.; Ezquerra-Romano, I. Using facial emotion recognition to research emotional phases in an inquiry-based science activity. *Res. Sci. Technol. Educ.* **2023**, 1–24. [CrossRef]
73. Chen, Y.; Qin, X.; Xu, X. Visual Analysis and Recognition of Virtual Reality Resolution Based on Pupil Response and Galvanic Skin Response. In Proceedings of the 4th International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI) 2023, Guangzhou, China, 4–6 August 2023; pp. 74–83. [CrossRef]
74. Muke, P.Z.; Kozierkiewicz, A.; Pietranik, M. Investigation and Prediction of Cognitive Load During Memory and Arithmetic Tasks. In *Computational Collective Intelligence. ICCCI 2023. Lecture Notes in Computer Science*; Nguyen, N.T., Botzheim, J., Gulyás, L., Núñez, M., Treur, J., Vossen, G., Kozierkiewicz, A., Eds.; Springer: Cham, Switzerland, 2023; Volume 14162. [CrossRef]
75. Amiri, S.S.; Masoudi, M.; Asadi, S.; Karan, E.P. A Quantitative Way for Measuring the Building User Design Feedback and Evaluation. In Proceedings of the 16th International Conference on Computing in Civil and Building Engineering (ICCCBE2016), Osaka, Japan, 6–8 July 2016; pp. 1–7.
76. Wilson, L. 30-Minute Conversion Rate Optimisation Actions. In *30-Minute Website Marketing*; Emerald Publishing Limited: Leeds, UK, 2019; pp. 131–141. [CrossRef]
77. Sood, S. Leveraging Web Analytics for Optimizing Digital Marketing Strategies. In *Big Data Analytics*; Chaudhary, K., Alam, M., Eds.; CRC Press (Auerbach Publications): Boca Raton, FL, USA, 2022; pp. 173–188.
78. Drivas, I.C.; Sakas, D.P.; Giannakopoulos, G.A. Display Advertising and Brand Awareness in Search Engines: Predicting the Engagement of Branded Search Traffic Visitors. In *Business Intelligence and Modelling. IC-BIM 2019. Springer Proceedings in Business and Economics*; Sakas, D.P., Nasiopoulos, D.K., Taratuhina, Y., Eds.; Springer: Cham, Switzerland, 2021. [CrossRef]
79. Hari, H.; Iyer, R.; Sampat, B. Customer Brand Engagement through Chatbots on Bank Websites–Examining the Antecedents and Consequences. *Int. J. Hum. Comput. Interact.* **2023**, *38*, 1212–1227. [CrossRef]
80. Makrydakis, N. SEO mix 6 O's model and categorization of search engine marketing factors for websites ranking on search engine result pages. *Int. J. Res. Mark. Manag. Sales* **2024**, *6*, 18–32. [CrossRef]
81. Shankar, B. Strategies for Deep Customer Engagement. In *Nuanced Account Management*; Palgrave Macmillan: Singapore, 2018; pp. 53–99. [CrossRef]
82. Chakrabortty, K.; Jose, E. Relationship Analysis between Website Traffic, Domain Age and Google Indexed Pages of E-commerce Websites. *IIM Kozhikode Soc. Manag. Rev.* **2018**, *7*, 171–177. [CrossRef]
83. Müller, O.; Fay, M.; vom Brocke, J. The Effect of Big Data and Analytics on Firm Performance: An Econometric Analysis Considering Industry Characteristics. *J. Manag. Inf. Syst.* **2018**, *35*, 488–509. [CrossRef]
84. Pejić Bach, M.; Krstić, Ž.; Seljan, S.; Turulja, L. Text Mining for Big Data Analysis in Financial Sector: A Literature Review. *Sustainability* **2019**, *11*, 1277. [CrossRef]
85. Gupta, S.; Justy, T.; Kamboj, S.; Kumar, A.; Kristoffersen, E. Big data and firm marketing performance: Findings from knowledge-based view. *Technol. Forecast. Soc. Change* **2021**, *171*, 120986. [CrossRef]
86. Ravi, V.; Kamaruddin, S. Big Data Analytics Enabled Smart Financial Services: Opportunities and Challenges. In *Big Data Analytics. BDA 2017. Lecture Notes in Computer Science*; Reddy, P., Sureka, A., Chakravarthy, S., Bhalla, S., Eds.; Springer: Cham, Switzerland, 2017; Volume 10721, pp. 15–39. [CrossRef]
87. Tichindelean, M.T.; Cetină, I.; Orzan, G. A Comparative Eye Tracking Study of Usability—Towards Sustainable Web Design. *Sustainability* **2021**, *13*, 10415. [CrossRef]
88. Bajaj, R.; Syed, A.A.; Singh, S. Analysing applications of neuromarketing in efficacy of programmatic advertising. *J. Consum. Behav.* **2023**, *23*, 939–958. [CrossRef]
89. Tirandazi, P.; Bamakan, S.M.H.; Toghroljerdi, A. A review of studies on internet of everything as an enabler of neuromarketing methods and techniques. *J. Supercomput.* **2022**, *79*, 7835–7876. [CrossRef]
90. Slijepčević, M.; Popović Šević, N.; Radojević, I.; Šević, A. Relative Importance of Neuromarketing in Support of Banking Service Users. *Marketing* **2022**, *53*, 131–142. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Navigation Based on Hybrid Decentralized and Centralized Training and Execution Strategy for Multiple Mobile Robots Reinforcement Learning

Yanyan Dai, Deokgyu Kim and Kidong Lee *

Robotics Department, Yeungnam University, Gyeongsan 38541, Republic of Korea; yanyan_dai@ynu.ac.kr (Y.D.)
* Correspondence: kdrhee@yu.ac.kr

Abstract: In addressing the complex challenges of path planning in multi-robot systems, this paper proposes a novel Hybrid Decentralized and Centralized Training and Execution (DCTE) Strategy, aimed at optimizing computational efficiency and system performance. The strategy solves the prevalent issues of collision and coordination through a tiered optimization process. The DCTE strategy commences with an initial decentralized path planning step based on Deep Q-Network (DQN), where each robot independently formulates its path. This is followed by a centralized collision detection the analysis of which serves to identify potential intersections or collision risks. Paths confirmed as non-intersecting are used for execution, while those in collision areas prompt a dynamic re-planning step using DQN. Robots treat each other as dynamic obstacles to circumnavigate, ensuring continuous operation without disruptions. The final step involves linking the newly optimized paths with the original safe paths to form a complete and secure execution route. This paper demonstrates how this structured strategy not only mitigates collision risks but also significantly improves the computational efficiency of multi-robot systems. The reinforcement learning time was significantly shorter, with the DCTE strategy requiring only 3 min and 36 s compared to 5 min and 33 s in the comparison results of the simulation section. The improvement underscores the advantages of the proposed method in enhancing the effectiveness and efficiency of multi-robot systems.

Keywords: multiple-robot navigation; hybrid DCTE strategy; reinforcement learning; DQN; effectiveness and efficiency

Citation: Dai, Y.; Kim, D.; Lee, K. Navigation Based on Hybrid Decentralized and Centralized Training and Execution Strategy for Multiple Mobile Robots Reinforcement Learning. *Electronics* **2024**, *13*, 2927. https://doi.org/10.3390/electronics13152927

Academic Editors: Chao Zhang, Wentao Li, Huiyan Zhang and Tao Zhan

Received: 28 June 2024
Revised: 19 July 2024
Accepted: 23 July 2024
Published: 24 July 2024

1. Introduction

Navigation for multiple mobile robots is an attractive research topic, with significant implications across various fields, including autonomous vehicles [1,2], robotics [3,4], and logistics [5,6]. Multiple robots can work together to accomplish complex tasks more efficiently than a single robot. Multiple robots can significantly increase efficiency and scalability in environments such as warehouses or manufacturing plants. They can be deployed to handle different tasks simultaneously, from transporting goods to assisting in assembly processes. The multi-robot systems offer greater resilience and redundancy than a single robot system. Integrating machine learning and artificial intelligence with multi-robot systems opens new avenues for autonomous decision making and adaptive learning. Robots can learn from their environment and each other, optimizing their strategies and behaviors for improved performance over time.

Navigation for multiple mobile robots using reinforcement learning (RL) [7–12] involves teaching robots how to optimally move through their environment and interact with each other to achieve goals. This approach leverages the principles of RL, where a robot learns to make decisions through trial and error, guided by rewards for achieving desired outcomes. Implementing RL in the multi-robot system is complex due to the dynamic interactions among robots and their environment. It needs sophisticated algorithms to handle coordination, communication, and obstacle avoidance. In RL, the environment includes everything that the robots interact with, including other robots, obstacles, and the target location. It is typically modeled as a state space that the robots can observe and act upon. Each robot is considered a robot with the ability to perceive the state of the environment through sensors, make decisions based on its policy, and execute actions to change its state or the environment. The state is the situation of a robot in the environment. In this paper, it includes the robot's location and the positions of other robots and obstacles. Actions are the set of movements a robot can make, such as moving forward, moving backward, turning left, turning right, or stopping. The choice of action at each step is based on the robot's policy. Rewards provide feedback to the robot for arriving at its goal point. Penalties are given for collisions. The policy is a strategy that the robot follows to decide its actions based on the current state. The goal of RL is to learn an optimal policy that maximizes the expected cumulative reward.

Applying RL to multiple robots' navigation is important, since it enhances the capabilities and efficiency of multi-robot systems. RL enables robots to learn optimal paths in complex environments. Using trial and error, robots can discover the most efficient routes to their destinations while avoiding obstacles. This is particularly important in dynamic environments where predefined paths may not be feasible [13]. RL allows robots to make autonomous decisions based on real-time environmental data. This reduces the need for constant human supervision and intervention, making the system more efficient and scalable [14]. Robots can adapt their strategies based on the environment and the behavior of other robots. This adaptability is crucial in dynamic settings where conditions and tasks can change rapidly. RL helps robots to continuously improve their performance over time [15]. In multi-robot systems, RL facilitates coordination and collaboration among robots. Robots can learn to work together to achieve common goals, such as avoiding collisions and optimizing task allocation. This leads to higher overall system efficiency [16]. Therefore, applying RL to multiple robots' navigation is essential for developing intelligent, efficient, and adaptable robotic systems.

The strategy of applying RL in multi-robot systems is of great importance. A well-defined RL strategy ensures that robots can work together well, optimizing their collective performance and achieving complex tasks [17]. Strategies that balance centralized and decentralized learning can optimize the use of computational resources [18]. A scalable RL strategy allows for the deployment of numerous robots in various environments, from small-scale operations to large, complex settings [19]. An effective RL strategy incorporates redundancy and adaptive learning, enabling robots to recover from failures and continue operations [20]. Therefore, this paper designs and applies a Hybrid Decentralized and Centralized Training and Execution (DCTE) Strategy of RL to solve complex path planning problems.

2. Related Works

There are several strategies for RL in multi-robot navigation. In centralized training with a decentralized execution strategy [21,22], robots are trained together in a centralized manner to learn how to coordinate their actions, but they execute their learned policies independently in the real world. This approach leverages the advantages of shared learning and optimizes the collective knowledge gained during training, while allowing for flexibility and individual decision making in execution. However, training multiple robots in a centralized manner can become computationally intensive as the number of robots increases, requiring significant processing power and memory. The complexity limits scalability and is not feasible for large-scale deployments. In Multi-Robot Reinforcement Learning [23], multiple robots learn and make decisions simultaneously. This strategy considers the interactions between robots in addition to the individual robot–environment interactions. This helps in developing policies that consider not just individual success but also the interdependencies between robots, essential for tasks requiring high coordination. However, the environment's dynamics can change as each robot learns and adapts, making the learning process inherently non-stationary. Algorithms may struggle to converge to a stable policy, leading to suboptimal performance and longer training times. It can be difficult to determine which actions by which robots led to a particular outcome, especially when actions are taken simultaneously by multiple robots. This complicates the learning process and can slow down the development of effective strategies. About communication mechanisms strategy [24], establishing robust communication channels between robots allows them to share real-time data about their status and the environment, such as detected obstacles or positional data. However, maintaining real-time communication between robots can introduce significant overhead in terms of latency and bandwidth usage. In environments where split-second decisions are critical, delays in communication can lead to inefficiencies or errors. As the number of robots increases, the amount of data that needs to be exchanged can grow exponentially, making it challenging to manage effectively. This can limit the maximum number of robots that can effectively collaborate within a system. The reward shaping and shared rewards strategy [25,26] is designed to motivate not only individual achievements but also collaborative efforts. Shared rewards promote teamwork, whereas individual rewards focus robots on personal efficiency and task completion. Such a dual reward system fosters an environment where robots learn to balance personal objectives with group objectives, enhancing the overall system performance. However, designing reward functions that accurately reflect both individual and collective goals is challenging and can often require extensive tuning and domain knowledge. Poorly designed rewards can lead to undesired behaviors, where robots exploit loopholes in the reward function rather than performing the intended task. If the balance between individual and shared rewards is not well managed, robots might either become too selfish, only optimizing their own performance, or too altruistic, where they fail to effectively optimize their own tasks. This can lead to inefficient overall performance and failure to achieve the collective optimally goal.

The contribution of this paper is described as below. In order to address the disadvantages discussed above, a Hybrid Decentralized and Centralized Training and Execution (DCTE) Strategy is proposed. It provides an effective method for solving complex path planning problems. This strategy adeptly handles collision and coordination issues common in multi-robot environments. The main idea of this method is to reduce computational complexity and improve the system efficiency through staged optimization. If implementing multi-robot navigation, the algorithm complexity and computation time will affect the system's scalability. This strategy can reduce computational complexity, making it feasible to scale up the system. Simulation comparison experiments show that, with this strategy, the computation time for robots to find safe paths is significantly reduced. The strategy implementation includes six steps: initial decentralized path planning, central collision detection, dynamic planning in collision areas, path re-optimization and connection, and execution. In the initial decentralized path planning step, each robot independently learns the optimal path from the starting point to the destination considering static obstacles within a static environment based on DRL [27,28]. In the central collision detection step, after all robots find their optimal paths, the central system performs path analysis to detect potential intersection or collision risks between these paths. If the path of a robot does not intersect with the paths of other robots, the path is confirmed to be safe and can be directly used for execution. At the dynamic planning in collision areas step, if path intersections are detected, the areas where these intersections occur will be marked as collision areas. Robots in these areas need to re-plan their paths and consider other robots as dynamic obstacles to avoid obstacles. During the re-learning process, the confirmed safe path will set an expanded safety area around it, prohibiting other robots from entering, and ensuring the security of the execution path. At the path re-optimization and connection step, once the avoidance paths are determined, they are connected to the initial safe paths to form a complete, secure route for execution. At the execution step, multiple robots will track the optimized paths and detect the environment again. If there are new obstacles, the implementation will return to step 1. This strategy effectively allocates computing resources and ensures the security of the execution process by first optimizing independently and then adjusting centrally. This hybrid strategy shows a high degree of adaptability in dealing with static and dynamic obstacles, and can flexibly respond to complex and changeable actual environments. The extensive experimental validation of the proposed strategy is conducted using the Gazebo Simulator and Agilex LIMO robots equipped with onboard LiDAR and camera sensors. The results confirm the advantages of the DCTE strategy, showcasing reduced reinforcement learning time (3 min and 36 s) compared to traditional methods (5 min and 33 s) and demonstrating efficient path planning and execution.

3. Hybrid Decentralized and Centralized Training and Execution Strategy for Multiple Mobile Robots Reinforcement Learning

This section will introduce the hybrid decentralized and centralized training and execution (DCTE) strategy for multiple-mobile-robots reinforcement learning (MRRL). In MRRL, robots must cooperate to achieve optimal collective behavior. Hybrid training and execution strategies combine decentralized and centralized approaches to optimize learning efficiency and strategy implementation. The hybrid DCTE strategy includes six steps: initial decentralized path planning based on DQN, central collision detection, dynamic planning in collision areas, path re-optimization and connection, execution, and checking whether there are new obstacles or not in the environment. The processing flowchart is shown in Figure 1.

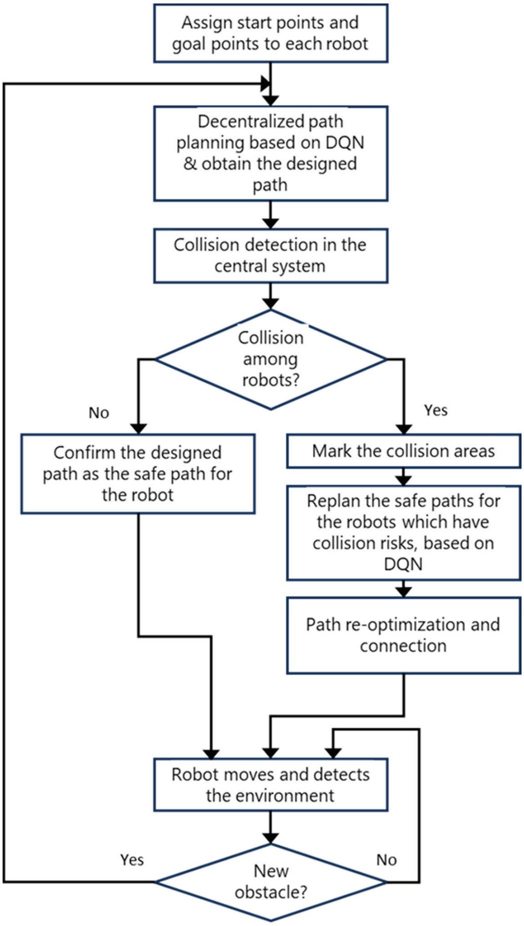

Assign start points and goal points to each robot
Decentralized path planning based on DQN & obtain the designed path
Collision detection in the central system
Collision among robots?
No
Yes
Confirm the designed path as the safe path for the robot
Mark the collision areas
Replan the safe paths for the robots which have collision risks, based on DQN
Path re-optimization and connection
Robot moves and detects the environment
Yes
No
New obstacle?

Figure 1. Hybrid DCTE strategy processing flowchart.

Initially, each robot is assigned with the start point and goal point. In the decentralized path planning step, each agent independently learns to find the optimal path from the starting point to the goal point, avoiding static obstacles, based on DRL. The states of the robot are the inputs of the neural network. There are two layers of the neural network. The first layer has 10 neurons, and the second layer has 4 neurons. The outputs of the neural network are the values of all actions. Each robot selects the action with the highest value among its possible actions (up, down, left, and right). After learning, the optimal safe path is obtained for each robot, and sent to the central system. The central system checks whether there is a collision risk or not among the robots. As shown in Figure 2, the black blocks represent static obstacles. After the decentralized path planning step, the red robot, blue robot, yellow robot, and orange robot obtain the solid black path, dash black path, dash–dot black path, and dot black path, separately. The arrow directions represent the moving directions. The yellow robot's path does not intersect with any other paths, so it is a safe path. Although the dot black path and the solid black path do interact, there is no collision risk, because the orange robot passes firstly. Therefore, the orange robot's dot path is confirmed to be safe and can be directly used for execution. However, the solid path and the dash path do interact, and the red robot and the blue robot also meet each other in the green block. Therefore, the green block is the collision area. The path in the collision area should be replanned.

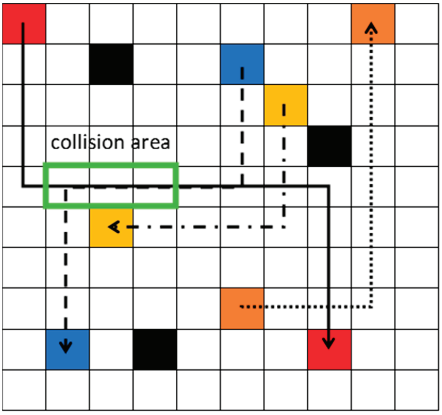


Figure 2. Three robots obtained optimal path after decentralized path planning step.

As shown in Figure 2, the collision area is marked. Robots in the collision area need to re-plan their paths. The robots should consider other robots as dynamic obstacles to avoid obstacles. As shown in Figure 3, the blue robot's start point (x_{bs}, y_{bs}) and goal point (x_{be}, y_{be}) are assigned to the robot. The red robot's start point (x_{rs}, y_{rs}) and goal point $(x_{\mathfrak{R}}, y_{\mathfrak{R}})$ are given to the robot. Because the path for the yellow robot is confirmed, the safe path of the yellow robot, which is marked as black blocks in Figure 3, will set as the static obstacles for the red and the blue robot. The blue and red robot should learn and find the optimal paths, avoiding static obstacles and collision with each other, based on [22]. Each robot has five possible actions: up, down, left, right, and stop. The Deep Q-Network (DQN) receives inputs based on the states of multiple robots. It then estimates the Q-value for each possible action of the robots. Each robot selects the action a associated with the highest Q-value, and then dispatches these actions to all robots. The function $f(a, n, k)$ is used to calculate the specific action a_k for robot $k(k = 1, 2, 3, \ldots)$. a is the selected action with the highest Q-value. $n(n \geq 2)$ is the number of the robot. For example. If there are three robots, then n is three. k (1,2,3, . . .) is the index number of the robot. The pseudo code is used to calculate a_k as Algorithm 1, where 5 means that the robot has four possible actions.

Algorithm 1. Function: : $f(a,n,k)$ is used to calculate the specific action a_k for a robot.

```
Define $a$ as the action with the highest Q-value.
Define n (n $\geq$ 2) as the number of the robot.
Define k (1, 2, 3, . . .) as the index number of the robot.
Define one variable base, which equals the integer part of the equation $a/5^{n-1}$.
if k equals n:
    $a_k$ equals base.
else if k equals n − 1:
    $a_k$ equals the integer part of base/5.
else if k equals n − 2:
    get the integer part of base/5 as base1.
    $a_k$ equals the reminder part of base1/5.
else if k equals n − 3:
    get the integer part of base/5 as base1.
get the reminder part of base1/5 as base2.
$a_k$ equals the integer part of base2/5.
```

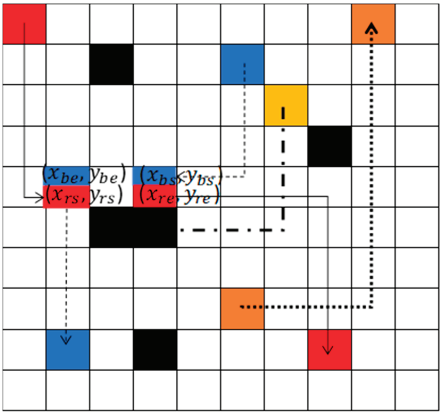


Figure 3. Red robot and blue robot re-planned the trajectory to avoid the collision.

The robot executes the selected action and observes the new state and reward. If the robot reaches the goal point, +1 rewards is obtained. If the robot moves to a free space without any incidents, 0 reward is obtained. If the robot either collides with the other robot, an obstacle, or returns to its starting point, −1 reward is obtained. The total reward r for all robots at each step is calculated by summing the individual rewards rwd_k:

$$r = \sum_{k=1}^{n} rwd_k \tag{1}$$

In the learning process, two neural networks are utilized: the evaluation network and the target network. The evaluation network is updated continuously and is used to predict the Q-values for current states and actions. The target network is a periodic copy of the evaluation network and is used to calculate the target Q-values. Each network has two layers. The first layer has 10 neurons, and the second layer has 4^n neurons. The method uses prioritized experience replay where the absolute temporal difference error is recorded in memory. It is calculated as the difference between the values from the target network and the evaluation network. A larger temporal difference error indicates significant room for improvement in prediction accuracy, thereby assigning higher priority to those samples for further learning. The formula used to train the evaluation network, representing the target Q-value, is given by:

$$Y_j = r_j + \gamma max\hat{Q} \tag{2}$$

r_j is the reward obtained after all robots taking actions $a_k(k = 1, 2, 3, \ldots\ldots)$ in state s_j. s_j is a single array stacking all robots' states. s_{j+1} is the resulting state after taking the action. $\gamma = 0.9$ is the discount factor, which determines the importance of future rewards. $max\hat{Q}$ represents the maximum predicted Q-value for the next state using the target network's parameters. a' represents the next action that maximizes the Q-value for the new state s_{j+1}. θ_j^- represents the parameters of the target network at step j.

After replanning the path, the optimal paths are shown in the green block in Figure 4. The solid black line and dash black are the paths for red robot and blue robot separately. The path re-optimization and connection step will be processed. These paths will be connected to the initial safe paths, obtained from decentralized path planning based on the DQN step, to form a complete, secure route for execution, as shown in Figure 5. Then, each robot will track the path and detect the environment. If new obstacles are detected, it will return to the step of decentralized path planning based on DQN.

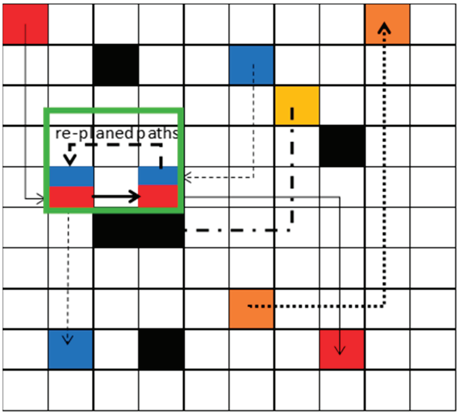


Figure 4. After replanning the path, the optimal paths are shown in the green block.

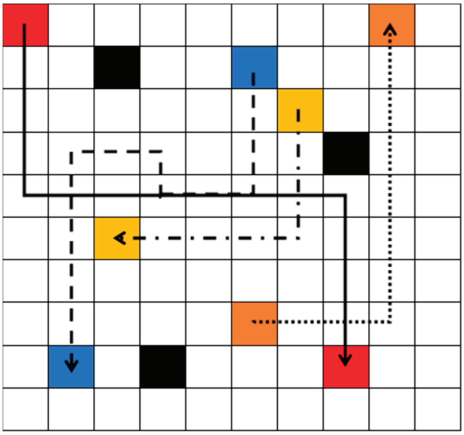


Figure 5. Path re-optimization and connection.

4. Simulations

Based on the proposed Hybrid DCTE strategy, three robots plan and track the safe path from start points to goal points. Initially, three robots plan the path based on DQN separately. There are four actions: forward, backward, left, and right. The parameters for the algorithm are given with a learning rate of 0.01, a reward decay of 0.9, an epsilon-greedy policy with an epsilon set to 0.9, and a target network replacement iteration of 200. The memory size and batch size were set to 2000 and 32. In Figure 6, the unit of each grid is 1 m. The black blocks represent the obstacles. The starting points for robots 1, 2, and 3 are marked by red, blue, and gray blocks, respectively. The goal points for robots 1, 2, and 3 are marked by yellow, green, and purple blocks, respectively. The red path is the planed path for robot 1. The blue path presents the planed path for robot 2. The gray path represents the planned path for robot 3. Since robots do not have the stop action to efficiently reach their goal points, robot 1 and robot 3 have a collision risk between each other. The risk area is shown in the green block.

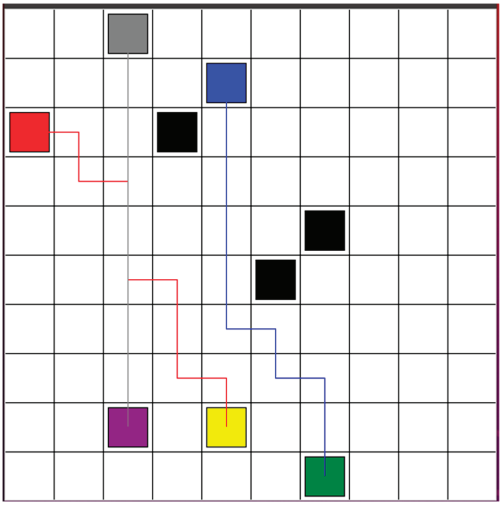


Figure 6. Three robots plan a path after decentralized path planning step, based on DQN.

Robot 1 and robot 3 replanned the optimal paths simultaneously, based on DQN RL algorithm. The learning rate, reward decay, and e-greedy were 0.01, 0.9, and 0.9, respectively. The target network is updated every 200 iterations. The memory size and batch size were set to 2000 and 32. The number training steps is 16,000 steps. There are five actions for each robot: forward, backward, left, right, and stop. In Figure 7, the red and gray blocks mark the starting points for robots 1 and 3, respectively, while the yellow and purple blocks designate their goal points, respectively. The black blocks present the static obstacles. The robot is recognized as the dynamic obstacle for the other robot. From the start point, robot 1 moves backward first and then move forward. The robot 3 moves forward directly.

After the path re-optimization and connection step, the optimized paths for robots 1, 2, and 3 are shown in Figure 8. There is no collision risk. If there is no new obstacle, three robots track the path to the goal points. If new obstacles are detected, robots will return to the step of decentralized path planning based on DQN. In total, the reinforcement learning time is 3 min and 36 s for three robots.

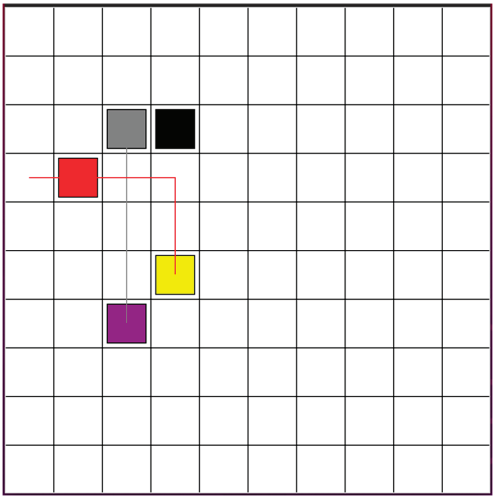


Figure 7. Robot 1 and robot 3 replan the optimal paths simultaneously, based on the DQN RL algorithm.

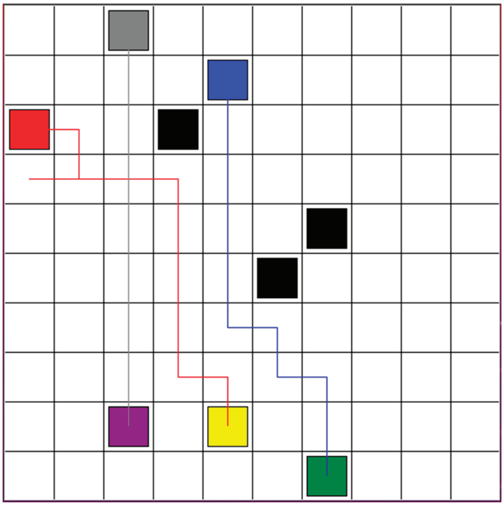


Figure 8. After path re-optimization and connection step, the optimized paths for robot 1, 2, and 3, respectively.

To illustrate the comparison between the hybrid DCTE strategy and the DQN-based approach from ref [22], we present the result where three robots are exploring safe paths in an environment, as depicted in Figure 6. Figure 9 presents the comparative results. The red, blue, and grey lines represent the safe path for robots 1, 2, and 3, separately. The paths are more complex and longer than the paths in Figure 9. In total, the reinforcement learning time is 5 min and 33 s for three robots.

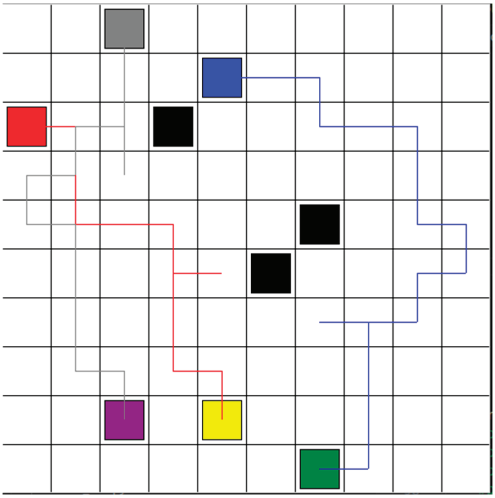
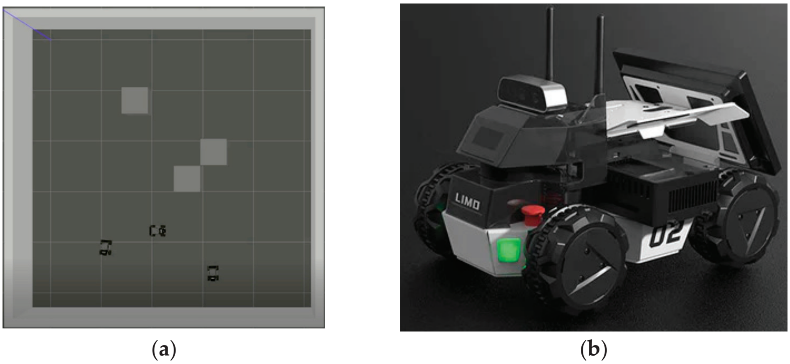


Figure 9. Optimal paths for robots 1, 2, and 3, based on DQN-based approach from ref [22].

The proposed hybrid DCTE strategy for multi-robot navigation was evaluated using the Gazebo Simulator. The simulation environment, consistent with the environment shown in Figure 10a, was designed to rigorously test the algorithm's effectiveness. LIMO robots from Agilex Robotics, as shown in Figure 10b, is equipped with onboard LiDAR and camera sensors. The obstacles are represented by gray boxes. Each cell measured 2*2 m. Three robots are added to the Gazebo simulation environment. Each robot is assigned a goal point. Robots applied the proposed DCTE Strategy to explore optimal paths. Robots subscribed to the "/odom" topic to acquire their x and y positions and the "/imu" topic to obtain the yaw angle value. Robots published the "/cmd_vel" topic to control movement using linear and angular velocities. Robots use linear velocity 0.02 m/s for forward actions, Robots use linear velocity −0.02 m/s for backward action. Robots calculate angular velocity, based on the difference of the angle between two waypoints and the robot's yaw angle.



(**a**) (**b**)

Figure 10. (**a**) Gazebo simulator environment; (**b**) Agilex LIMO robot.

Figure 11 demonstrates the safe path exploration and tracking process. At t = 0, robots initiate safe path exploration. At t = 39 s, three robots track their safe paths. At t = 66 s, robot 1 turns left to avoid collision with robot 3. At t = 96 s, robot 1 moves backward to track its optimal path. At t = 133 s, robot 2 avoids the static obstacle. At t = 162 s, three robots track their safe paths. At t = 201 s, robot 3 reaches its goal point. At t = 252 s, robot 2 arrives at its goal point. At t = 308 s, robot 1 finally reaches its goal point. The video can be seen in the link: https://www.dropbox.com/scl/fi/vq6tf7onj66vv9fc4625x/gazebo-result-DCTE-strategy.webm?rlkey=8hz2o8zn3fx9p7k5tqsx2lk9q&st=hxk66req&dl=0 (accessed on 26 June 2024).

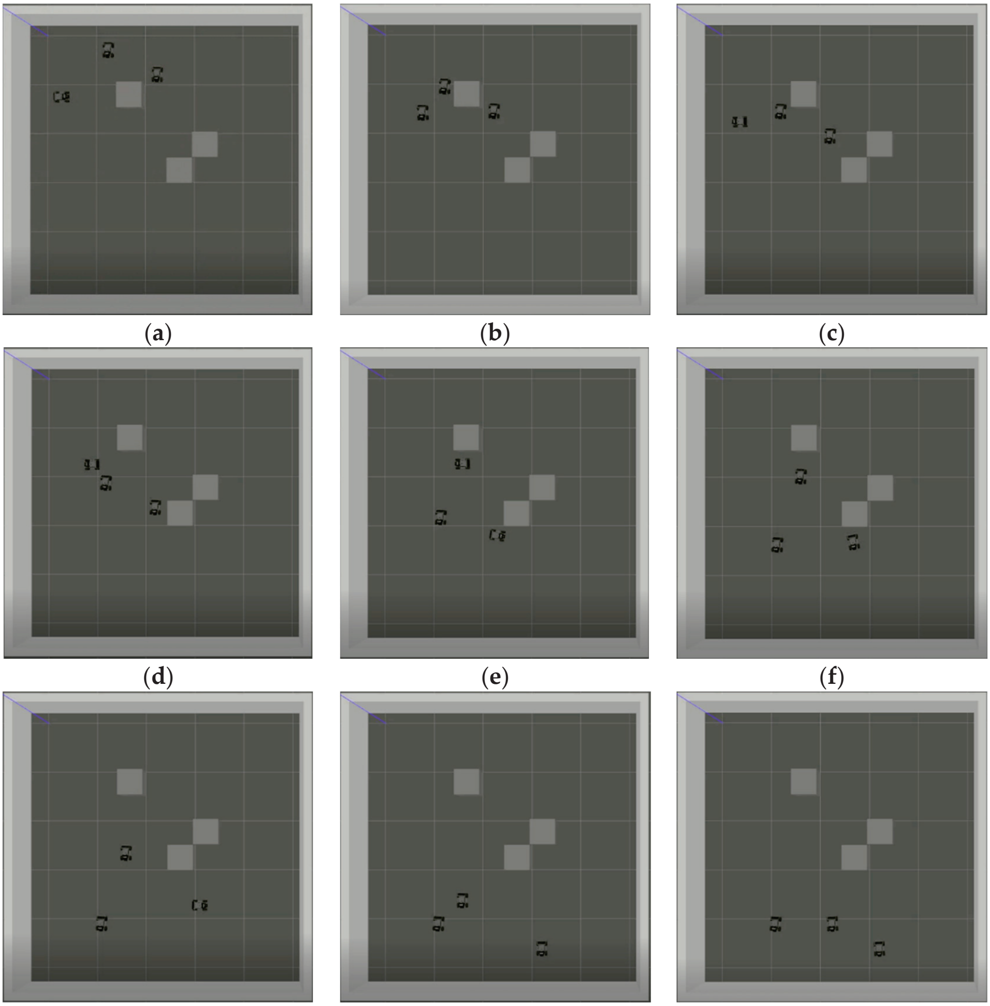

(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**) (**h**) (**i**)

Figure 11. Three robots explore and track the optimal paths, based on a hybrid DCTE strategy: (**a**) t = 0 s; (**b**) t = 39 s; (**c**) t = 66 s; (**d**) t = 96 s; (**e**) t = 133 s; (**f**) t = 162 s; (**g**) t = 201 s; (**h**) t = 252 s; (**i**) t = 308 s.

To validate the effectiveness of the proposed strategy, we deployed three robots in a more complex environment. By applying the hybrid DCTE strategy, the robots were able to reach their target points efficiently, quickly, and safely. In Figure 12, the robots need to avoid the black block obstacles. Robots 1 and 3 have the risk of colliding with each other and need to re-plan their optimal paths to avoid both static and dynamic obstacles.

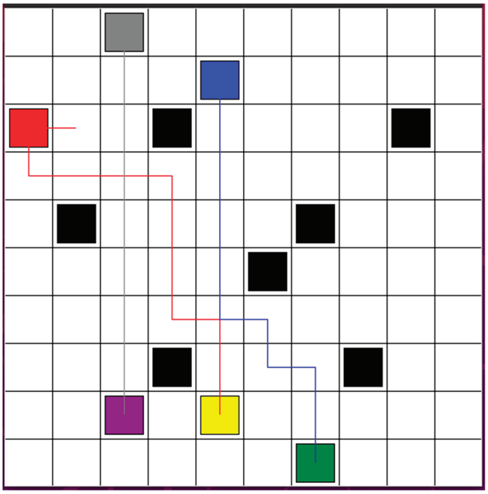


Figure 12. Three reach their target points efficiently, quickly, and safely, in a complex environment.

The complexity of the proposed control algorithm in real-time implementation can be discussed through theoretical analysis based on existing research and simulation results. The proposed Hybrid Decentralized and Centralized Training and Execution (DCTE) Strategy includes decentralized path planning and dynamic re-planning phases, which are expected to require substantial computational power. Each robot's Deep Q-Network (DQN) involves multi-layer neural network computations, typically necessitating powerful CPUs or GPUs. In large-scale deployments, distributed computing resources might be needed to ensure real-time performance. The DQN requires the storage of extensive training data, replay buffers, and neural network parameters, leading to significant memory usage. Especially when handling multiple robots, the memory requirements will grow exponentially. Based on the structure and training process of DQNs, the time complexity of the decentralized path planning and dynamic re-planning phases mainly depends on the number of robots and the complexity of the environment. In practical applications, the algorithm needs to complete path planning and adjustments within milliseconds. Although DQNs can theoretically achieve efficient computations, its actual performance will be influenced by hardware conditions and environmental complexity. The real-time processing of LiDAR sensor data requires efficient data stream management and rapid computational capability to ensure minimal latency. Results from the Gazebo simulator can provide valuable references for practical applications [29,30]. Therefore, the DCTE strategy performs well in simulated environments, suggesting it could achieve similar outcomes in real-world scenarios. The goal of the proposed algorithm is to reduce the computational load, improve the real-time effectiveness, and enhance scalability. We believe that the algorithm has great potential for practical applications. Through theoretical analysis, we preliminarily validate the feasibility and potential complexity of the proposed DCTE strategy in real-time multi-robot navigation. Further practical experiments and optimizations will help comprehensively evaluate and enhance the performance of this algorithm.

5. Conclusions

This paper has introduced a novel Hybrid Decentralized and Centralized Training and Execution (DCTE) Strategy to address the complex challenges of path planning in multi-robot systems. The initial decentralized approach, based on Deep Q-Networks (DQN), allows each robot to independently find a path in a static environment. The centralized collision detection step ensures that only non-intersecting paths proceed to execution, while paths at risk of collision are dynamically replanned, considering other robots as dynamic obstacles. Those robots in a collision area replan safe paths simultaneously, based on the DQN RL algorithm. After the re-optimization and connection step, all robots obtain the optimization paths. The hybrid nature of the DCTE strategy ensures that computing resources are allocated efficiently, with initial independent optimizations followed by necessary centralized adjustments. This approach not only maintains continuous operation but also adapts flexibly to both static and dynamic obstacles, ensuring robust performance in varied and unpredictable environments. Simulation results demonstrate that the DCTE strategy significantly improves the system performance by reducing the collision risks and optimizing path planning in a computationally efficient manner.

Author Contributions: Conceptualization, Y.D.; methodology, Y.D.; software, Y.D.; validation, Y.D.; formal analysis, Y.D.; investigation, Y.D.; resources, D.K.; data curation, Y.D.; writing—original draft preparation, Y.D.; writing—review and editing, Y.D. and K.L.; visualization, Y.D.; supervision, K.L.; project administration, K.L.; funding acquisition, D.K. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (RS-2024-00406796, HRD Program for Industrial Innovation).

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

DCTE	Decentralized and Centralized Training and Execution
DQN	Deep Q-Network
RL	Reinforcement Learning
MRRL	Multiple Mobile Robots Reinforcement Learning

References

1. Ort, T.; Paull, L.; Rus, D. Autonomous Vehicle Navigation in Rural Environments Without Detailed Prior Maps. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 2040–2047. [CrossRef]
2. Trinh, L.A.; Ekström, M.; Cürüklü, B. Dependable Navigation for Multiple Autonomous Robots with Petri Nets Based Congestion Control and Dynamic Obstacle Avoidance. *J. Intell. Robot. Syst.* **2022**, *104*, 69. [CrossRef]
3. Yew, C.H.; Ksm, S.; Leong, W.; Hong, F.; Nur, A.R.; Nurul, A.; Raad, H. Development of collision avoidance system for multiple autonomous mobile robots. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1–15. [CrossRef]
4. Qin, J.; Du, J. Robust adaptive asymptotic trajectory tracking control for underactuated surface vessels subject to unknown dynamics and input saturation. *J. Mar. Sci. Technol.* **2022**, *27*, 307–319. [CrossRef]
5. Elsanhoury, M.; Mäkelä, P.; Koljonen, J.; Välisuo, P.; Shamsuzzoha, A.; Mantere, T.; Elmusrati, M.; Kuusniemi, H. Precision Positioning for Smart Logistics Using Ultra-Wideband Technology-Based Indoor Navigation: A Review. *IEEE Access* **2022**, *10*, 44413–44445. [CrossRef]
6. Wang, L.; Liu, G. Research on multi-robot collaborative operation in logistics and warehousing using A3C optimized YOLOv5-PPO model. *Front. Neurorobot.* **2024**, *17*, 1329589. [CrossRef] [PubMed]
7. Wen, S.H.; Wen, Z.T.; Zhang, D.; Zhang, H.; Wang, T. A multi-robot path-planning algorithm for autonomous navigation using meta-reinforcement learning based on transfer learning. *Appl. Soft Comput.* **2021**, *110*, 107605. [CrossRef]
8. Marchesini, E.; Farinelli, A. Enhancing Deep Reinforcement Learning Approaches for Multi-Robot Navigation via Single-Robot Evolutionary Policy Search. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 5525–5531. [CrossRef]
9. Jestel, C.; Surmann, H.; Stenzel, J.; Urbann, O.; Brehler, M. Obtaining Robust Control and Navigation Policies for Multi-robot Navigation via Deep Reinforcement Learning. In Proceedings of the 2021 7th International Conference on Automation, Robotics and Applications (ICARA), Prague, Czech Republic, 4–6 February 2021; pp. 48–54. [CrossRef]
10. Escudie, E.; Matignon, L.; Saraydaryan, J. Attention Graph for Multi-Robot Social Navigation with Deep Reinforcement Learning. *arXiv* **2024**, arXiv:2401.17914.

11. Parnichkun, M. Multiple Robots Path Planning based on Reinforcement Learning for Object Transportation. In Proceedings of the 2022 5th Artificial Intelligence and Cloud Computing Conference (AICCC '22), Osaka, Japan, 17–19 December 2022; Association for Computing Machinery: New York, NY, USA, 2023; pp. 229–234. [CrossRef]
12. Wenzel, P.; Schön, T.; Leal-Taixé, L.; Cremers, D. Vision-Based Mobile Robotics Obstacle Avoidance with Deep Reinforcement Learning. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 14360–14366. [CrossRef]
13. Giusti, A.; Guzzi, J.; Cireşan, D.C.; He, F.L.; Rodríguez, J.P.; Fontana, F.; Faessler, M.; Forster, C.; Schmidhuber, J.; Di Caro, G.; et al. A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots. *IEEE Robot. Autom. Lett.* **2016**, *1*, 661–667. [CrossRef]
14. Nguyen, T.; Nguyen, N.D.; Nahavandi, S. Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Trans. Cybern.* **2020**, *50*, 3826–3839. [CrossRef] [PubMed]
15. de Zarzà, I.; de Curtò, J.; Roig, G.; Manzoni, P.; Calafate, C.T. Emergent Cooperation and Strategy Adaptation in Multi-Agent Systems: An Extended Coevolutionary Theory with LLMs. *Electronics* **2023**, *12*, 2722. [CrossRef]
16. Busoniu, L.; Babuska, R.; De Schutter, B. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2008**, *38*, 156–172. [CrossRef]
17. Orr, J.; Dutta, A. Multi-Agent Deep Reinforcement Learning for Multi-Robot Applications: A Survey. *Sensors* **2023**, *23*, 3625. [CrossRef] [PubMed]
18. Torbati, R.; Lohiya, S.; Singh, S.; Nigam, M.; Ravichandar, H. MARBLER: An Open Platform for Standardized Evaluation of Multi-Robot Reinforcement Learning Algorithms. In Proceedings of the 2023 International Symposium on Multi-Robot and Multi-Agent Systems (MRS), Boston, MA, USA, 4–5 December 2023; pp. 57–63. [CrossRef]
19. Zhang, Z.; Jiang, X.; Yang, Z.; Ma, S.; Chen, J.; Sun, W. Scalable Multi-Robot Task Allocation Using Graph Deep Reinforcement Learning with Graph Normalization. *Electronics* **2024**, *13*, 1561. [CrossRef]
20. Gu, S.; Kuba, J.G.; Chen, Y.; Du, Y.; Yang, L.; Knoll, A.; Yang, Y. Safe multi-agent reinforcement learning for multi-robot control. *Artif. Intell.* **2023**, *319*, 103905. [CrossRef]
21. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
22. Dai, Y.; Yang, S.; Lee, K. Sensing and Navigation for Multiple Mobile Robots Based on Deep Q-Network. *Remote Sens.* **2023**, *15*, 4757. [CrossRef]
23. Gronauer, S.; Diepold, K. Multi-agent deep reinforcement learning: A survey. *Artif. Intell. Rev.* **2022**, *55*, 895–943. [CrossRef]
24. Foerster, J.N.; Assael, Y.M.; Freitas, N.D.; Whiteson, S. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016.
25. Grzes, M. Reward Shaping in Episodic Reinforcement Learning. In Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems, São Paulo, Brazil, 8–12 May 2017.
26. Cipollone, R.; Giacomo, G.D.; Favorito, M.; Iocchi, L.; Patrizi, F. Exploiting Multiple Abstractions in Episodic RL via Reward Shaping. In Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 7227–7234.
27. Lee, M.R.; Yusuf, S.H. Mobile Robot Navigation Using Deep Reinforcement Learning. *Sensors* **2022**, *10*, 2748. [CrossRef]
28. Zeng, J.; Ju, R.; Qin, L.; Yin, Q.; Hu, C. Navigation in unknown dynamic environments based on deep reinforcement learning. *Sensors* **2019**, *19*, 18. [CrossRef] [PubMed]
29. Blum, P.; Crowley, P.; Lykotrafitis, G. Vision-based navigation and obstacle avoidance via deep reinforcement learning. *arXiv* **2022**, arXiv:2211.05243. [CrossRef]
30. Alborzi, Y.; Jalal, B.S.; Najafi, E. ROS-based SLAM and Navigation for a Gazebo-Simulated Autonomous Quadrotor. In Proceedings of the 21st International Conference on Research and Education in Mechatronics (REM), Cracow, Poland, 9–11 December 2020; pp. 1–5. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics MDPI

Article

An Online Review Data-Driven Fuzzy Large-Scale Group Decision-Making Method Based on Dual Fine-Tuning

Xuechan Yuan, Tingyu Xu, Shiqi He and Chao Zhang *

School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China;
202201001169@email.sxu.edu.cn (X.Y.); 202201001163@email.sxu.edu.cn (T.X.);
202201006111@email.sxu.edu.cn (S.H.)
* Correspondence: czhang@sxu.edu.cn

Abstract: Large-scale group decision-making (LSGDM) involves aggregating the opinions of participating decision-makers into collective opinions and selecting optimal solutions, addressing challenges such as a large number of participants, significant scale, and a low consensus. In real-world scenarios of LSGDM, various challenges are often encountered due to factors such as fuzzy uncertainties in decision information, the large size of decision groups, and the diverse backgrounds of participants. This paper introduces a dual fine-tuning-based LSGDM method using an online review. Initially, the sentiment analysis is conducted on online review data, and the identified sentiment words are graded and quantified into a fuzzy data set to understand the emotional tendencies of the text. Then, the Louvain algorithm is used to cluster the decision-makers. Meanwhile, a method combining Euclidean distances with Wasserstein distances is introduced to accurately measure data similarities and improve clustering performances. During the consensus-reaching process (CRP), a two-stage approach is employed to adjust the scores: to begin with, by refining the scores of the decision representatives via minor-scale group adjustments to generate a score matrix. Then, by identifying the scores corresponding to the minimum consensus level in the matrix for adjustment. Subsequently, the final adjusted score matrix is integrated with the prospect–regret theory to derive the comprehensive brand scores and rankings. Ultimately, the practicality and efficiency of the proposed model are demonstrated using a case study focused on the purchase of solar lamps. In summary, not only does the model effectively extract the online review data and enhance decision efficiency via clustering, but the dual fine-tuning mechanism in the model to improve consensus attainment also reduces the number of adjustment rounds and avoids multiple cycles without achieving the consensus.

Keywords: large-scale group decision-making; dual fine-tuning; online review data; prospect-regret theory; clustering analysis

Citation: Yuan, X.; Xu, T.; He, S.; Zhang, C. An Online Review Data-Driven Fuzzy Large-Scale Group Decision-Making Method Based on Dual Fine-Tuning. *Electronics* **2024**, *13*, 2702. https://doi.org/10.3390/electronics13142702

Academic Editor: Ping-Feng Pai

Received: 11 June 2024
Revised: 1 July 2024
Accepted: 8 July 2024
Published: 10 July 2024

1. Introduction

With the development of the digital economy, various online platforms, such as social media, online forums, and e-commerce websites, have become important channels for people to communicate and express opinions widely. The scale of decision members involved in these platforms has expanded continuously. When the scale of decision-makers exceeds a certain threshold, the group decision-making issue can be classified as an LSGDM challenge [1]. Overall, LSGDM has the following three characteristics: first, the decision group is large and diverse in opinions; second, the decision information involved in the process is highly uncertain and ambiguous; third, the participants in decision-making have different backgrounds, interests, and preferences, leading to low consensus and increasing decision complexity. In summary, LSGDM still faces many challenges.

In the sentiment analysis, the large scale of online reviews requires efficient data processing, integrating emotional elements into the LSGDM process. When evaluating options, people express emotions in various ways, such as positive, neutral, and negative emotions.

Therefore, accurately identifying and quantifying these emotional data is crucial. In the social network analysis, trust relationships among decision-makers may potentially influence clustering, opinion collection, and the group consensus process. Due to insufficient understanding of the internal structure of the group and the relationships among members, it is challenging to identify subgroups with similar characteristics or opinions. The clustering analysis is a crucial step in reducing the dimensionality of decision-makers and acquiring objective weight data. Therefore, effective clustering is one of the challenges of LSGDM. In the CRP, the LSGDM leads to a low initial consensus level. For instance, when purchasing solar garden lights, most decision-makers believe that the garden light has a long lighting time. However, some decision-makers think that the lighting time is short. Thus, achieving a high degree of consensus among experts in a single decision-making process is difficult. In summary, this paper intends to explore a dual fine-tuning LSGDM model. The following sections will introduce the current research status and research motivations from three aspects: the sentiment analysis, LSGDM, and behavioral decision-making.

Consumers' purchasing decisions are influenced not only by the attributes and evaluation standards of the products themselves but also by the online reviews and feedback from other consumers. As a result, consumers are now in the habit of looking up the experiences and feedback of prior buyers before making a purchase [2]. Specifically, how to effectively extract and analyze the sentiment factors contained in the review data and then accurately and effectively convert them into preference datasets is a focus of scholars. Preprocessing and the sentiment analysis are particularly important when selecting products, given that numerous online reviews are rich in emotional content and biases. Additionally, decision-makers must take into account various factors, such as price, precision, and convenience [3]. Through the sentiment analysis, insights can be gained into the emotional tendencies and attitudes of group members towards specific topics, issues, or decisions, which is significant for the formulation and implementation of large-scale group decisions. In summary, how to extract realistic decision data from online comments and reasonably depict the fuzzy uncertainty of the data in realistic decision scenarios, which is challenging.

The classic process of LSGDM, according to vertical research ideas, roughly includes clustering decision-makers, determining weights, and reaching a consensus. When dealing with LSGDM problems, integrating members' relationship information using social networks is considered an effective method. Most existing research constructs social networks based on the trust relationships between experts. Studies on social influence theory indicate that there is an interaction between similarity and social relationships, and similarity characteristics influence group relationships [4]. To address the issue of the large scale in the LSGDM, numerous scholars strive to address the LSGDM problems by employing clustering algorithms to reduce dimensions. Clustering algorithms can reduce decision complexity and make decision information among decision-makers within the same cluster more similar. This paper uses the improved Louvain algorithm. Compared with other graph-clustering algorithms, one of the most notable advantages of the Louvain algorithm is its efficiency and scalability; another advantage is its ability to handle weighted graphs, effectively dealing with the community structure division of complex networks while maintaining sound time complexity [5]. These advantages make the Louvain algorithm one of the preferred graph-clustering algorithms in many practical applications, especially in handling large-scale and complex network data. Specifically, the Wasserstein distance considers not only the position of the data points but also the shape and structure of their distribution. It performs well in handling probability distributions or non-continuous data, facilitating the processing of complex distributions and outliers [6]. During the decision-making process, the trust relationships and similarities in opinions among decision-makers are used to construct a relationship network among decision-makers, and the Louvain algorithm is adopted for the clustering analysis. The research gap between this paper and previous studies lies in the use of a mixed distance calculation formula for calculating the similarity of expert opinions, which improves the accuracy of the weight calculations.

Therefore, reducing the dimensionality of large-scale groups and objectively obtaining weight data to simplify the problem-solving process are important.

Determining weights is a key step to ensuring the accuracy and effectiveness of the LSGDM results [7]. By assigning different weights to different decision-makers or attributes, their importance or influence in the decision-making process can be more accurately reflected. Researchers have developed various methods to quantify and allocate weights to indicate the importance of different decision-makers' opinions or decision criteria. The calculation idea of expert weights originates from the weighted sum of the squared Euclidean distance and the squared Wasserstein distance to improve the accuracy and operability of the weight determination process.

The ultimate goal of LSGDM is to reach a group-satisfactory consensus, forming a final ranking of the options [8]. In fact, with the increase in the number of decision-makers, the number of adjustment rounds also increases, and the complexity of obtaining consensus opinions from a large-scale group simultaneously increases significantly. Therefore, this paper uses a two-stage consensus measurement and feedback mechanism to accelerate the CRP. In the first stage, the outliers are removed through mean and variance in the clustered decision groups, and the decision representatives in the decision groups are finally selected to reduce the decision scale. Secondly, to prevent the situation where there is more than one outlier in the decision-making process, a dual minimum consensus level is set. When the adjustments to the first minimum consensus level reach a certain number without reaching a consensus, it is considered that there is more than one decision-maker with a large decision difference from the others. The scores of decision-makers are adjusted for the second minimum consensus level. After obtaining the final scores, the prospect–regret theory is integrated to more comprehensively understand the impact of the decision-makers' bounded rationality on the decision results [4]. These two theories mitigate the influence of psychological factors before and after the decision-making process, effectively integrating the decision-makers' bounded rationality into the outcomes to produce the final decision result.

In an uncertain environment, rational decisions based on the expected utility cannot efficiently explain certain actual decision behaviors [9]. Therefore, within the framework of behavioral decisions, research on the cognitive limitations of decision-makers, the subjective psychological factors of decision-makers, and the psychological impact of the environment on decision-makers is becoming increasingly important. As scholars delved deeper, Kahneman [10] proposed the prospect theory, and Bell, Loomes, and Sugden [11] proposed the regret theory, providing new ideas for addressing uncertain decision problems considering decision-makers' psychological behaviors. This paper combines the prospect–regret theory [4] with the final results of the CRP to obtain the final ranking of the options. The prospect theory explains how decision-makers have varying attitudes towards gains and losses, highlighting the influence of emotions, but it does not directly address the regret that might occur after making decisions. The regret theory, on the other hand, emphasizes the future regret emotions without fully considering the balance between potential gains and losses, often resulting in more conservative decisions. By integrating these two approaches, the prospect–regret theory can address their individual limitations, resulting in decision outcomes that better align with people's behaviors in diverse situations. In summary, how to minimize the influence of outlier decision-makers after clustering to achieve satisfactory results for the group and effectively prevent situations where consensus requirements are not met even after multiple iterations.

In conclusion, in existing LSGDM methods, many studies on consensus-reaching mechanisms exist, but few use two-stage processing with secondary fine-tuning solutions. Owing to the large scale of LSGDM, experts have diverse backgrounds and cognitive differences, making it difficult to achieve a high consensus among experts on one decision-making process [12,13]. As the number of decision-makers increases, the number of adjustment rounds also increases, and the complexity of obtaining large-scale group consensus opinions significantly rises. Therefore, it is necessary to design a reasonable consensus feedback mechanism to improve group consensus levels and ensure the reliability of decision results.

The structure of this paper is as follows: the second section reviews the literature on LSGDM and the sentiment analysis. The third section provides an overview of the foundational knowledge framework for LSGDM methods and the prospect–regret theory. The fourth section introduces the improved Louvain algorithm for clustering and the CRP based on dual fine-tuning. The fifth section illustrates and tests our proposed LSGDM model through a case study of purchasing solar garden lights online and examines the strengths and weaknesses of our approach through both quantitative and qualitative comparative analyses, as well as a sensitivity analysis. The sixth section concludes the paper and offers suggestions for future research.

2. Related Work

In this section, we will break down the content into three distinct parts. Firstly, Section 2.1 offers a comprehensive literature review on the sentiment analysis. Following this, Section 2.2 delves into the existing research on the clustering analysis. Finally, Section 2.3 provides an in-depth literature review on the CRP.

2.1. Sentiment Analysis

The sentiment analysis plays a crucial role in LSGDM, especially when handling the vast amount of review data from social media, online forums, and e-commerce websites. Through the sentiment analysis, sentiment can be effectively introduced and quantified. These sentiment data help decision-makers or consumers understand the attitudes and tendencies toward a particular product or issue. Zhang et al. [14] established a multi-granularity probabilistic linguistic information system using probabilistic linguistic term sets. This method quantitatively analyzes users' emotional expressions through the sentiment analysis. Liang et al. [15] proposed an integrated decision support model that collects linguistic information from each review through the sentiment analysis and converts it into a linguistic intuitionistic standard cloud of the product, thus ranking hotels.

These models analyze the emotional tendencies within comments to help decision-makers understand user satisfaction and focal points. Compared with traditional small-scale datasets, online data collection can provide larger and richer samples, enhancing the accuracy and reliability of decisions. Additionally, online data collection can be updated in real-time, ensuring that the decision basis is always up-to-date. By analyzing the sentiment in online review data, decision-makers can better grasp user psychology, and optimize product design and marketing strategies, thereby gaining an advantage in the competitive market. In summary, sentiment analysis methods help to accurately examine online review data and provide reliability in interpreting the emotional classification of data.

2.2. Clustering Analysis

Conventional clustering approaches rely on the similarity of viewpoints among decision-makers, such as K-means clustering algorithms [16], hierarchical clustering algorithms [17], and vector space-based clustering algorithms [18].

Vincent et al. [19] pointed out that the Louvain algorithm, as an efficient community discovery algorithm, is particularly suitable for large-scale networks. Wu et al. [20] proposed an LSGDM model based on the Louvain algorithm using interval type-2 fuzzy sets, determining the weights of decision-makers and community submodules based on community network characteristics. It is noteworthy that most of the existing research on LSGDM problems based on social network relationships focuses on the clustering of LSGDM, with less improvement in the calculation process of expert similarity before the clustering analysis. This paper uses a mixed-distance method combining Euclidean distances with Wasserstein distances to calculate the opinion similarity between decision-makers.

Xu et al. [21] determined the weights of the subgroups based on the consistency degrees of the subgroup preference relationships. Wu et al. [20] divided a network into several communities and then obtained the centrality of the entire network and the communities by averaging the fused centrality of all the members, obtaining the community's weight based on the inverse relative distances between the community centrality and the overall network centrality. However, some methods of solving index weights have limitations and cannot cope with complex decision environments. Unlike the above literature, this paper also calculates the weight of each expert. This paper also calculates the weight of each expert to improve the accuracy and operability of the weight determination process.

2.3. CRP and Decision Result Generation

The CRP can effectively reduce the contradictions between decision-makers, facilitating the production of decision outcomes. The key to enhancing consensus lies in how to set the adjustment range for scores during the CRP to achieve a better decision consensus. Many scholars have offered various solutions. For example, Xu et al. [21] established a two-stage consensus method, where the two phases refer to the consensus within the group decision representatives and the consensus between groups. In addressing outliers within groups by modifying their evaluation values, the approach involves directly replacing the outliers' values with the desired ones. Tang et al. [22] constructed a subgroup adaptive CRP composed of mixed strategies. This model proposes different feedback mechanisms for varying degrees of subgroup inter-consensus and intra-consensus, which can be realized by increasing or decreasing the fixed values of outliers when modifying evaluation values. However, the aforementioned studies have considered adjustments to decision scores but lack flexibility, so are unable to make reasonable adjustments based on actual scores. In this paper, the size of the adjustment range is determined by the amount of scoring, effectively preventing a decrease in the consensus due to an excessive adjustment of scores. As the number of group decision-makers increases, the number of adjustment rounds also increases, significantly raising the complexity of obtaining large-scale group consensus opinions. Therefore, a two-stage consensus measurement and feedback mechanism can help to accelerate the consensus-reaching process.

The selection of adjustment subjects during the decision-making process also plays a crucial role in the decision outcome. Palomares et al. [23] designed an LSGDM model based on the FCM clustering algorithm, achieving effective dimensionality reduction by grouping large-scale decision-makers. Liu et al. [24] proposed a partial binary tree DEA-DA cyclical classification model to categorize decision-makers. However, existing methods may not pay attention to the adjustment of the decision scores themselves, and there may be situations where the adjustments are too large or too small. Palomares et al. [23] and Liu et al. [24] failed to fully consider the situation where an excessive number of decision adjustment rounds fails to reach a consensus level. These studies have not focused on situations where multiple cycles still fail to meet the consensus requirements during the decision-adjustment process. Zhang et al. [25] proposed a consensus model for MAGDM using multi-granular HFLTSs, optimizing preference adjustments. Li et al. [26] also proposed consensus models for ordinal classification-based GDM problems with heterogeneous preferences. Yuan et al. [27] optimized IFPRs for robust consensus in large-scale GDM problems. In fact, as the number of decision-makers increases, the complexity of obtaining consensus opinions from a large-scale group also significantly increases. To address this, this paper sets a dual minimum consensus level, reducing the time and adjustment costs required to reach a consensus level by increasing the number of adjusters.

Zhang et al. [28] explored a group consensus model method in the context of interval type-2 fuzzy sets. To alleviate the complexity of reaching a consensus among decision-makers, this model introduces random variables to complete the step of selecting consensus-level thresholds. Palomares et al. [23] designed an LSGDM model based on the FCM clustering algorithm, grouping group decision-makers to achieve effective dimensionality reduction. Liu et al. [24] proposed a partial binary tree DEA-DA cycle classification model to classify decision-makers. However, the above studies do not focus on the situation where multiple cycles still fail to meet the consensus requirements during the CRP of LSGDM.

The generation of the final decision is a key step in the CRP, and this paper incorporates the behavioral decision theory at this step. Within the behavioral decision theory, the prospect theory [29] mainly focuses on the decision makers' degree of delight in making decisions. Meanwhile, the regret theory has also yielded many excellent research results in LSGDM. For instance, Jin et al. [30] proposed a linguistic distribution LSGDM technique that applies statistical inference principles and incorporates the regret theory to address the regret-averse psychological characteristics among decision makers. However, relatively speaking, they have not considered the comprehensive impact of both information and regret on decision-making. The integration of these two aspects has also been explored by many scholars. For example, Wang et al. [31] studied a three-way decision model using the regret theory within a hesitant fuzzy environment. Furthermore, they introduced a novel regret–rejoice function in their research. Tian et al. [32] described a CRP for multi-criteria ranking issues with multiple experts based on probabilistic linguistic term sets, which takes into account the decision makers' regret–rejoice emotions during the decision-making process. Jin et al. [33] constructed a regret–rejoice PLMDEA model based on the regret theory, which considered the regretful attitudes of decision-makers. We incorporate the regret–elation theory into the process of ultimately reaching a decision, allowing the decision outcome to not only take into account the overall collective opinion but also to integrate the behavioral decision theory, facilitating the generation of decision results that are more aligned with objective reality.

3. Materials and Methods

This section will briefly introduce the sentiment analysis, the Louvain algorithm, the CRP, and the prospect–regret theory.

3.1. Text Preprocessing Techniques

The sentiment analysis is a technique that analyzes text data after segmentation to identify and understand the emotions and sentiments expressed therein [34,35]. This section will introduce the key steps required for the sentiment analysis.

3.1.1. Chinese Word Segmentation

This article uses ChatGPT 4.0 (Conversational Generative Pre-trained Transformer) for Chinese word segmentation. Compared with traditional segmentation methods, ChatGPT has a higher adaptability, and its deep learning-based model can effectively understand context and handle polysemy [36].

3.1.2. Creating an Emotion Dictionary

User evaluations often incorporate emotional language, utilizing adjectives, adverbs, and negations to convey their opinions and sentiments about products. Adjectives reflect attitudes toward items, while adverbs and negations indicate the extent of favorability or unfavorability. The sentiment analysis entails extracting these linguistic cues from user feedback and transforming them into valuable insights for assessing products across various dimensions. This article will employ the sentiment analysis utilizing emotion dictionaries to statistically gather collections of positive, neutral, and negative sentiment terms.

3.1.3. Translate the Quantitative Calculation of Emotions

LTP, a natural language processing tool developed in China, automates a range of tasks, including part-of-speech tagging and semantic role labeling [37]. By using LTP to conduct a dependency syntax analysis on product reviews, the identified sentiment words are categorized into three levels: negative (-1), neutral (0), and positive (1). The formula for sentiment quantification is as follows:

$$Score_{ij}^{q} = p(O_{ij}^{q}) \times deg(O_{ij}^{q}) \times \left[(-1)_{ij}^{q}\right]^{N}, \tag{1}$$

where O_{ij}^{q} denotes the sentiment word about attribute C_j of the product in the online review e_i^q, and $p(O_{ij}^{q})$ denotes the polarity of the sentiment word O_{ij}^{q}. $deg(O_{ij}^{q})$ represents the sentiment intensity of the sentiment word O_{ij}^{q} influenced by the degree adverb.

3.2. The Louvain Algorithm

The Louvain algorithm, as a method based on modularity optimization, is widely used in the discovery of community structures in complex networks [38]. Its core idea is to maximize the modularity of the network by iteratively optimizing the community assignment of the nodes. Through this locally optimized strategy, the Louvain algorithm can effectively discover community structures in networks and has a high computational efficiency, making it suitable for handling large-scale networks. In this study, a social relationship network is established based on the similarity of opinions among experts and trust relationships. The Louvain algorithm is then used to cluster large populations.

3.2.1. Modularity

In the Louvain algorithm, modularity is used as a metric to evaluate the quality of the network's community structure. The concept of modularity quantifies the difference between the density of connections within the modules and the expected density of random connections. Modularity serves as the objective function, and the algorithm discovers community structures by continuously optimizing this metric [39].

The definition of modularity is as follows:

$$Q = \frac{1}{2m}\sum_{i,j}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(C_i, C_j), \tag{2}$$

where A_{ij} represents the number of edges connecting the node i and the node j in the network, k_i and k_j represent the degrees of nodes i and j, respectively, m represents the total number of edges in the network, C_i and C_j represent the community labels of nodes i and j, respectively, and $\delta(C_i, C_j)$ is an indicator function that equals 1 when $C_i = C_j$ is true, and 0 otherwise.

3.2.2. The Euclidean Distance

The Euclidean distance considers the differences of the data points in each dimension, which has the characteristics of intuitiveness and ease of understanding. In the Louvain algorithm, using Euclidean distances helps to measure the similarity between nodes, thereby promoting community partitioning and clustering results. The Euclidean distance is applied in the calculation of node similarity [40] using the formula:

$$d_{ij} = \sqrt{\sum\nolimits_k ({A_{ij}}^2 - {A_{jk}}^2)}, \tag{3}$$

where d_{ij} represents the Euclidean distance between the node i and the node j, and A_{ik} and A_{jk}, respectively, represent the connection weights of nodes i and j in the adjacency matrix A.

3.2.3. The Process of the Louvain Algorithm

This section will introduce the process of using the Louvain algorithm, which consists of the following steps:

Step 1: Construct the social network. Let the set of nodes in the network be N, where each node i represents an individual or entity. The relationships between the nodes are represented by edges. Let the set of edges in the network be E, where each edge (i, j) represents some form of association between the node i and the node j. The relationships between the nodes and edges are represented using a graph structure $G = (N, E)$, typically implemented using an adjacency matrix or adjacency list. In the initialization phase, each node is initially considered as a separate community: $C_i = i, \forall i \in N$.

Step 2: Iterative optimization. Iterate over each node i and calculate the modularity gain when the node joins its neighboring community:

$$\Delta Q_{i\to j} = \frac{\sum_{in} + \frac{\sum_{tot} k_{i,in}}{2m}}{2m} - \left(\frac{\sum_{tot} k_{in}}{2m}\right)^2, \tag{4}$$

where $\sum_{in}$ represents the sum of weights of the edges from the node i to the interior of the community n, $\sum_{tot} k_{i,in}$ represents the sum of weights of the edges from the node i to all edges in community n, and m represents the total weight of all the edges in the network.

Step 3: If moving a node to a neighboring community yields the maximum modularity gain, execute the node movement operation. Merge the nodes with the same community label into a supernode.

Step 4: Repeat Steps 2 and 3 on the new network until further optimization of modularity is not possible. The final communities are the node groupings at the end of the iteration.

3.3. CRP

The ultimate goal of the LSGDM is to achieve a result satisfactory to the group, establish a consensus, and further obtain the group's decision. To this end, this paper divides the CRP into two stages [41,42].

Before adjusting the CRP, each decision-maker needs to express their preferences and provide subjective opinions on the decision-making matter.

During the initial stage, a consensus is attained within the decision groups by refining opinions to obtain the collective viewpoint of each group. Decision-makers' perspectives within each group are then combined to ascertain the group's preference and calculate the overall consensus. If the consensus does not meet the required standard, feedback mechanisms are used to adjust individual opinions. If a consensus is achieved through consensus measurement, the process moves to the second stage.

In the second stage, adjustments to opinion preferences continue based on the preferences provided by different groups in the first stage. If the ultimate consensus threshold is not achieved, group feedback is iteratively provided to refine the decision preferences within the groups, with the aim of enhancing the consensus levels within the group. If the required level of consensus is achieved, this preference is considered the final decision result.

The flowchart of the CRP is shown in Figure 1.

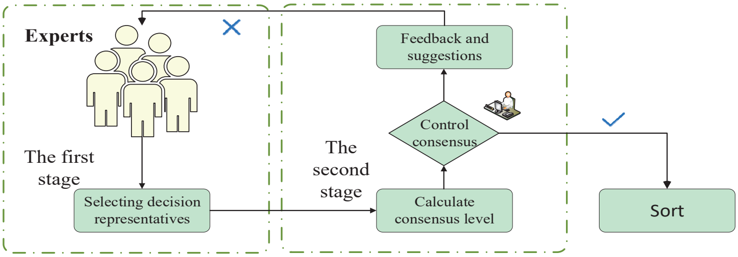


Figure 1. The Overall Plan for the CRP.

3.4. Prospect–Regret Theory

Using the prospect–regret theory, it is found that people evaluate decisions based on reference points, which can be the current state or expected benchmarks. People compare the differences between decision outcomes and reference points to assess the value of the decision and the likelihood of regret. The following are the core steps of using the prospect–regret theory:

Step 1: Use Formula (5) to calculate the value function for each decision-maker, which involves the difference between each decision-maker's score and the average score of all the decision-makers, serving as their losses and gains.

$$v(\Delta x_{ij}) = \begin{cases} (\Delta x_{ij})^{\alpha} & \Delta x_{ij} \geq 0 \\ -\lambda(-\Delta x_{ij})^{\beta} & \Delta x_{ij} < 0' \end{cases} \tag{5}$$

where λ represents the loss aversion coefficient, with a larger value indicating that the decision-maker is more sensitive to losses.

Step 2: Obtain the prospect value matrix V_{ij} according to Formula (6), where w represents the attribute weights. Additionally, the maximum prospect value V_i^{+} and the minimum prospect value V_i^{-} can be obtained.

$$V_{ij} = v(\Delta x_{ij})w. \tag{6}$$

Step 3: Obtain the delight value matrix R_{ij} according to Formula (7) and the Hamming distances.

$$Z_i(x) = \sum_{j}^{m} (R_{ij}(x) + G_{ij}(x)). \tag{7}$$

Step 4: Obtain the regret value matrix G according to Formula (8) and the Hamming distances.

$$G_{ij}(x) = 1 - \exp\left[-\delta\left|\frac{V_{ij}(x) - V_{ij}^{-}(x)}{V_{ij}^{+}(x) - V_{ij}^{-}(x)}\right|\right]. \tag{8}$$

Step 5: Calculate the delight–regret value using Formula (9).

$$Z_i(x) = \sum_{j}^{m} (R_{ij}(x) + G_{ij}(x)). \tag{9}$$

4. The LSGDM Method Based on Dual Fine-Tuning

The first step of this study involves a sentiment analysis based on online review data. Secondly, a social relationship network is constructed by integrating expert opinions' similarity and trust relationships. Based on this, an LSGDM with dual fine-tuning is adopted. Finally, ranking is conducted using the prospect–regret theory.

4.1. Data Processing and Sentiment Analysis

Since consumers are heavily influenced by online product reviews during the consumption process, the primary step is to effectively extract and analyze the sentiment factors from the review data and then accurately and effectively convert them into preference datasets. This section will extract five different brands of solar lighting products from online platforms as the solution set to verify the feasibility of the sentiment analysis methods. The extracted data include not only online reviews but also the star ratings given by consumers for the products.

4.1.1. Data Processing

Since the raw online review data may contain a large amount of noise, data processing can help to filter out irrelevant information and improve data quality, making it suitable for sentiment analysis models to enhance accuracy and efficiency. The following are the three steps of data processing:

(1) Text cleaning: Since the collected data may have some noise, this section first removes duplicate reviews and some emoticons to create a new text.

(2) Data tokenization: This section uses ChatGPT tokenization for extracting keywords and the sentiment analysis in the subsequent steps.

(3) Stopword filtering: The tokenized text may contain words like particles, numbers, mathematical symbols, English characters, etc., which do not affect the results. This paper filters out these stop words to avoid affecting the effectiveness of the sentiment analysis.

4.1.2. Constructing the Sentiment Dictionary

After the tokenization process described in Section 3.1.1, each sentence from the reviews is divided into individual words. The sentiment analysis primarily determines the sentiment expressed by the entire sentence based on the sentiment orientation of the words. This section assists the subsequent steps by constructing dictionaries of positive and negative sentiments. The positive emotion lexicon and negative emotion lexicon are shown in Tables 1 and 2, respectively.

Table 1. Building a positive emotion lexicon.

Broad range, affordable, convenient, satisfactory, likable, elegant, user-friendly, pleasant, patient, good review, value for money, thoughtful, worthy, reliable, beautiful, high brightness, long lifespan, attractive, repurchase, warm, highly recommended, cost-effectiveness, recommend, soft, good.

Table 2. Building a negative emotion lexicon.

Very dark, unpleasant, troublesome, missing parts, not bright, bad review, not in accordance, really bad, disappointed, useless, plastic packaging, deceptive, misleading, inferior, abnormal sound, not up to standard, mediocre, too dim, deceptive, cracked, glaring, return, very small, collapsed, deformed, damaged.

4.1.3. Sentiment Orientation Ratio

This section conducts a sentiment evaluation based on the star ratings collected from consumers for various brands. Ratings of 1 and 2 stars are classified as negative, 3 stars as neutral, and 4 and 5 stars as positive. The percentage of emotional tendency is shown in Table 3.

Table 3. The percentage of emotional tendency for five brands.

	Negative	**Neutral**	**Positive**
Xiang Zhe	27.3%	27.3%	45.4%
Shu FuJia	36.4%	36.4%	27.2%
Shuo Shi	18.2%	54.5%	27.3%
BELAN	30.0%	30.0%	40.0%
You Chi	10.0%	20.0%	70.0%

4.1.4. The Fuzzy Number Acquisition

Mapping the adverbs describing the degree of good or bad for a certain attribute of the product in the reviews to trapezoidal fuzzy numbers. The correspondence table between product ratings and trapezoidal fuzzy numbers is shown in Table 4.

Table 4. The correspondence table between product ratings and trapezoidal fuzzy numbers.

Product Ratings	**Trapezoidal Fuzzy Numbers**
Very Poor	(0.0, 0.0, 0.1, 0.2)
Poor	(0.1, 0.2, 0.2, 0.3)
Fairly Poor	(0.2, 0.3, 0.4, 0.5)
Moderate	(0.4, 0.5, 0.5, 0.6)
Fairly Good	(0.5, 0.6, 0.7, 0.8)
Good	(0.7, 0.8, 0.8, 0.9)
Very Good	(0.8, 0.9, 1.0, 1.0)

Human language is used to describe the advantages and disadvantages of product attributes. To convert them into fuzzy numbers, the following fuzzy number conversion process is conducted:

Symbols used in the conversion process: let us denote the score, ranging from $[-3,3]$. Trapezoidal fuzzy numbers are denoted as (a,b,c,d), triangular fuzzy numbers as (e,f,g), and fuzzy numbers as f. The specific steps for converting the product evaluations into fuzzy numbers are as follows:

Step 1: Assign scores to different brands and attributes based on adverbs of degree to ensure that the scores fall within the range of trapezoidal fuzzy numbers, calculated as $S = \frac{s}{3}$.

Step 2: Use Formula (10) to calculate the left triangular fuzzy number fl and the right triangular fuzzy number fr (replaced by f_1 in the following formulas). When calculating the value corresponding to the left triangular fuzzy number, input (a,b,c), and when calculating the value corresponding to the right triangular fuzzy number, pass in (b,c,d):

$$\begin{gathered}\text{If } S \le e,\ f_1 = 0;\\ \text{If } e \le S \le f,\ f_1 = \tfrac{S-e}{f_1-e};\\ \text{If } f \le S \le g,\ f_1 = \tfrac{g-S}{g-f_1};\\ \text{If } g \le S,\ f_1 = 0.\end{gathered} \tag{10}$$

Step 3: Utilize the following Formula (11) to determine the fuzzy number:

$$f = \frac{fl + fr}{2}. \tag{11}$$

4.2. The Improved Louvain Algorithm

Construct a relationship network among experts based on trust relationships and the similarity of opinions among decision-makers, and then utilize the improved Louvain algorithm on this network to cluster large populations and obtain aggregated weights.

4.2.1. The Problem Description

The formal representation of fuzzy large-group decision-making with dual fine-tuning in this paper is as follows:

Let $X = \{x_1, x_2, \ldots, x_m\}(m \ge 2)$ be the set of alternative solutions, where x_i represents the *i-th* solution; $C = \{c_1, c_2, \ldots, c_n\}(n \ge 2)$ be the set of attributes, where c_j represents the *j-th* attribute.

Similarly, let $E = \{e_1, e_2, \ldots, e_k\}(k \ge 20)$ be the set of decision-makers, where e_f denotes the f*-th* decision-makers; $\omega = \{\omega_1, \omega_2, \ldots, \omega_k\}^T$ be the weight vector of the decision-makers, where ω_f denotes the weight of the f*-th* decision-makers, satisfying $\omega_f \ge 0$ and $\sum\limits_{f=1}^{k} \omega_f = 1$.

This paper assumes that each decision-maker's score matrix during the CRP is denoted as w. The average score matrix is represented by avg_g, the variance matrix by var_g, the consensus matrix by con, and the group consensus by g_con. Additionally, the distances matrix between each pair of decision representatives is denoted by dis.

In the prospect–regret theory, it is assumed that the value function matrix is V, the prospect value matrix is pre, the joy value matrix is P, the regret value matrix is Q, and the joy–regret matrix is Re.

4.2.2. The Wasserstein Distance

The Wasserstein distance, alternatively referred to as the Earth Mover's distance, originates from transportation problems. In transportation problems, resources need to be transported from one location to another, but the distances and transfer costs between each pair of locations may vary. The calculation process of Wasserstein distances can be viewed as the optimal transportation of one probability distribution to another, where the cost of each transfer is associated with the difference between the probability distributions. Specifically, given two probability distributions, μ and ν, the Wasserstein distance is defined as the minimum total cost of transporting one probability distribution to another. This cost can be computed by finding the best matching between the two distributions, where the cost of transferring each element from one distribution to another is proportional to the distances between them. The calculation of Wasserstein distances can be achieved using linear programming or convex optimization-based methods [43].

Given two probability distributions, μ and ν, their Wasserstein distance is defined as:

$$W_p(\mu, v) = (\inf_{\gamma \in \Gamma(\mu, v)} \int_{R^d \times R^d} \|x - y\|^p d\gamma(x, y))^{1/p}, \tag{12}$$

where p is the order of the norm (usually 1 or 2), $\Gamma(\mu, v)$ is the set of all the joint distributions with marginal distributions μ and v, and $\gamma(x, y)$ represents the joint distribution where x and y are from the two probability distributions, respectively.

4.2.3. The Methodology of the Louvain Algorithm

Based on the list of fuzzy number matrices, the similarity of opinions among decision-makers is determined using a combination of the Euclidean distances formula and the Wasserstein distances formula. The distances formula is as follows:

$$d_{ij}^m = \alpha d_{ij}^e + (1 - \alpha) d_{ij}^w, \tag{13}$$

where α is a weight coefficient used to balance the importance of the Euclidean distances and the Wasserstein distances. Additionally, α is generally set to the average value to ensure equal relative contributions of both distances in the distance measurement.

The similarity matrix R is obtained according to Formula (14):

$$r_{ij} = 1 - \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij}^m. \tag{14}$$

By integrating the trust relationships among decision-makers with the similarity of opinions, we derive the relationship coefficients between decision-makers using Formula (15), yielding the relationship matrix among decision-makers:

$$y_{ij} = \frac{1}{2} p_{ij} + \frac{1}{2} r_{ij}. \tag{15}$$

When calculating the weights of the clusters and decision-makers, the aggregate scale and the degree centrality of the network are considered. The weights of each cluster $U = \{u_1, u_2, u_3, \ldots, u_q\}$ are obtained using Formula (16), followed by the calculation of the weights of each decision-maker $\omega = \{\omega_1, \omega_2, \omega_3, \ldots, \omega_k\}$ based on Formula (17):

$$u_q = \frac{\sum\limits_{i \in u_q} y_{ij}}{2 \sum y_{ij}} + \frac{N_q}{2k}; \tag{16}$$

$$\omega_k = \frac{\sum\limits_{j \in u_q} y_{kj}}{\sum\limits_{i \in u_q} \sum\limits_{j \in u_q} y_{ij}}. \tag{17}$$

Finally, the specific algorithm steps are presented in Algorithm A1 of the Appendix A.

4.3. The CRP Based on Dual Fine-Tuning

This section will introduce the two stages of the CRP based on dual fine-tuning in detail. The specific steps will be provided in Algorithm A2 of the Appendix A.

In the actual decision scoring process, decision-makers' scores for each product are often influenced by significant personal subjective emotions. To address this issue, a two-stage group CRP is established to obtain objective product ratings.

The CRP adopts two stages to effectively reduce the decision-making and adjustment process, improve decision-making efficiency, and reduce time complexity.

In the first stage, decision scores within different decision groups are derived:

After clustering using the improved Louvain algorithm, the matrix $w[i][j][k]$ is obtained. Then, the average score matrix $avg\_g[j][k]$ is calculated by averaging each row of matrix w. Next, the variance var_g is calculated using Formula (18) to indicate the dispersion of each score relative to the average value.

$$\left(\frac{1}{x}\sum_{j=0}^{x-1}\left(\frac{1}{y}\sum_{k=0}^{y-1}(w[i][j][k]-avg\_g[j][k])^2\right)\right). \tag{18}$$

In the second stage, adjustments are made for the dual minimal consensus. It is important to note that although the letter selection for the number of representatives remains unchanged in the second round, it now represents the decision representatives selected in the first round:

First, use Formula (19) to calculate the distance matrix $dis[n][n]$ between any two decision-makers, finding the Euclidean distance between them, where a and b are the identifiers of any two selected decision representatives:

$$\left(\frac{1}{x}\sum_{j=0}^{x-1}\left(\frac{1}{y}\sum_{k=0}^{y-1}(w[i][j][k]-avg\_g[j][k])^2\right)\right). \tag{19}$$

Next, calculate the consensus degree between a decision-maker and all other decision-makers using the distance matrix $dis[n][n]$ and Formula (20). Obtain an n-dimensional matrix $con[n]$:

$$1-\frac{1}{n}\sum_{j=0}^{n-1}\frac{dis[i][j]}{18}. \tag{20}$$

Then, use Formula (21) to calculate the average consensus degree matrix to obtain the final group consensus degree g_con:

$$\frac{1}{n}\sum_{i=0}^{n-1}con[i]. \tag{21}$$

Finally, Formula (22) is used for the score adjustment:

$$\begin{gathered}\text{If } w[i][j][k] > avg\_p[j][k],\\ \text{then } w[i][j][k]-=\tfrac{avg\_p[j][k]}{9};\\ \text{If } w[i][j][k] < avg\_p[j][k],\\ \text{then } w[i][j][k]+=\tfrac{avg\_p[j][k]}{9}.\end{gathered} \tag{22}$$

The flowchart of the dual fine-tuning CRP is shown in Figure 2.

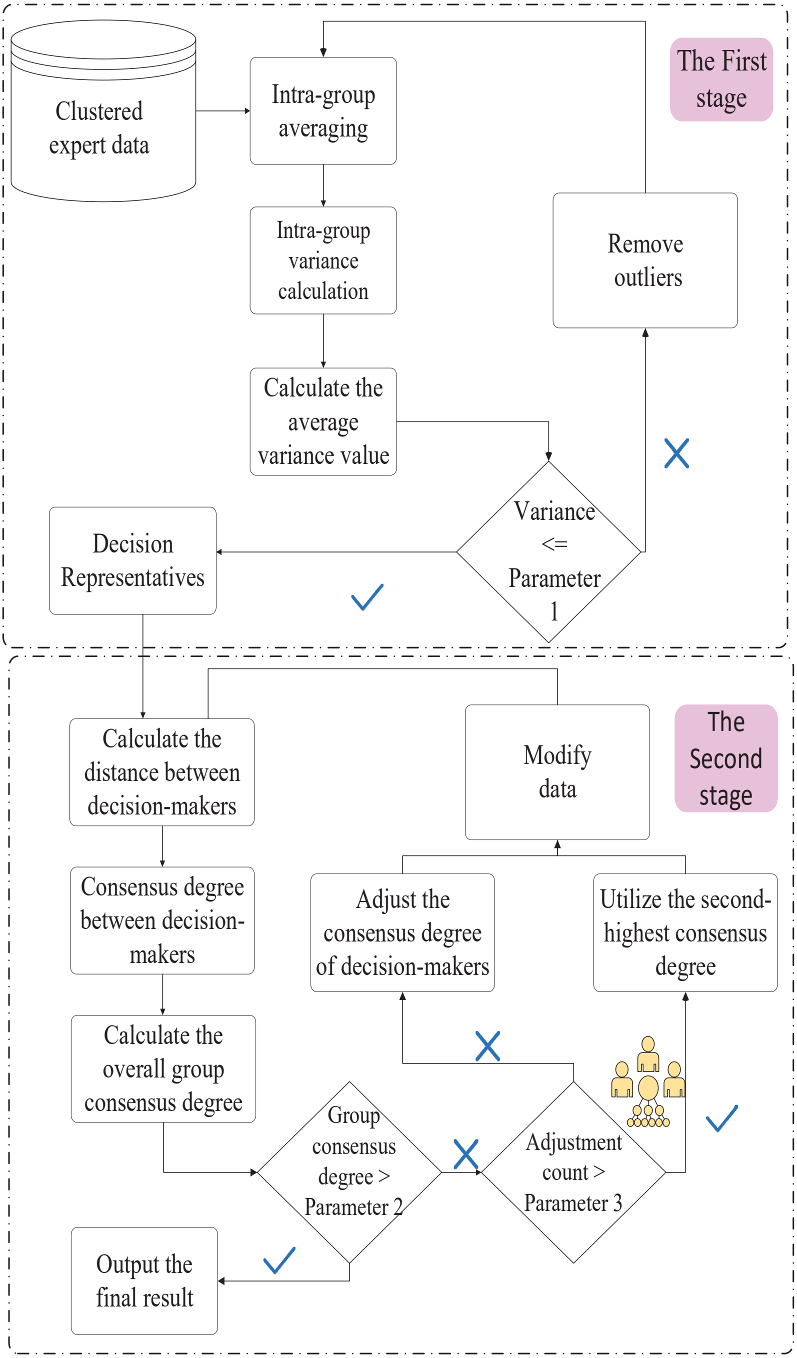

Clustered expert data
Intra-group averaging
Intra-group variance calculation
Calculate the average variance value
Remove outliers
The First stage
Variance <= Parameter 1
Decision Representatives
Calculate the distance between decision-makers
Modify data
The Second stage
Consensus degree between decision-makers
Adjust the consensus degree of decision-makers
Utilize the second-highest consensus degree
Calculate the overall group consensus degree
Group consensus degree > Parameter 2
Adjustment count > Parameter 3
Output the final result

Figure 2. The CRP with dual fine-tuning.

4.4. Prospect–Regret Theory

The prospect–regret theory can effectively prevent the regret or elation that decision-makers might experience after the decision results are generated. In this section, we will combine the existing knowledge of the prospect–regret theory with the final results of the CRP to obtain the ranking of the options. This method can effectively address the impact of decision-makers' regret or elation on the experimental results. The specific experimental steps are detailed in Algorithm A3 of the Appendix A.

5. Instance Analysis

This section will verify the decision-making model based on dual minimum value fine-tuning using the case of solar light selection, and present the steps of the specific algorithm. The overall experimental process is shown in Figure 3. The overall experimental steps are seen in Algorithm A4 of the Appendix A. In addition, this section not only compares the classical multi-attribute decision-making methods with the method presented in this paper, but it also conducts a sensitivity analysis on the method presented in this paper using different parameter values.

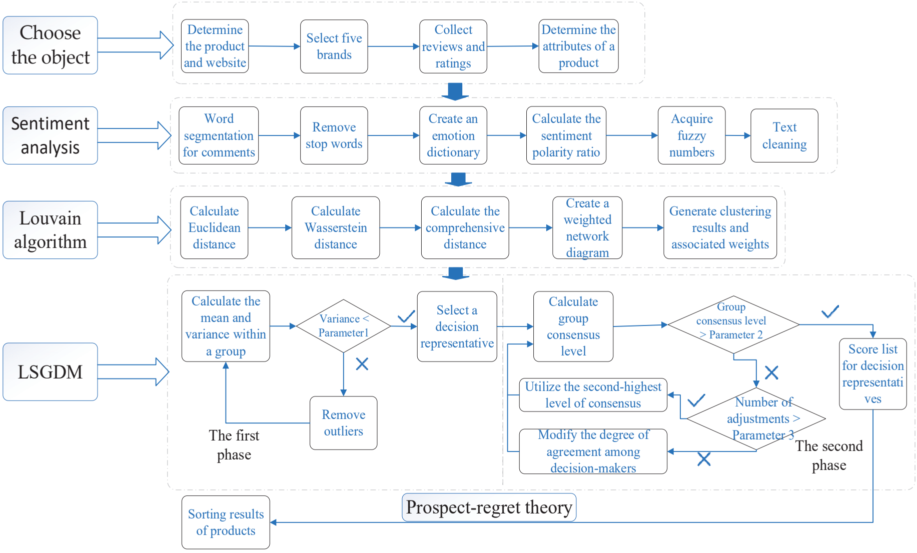

Choose the object
Determine the product and website
Select five brands
Collect reviews and ratings
Determine the attributes of a product
Sentiment analysis
Word segmentation for comments
Remove stop words
Create an emotion dictionary
Calculate the sentiment polarity ratio
Acquire fuzzy numbers
Text cleaning
Louvain algorithm
Calculate Euclidean distance
Calculate Wasserstein distance
Calculate the comprehensive distance
Create a weighted network diagram
Generate clustering results and associated weights
LSGDM
Calculate the mean and variance within a group
Variance < Parameter1
✓
✗
Remove outliers
The first phase
Select a decision representative
Calculate group consensus level
Group consensus level > Parameter 2
✓
✗
Score list for decision representati ves
Utilize the second-highest level of consensus
✓
Number of adjustments > Parameter 3
✗
Modify the degree of agreement among decision-makers
The second phase
Prospect-regret theory
Sorting results of products

Figure 3. The experimental procedure flowchart.

5.1. Instance Description

The data used in this section come from reviews of solar streetlights sold on an online platform. To ensure the data are broadly representative, this article selects five solar light brands with a high sales volume and rich reviews, including Xiangzhe, Shufujia, Shuoshi, BELAN, and Youchi.

First, assume $X = \{x_1, x_2, x_3, x_4, x_5\}$ to be the set of alternative solutions, which are "Xiang Zhe", "Shu Fu Jia", "Shuo Shi", "BELAN", and "You Chi".$C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$. The meanings they represent are brightness, duration of light, price, appearance, service attitude, and product quality. Let $E = \{e_1, e_2, e_3, e_4, \ldots, e_{30}\}$ represent the decision-makers among them. Use $V = \{v_1, v_2, \ldots, v_n\}$ to represent the different groups after clustering the decision-makers.

Apply this dataset to the decision-making model proposed in Section 3 for decision-making, with the following steps:

Step 1: Process the crawled review data according to the first part of Section 3, obtaining a score matrix for different decision-makers for different brands. Use the formula to derive a fuzzy number matrix from the score matrix.

Step 2: Cluster all the obtained fuzzy number matrices as described in the second part of Section 3, deriving inter-group weights, intra-group weights, and clustering group results, as shown in Table 5. The results of clustering all the obtained fuzzy number matrices are shown in Figure 4.

Table 5. Clustering and decision-maker weight results.

Clustering Result	Expert Individuals and Weights	Cluster Weighting
e_1	$\mathrm{v}_4(0.08), \mathrm{v}_{10}(0.07),$ $\mathrm{v}_0(0.08), \mathrm{v}_{16}(0.08),$ $\mathrm{v}_{20}(0.08), \mathrm{v}_{21}(0.06),$ $\mathrm{v}_{23}(0.08), \mathrm{v}_{28}(0.08),$ $\mathrm{v}_{15}(0.06), \mathrm{v}_{12}(0.06),$ $\mathrm{v}_{17}(0.08), \mathrm{v}_{22}(0.06),$ $\mathrm{v}_{30}(0.07),$	0.41
e_2	$\mathrm{v}_1(0.08), \mathrm{v}_{14}(0.09),$ $\mathrm{v}_8(0.08), \mathrm{v}_{11}(0.09),$ $\mathrm{v}_5(0.08), \mathrm{v}_{18}(0.08),$ $\mathrm{v}_6(0.09), \mathrm{v}_{27}(0.08),$ $\mathrm{v}_{26}(0.08), \mathrm{v}_{13}(0.08),$ $\mathrm{v}_{19}(0.09), \mathrm{v}_{29}(0.08)$	0.39

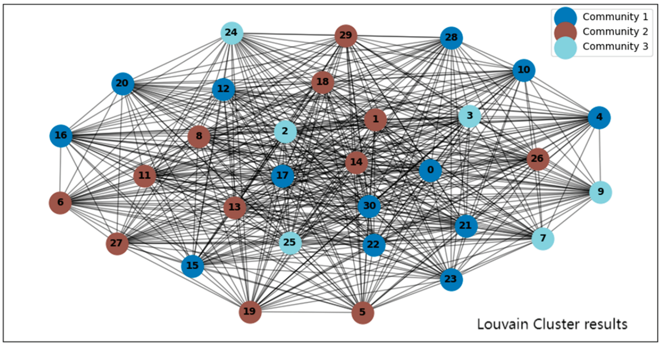


Figure 4. The clustering result graph.

Step 3: Based on the clustering groups and weights obtained in Step 2, conduct the first round of the consensus adjustment. In this round, the scores, means, and variances will be calculated to remove the outliers within each group. Finally, the decision-maker score matrix for each group is selected as follows:

$$\begin{bmatrix} 2.45 & 2.20 & 0.30 & 0.20 & 0.10 & 1.70 \\ 2.05 & -0.20 & 0.50 & 1.40 & 0.20 & 2.20 \\ 2.05 & 2.05 & 0.15 & 0.85 & 0.05 & 2.60 \\ 2.00 & 1.25 & 0.10 & 2.00 & 0.10 & 2.70 \\ 2.05 & 0.89 & 0.00 & 0.50 & 0.15 & 0.55 \end{bmatrix},$$

$$\begin{bmatrix} 0.50 & 1.80 & 0.00 & 2.30 & 0.10 & 1.60 \\ 2.40 & 0.80 & 0.20 & -0.30 & 0.00 & 0.10 \\ -0.20 & 0.50 & 2.60 & 1.60 & 0.10 & 2.30 \\ 0.20 & 2.10 & 0.00 & 0.40 & 0.10 & 1.60 \\ 0.20 & 0.40 & 0.00 & 1.50 & 0.10 & 0.30 \end{bmatrix},$$

$$\begin{bmatrix} 2.20 & 1.50 & 0.00 & 1.40 & 1.90 & 0.10 \\ 1.50 & 2.40 & 0.10 & 1.10 & 0.00 & 1.30 \\ 2.50 & 2.30 & 0.00 & 0.30 & 0.10 & 1.60 \\ 2.40 & 2.80 & 0.00 & 1.60 & 1.70 & 2.10 \\ 2.50 & 2.30 & 0.00 & 0.10 & 1.40 & 1.40 \end{bmatrix}.$$

Step 4: In the second round, the consensus degree of the decision-making representatives was -0.024498147569775024, which did not meet the consensus requirements. Subsequently, consensus adjustments were made based on the above scores, and after ten rounds of adjustments, focusing on the representative with the lowest consensus degree, a consensus level was reached, with a final consensus degree of 0.8019555217792464. This resulted in the final adjusted decision-making representative score matrix.

Step 5: According to prospect–regret theory and combined with the decision-making representative score matrix obtained above, as follows:

$$\begin{bmatrix} 89.00 & 77.67 & 41.65 & 106.23 & 46.32 \end{bmatrix}.$$

According to the final score matrix, the ranking of the final products is as follows:

$$x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3.$$

Drawing a brand score and ranking chart as shown in Figure 5, based on the above rankings and scores.

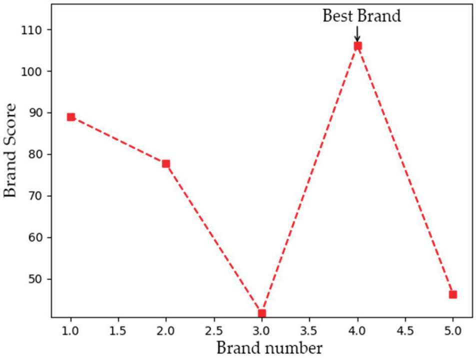


Figure 5. Our method ranks the comprehensive scores of the commodities.

Based on Figure 5 and the ranking, it can be concluded that BELAN's product is the best, after synthesizing the scores of six evaluation attributes for each solar light.

5.2. Advantages of Our Approach over Other Methods

In this section, the comparison will be divided into three parts: initially, we will delve into the significance and impact of the methodologies introduced in this paper for the purpose of decision-making. Secondly, we will undertake a comparative analysis using traditional multi-attribute decision-making techniques, including TOPSIS, TODIM, and VIKOR. Finally, a comparison by integrating the prospect–regret theory used in this paper with the above classic multi-attribute decision-making methods.

The corresponding methods and symbols used in this section are shown in Table 6.

Table 6. The comparison method corresponds to the symbolic diagram of the method representative.

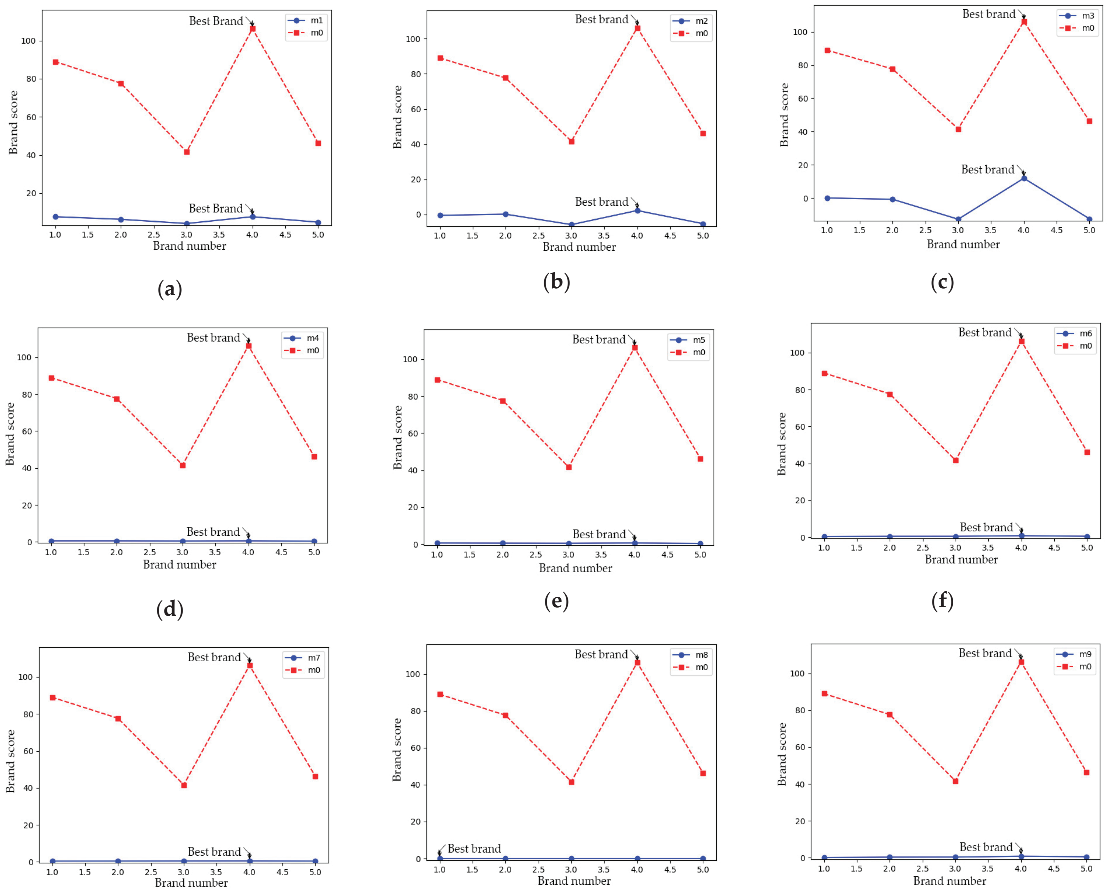
Comparative Method	Method Symbol
Two-stage dual minimum consensus degree adjustment combined with the prospect–regret theory	m0
Remove the prospect–regret theory from m0	m1
Replace the prospect–regret theory in m0 with the prospect theory	m2
Replace the prospect–regret theory in m0 with the regret theory	m3
Classic multi-attribute decision-making method TOPSIS	m4
Classic multi-attribute decision-making method TODIM	m5
Classic multi-attribute decision-making method VIKOR	m6

Table 6. *Cont.*

Comparative Method	Method Symbol
Two-stage double minimum consensus degree adjustment combined with TOPSIS	m7
Two-stage double minimum consensus degree adjustment combined with TODIM	m8
Two-stage dual minimum consensus degree adjustment combined with VIKOR	m9

5.2.1. Compare Different Methods with Our Method

To clarify the impact of the important steps in the method proposed in this paper on decision-making outcomes, a comparison is made between the method used and its replacement methods during decision-making in m1 to m3. The focus is on three treatments of the prospect–regret theory proposed in this paper: removing the prospect–regret theory directly, replacing the prospect–regret theory with the prospect theory, and replacing the prospect–regret theory with the regret theory. The method m0 is compared with the replaced methods m1, m2, and m3, respectively. The experimental result graphs for these three treatments correspond to Figures 6a, 6b and 6c, respectively.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

(**g**) (**h**) (**i**)

Figure 6. Comparison chart of this method with other methods.

To compare with the classic multi-attribute decision-making methods and identify the advantages of the method proposed in this paper, the classic multi-attribute decision-making methods TOPSIS, TODIM, and VIKOR are used for comparison, that is, method m0 is compared with classic methods m4, m5, and m6, respectively. The experimental result graphs obtained from the comparison correspond to Figures 6a, 6b and 6c, respectively.

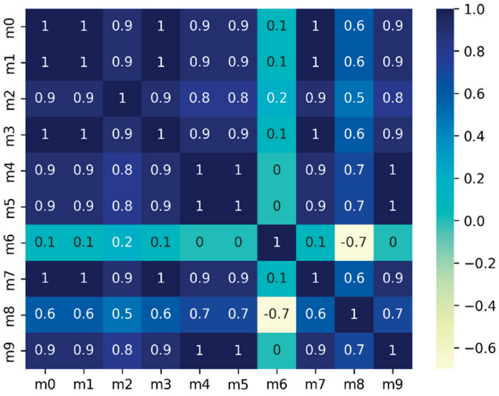
To further control the variables, the two-stage minimum consensus-reaching process proposed in this paper is combined with the classic multi-attribute decision-making methods TOPSIS, TODIM, and VIKOR, respectively, and methods m7, m8, and m9 are compared with the method m0 proposed in this paper, respectively, in order to obtain more convincing comparative results.

It can be observed from figures a–i that the method proposed in this paper generally exhibits higher final product score differentiation compared to other methods. particularly in figures d–i, the product scores for methods m4 to m9 show overall less differentiation, whereas the scores obtained from our method have increased gaps, reducing the bias in final decision-making. moreover, the optimal brand selected by the method proposed in this paper aligns with the majority of other methods, except for a deviation with method m8 in figure h, which indicates the accuracy of the results produced by our proposed method.

According to the experimental results, it can be seen that the decision-making method proposed in this article is more conducive to producing decision outcomes. The significant differences between brands can effectively reduce user hesitation when choosing a brand and minimize deviations in selecting products. Moreover, the optimal product selected by the method proposed in this paper is basically consistent with the optimal product selected under the other decision-making methods, which also proves the correctness of this method for decision outcomes.

5.2.2. Similarity Analysis of Final Decision Outcomes Using Different Methods

This section presents two comparative experiments. Firstly, it compares the similarities between the sorting results of the method proposed in this article with those of the comparative methods, demonstrating that the method introduced in this paper aligns well with the decision-making outcomes. The results of this comparison are shown in Figure 7. Subsequently, it compares the final scores assigned to each product by the method proposed in this article with the scores assigned by the comparative methods. The results of this comparison are shown in Figure 8.



	m0	m1	m2	m3	m4	m5	m6	m7	m8	m9
m0	1	1	0.9	1	0.9	0.9	0.1	1	0.6	0.9
m1	1	1	0.9	1	0.9	0.9	0.1	1	0.6	0.9
m2	0.9	0.9	1	0.9	0.8	0.8	0.2	0.9	0.5	0.8
m3	1	1	0.9	1	0.9	0.9	0.1	1	0.6	0.9
m4	0.9	0.9	0.8	0.9	1	1	0	0.9	0.7	1
m5	0.9	0.9	0.8	0.9	1	1	0	0.9	0.7	1
m6	0.1	0.1	0.2	0.1	0	0	1	0.1	-0.7	0
m7	1	1	0.9	1	0.9	0.9	0.1	1	0.6	0.9
m8	0.6	0.6	0.5	0.6	0.7	0.7	-0.7	0.6	1	0.7
m9	0.9	0.9	0.8	0.9	1	1	0	0.9	0.7	1

Figure 7. The Spearman's rank correlation plot of the similarity of product rankings.

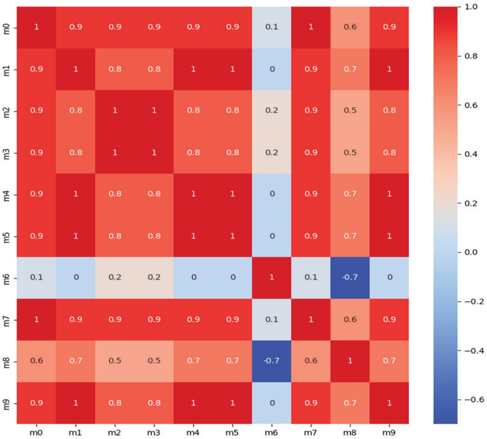


Figure 8. Spearman's rank correlation plot of product score similarity.

5.2.3. Analysis of Advantages Compared with Other Methods

To intuitively compare the advantages with other methods, comparisons were made in four aspects: the sentiment analysis, CRP, risk assessment, and classification ability. The comparison results are shown in Table 7.

Table 7. The advantages compared with other methods.

	Sentiment Analysis	CRP	Risk Assessment	Classification Ability	Accuracy
m1	√	√	×	√	×
m2	√	√	√	√	×
m3	√	√	√	√	×
m4	×	×	×	×	×
m5	×	×	×	×	×
m6	×	×	×	×	×
m7	×	×	×	√	×
m8	×	×	×	√	×
m9	×	×	×	√	×
m0	√	√	√	√	√

5.3. Performance Testing and Sensitivity Analysis

To enhance the practicality of the method, an analysis of its execution performance and sensitivity is conducted. The performance analysis includes the adjustment time when the decisions are made and the number of adjustment rounds for decision-makers. The sensitivity analysis mainly focuses on the parameters used in prospect–regret theory. By following these procedures, the merits of the method presented in this paper become even more evident.

5.3.1. Performance Analysis

As the decision-making process incorporates a growing number of experts, the CRP model introduced in this paper demonstrates its ability to swiftly converge on a consensus within a reasonable timeframe. Even when the number of experts balloons to 1000, the execution time remains under 70 s, as illustrated in Figure 9. This graphical representation highlights the efficient relationship between the number of decision-makers and the corresponding execution time. These findings underscore the practicality and dependability of the proposed method, particularly in scenarios where the decisions involve significant numbers of experts, extending into the thousands.

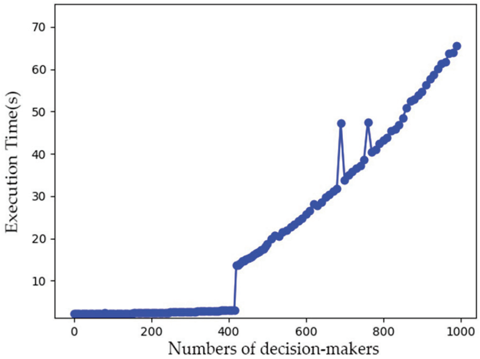


Figure 9. Performance analysis experimental chart.

The above simulation experiment was conducted using PyCharm Community Edition 2023.3.2, executed on a computer equipped with an Intel® Core™ i7-12700 processor from the 12th generation, with a running frequency of 3.90 GHz.

In order to ensure that within a certain range of decision-makers, the adjustment of the decision-maker scores can reach the required consensus level in a limited number of rounds, without the situation where consensus cannot be adjusted. The number of adjustment rounds for the least and second-least consensus degrees among 1000 decision-makers is counted. The results are shown in Figure 10 and Figure 11, respectively:

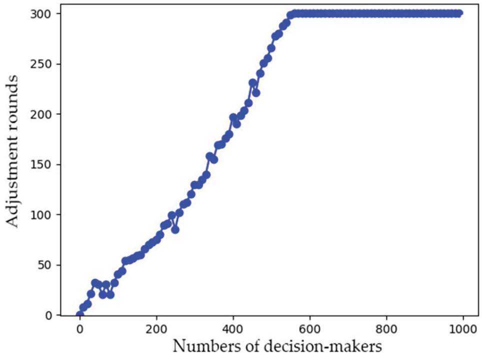


Figure 10. The number of adjusting rounds for the first minimum.

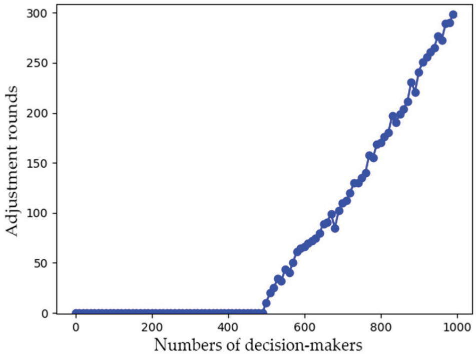


Figure 11. The number of adjusting rounds for the second.

5.3.2. Sensitivity Analysis

In this segment, the robustness of the proposed method is tested by conducting sensitivity analyses on the parameters α, β, and λ involved in the decision-making process.

In the content of the previous section, the value of α was set to 1.21. In this section, its value is set to 2.42 and 3.63, respectively. The ranking results of the goods under different values are shown in Table 8, and the sensitivity test results are shown in Figure 12.

Table 8. The sorting results when the parameters α take different values.

Parameter α Value	Sorting Result
$\alpha = 1.21$	$x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3$
$\alpha = 2.42$	$x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3$
$\alpha = 3.63$	$x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3$

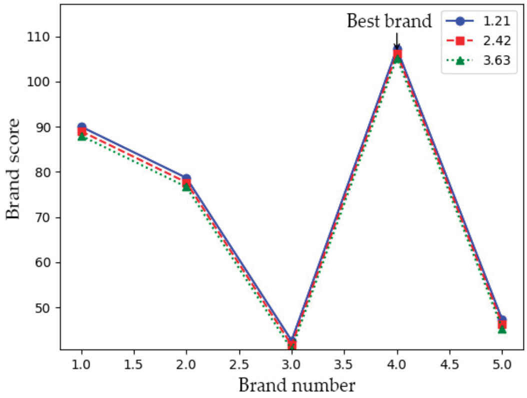


Figure 12. The sensitivity test for parameters α.

During the course of making a decision, the value of β is set to 1.02. To verify the stability of the results, its values are set to 2.04 and 3.06, respectively. The ranking results of the goods with different values are shown in Table 9, and the outcomes of the sensitivity analysis are presented in Figure 13.

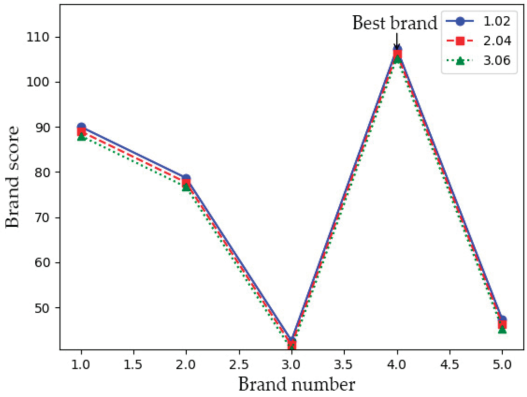


Figure 13. The sensitivity test for parameters β.

Table 9. The sorting results when the parameters β take different values.

Parameter β Value	Sorting Result
$\beta = 1.02$	$x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3$
$\beta = 2.04$	$x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3$
$\beta = 3.06$	$x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3$

During the calculation, the value of λ is taken as 2.25. In this section, its values are respectively taken as 4.5 and 6.75. The sorting results of the goods under different values are shown in Table 10, and the findings of the sensitivity test are depicted in Figure 14.

Table 10. The sorting results when the parameters λ take different values.

Parameter λ Value	Sorting Result
$\lambda = 2.25$	$x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3$
$\lambda = 4.5$	$x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3$
$\lambda = 6.75$	$x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3$

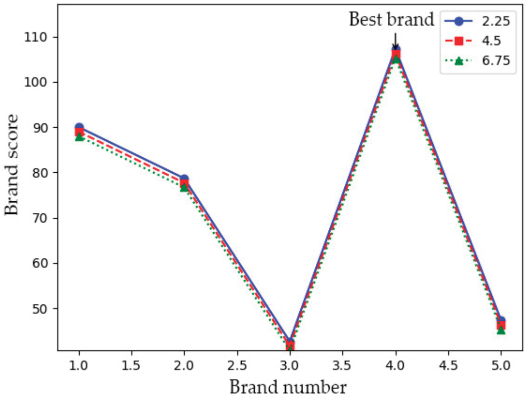


Figure 14. The sensitivity test for parameters λ.

6. Conclusions

This paper first converts the sentiments contained in online reviews into fuzzy numbers through the sentiment analysis. Next, it improves the Louvain algorithm using a mixed distance of Euclidean distances and Wasserstein distances. Then, a two-stage dual fine-tuning CRP model is used to adjust the scores of decision-makers. Additionally, the prospect–regret theory is utilized to address the potential joy and regret psychological issues that decision-makers might experience during the decision-making process. Ultimately, the model's applicability and operationality are confirmed via a case study involving the purchase of solar lights.

Considering the above analysis, the contributions of this paper are as follows:

(1) Incorporating a sentiment analysis into the LSGDM model to accurately quantify and analyze the fuzzy dataset of decision-makers' linguistic preferences.

(2) Using a mixed distance of Euclidean distances and Wasserstein distances to calculate the similarity between experts when constructing social networks based on the Louvain algorithm.

(3) Utilizing a two-stage process to reduce the decision scale while minimizing adjustments to decision-makers. Setting dual minimum consensus levels avoids multiple outlier situations and excessive adjustment times.

In future research, several aspects of the proposed method require further investigation to address its current limitations. First, the method lacks the capability to handle dynamically changing online review data effectively. Incorporating real-time monitoring and analysis processes is essential to adapt to continuous changes in review data. Techniques such as a time series analysis could explore trends and periodic changes, providing more timely and accurate information for decision-making. Second, the method does not adequately consider decision-makers' overconfidence during the CRP. Overconfidence can significantly influence decision outcomes, especially under uncertain conditions. A deeper analysis of how overconfidence affects the CRP is needed, focusing on how varying confidence levels among decision-makers impact consensus and decision quality. Addressing these limitations will enhance the method's robustness and applicability in real-world decision-making scenarios.

Author Contributions: Conceptualization, X.Y. and C.Z.; Data curation, X.Y.; Methodology, T.X. and S.H.; Software, T.X. and S.H.; Validation, T.X. and S.H.; Writing—original draft, X.Y. and S.H.; Writing—review & editing, X.Y. and C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Special Fund for Science and Technology Innovation Teams of Shanxi (202204051001015), the Training Program for Young Scientific Researchers of Higher Education Institutions in Shanxi, the Wenying Young Scholars of Shanxi University, the 22nd Undergraduate Innovation and Entrepreneurship Training Program of Shanxi University, and the Cultivate Scientific Research Excellence Programs of Higher Education Institutions in Shanxi (2019SK036).

Data Availability Statement: The dataset we used is already included in Appendix B at the end of the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Algorithm A1 is related to the methodological steps of Section 4.2.3, and it is the Louvain Algorithm.

Algorithm A1: The Louvain Community Clustering Algorithm Based on Mixed Distances

```
Input: The List of Fuzzy Number Matrices $|fuzzy\_numbers|$
Output: The Clustered Threshold Matrix
Loop $i$ from 1 to $|fuzzy\_numbers|$:
    Loop for $j$ from 1 to $|fuzzy\_numbers|$:
        Calculate the Euclidean distance according to Formula (3)
end
Loop for $i$ from 1 to $|fuzzy\_numbers|$:
    Loop for $j$ from 1 to $|fuzzy\_numbers|$:
        Calculate the Wasserstein distance according to Formula (12).
end
Loop $i$ from 1 to $|fuzzy\_numbers|$:
    Loop for $j$ from 1 to $|fuzzy\_numbers|$:
        Calculate the comprehensive distances according to Formula (13).
        Add an edge connecting node $i$ and node $j$ in the weighted network graph $G$.
        Calculate the decision-makers' weights according to Formulas (16) and (17).
end
Final Result: Clustering Results
```

Algorithm A2 is related to the methodological steps of Section 4.3, and it is the CRP based on dual fine-tuning.

Algorithm A2: The CRP Based on Dual Fine-Tuning

```
Input: The decision group (containing scores of different decision-makers for different products),
inter-group weights, intra-group weights
Output: The consensus decision representative score matrix
1 Initialize
2 Initialize attributes and brands with equal weights
3 Calculate the weight of each decision-makers for each attribute of each product, obtaining
        the weight matrix
4 while ($var\_g$ <parameter 1):
5 Calculate the average score matrix $avg\_g$ for each attribute of each product in each
decision group
6 Calculate the variance $var\_g$ for each decision group according to Formula (18)
        Remove decision-makers with larger variances:
7 Find the decision-makers with the largest variance
8 Remove the decision-makers from the decision group
9 end
10 Take the average score of the decision-makers as the decision representative score for that
decision group, obtaining the decision representative score matrix $w$ for all groups
11 Calculate the distances between each group leader and other group leaders according to
Formula (19), obtaining the distance matrix $dis$
12 Calculate the consensus level of each group leader according to Formula (20), obtaining the
consensus level matrix $con$
13 Calculate the group consensus level $g\_con$ according to Formula (21)
14 Find the group leader with the lowest consensus level
15 while ($g\_con <$ parameter 2 && the adjustment times for the group leader with the lowest
consensus level are less than parameter 3):
16  Adjust the score of the group leader with the lowest consensus level using Formula (22)
17 Repeat steps 12–15
18 end
19 while ($g\_con <$ parameter 2)
20 Adjust the score of the group leader with the second lowest consensus level using Formula (22)
21 Repeat steps 12–14
22 end
23 Final result: The adjust decision representative score list $W$
```

Algorithm A3 is related to the methodological steps of Section 4.4, and it is the final scheme ranking decision based on the prospect–regret theory.

Algorithm A3: Final Scheme Ranking Decision Based on Prospect-Regret Theory

```
Input: Weight matrix, decision representative score matrix $w$
Output: Scheme ranking
1 Calculate the average value matrix of the score matrix $w$
2 Calculate the loss and gain matrix based on the average value matrix and the decision
representative score matrix
3 Use Formula (5) from Section 3.4 to obtain the value function matrix $V$
4 Use Formula (6) from Section 3.4 to obtain the prospect value matrix $pre$
5 Compare the values in $pre$ one by one to find max$V$ and min$V$
6 $d = \max V - \min V$
7 Use Formula (7) from Section 3.4 to calculate the joy value matrix $P$
8 Use Formula (8) from Section 3.4 to calculate the regret value matrix $Q$
9 Use Formula (9) from Section 3.4 to calculate the joy–regret matrix $Re$
10 Obtain the transpose matrix $N$ of $w$
11 The comprehensive score is $Re \times N$
12 Rank the comprehensive scores
13 Final result: the scheme ranking from highest to lowest
```

Algorithm A4 is related to the LSGDM method based on double fine-tuning driven by online review data.

Algorithm A4: LSGDM Method Based on Double Fine-Tuning Driven by Online Review Data

Input: Fuzzy number matrix list, decision group (including different decision-makers' scores for different products), weight matrix
Output: Solution ranking

```
1 Initialize the fuzzy number dataset
2 Euclidean distance = calculate_euclidean_distance(fuzzy_number_set, metric = 'euclidean')
3 Wasserstein distance = create_zero_matrix(len(fuzzy_number_set), len(fuzzy_number_set))
4 for i in range(0, len(fuzzy_number_set) − 1):
    5 for j in range(i + 1, len(fuzzy_number_set) − 1):
        6 Wasserstein_distance[i][j] = calculate_wasserstein_distance(fuzzy_number_set[i], fuzzy_number_set[j])
        7 Wasserstein_distance[j][i] = Wasserstein_distance[i][j]
8 Network graph = create_empty_graph()
9 for i in range(0, len(fuzzy_number_set) − 1):
    10 for j in range(i + 1, len(fuzzy_number_set) − 1):
        11 weight = (Euclidean_distance[i][j] + Wasserstein_distance[i][j])/2
        12 add_edge(network_graph, i, j, weight)
13 Community division = detect_community_structure(network_graph, resolution = 0.85)
14 Community weights = {}
15 Total weight = calculate_sum_of_all_community_weights(community_division)
16 Initialize inter-group weights, intra-group weights, brand weights, and attribute weights
16 Initialize inter-group weights, intra-group weights, brand weights, and attribute weights
17 Weight result = calculate_weight(intra_group_weights, brand_weights, attribute_weights)
18 Average score = calculate_average(decision_group)
19 Variance result = calculate_variance(decision_group)
20 while (variance_result > parameter1):
    21 remove decision maker with the highest variance
    22 recalculate variance_result
23 Leader = average(decision_group)
24 Distance = calculate_distance(leader)
25 Consensus degree = calculate_consensus_degree(leader)
26 Group consensus degree = calculate_group_consensus_degree(leader, inter-group)
27 while (group_consensus_degree < parameter2 && adjustment_count < parameter3):
    28 adjust leader's score
    29 recalculate group_consensus_degree
30 If adjustment_count >= parameter3:
    31 then use the second largest consensus degree
    32 repeat steps 27–29
33 Calculate regret value matrix
32 Calculate delight–regret value matrix
33 Calculate transpose of the scoring matrix
34 Calculate comprehensive scoring matrix
35 Sort comprehensive scoring matrix
36 Output sorting results
36 Output sorting results
```

Appendix B

The data used in this section are as shown in the table below. It is defined that the final fuzzy number range for the product evaluation is from −3 to 3. In the table, each matrix represents a decision-making representative, and each row of it, respectively, represents a kind of product, which are "Xiang Zhe", "Shu Fu Jia", "Shuo Shi", "BELAN", and "You Chi" from top to bottom. Each column respectively represents different aspects of evaluation, which are the duration of light, price, appearance, service attitude, and product quality from left to right.

$\begin{bmatrix} 3.0 & 2.6 & 1.9 & 2.1 & 1.0 & 1.7 \\ 0.2 & 2.2 & 0.0 & 0.1 & 0.0 & 1.4 \\ 2.9 & 0.5 & 0.0 & 0.3 & 0.0 & 0.8 \\ 2.8 & 2.9 & 1.1 & 0.1 & 0.9 & 0.3 \\ 0.5 & 0.7 & 0.0 & 0.2 & 0.7 & -0.9 \end{bmatrix}$	$\begin{bmatrix} 0.5 & 0.7 & 0.0 & 0.2 & 0.7 & -0.9 \\ 0.1 & 0.0 & 0.1 & 1.3 & 0.0 & -0.1 \\ 2.8 & 1.9 & 0.1 & 2.6 & 0.2 & 2.3 \\ 1.4 & 1.9 & 0.2 & 0.3 & 0.3 & 0.0 \\ 2.5 & 2.5 & 0.2 & 0.8 & 0.0 & 2.1 \end{bmatrix}$
$\begin{bmatrix} 2.8 & 1.9 & 0.6 & 0.0 & 0.0 & 1.6 \\ 2.2 & -0.4 & 1.1 & 2.2 & 0.2 & 2.2 \\ 1.8 & 1.6 & 0.3 & 0.1 & 0.0 & 2.3 \\ 2.1 & 0.3 & 0.2 & 2.2 & 0.1 & 3.0 \\ 1.9 & 1.4 & 0.0 & 0.9 & 0.1 & 0.0 \end{bmatrix}$	$\begin{bmatrix} 2.1 & 2.5 & 0.0 & 0.4 & 0.2 & 1.8 \\ 1.9 & 0.0 & -0.1 & 0.6 & 0.2 & 2.2 \\ 2.3 & 2.5 & 0.0 & 1.6 & 0.1 & 2.9 \\ 1.9 & 2.2 & 0.0 & 1.8 & 0.1 & 2.4 \\ 2.2 & 0.4 & 0.0 & 0.1 & 0.2 & 1.1 \end{bmatrix}$
$\begin{bmatrix} 2.9 & 0.3 & 0.0 & 2.0 & 0.0 & 1.8 \\ 1.8 & 2.3 & 0.0 & 2.4 & 0.1 & 1.5 \\ -0.1 & 1.8 & 0.1 & 1.6 & 0.0 & 2.2 \\ 0.2 & 0.7 & 0.0 & 0.3 & 0.1 & 0.0 \\ 2.6 & 0.1 & 0.0 & 1.7 & 0.2 & 2.3 \end{bmatrix}$	$\begin{bmatrix} 0.5 & 1.8 & 0.0 & 2.3 & 0.1 & 1.6 \\ 2.4 & 0.8 & 0.2 & -0.3 & 0.0 & 0.1 \\ -0.2 & 0.5 & 2.6 & 1.6 & 0.1 & 2.3 \\ 0.2 & 2.1 & 0.0 & 0.4 & 0.1 & 1.6 \\ 0.2 & 0.4 & 0.0 & 1.5 & 0.1 & 0.3 \end{bmatrix}$
$\begin{bmatrix} 1.8 & 2.6 & 1.9 & 1.6 & 0.0 & 2.3 \\ 0.5 & 0.3 & 0.0 & 1.8 & 0.1 & 0.0 \\ 2.6 & 1.5 & 0.4 & 1.7 & 0.0 & 0.8 \\ 2.2 & 2.3 & 0.4 & 0.1 & 0.0 & 2.1 \\ 0.5 & 2.2 & 0.0 & 1.7 & 0.1 & 0.9 \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0.8 & 0.0 & 1.8 & 2.5 & 2.3 \\ 2.5 & 0.3 & 0.1 & 1.2 & 0.0 & 2.1 \\ 1.8 & 0.3 & 2.1 & 1.6 & 0.0 & 2.3 \\ -0.2 & 0.0 & 1.5 & 0.1 & 2.3 & 0.1 \\ 0.1 & 0.0 & 1.6 & -0.3 & 0.0 & -0.1 \end{bmatrix}$
$\begin{bmatrix} 0.9 & -0.2 & 0.0 & 1.9 & 1.6 & 2.3 \\ 2.9 & 1.8 & 0.0 & 2.4 & 0.0 & 0.3 \\ 0.9 & 0.2 & 0.0 & 1.7 & 2.2 & 1.9 \\ 2.3 & 0.0 & 1.9 & 2.3 & 1.6 & 1.5 \\ 2.6 & 2.2 & 1.7 & 0.4 & 0.0 & 1.5 \end{bmatrix}$	$\begin{bmatrix} 2.4 & 0.4 & 0.0 & 0.2 & 0.0 & 2.1 \\ 2.2 & 2.9 & 1.3 & 0.0 & 1.8 & 1.6 \\ 1.5 & 2.2 & 0.1 & 0.0 & 0.0 & 1.4 \\ 2.2 & 0.0 & 1.4 & 0.1 & 1.7 & 2.5 \\ 2.8 & 0.8 & 1.6 & 1.4 & 0.7 & 0.9 \end{bmatrix}$
$\begin{bmatrix} 2.2 & 1.5 & 0.0 & 1.4 & 1.9 & 0.1 \\ 1.5 & 2.4 & 0.1 & 1.1 & 0.0 & 1.3 \\ 2.5 & 2.3 & 0.0 & 0.3 & 0.1 & 1.6 \\ 2.4 & 2.8 & 0.0 & 1.6 & 1.7 & 2.1 \\ 2.5 & 2.3 & 0.0 & 0.1 & 1.4 & 1.4 \end{bmatrix}$	$\begin{bmatrix} 2.8 & 2.6 & 1.9 & -0.1 & 1.0 & 1.7 \\ 0.2 & 1.0 & 0.8 & 1.9 & 0.3 & 1.4 \\ 2.9 & 0.5 & 0.0 & 0.3 & 1.2 & 0.8 \\ 2.8 & 1.6 & 0.4 & 0.1 & 0.8 & 0.3 \\ 0.5 & 0.7 & 0.1 & 0.2 & 0.7 & 0.8 \end{bmatrix}$
$\begin{bmatrix} 0.1 & 0.0 & 0.1 & 1.3 & 0.0 & -0.1 \\ 1.8 & 2.5 & 0.0 & 0.3 & 0.9 & 0.5 \\ 2.2 & 1.5 & 0.0 & 0.1 & 1.8 & 2.3 \\ 0.5 & 1.2 & 1.3 & 1.6 & 2.9 & -0.5 \\ 0.5 & 0.2 & 1.5 & 0.2 & 2.4 & 1.6 \end{bmatrix}$	$\begin{bmatrix} -0.9 & 1.4 & 2.8 & -0.9 & 0.5 & 1.3 \\ 0.9 & 0.3 & 2.6 & 2.4 & 1.9 & -0.6 \\ -0.9 & 2.6 & 2.8 & -2.6 & 0.2 & 2.3 \\ 0.3 & 0.9 & 0.7 & 0.6 & 2.9 & 0.0 \\ -1.5 & 2.5 & 0.2 & -2.4 & 1.4 & 2.1 \end{bmatrix}$
$\begin{bmatrix} -1.2 & 0.2 & 1.8 & 2.1 & 0.2 & 1.7 \\ 1.5 & 0.4 & 0.3 & 1.4 & 2.4 & -0.3 \\ 0.5 & 1.6 & 0.6 & 1.0 & 1.3 & 0.8 \\ 1.2 & 0.3 & -2.8 & -0.4 & 0.5 & 0.3 \\ 2.2 & 1.4 & 0.0 & 0.2 & 0.7 & 1.5 \end{bmatrix}$	$\begin{bmatrix} -0.6 & 2.5 & 1.4 & 2.7 & 1.9 & -0.6 \\ 2.0 & 0.3 & 0.9 & 1.4 & -0.5 & 1.2 \\ 0.4 & 1.8 & 1.5 & 0.8 & 1.7 & 0.7 \\ 1.5 & 1.2 & 1.9 & 2.7 & -0.2 & 0.0 \\ 0.2 & 0.3 & 1.4 & 0.6 & 2.7 & 1.1 \end{bmatrix}$
$\begin{bmatrix} 0.8 & 1.6 & 1.3 & 2.3 & 1.0 & 1.7 \\ 1.3 & 1.8 & 0.6 & 0.1 & 2.4 & 1.4 \\ 1.2 & 1.6 & 2.7 & 1.0 & 0.4 & 0.8 \\ 1.0 & 1.2 & 0.9 & 0.1 & 0.7 & 0.3 \\ 2.6 & 2.5 & 1.7 & 0.2 & 1.4 & -0.9 \end{bmatrix}$	$\begin{bmatrix} 2.7 & 2.1 & 1.9 & -1.8 & 0.7 & 0.8 \\ 0.1 & 2.5 & 2.3 & 1.7 & 0.0 & -0.1 \\ 2.7 & 0.4 & 0.1 & 2.6 & 1.6 & 2.2 \\ 1.4 & 2.8 & 0.2 & 1.3 & 0.3 & 0.7 \\ 2.5 & 2.1 & 0.0 & 0.8 & 0.0 & 2.1 \end{bmatrix}$
$\begin{bmatrix} 0.2 & 1.3 & 0.5 & 0.3 & 0.7 & 0.8 \\ -0.1 & 1.0 & -1.2 & 0.1 & 2.9 & 1.4 \\ 2.4 & 0.5 & 1.1 & 0.3 & 0.0 & -0.2 \\ 2.8 & 2.9 & 1.1 & 3.0 & 0.9 & 0.3 \\ 0.0 & 0.7 & 1.2 & 0.2 & 0.7 & 0.5 \end{bmatrix}$	$\begin{bmatrix} 2.9 & 0.2 & 0.5 & 0.2 & 0.7 & -0.9 \\ 0.1 & 1.7 & 0.4 & 1.3 & 0.0 & 0.6 \\ 2.2 & 1.9 & 0.1 & 1.6 & 0.2 & 2.3 \\ 2.1 & 2.6 & -0.1 & 0.3 & 1.3 & 2.7 \\ 1.9 & 1.8 & 0.2 & 0.8 & 0.0 & 2.1 \end{bmatrix}$
$\begin{bmatrix} 0.2 & 1.3 & 0.5 & 0.3 & 0.7 & 0.1 \\ 0.5 & 1.0 & 2.5 & 0.1 & 2.9 & 1.4 \\ 0.6 & 0.5 & 1.1 & 0.3 & 0.0 & -0.2 \\ 2.0 & 2.9 & 1.3 & -0.1 & 2.6 & 2.9 \\ 2.6 & 0.7 & 1.2 & 0.2 & 0.7 & 0.5 \end{bmatrix}$	$\begin{bmatrix} 0.3 & 0.2 & 0.5 & 0.2 & 0.7 & -0.9 \\ 0.1 & 0.4 & 1.2 & 1.5 & 0.0 & 0.6 \\ 2.2 & 1.9 & 0.1 & 1.6 & 0.2 & 2.3 \\ 2.1 & 2.0 & 1.8 & 2.5 & 1.3 & 2.7 \\ 1.6 & 2.0 & -0.1 & 0.8 & 0.0 & 1.5 \end{bmatrix}$

$\begin{bmatrix} 2.2 & 1.0 & -0.2 & 0.5 & 1.5 & 2.0 \\ 1.5 & 0.5 & 0.5 & 0.0 & 1.5 & 0.0 \\ 2.4 & 0.5 & 1.1 & 0.3 & 0.0 & -0.2 \\ 2.8 & 2.9 & 1.1 & 3.0 & 0.9 & 0.3 \\ 0.0 & 0.5 & 1.2 & 0.2 & 0.7 & 0.5 \end{bmatrix}$	$\begin{bmatrix} 2.9 & 0.2 & 0.5 & 0.2 & 0.7 & -0.9 \\ 0.1 & 1.7 & 0.4 & 1.3 & 0.0 & 0.6 \\ 2.2 & 1.9 & 0.1 & 1.6 & 0.2 & 2.3 \\ 2.1 & 2.6 & -0.1 & 0.3 & 1.3 & 2.7 \\ 1.9 & 1.8 & 0.2 & 0.8 & 0.0 & 2.1 \end{bmatrix}$
$\begin{bmatrix} 2.9 & 2.1 & 2.0 & 1.9 & 1.1 & 1.6 \\ 0.3 & 2.1 & 0.1 & -0.1 & 0.2 & 0.7 \\ 3.0 & 0.4 & -0.5 & 0.7 & 0.0 & 0.5 \\ 0.4 & 2.0 & 2.9 & 0.1 & 2.7 & 0.3 \\ 0.6 & 0.8 & -0.1 & 0.5 & 0.7 & -0.5 \end{bmatrix}$	$\begin{bmatrix} 0.6 & 0.4 & 0.5 & 1.0 & 0.0 & 0.1 \\ 0.5 & -0.1 & 0.5 & 0.8 & 0.0 & 1.4 \\ 2.4 & 1.7 & 0.1 & 0.9 & 0.5 & 2.5 \\ -0.1 & 1.9 & 1.2 & 2.3 & 0.3 & 2.5 \\ 2.1 & 2.5 & 0.9 & 0.5 & 0.0 & 2.6 \end{bmatrix}$
$\begin{bmatrix} 2.0 & 2.5 & 0.6 & 0.1 & 0.0 & 1.2 \\ 1.2 & 0.2 & 1.1 & 2.2 & 0.5 & 0.5 \\ 0.5 & 1.8 & 0.3 & 2.2 & 0.0 & 0.5 \\ 2.5 & 0.3 & 2.4 & 1.3 & 0.1 & 2.9 \\ 1.8 & 1.4 & 0.4 & 1.1 & 0.5 & 0.0 \end{bmatrix}$	$\begin{bmatrix} 2.6 & 1.2 & 0.0 & 0.5 & 0.2 & 1.0 \\ 1.9 & 1.4 & 0.5 & -0.1 & 0.5 & 1.2 \\ 2.3 & 2.1 & 0.0 & 1.6 & 0.1 & 2.9 \\ 1.9 & 2.2 & 0.0 & 1.8 & 2.0 & 2.9 \\ 0.5 & 0.4 & 2.3 & 0.1 & 0.2 & 1.1 \end{bmatrix}$
$\begin{bmatrix} 2.1 & 1.3 & 0.1 & 1.2 & 0.0 & 1.8 \\ 1.8 & 2.3 & 0.0 & 2.4 & 0.1 & 1.5 \\ 0.0 & -1.0 & 0.1 & 0.2 & 1.0 & 1.2 \\ 2.9 & 2.2 & 0.0 & 0.8 & 1.5 & 1.8 \\ 2.6 & 0.1 & 0.0 & 1.9 & 0.2 & 1.5 \end{bmatrix}$	$\begin{bmatrix} 0.8 & 1.5 & 0.0 & 1.5 & 0.1 & 0.5 \\ 2.4 & 0.8 & 0.2 & -0.3 & 0.0 & 0.1 \\ -0.2 & 0.4 & 1.5 & 1.6 & 1.4 & 2.3 \\ 1.0 & 2.4 & 0.0 & 0.4 & 0.8 & 1.8 \\ 0.2 & 1.0 & 0.0 & 2.4 & 0.1 & -0.1 \end{bmatrix}$

References

1. Villegas-Ch, W.; García-Ortiz, J.; Sánchez-Viteri, S. Identification of the factors that influence university learning with low-code/no-code artificial intelligence techniques. *Electronics* **2021**, *10*, 1192. [CrossRef]
2. Chen, Y.; Wu, X.; Liao, H.; Kou, G. Consumer preference disaggregation based on online reviews to support new energy automobile purchase decision. *Procedia Comput. Sci.* **2023**, *221*, 1485–1492. [CrossRef]
3. Cheng, Y.; Zhang, C.; Sangaiah, A.K.; Fan, X.; Wang, A.; Wang, L.; Liu, Y. Efficient low-resource medical information processing based on semantic analysis and granular computing. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2023**. [CrossRef]
4. Hua, Z.; Jing, X.; Martínez, L. Consensus reaching for social network group decision making with ELICIT information: A perspective from the complex network. *Inf. Sci.* **2023**, *627*, 71–96. [CrossRef]
5. Arebi, P.; Fatemi, A.; Ramezani, R. Event stream controllability on event-based complex networks. *Expert Syst. Appl.* **2023**, *213*, 118886. [CrossRef]
6. Tong, X.; Liu, X.; Tan, X.; Li, X.; Jiang, J.; Xiong, Z.; Xu, T.; Jiang, H.; Qiao, N.; Zheng, M. Generative models for de novo drug design. *J. Med. Chem.* **2021**, *64*, 14011–14027. [CrossRef] [PubMed]
7. Liu, P.; Zhu, B.; Wang, P. A weighting model based on best–worst method and its application for environmental performance evaluation. *Appl. Soft Comput.* **2021**, *103*, 107168. [CrossRef]
8. Li, D.; Hu, S. Adaptive consensus reaching process with dynamic weights and minimum adjustments for group interactive portfolio optimization. *Comput. Ind. Eng.* **2023**, *183*, 109491. [CrossRef]
9. Glimcher, P.W.; Tymula, A.A. Expected subjective value theory (ESVT): A representation of decision under risk and certainty. *J. Econ. Behav. Organ.* **2023**, *207*, 110–128. [CrossRef]
10. Kahneman, D. Prospect theory: An analysis of decisions under risk. *Econometrica* **1979**, *47*, 278. [CrossRef]
11. Bell, D.E. Regret in decision making under uncertainty. *Oper. Res.* **1982**, *30*, 961–981. [CrossRef]
12. Wu, X.; Nie, S.; Liao, H.; Gupta, P. A large-scale group decision making method with a consensus reaching process under cognitive linguistic environment. *Int. Trans. Oper. Res.* **2023**, *30*, 1340–1365. [CrossRef]
13. Sáenz-Royo, C.; Chiclana, F.; Herrera-Viedma, E. Steering committee management. Expertise, diversity, and decision-making structures. *Inf. Fusion* **2023**, *99*, 101888. [CrossRef]
14. Zhang, C.; Zhang, J.; Li, W.; Castillo, O.; Zhang, J. Exploring static rebalancing strategies for dockless bicycle sharing systems based on multi-granularity behavioral decision-making. *Int. J. Cogn. Comput. Eng.* **2024**, *5*, 27–43. [CrossRef]
15. Liang, X.; Liu, P.; Wang, Z. Hotel selection utilizing online reviews: A novel decision support model based on sentiment analysis and DL-VIKOR method. *Technol. Econ. Dev. Econ.* **2019**, *25*, 1139–1161. [CrossRef]
16. Wu, Z.; Xu, J. A consensus model for large-scale group decision making with hesitant fuzzy information and changeable clusters. *Inf. Fusion* **2018**, *41*, 217–231. [CrossRef]
17. Zhao, Y.; Karypis, G.; Fayyad, U. Hierarchical clustering algorithms for document datasets. *Data Min. Knowl. Discov.* **2005**, *10*, 141–168. [CrossRef]
18. Xu, X.H.; Du, Z.J.; Chen, X.H.; Cai, C.G. Confidence consensus-based model for large-scale group decision making: A novel approach to managing non-cooperative behaviors. *Inf. Sci.* **2019**, *477*, 410–427. [CrossRef]
19. Belkhir, L.; Elens, L.; Zech, F.; Panin, N.; Vincent, A.; Yombi, J.C.; Vandercam, B.; Haufroid, V. Interaction between darunavir and etravirine is partly mediated by CYP3A5 polymorphism. *PLoS ONE* **2016**, *11*, e0165631. [CrossRef]

20. Wu, T.; Liu, X.; Liu, F. An interval type-2 fuzzy TOPSIS model for large-scale group decision making problems with social network information. *Inf. Sci.* **2018**, *432*, 392–410. [CrossRef]
21. Xu, Y.; Wen, X.; Zhang, W. A two-stage consensus method for large-scale multi-attribute group decision making with an application to earthquake shelter selection. *Comput. Ind. Eng.* **2018**, *116*, 113–129. [CrossRef]
22. Tang, M.; Liao, H.; Xu, J.; Streimikiene, D.; Zheng, X. Adaptive consensus reaching process with hybrid strategies for large-scale group decision making. *Eur. J. Oper. Res.* **2020**, *282*, 957–971. [CrossRef]
23. Palomares, I.; Martinez, L.; Herrera, F. A consensus model to detect and manage noncooperative behaviors in large-scale group decision making. *IEEE Trans. Fuzzy Syst.* **2013**, *22*, 516–530. [CrossRef]
24. Liu, B.; Shen, Y.; Chen, X.; Chen, Y.; Wang, X. A partial binary tree DEA-DA cyclic classification model for decision makers in complex multi-attribute large-group interval-valued intuitionistic fuzzy decision-making problems. *Inf. Fusion* **2014**, *18*, 119–130. [CrossRef]
25. Yu, W.; Zhang, Z.; Zhong, Q. Consensus reaching for MAGDM with multi-granular hesitant fuzzy linguistic Term sets: A minimum adjustment-based approach. *Ann. Oper. Res.* **2021**, *300*, 443–466. [CrossRef]
26. Li, Z.; Zhang, Z.; Yu, W. Consensus reaching for ordinal classification-based group decision making with heterogeneous preference information. *J. Oper. Res. Soc.* **2024**, *75*, 224–245. [CrossRef]
27. Yuan, R.; Wu, Z.; Tu, J. Large-scale group decision-making with incomplete fuzzy preference relations: The Perspective of Ordinal Consistency. *Fuzzy Sets Syst.* **2023**, *454*, 100–124. [CrossRef]
28. Zhang, C.; Wang, B.; Li, W.; Li, D. Incorporating artificial intelligence in detecting crop diseases: Agricultural decision-making based on group consensus model with MULTIMOORA and evidence theory. *Crop Prot.* **2024**, *179*, 106632. [CrossRef]
29. Kahneman, D.; Tversky, A. Prospect theory: An analysis of decision under risk. In *Handbook of the Fundamentals of Financial Decision Making: Part I*; World Scientific: Singapore, 2013; pp. 99–127.
30. Jin, F.; Liu, J.; Zhou, L.; Martínez, L. Consensus-based linguistic distribution large-scale group decision making using statistical inference and regret theory. *Group Decis. Negot.* **2021**, *30*, 813–845. [CrossRef]
31. Wang, J.; Ma, X.; Xu, Z.; Zhan, J. Regret theory-based three-way decision model in hesitant fuzzy environments and its application to medical decision. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 5361–5375. [CrossRef]
32. Tian, X.; Xu, Z.; Gu, J.; Herrera, F. A consensus process based on regret theory with probabilistic linguistic term sets and its application in venture capital. *Inf. Sci.* **2021**, *562*, 347–369. [CrossRef]
33. Jin, F.; Cai, Y.; Zhou, L.; Ding, T. Regret-rejoice two-stage multiplicative DEA models-driven cross-efficiency evaluation with probabilistic linguistic information. *Omega* **2023**, *117*, 102839. [CrossRef]
34. Alslaity, A.; Orji, R. Machine learning techniques for emotion detection and sentiment analysis: Current state, challenges, and future directions. *Behav. Inf. Technol.* **2024**, *43*, 139–164. [CrossRef]
35. Hajek, P.; Munk, M. Speech emotion recognition and text sentiment analysis for financial distress prediction. *Neural Comput. Appl.* **2023**, *35*, 21463–21477. [CrossRef]
36. Su, Y.; Kabala, Z.J. Public perception of ChatGPT and transfer learning for tweets sentiment analysis using wolfram mathematica. *Data* **2023**, *8*, 180. [CrossRef]
37. Liao, Y.; Hua, J.; Luo, L.; Ping, W.; Lu, X.; Zhong, Y. APRCOIE: An open information extraction system for Chinese. *SoftwareX* **2024**, *26*, 101649. [CrossRef]
38. Li, D.; Hu, S. Adaptive large-scale group interactive portfolio optimization approach based on social network with multi-clustering analysis and minimum adjustment. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108403. [CrossRef]
39. Brooks, S.J.; Jones, V.O.; Wang, H.; Deng, C.; Golding, S.G.H.; Lim, J.; Gao, J.; Daoutidis, P.; Stamoulis, C. Community detection in the human connectome: Method types, differences and their impact on inference. *Hum. Brain Mapp.* **2024**, *45*, e26669. [CrossRef]
40. Wang, L.; Zhang, Y.; Feng, J. On the euclidean distance of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1334–1339. [CrossRef]
41. Wu, P.; Li, F.; Zhao, J.; Zhou, L.; Martínez, L. Consensus reaching process with multiobjective optimization for large-scale group decision making with cooperative game. *IEEE Trans. Fuzzy Syst.* **2022**, *31*, 293–306. [CrossRef]
42. Yao, S.; Gu, M. An influence network-based consensus model for large-scale group decision making with linguistic information. *Int. J. Comput. Intell. Syst.* **2022**, *15*, 1–17. [CrossRef]
43. Fanchiang, K.H.; Huang, Y.C.; Kuo, C.C. Power electric transformer fault diagnosis based on infrared thermal images using Wasserstein generative adversarial networks and deep learning classifier. *Electronics* **2021**, *10*, 1161. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

An Architecture as an Alternative to Gradient Boosted Decision Trees for Multiple Machine Learning Tasks

Lei Du †, Haifeng Song, Yingying Xu and Songsong Dai *,†

School of Electronics and Information Engineering, Taizhou University, Taizhou 318000, China; dulei2109@tzc.edu.cn (L.D.); isshf@tzc.edu.cn (H.S.); yyxu@tzc.edu.cn (Y.X.)

* Correspondence: ssdai@tzc.edu.cn

† These authors contributed equally to this work.

Abstract: Deep networks-based models have achieved excellent performances in various applications for extracting discriminative feature representations by convolutional neural networks (CNN) or recurrent neural networks (RNN). However, CNN or RNN may not work when handling data without temporal/spatial structures. Therefore, finding a new technique to extract features instead of CNN or RNN is a necessity. Gradient Boosted Decision Trees (GBDT) can select the features with the largest information gain when building trees. In this paper, we propose an architecture based on the ensemble of decision trees and neural network (NN) for multiple machine learning tasks, e.g., classification, regression, and ranking. It can be regarded as an extension of the widely used deep-networks-based model, in which we use GBDT instead of CNN or RNN. This architecture consists of two main parts: (1) the decision forest layers, which focus on learning features from the input data, (2) the fully connected layers, which focus on distilling knowledge from the decision forest layers. Powered by these two parts, the proposed model could handle data without temporal/spatial structures. This model can be efficiently trained by stochastic gradient descent via back-propagation. The empirical evaluation results of different machine learning tasks demonstrate the the effectiveness of the proposed method.

Keywords: Gradient Boosted Decision Trees; neural network; decision forest layer; back-propagation

Citation: Du, L.; Song, H.; Xu, Y.; Dai, S. An Architecture as an Alternative to Gradient Boosted Decision Trees for Multiple Machine Learning Tasks. *Electronics* **2024**, *13*, 2291. https://doi.org/10.3390/electronics13122291

Academic Editors: Ping-Feng Pai and Chunping Li

Received: 20 April 2024
Revised: 25 May 2024
Accepted: 8 June 2024
Published: 12 June 2024

1. Introduction

In the past few years, we have witnessed dramatic progress in the development of deep neural networks. These models have achieved excellent performance in various applications, e.g., object recognition in images [1–5], speech recognition [6,7], and natural language processing [8–10]. One of the reasons that deep networks-based approaches (e.g., CNN [1,2,11], RNN [6,8,12]) succeed in these applications is that they are effective in extracting discriminative feature representations from data with spatial/temporal structures. Li et al. [3] proposed a deep architecture reversible autoencoder for image reconstruction, which integrated operators based on CNN. To refine the noise and speech, Lu et al. [7] proposed a dual-stream spectrogram refine network (DSRNet), which was a deep model.

The attractive feature of CNN is its ability to exploit spatial or temporal correlation in data [2,11], and it has turned out to be very good at learning intricate structures in high-dimensional data [12,13]. The RNN unit holds a hidden state which empowers it to process sequential data [14]. However, in many machine learning tasks, each of the data samples is represented by a vector with effective hand-crafted features, where this vector does not have any temporal/spatial structures [6,11,15]. For example, as shown in Table 1, each instance of the Ecoli dataset (http://archive.ics.uci.edu/dataset/39/ecoli, accessed on 15 January 2024) is a vector, each element of which is a feature extracted by different methods. The dimension of each vector is one and the vector has no temporal structure, so CNN or RNN cannot extract discriminative feature representations from such kinds of data. Therefore, creating an effective model for data without temporal/spatial structures has become a necessity.

Representative approaches for these kinds of data are tree ensemble methods and NN. The tree ensemble methods (e.g., random forest [16,17], GBDT [18–20]) have achieved excellent performance in different tasks, e.g., learning-to-rank (LambdaMART [21]) and binary classification (AdaBoost [22]). GBDT [18] is a regression model which consists of a number of regression trees. It automatically selects the features with largest statistical information gain and combines the selected features to fit the training targets well when building trees, which allows it to effectively handle datasets with dense numerical features. The difference between GBDT and random forest is that the trees in GBDT are trained sequentially, while trees in random forest are trained independently. The first weakness of GBDT is that the learned trees are not differentiable, which prevents these methods from learning over large scale data. Another weakness of GBDT is its ineffectiveness when learning datasets with sparse categorical features [23]. The information gain becomes small when converting the sparse features into high-dimensional one-hot encodings. Although there are some encoding methods which convert the sparse categorical features into dense features directly, these methods will hurt the raw information as it is hard to distinguish the encode values of different categories [23]. The two weaknesses cause GBDT to fail in many machine learning tasks.

Table 1. The first 5 lines of the Ecoli dataset. Each line represents an instance and does not have any temporal/spatial structures.

Mcg	Gvh	Lip	Chg	Aac	Alm1	Alm2
0.29	0.30	0.48	0.50	0.45	0.03	0.17
0.22	0.36	0.48	0.50	0.35	0.39	0.47
0.23	0.58	0.48	0.50	0.37	0.53	0.59
0.47	0.47	0.48	0.50	0.22	0.16	0.26
0.54	0.47	0.48	0.50	0.28	0.33	0.42

NN [24] is a feedforward neural network, which makes up of a number of interconnected processing elements and processes information based on their dynamic state response to external inputs. A number of studies have revealed the success of using NN in different machine learning tasks, e.g., recommender systems [25] and online prediction [23]. The reason why NN is unique is that it can make models more accurate when handling complex problems [23]. Therefore, the advantages of NN are efficient learning over large-scale data and capability in learning over sparse categorical features by embedding structure. Nevertheless, when learning dense numerical features, NN cannot outperform tree ensemble methods. Although a fully connected neural network can directly handle the datasets with dense numerical features, the performance may be unsatisfactory because the structure makes it susceptible to falling into local optimums when solving complex problems. This is the reason why NN cannot outperform GBDT when handling datasets with dense numerical features [26].

In this paper, we take a step forward to extend GBDT to a deep model. Specifically, we propose a deep architecture that combines GBDT with the multi-layer perceptron network [24]. As shown in Figure 1, from bottom to top, the proposed architecture has four parts: (1) The input data layer in which each data sample is represented by a vector; (2) The decision forest layer that consists of multiple hidden nodes. Each node is fitted by an ensemble of decision trees. This layer focuses on learning features from the input data. (3) Multiple fully connected layers, which focus on distilling knowledge from the decision forest layers. (4) The loss layer. For each node in the decision forest layer, we develop a GBDT-like method to generate a weighted sum of regression trees. Each tree is an estimated approximate gradient for an iteration, making the decision forest layer to compatible with back-propagation. Hence, the proposed deep architecture can be efficiently trained by stochastic gradient descent via back-propagation, which allows the proposed method to outperform tree ensemble methods. Compared with NN and the deep models based on convolutional networks or recurrent networks, the proposed method can select the useful numerical features when handling data without temporal/spatial structures. The learning task of DeepGBM is online prediction, which has to adapt the learning model to the online data generation, while it is not necessary to consider the new arrival data for the proposed method. There are a lot of models dedicated to solving concrete tasks, while our method is used for multiple tasks.

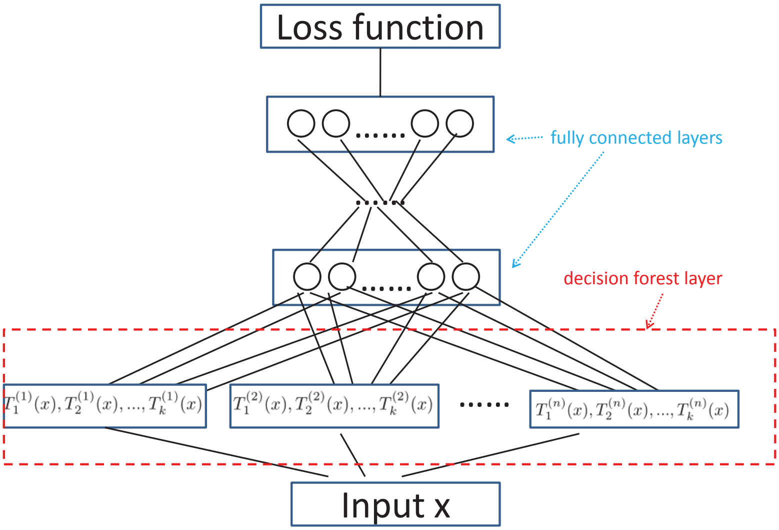

Loss function
fully connected layers
decision forest layer
$T_1^{(1)}(x), T_2^{(1)}(x), ..., T_k^{(1)}(x)$
$T_1^{(2)}(x), T_2^{(2)}(x), ..., T_k^{(2)}(x)$
$T_1^{(n)}(x), T_2^{(n)}(x), ..., T_k^{(n)}(x)$
Input x

Figure 1. Overview of the proposed architecture. From bottom to top, this architecture consists of four parts: (1) the input data layer; (2) the decision forest layer, which focuses on learning features from the input data; (3) multiple fully connected layers, which focuses on distilling knowledge from the decision forest layers; (4) the loss layer.

We conduct several experiments using eight datasets for multiple machine learning tasks, where four are used for a classification task, two are used for a regression task, and two are used for a ranking task. Comprehensive experimental results verified the effectiveness of the proposed architecture, e.g., it shows 2.22%, 2.16% and 5.60% increase against the corresponding second-best baseline on the Protein, Seismic and Isolet datasets, respectively. In summary, the fundamental contributions of this paper are listed as follows: (1) We propose an architecture based on GBDT and NN for multiple machine learning tasks, which could handle data without temporal/spatial structures. (2) We present a back-propagation procedure which will allow us to update the parameters by stochastic gradient descent. (3) We conduct empirical evaluations on several datasets for classification, regression, and ranking. The experimental results show that the proposed method can outperform the other methods.

The rest of this paper is organized as follows. We present the related work in Section 2 and present the different parts of the proposed architecture in Section 3. In Section 4, we present a back-propagation procedure to update the parameters by stochastic gradient descent for different machine learning tasks. All the experimental results are shown in Section 5. In Section 6, we present the complexity of the proposed architecture. Finally, we conclude our work in the Section 7.

2. Related Work

Random forest [16,17] is a widely used procedure which generates a number of regression or classification trees. Each tree is constructed by randomly selecting subsets of features. The results from the trees are aggregated to provide a prediction for each data, which leads to random forest providing higher accuracy in contrast with a single decision tree. Boosting is a successful technique for different machine learning problems, which was first proposed by Freund et al. in the form of the AdaBoost method [22]. This technique has been widely used in data analysis and real-world applications. AdaBoost uses decision trees as weak learners and integrates the decision trees into a strong classifier with the boosting technique. Recently, different machine learning systems have been proposed to train GBDT, e.g., XGBoost [26], LightGBM [27] and DimBoost [28].

It is worth noting that different models have been proposed [23,29,30] by combining tree ensemble methods and NN. Sethi [31] shows how to map a decision tree into a multi-layer perceptron network structure. Scornet et al. [29] show that each regression tree can be regarded as a particular multi-layer perceptron network and the random forest can be reformulated into a multi-layer perceptron network by restructuring several randomized regression trees as a collection of the multi-layer perceptron networks with particular connection weights. They first learn a random forest, and then extracted all the split directions and split positions to build the multi-layer perceptron network initialization parameters. Wang et al. [30] propose a novel model of decision-tree like the multi-layer perceptron network (NNRF), which has similar properties as a decision tree. Similar with random forest, NNRF has one path activated for each input, which leads it is efficient when performing forward and backward propagation. This technique also makes it easy to handle small datasets as the number of parameters is relatively small. NNRF learns complex functions when choosing the relevant paths of each node, which leads it outperforming random forest. Integrating GBDT and NN, Ke et al. [23] propose a learning framework DeepGBM for online prediction tasks. The framework consists of two parts: CatNN focuses on sparse categorical features and GBDT2NN focuses on handling dense numerical features. Powered by the above parts, this model has a strong learning capacity over numerical tabular features and categorical features while still having the ability to achieve efficient online learning.

3. Methodology

3.1. GBDT Algorithm

In this paper, we take a step forward by extending GBDT to a deep model. It is necessary to give the specific steps of GBDT. Given a training set $\{(x_j, y_j)\}_{m=1}^{M}$, x_m is a vector which represents the m_{th} data point, y_m is the predicted label. Each step is indicated as follows [20]:

Step 1: Compute the initial constant value γ by

$$F_0(x) = argmin_{\gamma} \sum_{i=1}^{m} L(y_i, \gamma), \tag{1}$$

where $L(y_i, \gamma)$ represents the loss function.

Step 2: The residual along the gradient direction is defined as

$$\hat{y_i} = -[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}]_{f(x)=f_{n-1}(x)}, \tag{2}$$

where n represents the number of iterations.

Step 3: By fitting the data into the initial model $T(x_i)$, we can construct a tree, where the parameter α_n of the tree is based on the following expression as

$$\alpha_n = argmin_{\alpha,\beta} \sum_{i=1}^{m} (\hat{y_i} - \beta T(x_i))^2. \tag{3}$$

Step 4: After minimizing the following loss function, we obtain the weight of the current model

$$\gamma_n = argmin_\gamma \sum_{i=1}^{m} L(y_i, F_{n-1}(x) + \gamma T(x_i)). \tag{4}$$

Step 5: The new model is updated as

$$F_n(x) = argmin_\gamma \sum_{i=1}^{m} L(y_i, F_{n-1}(x) + \gamma T(x_i)). \tag{5}$$

The residual of the former decision tree is the input for the next decision tree, which can reduce the residual, and the loss decreases following the negative gradient direction in each iteration. After n iterations, we will learn a model which consists of n decision trees. The final result is determined by the sum of results from all the decision trees.

3.2. Proposed Architecture

The proposed architecture is a deep neural network which consists of an input layer, a decision forest layer, multiple fully connected layers, and a loss layer. In this section, we will present these parts, respectively.

3.2.1. Input Layer

The motivation of the proposed architecture is to tackle the machine learning tasks whose data samples are represented by vectors with effective hand-crafted features, where each vector does not have any temporal/spatial structures. Hence, the input layer accepts data points in the form of vectors. We denote L^{in} as the input layer.

3.2.2. Decision Forest Layer

The input layer is a decision forest layer, which is the crucial part of the proposed architecture and focuses on learning features from the input data. This layer consists of n hidden nodes, where each node is an ensemble of k decision trees. Specifically, we denote L^f as the decision forest layer and a^f as the output vector of the decision forest layer. For an input point x, the i-th ($i = 1, 2, \ldots, n$) element in a^f (i.e., the i-th hidden node in L^f) can be calculated by

$$a_i^f = g_i(x) = r_i + s \sum_{j=1}^{k} T_j^{(i)}(x), \tag{6}$$

where r_i is the initial value of a_i^f, $T_j^{(i)}$ ($j = 1, 2, \ldots, k$) represents a regression tree, and s is a pre-defined shrinkage coefficient. The structure and parameters in each regression tree can by determined during the back-propagation process (see Section 4 for details).

3.2.3. Fully Connected Layers

On top of the decision forest layer, we construct multiple fully connected layers which are widely used in neural networks. The input of layer is the output of the decision forest layer. We denote $L_{d_1}, L_{d_2}, \ldots, L_{d_m}$ as the fully connected layers on top of the decision forest layer. Suppose z^{d_i}, a^{d_i} are the input and output (after activation) of L_{d_i} ($i = 1, 2, \ldots, m$). Then, we have the following:

$$\begin{aligned} z^{d_1} &= W^f a^f, \\ a^{d_i} &= f(z^{d_i}) \ \ (i = 1, 2, \ldots, m), \\ z^{d_{i+1}} &= W^{d_i} a^{d_i} + b^{d_i} \ \ (i = 1, 2, \ldots, m-1), \end{aligned} \tag{7}$$

where function $f()$ represents the activation function and the matrix W^{d_i} and the vector b^{d_i} are the parameters in the i-th fully connected layer.

3.2.4. Loss Layer

On top of the last fully connected layer L_{d_m}, we can define a loss layer for a specific machine learning task.

For classification, the output is $a^{d_m} = h(x) = (h_1(x), h_2(x), h_3(x), \ldots h_K(x))$, where K is the number of classes. Follow softmax [32], loss layer is defined by [33]:

$$J(h(x), y) = -\sum_{k=1}^{K} I(y = k) \log p(y = k|x), \tag{8}$$

where I is an indicator function and

$$p(y = k|x) = \frac{e^{h_k(x)}}{\sum_{j=1} e^{h_j(x)}}. \tag{9}$$

The gradient is as follows:

$$\delta^{d_m} = \frac{\partial}{\partial z^{d_n}} J(h(x), y) = p(y = k|x) - I(y = k). \tag{10}$$

For regression, square loss is used to define the loss layer:

$$J(a^{d_m}, y) = \frac{1}{2} ||a^{d_m} - y||^2. \tag{11}$$

The gradient is as follows:

$$\delta^{d_m} = \frac{\partial}{\partial z^{d_n}} J(h(x), y) = z^{d_n} - y. \tag{12}$$

For ranking, we use LambdaRank [21] to define the loss layer, which was based on RankNet [34] and directly optimizes Normalized Discounted Cumulative Gain (NDCG) [35]. NDCG is an evaluation metric and will be introduced later. Given a query set $x_1, x_2 \ldots, x_n$, the output is $s_1, s_2 \ldots, s_n$. The cost function of RankNet is as follows:

$$J(s_i, s_j) = \sum_{R_i > R_j} \log(1 + e^{-\sigma(s_i - s_j)}), \tag{13}$$

where σ is a parameter that determines the shape of the sigmoid and $R_i > R_j$ represents document x_i, which ranks higher than x_j. The gradient is

$$\frac{\partial J}{\partial s_i} = \lambda_i = \sum_{R_i > R_j} \lambda_{ij} - \sum_{R_j > R_i} \lambda_{ij}, \tag{14}$$

where

$$\lambda_{ij} = \frac{\partial J(s_i - s_j)}{\partial s_i} = \sigma(I(R_i > R_j) - \frac{1}{1 + e^{-\sigma(s_i - s_j)}}). \tag{15}$$

The LambdaRank's key observation is that the costs themselves are unnecessary, except for the gradients. Following LambdaRank, we have

$$\begin{aligned} \lambda_{ij} &= \frac{\partial J(s_i - s_j)}{\partial s_i} \\ &= \sigma(I(R_i > R_j) - \frac{1}{1 + e^{-\sigma(s_i - s_j)}}) \mid \Delta NDCG \mid, \end{aligned} \tag{16}$$

where $\mid \Delta NDCG \mid$ is the size of change in metric NDCG by swapping the data point x_i and x_j. The gradient is as follows:

$$\begin{aligned}\delta^{d_m} &= \sum_{R_i > R_j} \lambda_{ij} - \sum_{R_j > R_i} \lambda_{ij} \\ &= \sum_{R_i > R_j} \frac{-\sigma \mid \Delta NDCG \mid}{1 + e^{-\sigma(s_i - s_j)}} - \sum_{R_j > R_i} \frac{-\sigma \mid \Delta NDCG \mid}{1 + e^{-\sigma(s_i - s_j)}}.\end{aligned} \tag{17}$$

4. Optimization

Training the proposed architecture requires trying to find a solution to the parameters $(\{T_i^{(j)}\}_{i=1,2,\dots,k}^{j=1,2,\dots,n}, \{W_l, b_l\}_{l=1,2,\dots,m})$ that minimizes the loss function. As shown in Algorithm 1, we present a back-propagation procedure to update these parameters by stochastic gradient descent.

Algorithm 1 The back-propagation procedure for the proposed architecture

```
Input: training set $\{(x_j, y_j)\}_{j=1}^{M}$, maximal iteration number $k$, number
$n$ of hidden nodes in the decision forest layer, the shrinkage $s$.
Output: $\{T_i^{(j)}\}_{i=1,2,\dots,k}^{j=1,2,\dots,n}$, $\{W_l, b_l\}_{l=1,2,\dots,m}$
1. Initialize: $p \leftarrow 0$ and randomly initialize $r_i$ $(i = 1, 2, \dots, n)$,
$\{W_l, b_l\}_{l=1,2,\dots,m}$ by Gaussian distribution.
2. Repeat:
3.    Calculate $a^f$, $z^{d_i}$, $a^{d_i}$, $J$ $(i = 1, 2, \dots, m)$ by Equations (6)–(8) via
forward propagation.
4.    Update $\{W_l, b_l\}_{l=1,2,\dots,m}$ by Equation (20).
5.    For $i = 1, 2, \dots, n$, add a new tree $T_p^{(i)}$ to the ensemble of the $i$-th
hidden node in the decision forest layer by Equation (22).
6.    $p \leftarrow p + 1$.
7. Until $p = T$.
```

Since the fully connected layer is an easy studied component in neural networks, the parameters $\{W_l, b_l\}_{l=1,2,\dots,m}$ in the fully connected layers can be updated by stochastic gradient descent via back-propagation. Specifically, suppose

$$\delta^i = \frac{\partial J}{\partial z^{d_i}} \quad (i = 1, 2, \dots, m), \tag{18}$$

then we have:

$$\begin{aligned}\delta^{d_i} &= ((W^{d_i})^T \delta^{d_{i+1}}) \odot f'(z^{d_i}), \\ \nabla_{W^{d_i}} &= \delta^{d_{i+1}} (a^{d_i})^T, \\ \nabla_{b^{d_i}} &= \delta^{d_{i+1}},\end{aligned} \tag{19}$$

and

$$\begin{aligned}W^{d_i} &\leftarrow W^{d_i} - \alpha \nabla_{W^{d_i}}, \\ b^{d_i} &\leftarrow b^{d_i} - \alpha \nabla_{b^{d_i}},\end{aligned} \tag{20}$$

where $\odot$ represents the element-wise multiplication for matrices and α is the learning rate.

In the decision forest layer, the regression trees in each of the n hidden nodes are constructed one by one. For each hidden node, we add one tree in an iteration of back-propagation. Specifically, in the p-th iteration of back-propagation $(p = 1, 2, \dots, k)$, the negative gradient for the decision forest layer can be calculated as follows:

$$r^f = -\delta^f = (W^f)^T \delta^{d_1}. \tag{21}$$

We denote r_i^f as the i-th element in r^f $(i = 1, 2, \ldots, n)$. Then, r_i^f is the residual for the i-th hidden node in the decision forest layer. For a specific input x, we can obtain a residual $r_i^f(x)$. Then, for all of the training samples $x_1, x_2, \ldots, x_M$, we can obtain $\{(x_j, r_i^f(x_j))\}_{j=1}^M$. After that, following the third step of GBDT [20], we construct a new regression tree $T_p^{(i)}(.)$ by solving the problem $\min_{T_p^{(i)}} \frac{1}{2}\sum_{j=1}^M (T_p^{(i)}(x_j) - r_i^f(x_j))^2$. The ensemble of the tree in the i-th hidden node can be updated as follows:

$$g_i(x) \leftarrow g_i(x) + sT_p^{(i)}(x), \tag{22}$$

where s is the shrinkage coefficient.

5. Experiments

We evaluate the proposed method (The source code has been released at https://github.com/dllinhe2017/Dgbdt, accessed on 1 January 2024) using several datasets for different tasks in this section. For classification, we compare the proposed method against the three most closely related baselines:

- GBDT [18], which consists of an ensemble of regression trees. It automatically selects the features with largest statistical information gain and combines the selected features to fit the training targets well when building trees.
- MLP [24], which makes up a number of interconnected processing elements and processes information by their dynamic state response to external inputs. We report the experimental results of two different models: MLP_1 and MLP_2, where MLP_1 includes one hidden-layer and MLP_2 includes two hidden layers.
- NNRF [30], a novel model of decision-tree like the multi-layer perceptron network, which has similar properties to a decision tree (The author of NNRF only reported the experimental results for the classification task, so we neglected to select NNRF as the baseline method for the regression task). It has one path activated for each input.

For regression, we compare our method against two most closely related baselines: GBDT and MLP.

For ranking, we compare our method against the two most closely related baselines:

- Ranknet [34], which is a pair-based neural network method and learns the ranking functions by a probabilistic cost. Two sentences from the same document generated a pair and the label is determined by the scores of sentences.
- LambdaMART [21], which is based on Ranknet and directly optimizes NDCG. LambdaMART defines a weight parameter to represent the difference in NDCG when swapping a pair of documents. It is then used to update the weight in the next iteration.

5.1. Datasets

We evaluate the proposed methods and the baselines on four datasets for classification (https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass, accessed on 21 January 2023) and two for regression (http://archive.ics.uci.edu/, accessed on 22 January 2023), respectively. The statistics information of these datasets is shown in Table 2. In particular, Isolet is a dataset that was used to predict which letter was spoken. Gesture is a dataset that was used to study gesture phase segmentation. The features of the datasets are hand-crafted and the structures of the datasets are not temporal/spatial.

Table 2. Statistics information of the datasets for classification or regression.

Dataset	Instances	Features	Train/Test Split	Task
Protein	17,766	357	12,436/5330	Classification
Seismic	581,012	54	406,712/174,300	Classification
Isolet	7797	617	5458/2339	Classification
Gesture	9873	32	6911/2962	Classification
Slices	53,500	386	48,150/5350	Regression
YearPredictMSD	515,345	90	463,811/51,534	Regression

For ranking, we use MQ2007 and MQ2008 (https://www.microsoft.com/en-us/research/project/letor-learning-rank-information-retrieval/letor-4-0/, accessed on 24 January 2023) to evaluate the proposed methods and the baselines. The statistical information is shown in Table 3.

Table 3. Statistical information of the datasets for ranking.

Dataset	Number of Documents	Number of Queries	Features
MQ2007	6,962,598	1700	46
MQ2008	15,211	800	46

5.2. Evaluation Metrics

For classification, we use the accuracy as the evaluation metric. Specifically, for a test set $\{x_i, y_i\}_{i=1}^N$, let p_i be the predicted class label of x_i ($i = 1, 2, \ldots, N$), then let the accuracy rate be defined by

$$Accuracy = \frac{\sum_{i=1}^N I(y_i = p_i)}{N}, \tag{23}$$

where $I(condition)$ is an indicator function that $I(condition) = 0$ if the *condition* is false; otherwise, $I(condition) = 1$.

For regression, we use the Root of Mean Square Error (RMSE) as the evaluation metric. Specifically, for a test set $\{x_i, y_i\}_{i=1}^N$, let p_i be the predicted value of x_i ($i = 1, 2, \ldots, N$), then let the RMSE be defined by

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (y_i - p_i)^2}{N}}. \tag{24}$$

For ranking, we use NDCG and Mean Average Precision (MAP) [36,37] as the evaluation metrics. For a given set of search results, the Discounted Cumulative Gain (DCG) is

$$DCG@T = \sum_{i=1}^{T} \frac{2^{l_i} - 1}{log(1+i)}, \tag{25}$$

where T is a value representing the truncation level (usually less 10), and l_i is the label of the ith listed URL. And the NDCG is defined by

$$NDCG@T = \frac{DCG@T}{\max DCG@T}. \tag{26}$$

MAP is defined by

$$MAP = \frac{\sum_{q=1}^Q AveP(q)}{Q}, \tag{27}$$

where Q represents the number of queries,

$$AveP(q) = \frac{\sum_{k=1}^n (P(k) \times rel(k))}{NumD}, \tag{28}$$

where $P(k)$ is the list's precision at cut-off k. $rel(k)$ is also an indicator function. It is equal to 0 if the item at rank k is a irrelevant document, and is equal 1 otherwise, and $NumD$ is the number of relevant documents.

5.3. Experiment Settings

As shown in Figure 2, there are several parameters (the number of hidden nodes in forestnet is J_1, the number of trees is K, and the number of hidden nodes in the fully connected layer is J_2 in the proposed model. The sensitivity of the three parameters will be analyzed in detail in Section 5.5.

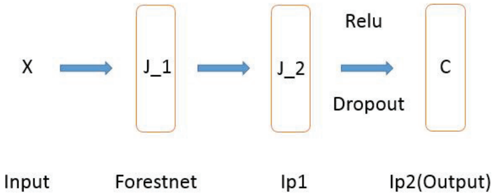


Figure 2. Structure of the proposed model for classification or regression tasks: (1) The decision forest layer (Forestnet) consists of J_1 hidden nodes. Each node has an ensemble of K. (2) The fully connected layer (Ip1) has J_2 hidden nodes. (3) The fully connected layer (Ip2) has C hidden nodes (for classification, C is the number of classes, and for regression, $C = 1$).

In the parameter tuning procedure for both the proposed and competitor methods, 10-fold cross validation is used. Note that no test data are involved in the parameter tuning. Given a training set, we divide it into 10 parts. One is chosen to be a validation set, while the remaining nine parts are used as the corresponding training set. The experiments are repeated 10 times and the 10 experiments are averaged to produce the best values of different parameters.

For all the classification tasks or the regression tasks, it is worth noting that the parameters J_1 and J_2 are chosen in the set $\{50, 100, 150, 200, 250, 300\}$ by cross-validation. The parameter K is chosen in the set $\{100, 200, 300, 400, 500\}$. The shrinkage rate is 0.01. We use ReLU as the activation function. The proposed model is trained by minibatch gradient descent, in which we set the learning rate to 0.01, the momentum to 0.9, and the weight decay to 0.0001.

There are several parameters to set in the baseline algorithms. To ensure a fair comparison with the proposed model, it is worth noting that the parameter K in GBDT is chosen in the set $\{100, 200, 300, 400, 500\}$ by cross-validation; the parameter J_1 in MLP_1 is chosen in the set $\{50, 100, 150, 200, 250, 300\}$; parameters J_1 and J_2 in MLP_2 are chosen in the set $\{50, 100, 150, 200, 250, 300\}$; the number of layers d in NNRF is chosen in the set $\{3, 4, 5\}$; the parameter J_1 is chosen in the set $\{50, 100, 150, 200, 250, 300\}$ by cross validation, the depth of regression trees is set to be 3.

For the ranking tasks, in the decision forest layer, the number of the hidden nodes is $J\_1 = 24$, each node has an ensemble of $K = 700$, and the shrinkage rate is 0.1. In the fully connected layer, the number of the hidden nodes is 12. Every fully connected layer uses maxout with a rate of 0.5. The proposed model is also trained by stochastic gradient descent, in which the learning rate is 0.01, the momentum is 0, and the weight decay is 0.05.

5.4. Results

Table 4 shows the comparison results for four datasets for classification tasks. We can observe that the proposed method shows superior performance gains over the baselines on the four datasets. Some statistics are listed below. For Gesture, the results of our method indicate a relative increase of 25.20% compared to the corresponding second-best baseline. The proposed method also shows 2.22%, 2.16% and 5.60% increase against the corresponding second-best baseline for the Protein, Seismic and Isolet datasets, respectively.

Table 5 shows the comparison results on two datasets for regression tasks. We can observe that the proposed method performs better than the baselines. Next, we will list some statistics. The proposed method shows 54.52% and 0.97% decreases against the corresponding second-best baseline on the Slices dataset and YearPredictMSD dataset, respectively.

Table 4. Comparison results with respect to the classification accuracy rate. On each dataset, 10 test runs were conducted and the average performance as well as the variance (numbers in parentheses) is reported.

Methods	**Protein**	**Seismic**	**Isolet**	**Gesture**
GBDT	0.702 (0.00078)	0.713 (0.00172)	0.813 (0.00067)	0.488 (0.00096)
MLP_1	0.803 (0.00175)	0.737 (0.00084)	0.596 (0.00093)	0.456 (0.00087)
MLP_2	0.811 (0.00162)	0.742 (0.00178)	0.616 (0.00128)	0.477 (0.00114)
NNRF	0.693 (0.00074)	0.711 (0.00076)	0.857 (0.00145)	0.462 (0.00098)
Ours	0.829 (0.00093)	0.768 (0.00069)	0.905 (0.00081)	0.611 (0.00078)

Table 5. Comparison Results with respect to RMSE. On each dataset, 10 test runs were conducted and the average performance as well as the variance (numbers in parentheses) is reported.

Methods	**Slices Dataset**	**YearPredictMSD Dataset**
GBDT	5.18 (0.00193)	9.38 (0.00487)
MLP_1	5.78 (0.00253)	9.33 (0.00284)
MLP_2	5.63 (0.00272)	9.31 (0.00354)
Ours	2.56 (0.00383)	9.22 (0.00283)

Tables 6 and 7 list the results for two datasets. For MQ2007, we can observe that our method outperforms RankNet and LambdaMART on all metrics. For MQ2008, we can also observe our method has superior performance gains over RankNet and LambdaMART on the metrics of NDCG@1 and mean NDCG, although it performs worse than LambdaMART on the metrics of NDCG@3 and MAP. The difference is negligible.

Table 6. Comparison results for MQ2007. Ten test runs were conducted and the average performance, as well as the variance (numbers in parentheses), is reported.

Methods	**NDCG@1**	**NDCG@3**	**Mean NDCG**	**MAP**
RankNet	0.3418 (0.00074)	0.3519 (0.00074)	0.4518 (0.00074)	0.4224 (0.00074)
LambdaMART	0.4137 (0.00074)	0.4157 (0.00074)	0.5035 (0.00074)	0.4684 (0.00074)
Ours	0.4157 (0.00074)	0.4178 (0.00074)	0.5061 (0.00074)	0.4712 (0.00074)

Table 7. Comparison Results on MQ2008. Ten test runs were conducted and the average performance, as well as the variance (numbers in parentheses), is reported.

Methods	**NDCG@1**	**NDCG@3**	**Mean NDCG**	**MAP**
RankNet	0.3400 (0.00104)	0.4000 (0.00058)	0.4599 (0.00062)	0.4515 (0.000118)
LambdaMART	0.3753 (0.00068)	0.4312 (0.00084)	0.4879 (0.00058)	0.4765 (0.00099)
Ours	0.3766 (0.00056)	0.4304 (0.00074)	0.4882 (0.00083)	0.4764 (0.00089)

5.5. Parameter Analysis

The most important parameters of the proposed framework are the number of hidden nodes *J*_1 in the decision forest layer, the number of trees *K* of each node, and the number of hidden nodes *J*_2 in the fully connected layer. We conduct experiments to investigate the effects on classification performances with different values of the three parameters. In the experiment, we use five-fold cross validation on the training data to set the parameters. Note that no test data are involved.

To investigate the effects of the number of hidden nodes J_1 in the decision forest layer, we first fix K and J_2 to be 100. We then vary the values of J_1 as $\{50, 100, 150, 200, 250, 300\}$. We report the results of the proposed method with different values of J_1 on Seismic and Protein. For comparison, we also provide the result of the best baseline. The results are shown in Figures 3 and 4. Based on the figures, we make the following observations: (i) The performance of the proposed method is fairly good when J_1 is from 50 to 300. (ii) From the point of view of trade-offs between training cost and performance, it seems like a reasonable point when setting J_1 to be a value within $[100, 200]$. (iii) The performance of the proposed method increases a little on Seismic and Protein when J_1 is set to be bigger than 100.

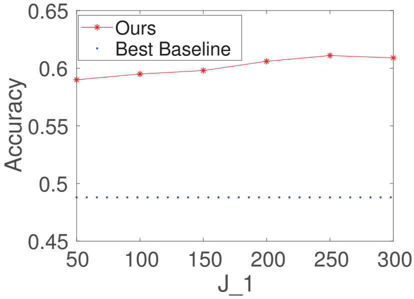


Figure 3. Results with different values of the number of hidden nodes J_1 in the decision forest layer on Gesture ($K = 100$, $J\_2 = 100$).

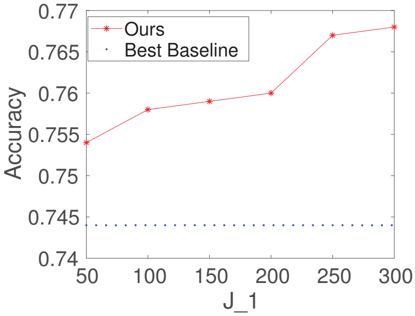


Figure 4. Results with different values of the number of hidden nodes J_1 in the decision forest layer on Seismic ($K = 100$, $J\_2 = 100$).

To investigate the effects of the number of hidden nodes J_2 in the fully connected layer, we first select K and J_1 to be 100. We then vary the values of J_2 as $\{50, 100, 150, 200, 250, 300\}$. We report the results of the proposed method with different values of J_2 for Seismic and Protein. For comparison, we also present the result of the best baseline. The results are shown in Figures 5 and 6. Based on the figures, we make the following observations: (i) the performance of the proposed method is fairly good when J_2 is from 50 to 300; (ii) from the point of view of trade-offs between training cost and performance, it seems like a reasonable point when setting J_1 to be 100.

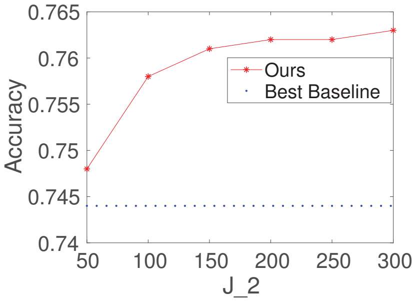


Figure 5. Results with different values of the number of hidden nodes J_2 in the fully connected layer on Seismic ($K = 100$, $J\_1 = 100$).

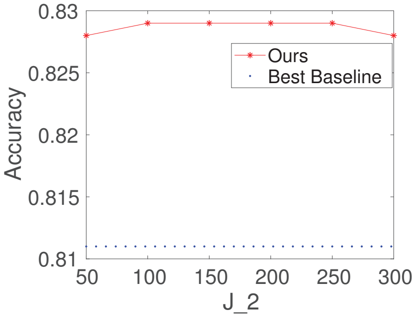


Figure 6. Results with different values of the number of hidden nodes J_2 in the fully connected layer on Protein ($K = 100$, $J\_1 = 100$).

To investigate the effects of the number of trees K of each node in the decision forest layer, we first set J_1 and J_2 to be 100 and set the value of J_2 to be 50 to 500. We report the results of the proposed method with different values of K on Gesture. The results are shown in Figure 7. Based on the figure, we make the following observations: (i) the performance of the results increases at first and then increases a little; (ii) from the point of view of trade-offs between training cost and performance, it seems like a reasonable point when setting K to be 100.

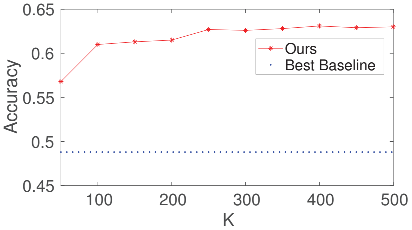


Figure 7. Results with different values of the number of trees K of each node in the decision forest layer on Gesture ($J\_1 = 100$, $J\_2 = 100$).

6. Complexity

The proposed method uses an architecture with a decision forest layer, followed by two fully connected layers and then the loss layer. Compared to GBDT, the proposed model has additional fully connected layers when the number of hidden nodes in the decision forest layer is set to be one. Compared to MLP, the proposed model has an extra decision forest layer.

7. Conclusions

The deep networks-based approaches do not work when handling data without spatial/temporal structures. To address this challenge, we propose a deep architecture that combines NN with GBDT, which takes advantage of GBDT's ability to learn dense numerical features and NN's strength in learning sparse categorical features by an embedding structure. Specifically, the architecture consists of two major parts: the decision forest layers focus on learning features from the input data and the fully connected layers focus on distilling knowledge from the decision forest layers. We designed an optimization procedure to train the proposed model via back-propagation. Although comprehensive experimental results verified the effectiveness of the proposed architecture, there are still some limitations: (1) it is not effective for all datasets, especially when learning datasets with sparse categorical features, because the experimental result is influenced by the specific context; (2) it has higher computational complexity in contrast to GBDT.

Author Contributions: Conceptualization, L.D. and S.D.; methodology, L.D. and S.D.; software, Y.X.; validation, Y.X., L.D. and S.D.; formal analysis, S.D.; investigation, H.S.; resources, H.S.; data curation, H.S.; writing—original draft preparation, Y.X. and H.S.; writing—review and editing, L.D. and S.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available in this article.

Conflicts of Interest: All authors were teachers of Taizhou University. They declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Khan, H.; Wang, X.; Liu, H. Handling missing data through deep convolutional neural network. *Inf. Sci.* **2022**, *595*, 278–293. [CrossRef]
2. Zhou, S.; Deng, X.; Li, C.; Liu, Y.; Jiang, H. Recognition-Oriented Image Compressive Sensing With Deep Learning. *IEEE Trans. Multimed.* **2023**, *25*, 2022–2032. [CrossRef]
3. Li, S.; Dai, W.; Zheng, Z.; Li, C.; Zou, J.; Xiong, H. Reversible Autoencoder: A CNN-Based Nonlinear Lifting Scheme for Image Reconstruction. *IEEE Trans. Signal Process.* **2021**, *69*, 3117–3131. [CrossRef]
4. Rasheed, M.T.; Guo, G.; Shi, D.; Khan, H.; Cheng, X. An empirical study on retinex methods for low-light image enhancemen. *Remote Sens.* **2022**, *14*, 4608. [CrossRef]
5. Rasheed, M.T.; Shi, D.; Khan, H. A comprehensive experiment-based review of low-light image enhancement methods and benchmarking low-light image quality assessment. *IEEE Trans. Signal Process.* **2023**, *204*, 108821. [CrossRef]
6. Soleymanpour, M.; Johnson, M.T.; Soleymanpour, R.; Berry, J. Synthesizing Dysarthric Speech Using Multi-Speaker Tts For Dysarthric Speech Recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Singapore, 23–27 May 2022; pp. 7382–7386.
7. Lu, H.; Li, N.; Song, T.; Wang, L.; Dang, J.; Wang, X.; Zhang, S. Speech and Noise Dual-Stream Spectrogram Refine Network With Speech Distortion Loss For Robust Speech Recognition. In Proceedings of the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
8. Liu, J.; Fang, Y.; Yu, Z.; Wu, T. Design and Construction of a Knowledge Database for Learning Japanese Grammar Using Natural Language Processing and Machine Learning Techniques. In Proceedings of the 2022 4th International Conference on Natural Language Processing (ICNLP), Xi'an, China, 25–27 March 2022; pp. 371–375.
9. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 160–167.
10. Collobert, R.; Weston, J.; Bottou, L. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
11. Asifullah, K.; Anabia, S.; Umme, C.Z.; Saeed, Q.A. A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* **2020**, *53*, 5455–5516.
12. Wang, X.; Gao, L.; Song, J. Beyond Frame-level CNN: Saliency-Aware 3-D CNN With LSTM for Video Action Recognition. *IEEE Signal Process. Lett.* **2017**, *24*, 510–514. [CrossRef]
13. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
14. Wang, F.; Tax, D.M. Survey on the attention based RNN model and its applications in computer vision. *arXiv* **2016**, arXiv:1601.06823.
15. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
16. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]

17. Díaz-Uriarte, R.; De Andres, S.A. Gene selection and classification of microarray data using random forest. *BMC Bioinform.* **2006**, *7*, 3. [CrossRef] [PubMed]
18. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2011**, *29*, 1189–1232. [CrossRef]
19. Mohan, A.; Chen, Z.; Weinberger, K. Web-search ranking with initialized gradient boosted regression trees. *Proc. Learn. Rank. Chall.* **2011**, *14*, 77–89.
20. Rao, H.; Shi, X.; Rodrigue, A.; Feng, J.; Xia, Y.; Elhoseny, M.; Yuan, X.; Gu, L. Feature selection based on artificial bee colony and gradient boosting decision tree. *Appl. Soft Comput.* **2019**, *74*, 634–642. [CrossRef]
21. Burges, C.J.C. From ranknet to lambdarank to lambdamart: An overview. *Learning* **2010**, *11*, 81.
22. Freund, Y.; Schapire, R. A short introduction to boosting. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, San Francisco, CA, USA, 31 July–6 August 1999.
23. Ke, G.; Xu, Z.; Zhang, J. DeepGBM: A deep learning framework distilled by GBDT for online prediction tasks. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 384–394.
24. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **1958**, *65*, 86. [CrossRef] [PubMed]
25. Paul, C.; Jay, A.; Emre, S. Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 191–198.
26. Chen, T.; Carlos, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
27. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3149–3157.
28. Jiang, J.; Cui, B.; Zhang, C.; Fu, F. Dimboost: Boosting gradient boosting decision tree to higher dimensions. In Proceedings of the 2018 International Conference on Management of Data, Houston, TX, USA, 10–15 June 2018; pp. 1363–1376.
29. Biau, G.; Scornet, E.; Welbl, J. Neural random forests. *arXiv* **2016**, arXiv:1604.07143v1.
30. Wang, S.H.; Aggarwal, C.C.; Liu, H. Using a Random Forest to Inspire a Neural Network and Improving on It. In Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, TX, USA, 27–29 April 2017; pp. 1–9.
31. Sethi, I.K. Entropy nets: From decision trees to neural networks. *Proc. IEEE* **1990**, *78*, 1605–1613. [CrossRef]
32. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, PA, USA, 27–29 July 1982; pp. 144–152.
33. De Boer, P.T.; Kroese, D.P.; Mannor, S. A tutorial on the cross-entropy method. *Ann. Oper. Res.* **2005**, *134*, 19–67. [CrossRef]
34. Burges, C.; Shaked, T.; Renshaw, E. Learning to rank using gradient descent. In Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 7–11 August 2005; pp. 89–96.
35. Järvelin, K.; Kekäläinen, J. IR evaluation methods for retrieving highly relevant documents. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece, 24–28 July 2000; pp. 41–48.
36. Baeza-Yates, R.; Ribeiro-Neto, B. *Modern Information Retrieval*; ACM Press: New York, NY, USA, 1999.
37. Ganjisaffar, Y.; Caruana, R.; Lopes, C.V. Bagging gradient-boosted trees for high precision, low variance ranking models. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, 24–28 July 2011; pp. 85–94.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

An Advanced Approach to Object Detection and Tracking in Robotics and Autonomous Vehicles Using YOLOv8 and LiDAR Data Fusion

Yanyan Dai, Deokgyu Kim and Kidong Lee *

Robotics Department, Yeungnam University, Gyeongsan 38541, Republic of Korea;
daiyanyan1011@gmail.com (Y.D.)
* Correspondence: kdrhee@yu.ac.kr

Abstract: Accurately and reliably perceiving the environment is a major challenge in autonomous driving and robotics research. Traditional vision-based methods often suffer from varying lighting conditions, occlusions, and complex environments. This paper addresses these challenges by combining a deep learning-based object detection algorithm, YOLOv8, with LiDAR data fusion technology. The principle of this combination is to merge the advantages of these technologies: YOLOv8 excels in real-time object detection and classification through RGB images, while LiDAR provides accurate distance measurement and 3D spatial information, regardless of lighting conditions. The integration aims to apply the high accuracy and robustness of YOLOv8 in identifying and classifying objects, as well as the depth data provided by LiDAR. This combination enhances the overall environmental perception, which is critical for the reliability and safety of autonomous systems. However, this fusion brings some research challenges, including data calibration between different sensors, filtering ground points from LiDAR point clouds, and managing the computational complexity of processing large datasets. This paper presents a comprehensive approach to address these challenges. Firstly, a simple algorithm is introduced to filter out ground points from LiDAR point clouds, which are essential for accurate object detection, by setting different threshold heights based on the terrain. Secondly, YOLOv8, trained on a customized dataset, is utilized for object detection in images, generating 2D bounding boxes around detected objects. Thirdly, a calibration algorithm is developed to transform 3D LiDAR coordinates to image pixel coordinates, which are vital for correlating LiDAR data with image-based object detection results. Fourthly, a method for clustering different objects based on the fused data is proposed, followed by an object tracking algorithm to compute the 3D poses of objects and their relative distances from a robot. The Agilex Scout Mini robot, equipped with Velodyne 16-channel LiDAR and an Intel D435 camera, is employed for data collection and experimentation. Finally, the experimental results validate the effectiveness of the proposed algorithms and methods.

Keywords: object detection and tracking; ground threshold; calibrations; onboard sensors

Citation: Dai, Y.; Kim, D.; Lee, K. An Advanced Approach to Object Detection and Tracking in Robotics and Autonomous Vehicles Using YOLOv8 and LiDAR Data Fusion. *Electronics* **2024**, *13*, 2250. https://doi.org/10.3390/electronics13122250

Academic Editors: Chao Zhang, Wentao Li, Huiyan Zhang and Tao Zhan

Received: 24 May 2024
Revised: 1 June 2024
Accepted: 6 June 2024
Published: 7 June 2024

1. Introduction

Object detection and tracking are important concepts in robotics [1–3] and self-driving [4,5]. There are several applications of object detection and tracking, such as in robotics for obstacle detection [5,6], in retail for tracking customer movement [7], in sports for analyzing player movements [8], and in security systems for monitoring and surveillance [9]. Object detection and tracking are important components in the fields of robotics and self-driving vehicles. In robotics, object detection helps robots to understand their environment better. It helps robots build more accurate maps and effectively avoid obstacles and other robots. For self-driving vehicles, detecting and tracking objects, like other vehicles, pedestrians, road barriers, road signs, traffic lights, and lane marking, helps self-driving vehicles understand their surroundings, make decisions, and navigate safely and correctly. In addition, object detection and tracking can help the robots or self-driving vehicles gather amounts of data, which can be used for optimizing paths, improving algorithms and models, and enhancing the safety and efficiency of autonomous driving systems. Computational intelligence techniques [10,11], particularly those involving machine learning and deep learning, play a significant role in these systems.

Current object detection and tracking approaches have the challenges of speed, accuracy, and reliability, particularly in dynamic and complex environments [12,13]. Addressing these challenges is essential for advancing autonomous technology. This paper proposes an integrated approach that combines LiDAR and camera data to achieve a comprehensive understanding of the environment, enhancing object localization and mapping. By leveraging the speed and accuracy of YOLOv8 alongside the precise spatial analysis capabilities of LiDAR, the proposed system aims to improve detection and tracking performance, meeting the demands of real-time applications. Maintaining the identity of objects in motion, despite changes in appearance or partial occlusion, remains a significant challenge [14–16]. This paper introduces a 3D tracking algorithm designed to calculate the 3D poses of objects, ensuring robust detection and tracking even when objects are partially occluded or missing parts. Enhanced tracking methods aim to improve the robustness and reliability of tracking systems, ensuring consistent object identification.

Combining LiDAR and camera data to achieve object detection and tracking requires several technologies: LiDAR data processing, object detection based on camera data, LiDAR and camera data calibration and fusion, and object tracking.

LiDAR point cloud data consist of millions of points that map the surroundings in three dimensions, providing a detailed view beyond 2D images or videos. This is critical for accurately identifying and distinguishing objects in a scene. LiDAR provides precise distance measurements, and being able to accurately determine the position, size, and shape of objects is critical for applications such as autonomous driving, where understanding the spatial arrangement of objects is crucial. LiDAR performs well in different lighting and weather conditions, such as low light, fog, or rain. This robustness enables LiDAR to reliably detect and track objects in a variety of environments. Point cloud data generated in real time can be analyzed and responded to instantly. This is important for applications that require fast decision making, such as collision avoidance systems in self-driving cars; moreover, these are significant in LiDAR point cloud data for object detection and tracking. LiDAR ground thresholding [17,18] is an important step in processing LiDAR point cloud data, especially for applications like navigation and mapping. The goal is to separate ground points from non-ground points. In this paper, we propose a simple ground thresholding algorithm to quickly segment the ground points and non-ground points. In the algorithm, a threshold height is used to determine whether a LiDAR point is classified as a ground point or a non-ground point. The challenge is to select the correct threshold height for effective ground point segmentation. In this paper, to solve this problem, three conditions are considered to determine the threshold height: flat terrain, uphill terrain, and downhill terrain.

The object detection process involves identifying objects within a single image or frame. It includes recognizing the object's type and drawing a bounding box or similar marker around the object to highlight its location in the image. The techniques for object detection include YOLO (You Only Look Once) [19], SSD (Single Shot MultiBox Detector) [20], and Faster R-CNN [21]. YOLO system is highly efficient and has revolutionized the field of computer vision due to its speed and accuracy. Unlike traditional object detection systems that first propose regions and then classify each region separately, YOLO applies a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region simultaneously. YOLO's unique approach allows it to perform detection at a much faster rate compared to other methods. This makes it particularly suitable for real-time applications, such as video surveillance, self-driving cars, and robotics. YOLO also achieves high accuracy in detecting objects, although it may not be as precise as some slower, region-proposal-based methods. It balances speed and accuracy effectively, making it a popular choice in many practical applications. Compared with earlier YOLO versions [22–24], YOLOv8 is faster, more accurate, and more flexible [25,26]. These advantages make YOLOv8 particularly suitable for fusion with LiDAR data, providing a powerful solution for real-time object detection and tracking in complex dynamic environments. This fusion solves the key challenges faced by current object detection and tracking systems and improves overall performance and reliability.

The extrinsic calibration between a LiDAR sensor and a camera is a crucial step in integrating data from both sensors for applications in computer vision and mobile robotics [27,28]. LiDAR sensors provide depth information by measuring the time it takes for laser pulses to return after hitting an object. On the other hand, cameras capture color and texture information. By combining these two sources of data, a more comprehensive understanding of the environment can be achieved. This fusion is especially powerful in tasks like object recognition and navigation. The fusion of LiDAR and camera data enables more accurate localization of objects in the environment. While LiDAR provides precise distance measurements, cameras contribute additional information about the appearance of objects. There are two parts of calibration, intrinsic calibration and extrinsic calibration. For the camera, intrinsic calibration estimates the intrinsic parameters such as focal length, skew, and image center. This is often carried out using calibration patterns or dedicated calibration procedures. LiDAR sensors usually do not have intrinsic parameters like cameras, but checking for any potential distortions or biases in the LiDAR data is essential. Extrinsic calibration estimates the rigid body transformation between the LiDAR and the camera [29]. Extrinsic calibration ensures that data from the LiDAR and the camera are aligned correctly in a shared coordinate system. In autonomous vehicles and robotic systems, extrinsic calibration is essential for tasks like obstacle avoidance, path planning, and navigation. In this paper, a simple LiDAR and camera fusion process is proposed to translate the coordinates of objects detected by LiDAR into the coordinate system of the camera image.

Once objects are detected, the object tracking process involves following the objects and calculating the pose of the object in a video. The challenge in object tracking is to maintain the identity of the object even when it moves, changes in appearance, or is partially obscured. For object tracking, Kalman Filter [30], Mean-shift [31], and tracking-by-detection approaches are used. This paper describes an advanced approach for object detection and tracking, integrating YOLOv8 for image-based object detection with LiDAR data for precise spatial analysis. According to the object detection result and LiDAR points calibration result, a method is proposed for LiDAR points segmentation to cluster different objects. Then, the object tracking algorithm is proposed to calculate the object's 3D poses and to calculate the relative distance between the robot and the object.

2. Related Works

The related work of object detection and tracking is as in [12,32,33]. Ref. [10] introduced an approach to object detection and tracking based on YOLO and Kalman Filter algorithms. However, the approach lacks two sensors' calibration processes and it does not provide the 3D poses of the object. Ref. [32] focuses on fusing the LiDAR and camera to achieve better detection performance. However, there is no LiDAR data segment process to increase efficiency. The authors ignored the object tracking issue. Object detection and tracking are critical and interrelated components in the fields of robotics and autonomous vehicles. Ref. [33] improves the object detection performance by designing a feature switch layer, based on camera–lidar fusion. However, it did not apply the method for 3D object detection.

The related work of ground thresholding is as in [34,35]. Ref. [34] proposes a Jump–Convolution Process (JCP) to convert the 3D point cloud segmentation problem into a 2D image smoothing problem. The method improves segmentation accuracy and terrain adaptability while maintaining low time costs, making it suitable for real-time applications in autonomous vehicles. While the method is designed to be fast, it still requires significant computational resources, particularly for real-time applications. The convolution operations and the iterative nature of the jump process can be computationally intensive. The approach involves projecting 3D point cloud data onto a 2D image for processing. This projection can lead to a loss of spatial information and might result in inaccuracies. Ref. [35] uses PointNet and Pillar Feature Encoding to estimate ground plane elevation and segment ground points in real time. The accuracy of the 2D elevation maps used in GroundGrid depends on the resolution of the grid. High-resolution grids provide better detail but at the cost of increased computational load and memory usage. The approach might face scalability issues when applied to very large-scale point clouds or environments.

The contributions of this paper are as follows: (1) Compared with [34,35], this paper presents a simple ground thresholding algorithm to quickly segment ground points and non-ground points in LiDAR data. This algorithm considers different terrain conditions (flat, uphill, and downhill) to determine the correct threshold height, enhancing the efficiency and effectiveness of ground point segmentation. (2) A method for extrinsic calibration between LiDAR and camera data is proposed. This process ensures accurate alignment of data from both sensors in a shared coordinate system, which is essential for tasks such as obstacle avoidance, path planning, and navigation. (3) This paper proposes an advanced system that integrates YOLOv8 for image-based object detection with LiDAR data for precise spatial analysis. This integration applies YOLOv8's speed and accuracy and LiDAR's precise measurements to enhance object detection and tracking performance, meeting the demands of real-time applications. (4) A novel 3D tracking algorithm is introduced to calculate the 3D poses of objects. This algorithm ensures robust detection and tracking even when objects are partially occluded or missing parts. Compared with [10,32,33], the algorithms proposed in this paper help detect 3D objects and calculate 3D poses of objects.

3. Multiple Object Detection and Tracking

3.1. System Overview

The goal of this work is 3D object detection and tracking. The system structure addresses the object detection and tracking problem as shown in Figure 1. There are LiDAR point cloud input and camera RGB image input. After receiving the LiDAR point cloud data, the ground point is filtered. Then, the non-ground LiDAR points undergo the calibration process. After receiving the camera RGB image, YOLOv8 is used for object detection, based on a customized dataset. Then, the image with 2D bounding boxes undergoes the calibration process. In the calibration of the LiDAR–camera process, a simple LiDAR and camera fusion process is proposed to translate the coordinates of objects detected by LiDAR into the coordinate system of the camera image. The process includes data collection, coordinate transformation, feature extraction, calibration, and validation. The conversion between the LiDAR coordinate and the image pixel coordinate will be determined. Then, the LiDAR Point Cloud Clustering is processed. The pixel coordinates are in the 2D bounding boxes, based on the YOLO object detection result, and are recognized as the same object. According to the calibration results, the LiDAR points are segmented, with their transformed pixel coordinates within the 2D bounding boxes. Based on the top view result of the LiDAR points and LiDAR segmentation result, 3D bounding boxes for the LiDAR points can be obtained. The center of the bounding boxes is calculated. Then, we approach the object tracking algorithm to calculate the object's 3D poses and determine the relative distance between the robot and the object.

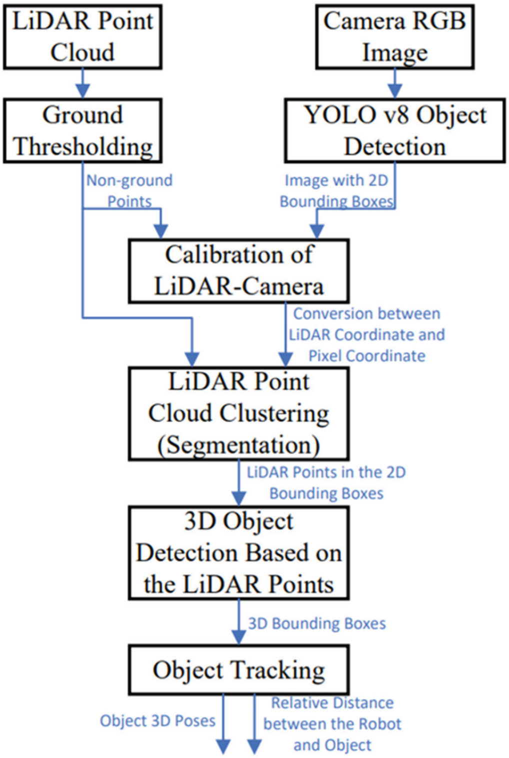

LiDAR Point Cloud
Camera RGB Image
Ground Thresholding
YOLO v8 Object Detection
Non-ground Points
Image with 2D Bounding Boxes
Calibration of LiDAR-Camera
Conversion between LiDAR Coordinate and Pixel Coordinate
LiDAR Point Cloud Clustering (Segmentation)
LiDAR Points in the 2D Bounding Boxes
3D Object Detection Based on the LiDAR Points
3D Bounding Boxes
Object Tracking
Object 3D Poses
Relative Distance between the Robot and Object

Figure 1. System structure of 3D object detection and tracking.

3.2. LiDAR Data Ground Segmentation

As shown in Figure 2, the LiDAR sensor reports points in spherical coordinates (detection distance d, elevation ω, and azimuth α). There is a detected LiDAR point $P_i(x_i, y_i, z_i)$, where i is the point's index number. The detection distance OP_i is calculated as d_i as follows:

$$d_i = \sqrt{x_i^2 + y_i^2 + z_i^2} \tag{1}$$

The elevation ω_i is the angle between the horizontal plane and the line to the target. ω_i is calculated as follows:

$$\omega_i = \arccos\left(\frac{z_i}{d_i}\right) \tag{2}$$

The azimuth is the angular measurement in a spherical coordinate system. It is the angle between a reference direction and the line from the observer to the point of interest, projected onto the horizontal plane. The azimuth is measured in degrees, with values ranging from 0 to 360 degrees, usually in a clockwise direction from the reference direction. α_i is calculated as follows:

$$\alpha_i = \text{atan2}(x_i, y_i) \tag{3}$$

Convert the spherical data of point P_i to the Cartesian coordinates. The transfer equations are as follows:

$$x_i = d_i * \cos(\omega_i) * \sin(\alpha_i) \tag{4}$$

$$y_i = d_i * \cos(\omega_i) * \cos(\alpha_i) \tag{5}$$

$$z_i = d_i * \sin(\omega_i) \tag{6}$$

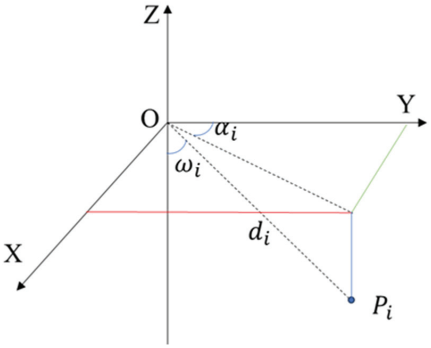


Figure 2. LiDAR point in the coordinate system to segment the ground point.

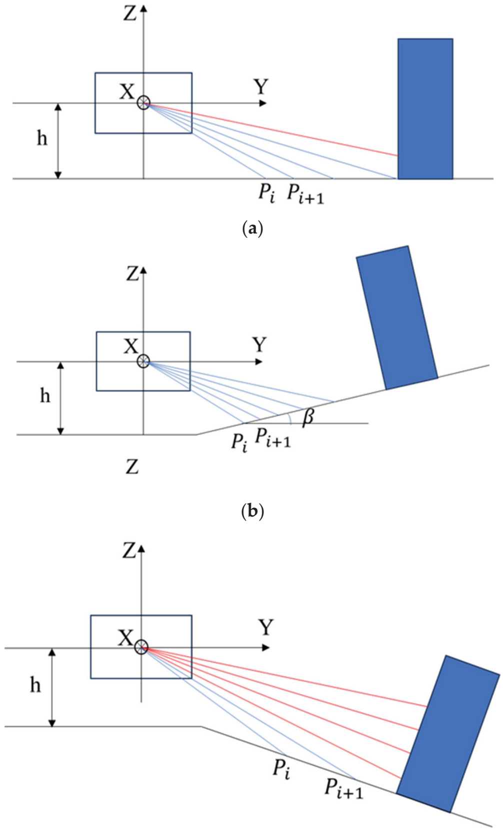
In order to quickly process the ground segmentation algorithm, the LiDAR points with negative *z*-axis values are taken into account. A threshold height h_t is defined to determine whether a point P_i is classified as a ground point or a non-ground point. If the absolute of z_i at point P_i is greater than or equal to the threshold height h_t, then the point P_i is considered a detected ground point. Conversely, if the absolute of z_i is less than the threshold height h, the point P_i is classified as a non-ground point. However, the challenge lies in selecting the correct threshold height h_t for effective ground point segmentation. Different road surfaces exhibit distinct characteristics. As shown in Figure 3, three conditions are considered to determine the threshold height h_t: flat terrain, uphill terrain, and downhill terrain. In order to calculate the threshold height h_t, two points, P_i and P_{i+1}, with the same azimuth are considered. The angle β is calculated as follows:

$$\beta = \text{atan2}((z_{i+1} - z_i), (y_{i+1} - y_i)) \tag{7}$$

As shown in Figure 4, h represents the mounting height of the LiDAR above the ground. It is related to the height of the robot, the sensor bracket, and the size of the LiDAR sensor. A constant compensatory distance is defined as d_c and the threshold height h_t is calculated as (8), where β is calculated from (7). d_c is a constant value, which is defined based on the distance between LiDAR and the object. It is equal to the average y-value of points with the same azimuth angle minus the minimum y-value.

$$h_t = h + d_c * \tan(-\beta) \tag{8}$$

As shown in Figure 3a, the default road is a flat terrain road and β is zero. Therefore, the threshold height h_t is equal to h. For the uphill condition, h_t is lower than h. For the downhill condition, h_t is higher than h. For the uphill and downhill conditions, d_c is defined based on the real environment, for example, the distance of the detected object in front of the LiDAR.



(**a**)

(**b**)

(**c**)

Figure 3. The three conditions to determine the threshold height: (**a**) flat terrain, (**b**) uphill terrain, and (**c**) downhill terrain.

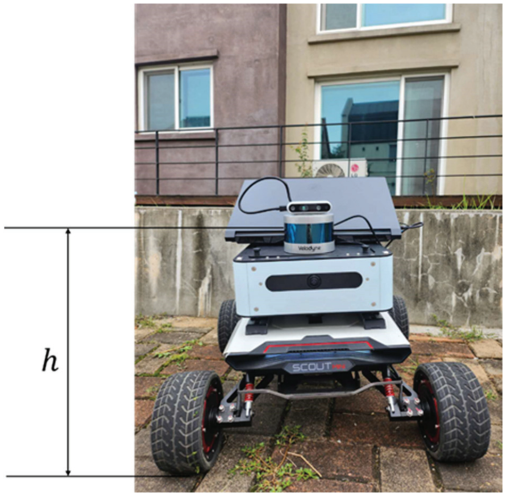


Figure 4. The height of the LiDAR sensor from the ground in a real environment.

3.3. 3D LiDAR and Camera Data Fusion

In this section, a simple LiDAR and camera fusion process is proposed to translate the coordinates of objects detected by LiDAR into the coordinate system of the camera image. The process includes data collection, coordinate transformation, feature extraction, calibration, and validation. In the data collection process, based on the camera's RGB Field of View (FOV), the right range of LiDAR points are collected. For example, as shown in Figure 4, When a Velodyne LiDAR and an Intel D435 camera are both facing forward and the camera's RGB FOV is $69^\circ \times 42^\circ$, it implies that the LiDAR's horizontal detection angle range should be matched with the camera's horizontal FOV for effective data fusion. In this scenario, the LiDAR's azimuth α is confined between 0 to 34.5 degrees and 315.5 to 360 degrees. This range ensures that the area covered by the LiDAR overlaps with the camera's field of vision. This fusion enhances the accuracy of object detection and is highly beneficial for understanding the surroundings in complex environments. After collecting the right range of LiDAR data, the LiDAR point positions in Cartesian coordinates are calculated based on (4)–(6). LiDAR Data Ground Segmentation in Section 3.2 is used to filter the ground data. Then, among the non-ground in-range points, based on the distance from the LiDAR to the detection points and the value of inflection, some points are selected for calibration. As shown in Figure 5, the car is utilized for calibration. The distance between the LiDAR sensor and the car is measurable. The color of the car differs from that of its wheel. This distinction is significant because white and black surfaces reflect light differently, influencing the LiDAR reflection value. These contrasting colors allow for the selection of characteristic points for calibration. The positions of these selected LiDAR points are then recorded in both the LiDAR coordinate system and the camera's pixel coordinate system.

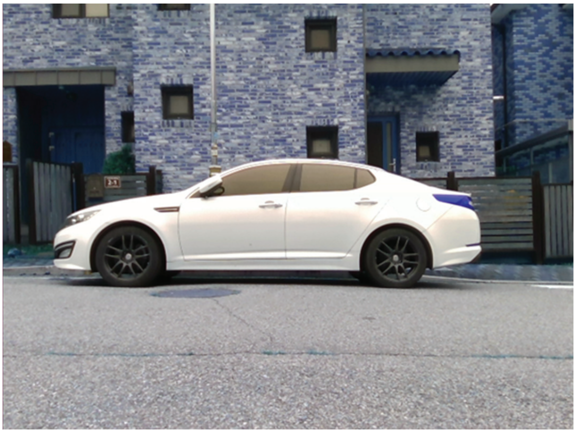


Figure 5. Environment for LiDAR and camera data fusion.

In the calibration process, define a transformation matrix M, which converts the coordinate system of LiDAR to the coordinate system of the camera. The transformation matrix includes rotation, scaling, and translation.

$$M = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

The homogeneous coordinate of LiDAR is $P = [x, y, z, 1]^T$. Then, the coordinates $P' = [x', y', z', 1]^T$ are obtained as follows:

$$P' = MP = \begin{bmatrix} ax + by + cz + d \\ ex + fy + gz + h \\ ix + jy + kz + l \\ 1 \end{bmatrix} \tag{10}$$

To convert the homogeneous coordinates P' into two-dimensional pixel coordinates $\begin{bmatrix} u & v \end{bmatrix}^T$, firstly use perspective division to convert homogeneous coordinates into three-dimensional Cartesian coordinates, and then use the intrinsic parameter of the camera to transfer. The intrinsic parameter matrix of the camera is defined as (11).

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

f_x and f_y are the focal lengths along the x and y axes, respectively, measured in pixels. c_x and c_y are the coordinates of the image's center point. The two-dimensional pixel coordinates $\begin{bmatrix} u & v \end{bmatrix}^T$ can be obtained as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \tag{12}$$

The least squares method is used to find the parameters $a, b, c, d, e, f, g, h, i, j, k, l$ in the transformation matrix M. An objective function $f(M)$ is created to measure the difference between the LiDAR coordinates transformed by these parameters, based on (9)–(12), and the actual camera pixel coordinates $\begin{bmatrix} u_c & v_c \end{bmatrix}^T$. The function $f(M)$ is defined as follows:

$$f(M) = \sum_{j=1}^{n} \left[\left(u_j - u_{c\_j} \right)^2 + \left(v_j - v_{c\_j} \right)^2 \right] \tag{13}$$

where n is the total number of calibration points and j is the index of the jth calibration point. Through iterative optimization algorithms, the parameters of M are adjusted according to the gradient of the objective function, until the convergence criteria are met or the predetermined number of iterations is completed.

3.4. Object Detection and Tracking

In order to use YOLO v8 to detect the objects, a customized dataset is created. As shown in Figure 4, using Agilex's Scout Mini robot equipped with an Intel D435 camera, images are collected in a residential community. The target categories include pedestrians, cars, motorcycles, bicycles, street lights, trees, and houses. Two lighting conditions are considered: bright light and darkness. Totally, 1694 images are used for training and 424 images are used for testing. LabelImg is used to create bounding boxes and label categories. The .txt file is created with the same name for each image, containing the labeled information. Set the configuration file and the training environment, and then the model is trained.

Based on the calibration process in Section 3.3, the LiDAR points are transformed in the pixel coordinates. For object tracking, LiDAR Point Cloud Clustering is integrated with the YOLO object detection result. As shown in Figure 6, the LiDAR data are used in conjunction with YOLO-detected objects. The pixel coordinates of these objects within 2D bounding boxes are identified as belonging to the same object. Based on the top view result of the LiDAR points, the LiDAR points $P_n = [x_n, y_n, z_n] (n = 1, 2, 3, \ldots, n)$ detected from the same object can be separated. Indicated as the red point shown in Figure 6, the center point $CP = [x_c, y_c, z_c]$ of the object can be calculated as follows:

$$x_c = \frac{(\max(x_n) - \min(x_n))}{2} + \min(x_n) \tag{14}$$

$$y_c = \frac{(\max(y_n) - \min(y_n))}{2} + \min(y_n) \tag{15}$$

$$z_c = \frac{(\max(z_n) - \min(z_n))}{2} + \min(z_n) \tag{16}$$

The center point is defined as the object point for tracking. The relative distance between the center point of the object and the LiDAR is as shown in (17):

$$d_{ol} = \sqrt{x_c^2 + y_c^2 + z_c^2} \tag{17}$$

The elevation ω_{ol} is calculated as follows:

$$\omega_{ol} = \arccos\left(\frac{z_c}{d_{ol}}\right) \tag{18}$$

Based on (1) in [36], the LiDAR coordinate can be transferred to the world coordinate. Therefore, based on (14)–(18), the coordinates of the center point in the world coordinate system can be calculated. In order to obtain the 3D bounding box of the objects, the length, width, and height of the box can be calculated as follows:

$$length = \max(x_n) - \min(x_n) + m \tag{19}$$

$$width = \max(y_n) - \min(y_n) + m \tag{20}$$

$$height = \max(z_n) - \min(z_n) + m \tag{21}$$

where m is the extended distance. The corner coordinates of the box are obtained as follows:

$$corners = \begin{bmatrix} x_c - \frac{length}{2} & y_c - \frac{width}{2} & z_c - \frac{height}{2} \\ x_c + \frac{length}{2} & y_c - \frac{width}{2} & z_c - \frac{height}{2} \\ x_c + \frac{length}{2} & y_c + \frac{width}{2} & z_c - \frac{height}{2} \\ x_c - \frac{length}{2} & y_c + \frac{width}{2} & z_c - \frac{height}{2} \\ x_c - \frac{length}{2} & y_c - \frac{width}{2} & z_c + \frac{height}{2} \\ x_c + \frac{length}{2} & y_c - \frac{width}{2} & z_c + \frac{height}{2} \\ x_c + \frac{length}{2} & y_c + \frac{width}{2} & z_c + \frac{height}{2} \\ x_c - \frac{length}{2} & y_c + \frac{width}{2} & z_c + \frac{height}{2} \end{bmatrix} \tag{22}$$

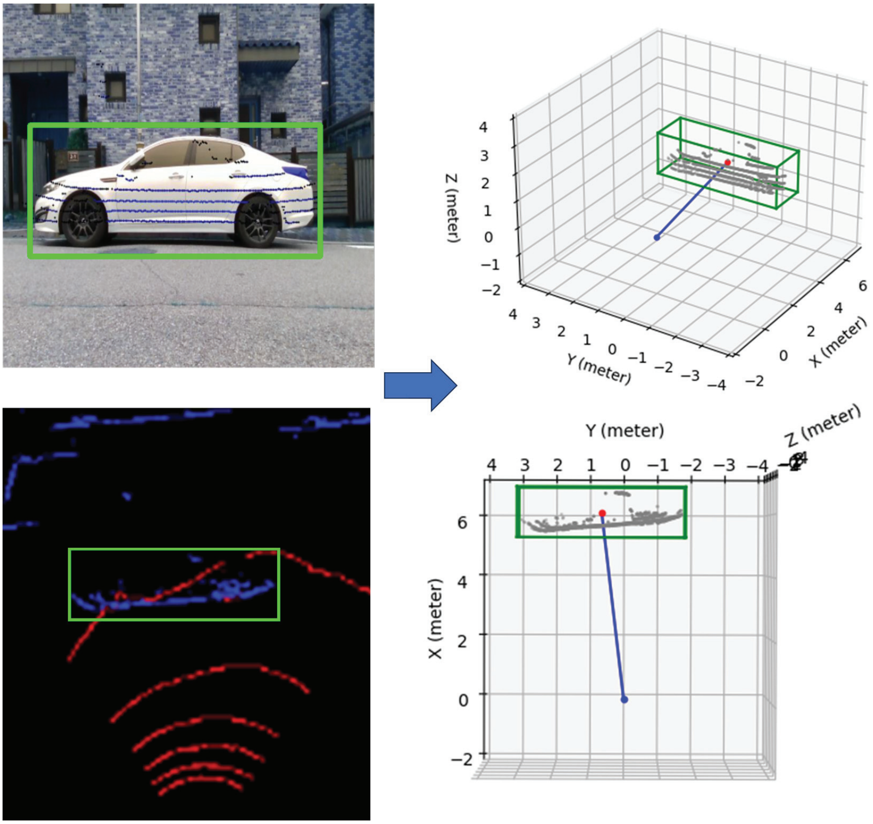


Figure 6. Object detection and tracking. On the left side, the detected car is in the green block. On the right side, the detected car's 3D bounding box is shown as a green box. The red point is used to calculate the detected car's pose.

4. Simulations and Experiment

In this section, the data are collected by using the Agilex Scout Mini onboard PC. As shown in Figure 4, a Velodyne LiDAR and an Intel D435 camera are mounted on the robot. Both sensors face forward, the x-axis of the robot's moving direction. The camera's RGB FOV is $69^\circ \times 42^\circ$, and it implies that the LiDAR's horizontal detection angle range should be matched with the camera's horizontal FOV for effective data fusion. In the experiment, the LiDAR's azimuth α is confined between 0 to 34.5 degrees and 315.5 to 360 degrees. The mounting height of the LiDAR above the ground is 0.412 m. d_c is defined based on the real environment, for example, the distance of the detected object in front of the LiDAR. The parameters $a, b, c, d, e, f, g, h, i, j, k, l$ in the transformation matrix M are obtained as [28.6197, −93.72131, 156.6157, 291.09294, −52.93593, 27.92016, −310.95358, 407.69218, 0.8058, 0.45731, 0.77985, 0.82676]. For object detection using YOLOv8, the target categories include pedestrians, cars, motorcycles, bicycles, street lights, trees, and houses. Two lighting conditions are considered: bright light and darkness. Totally, 1694 images are used for training and 424 images are used for testing. The model is selected as the YOLOv8n model.

The experimental design involves the following steps: LiDAR data ground segmentation, object detection, LiDAR–camera data calibration, object tracking, and 3D bounding box construction.

Figure 7 shows the results of the LiDAR data ground segmentation. Figure 7a is the top-view result of all LiDAR point cloud data. Figure 7b displays the results of ground segmentation, where red points represent ground points and blue points represent non-ground points. As shown in Figure 4, the mounting height of the LiDAR above the ground is 0.412 m. The ground points are correctly identified and separated from non-ground points.

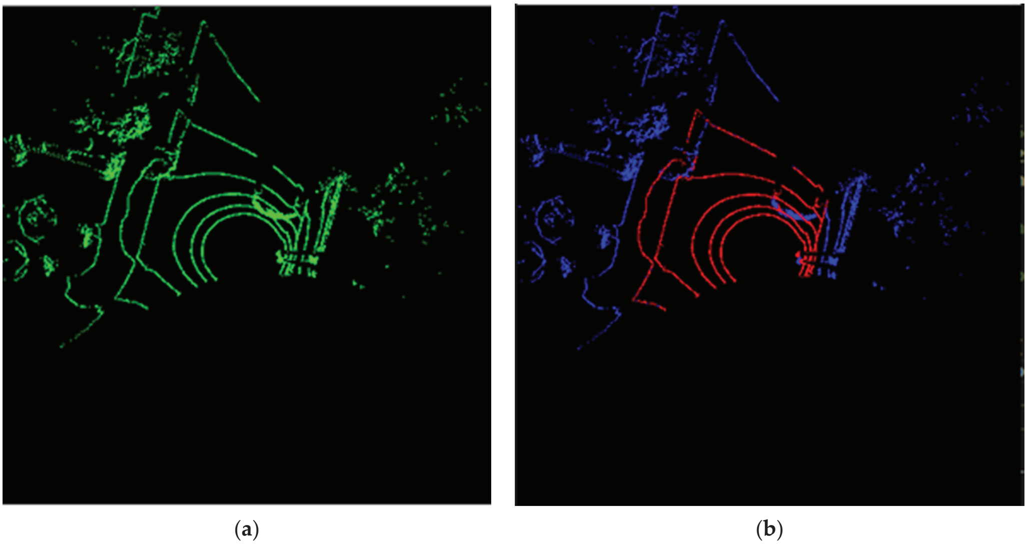


Figure 7. The results of the LiDAR data ground segmentation. (**a**) Top-view result of all LiDAR point cloud data; (**b**) ground segmentation result: red points represent ground points and blue points represent non-ground points.

The object detection process, based on YOLOv8, efficiently and accurately identifies two cars and one truck, as shown in Figure 8. The object tracking algorithm uses the detection results from Figure 8.

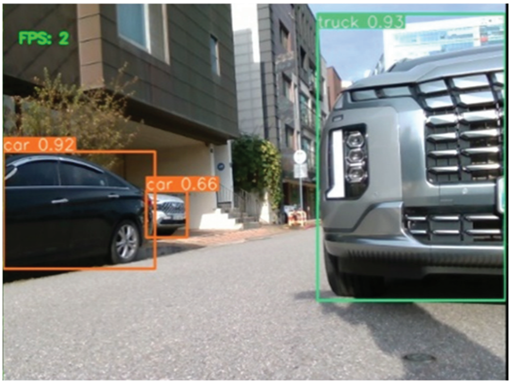


Figure 8. Object detection result using YOLOv8.

As shown in Figure 8, three objects are detected. Based on the calibration process in Section 3.3, the LiDAR points are transformed in the pixel coordinates. The LiDAR data are used in conjunction with the object detection results. The pixel coordinates of the truck and two cars within 2D bounding boxes are identified as belonging to the same object. The results are shown in Figure 9. The green points, red points, and blue points are the detected LiDAR points of the truck, black car, and white car, separately. The transformed lidar points coincide with the actual pixel coordinates of the detected object. Therefore, the calibration precision is acceptable. Despite occlusions and missing parts between the cars and truck, the LiDAR points can still be well segmented and accurately matched with their respective detected objects.

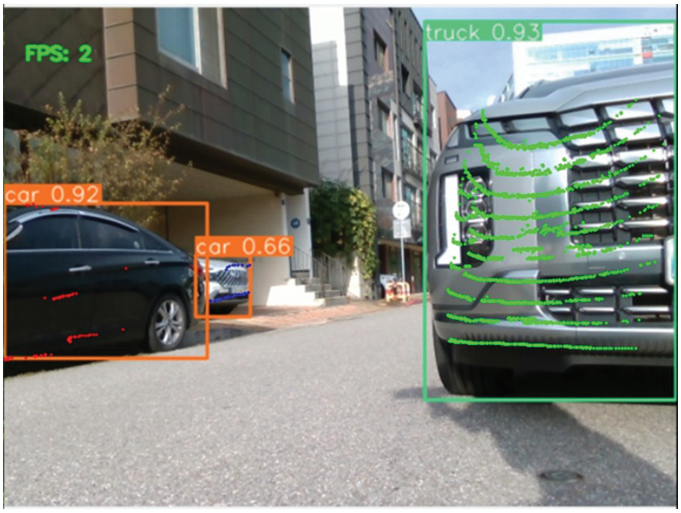


Figure 9. Object detection result and LiDAR point calibration results.

Based on the object detection and calibration results in Figure 9 and the top-view result of all LiDAR point cloud data in Figure 7b, 3D bounding boxes are constructed to represent each object, as depicted in Figure 10. The red points indicate the central points for the 3D bounding boxes. The coordinates for the central points of the truck, Car1, and Car2, relative to the robot's zero point, are as follows: $\begin{bmatrix} 1.96 & -1.01 & 0.25 \end{bmatrix}^T$, $\begin{bmatrix} 3.92 & 3.63 & 0.14 \end{bmatrix}^T$, and $\begin{bmatrix} 6.76 & 2.03 & -0.5 \end{bmatrix}^T$, separately. The corners' coordinates for the bounding box of the truck are as in (23). The length, width, and height of the truck bounding box are 1.96 m, 1.15 m, and 0.96 m.

$$corners_{truck} = \begin{bmatrix} 1.38 & -1.99 & -0.23 \\ 2.53 & -1.99 & -0.23 \\ 2.53 & -0.03 & -0.23 \\ 1.38 & -0.03 & -0.23 \\ 1.38 & -1.99 & 0.73 \\ 2.53 & -1.99 & 0.73 \\ 2.53 & -0.03 & 0.73 \\ 1.38 & -0.03 & 0.73 \end{bmatrix} \tag{23}$$

The corners' coordinates for the bounding box of Car1 are as in (24). The length, width, and height of the truck bounding box are 1.86 m, 1.88 m, and 1.04 m.

$$corners_{car1} = \begin{bmatrix} 2.98 & 2.71 & -0.38 \\ 4.86 & 2.71 & -0.38 \\ 4.86 & 4.57 & -0.38 \\ 2.98 & 4.57 & -0.38 \\ 2.98 & 2.71 & 0.66 \\ 4.86 & 2.71 & 0.66 \\ 4.86 & 4.57 & 0.66 \\ 2.98 & 4.57 & 0.66 \end{bmatrix} \tag{24}$$

The corners' coordinates for the bounding box of Car2 are as in (25). The length, width, and height of the truck bounding box are 1.31 m, 0.54 m, and 0.97 m.

$$corners_{car2} = \begin{bmatrix} 6.11 & 2.04 & -0.5 \\ 7.42 & 2.04 & -0.5 \\ 7.42 & 2.58 & -0.5 \\ 6.11 & 2.58 & -0.5 \\ 6.11 & 2.04 & 0.47 \\ 7.42 & 2.04 & 0.47 \\ 7.42 & 2.58 & 0.47 \\ 6.11 & 2.58 & 0.47 \end{bmatrix} \tag{25}$$

The dimensions of the 3D bounding boxes closely match the dimensions of the objects detected by the 3D LiDAR.

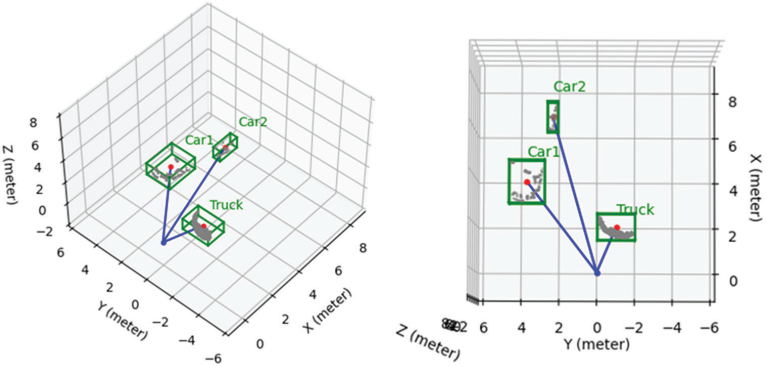


Figure 10. Object tracking result.

5. Conclusions

This paper presents an innovative and effective approach to object detection and tracking. A novel algorithm is proposed to filter ground points from LiDAR data, which is critical for the accuracy of subsequent detection processes. YOLOv8 was used for object detection, which was trained on a customized dataset. It outputs the image through 2D bounding boxes. The development of a calibration algorithm that transforms 3D LiDAR coordinates to image pixel coordinates is a key contribution, enabling the effective correlation of LiDAR data with object detection results. This is further enhanced by a proposed method for object clustering based on combined data from object detection and LiDAR calibration. An object tracking algorithm is proposed to compute the 3D poses and relative distances of objects in relation to a robot. The Agilex Scout Mini robot, equipped with Velodyne 16-channel LiDAR and an Intel D435 camera, was applied for data collection and experimentation. The experimental results show the efficiency and effectiveness of the algorithms. Future research will focus on enhancing the diversity of object recognition and tracking, as well as improving the accuracy of object tracking.

Author Contributions: Conceptualization, Y.D.; methodology, Y.D.; software, Y.D.; validation, Y.D.; formal analysis, Y.D.; investigation, Y.D. and D.K.; resources, Y.D.; data curation, Y.D.; writing—original draft preparation, Y.D.; writing—review and editing, Y.D. and K.L.; visualization, Y.D.; supervision, Y.D.; project administration, Y.D.; funding acquisition, K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (RS-2024-00406796, HRD Program for Industrial Innovation).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

LiDAR	Light Detection and Ranging
YOLO	You Only Look Once
SSD	Single Shot MultiBox Detector
FOV	Field of View
2D	Two dimensional
3D	Three dimensional

References

1. Mehdi, S.M.; Naqvi, R.A.; Mehdi, S.Z. Autonomous object detection and tracking robot using Kinect v2. In Proceedings of the 2021 International Conference on Innovative Computing (ICIC), Lahore, Pakistan, 9–10 November 2021; pp. 1–6. [CrossRef]
2. Lee, M.-F.R.; Chen, Y.-C. Artificial Intelligence Based Object Detection and Tracking for a Small Underwater Robot. *Processes* **2023**, *11*, 312. [CrossRef]
3. Xu, Z.; Zhan, X.; Xiu, Y.; Suzuki, C.; Sh, K. Onboard Dynamic-object Detection and Tracking for Autonomous Robot Navigation with RGB-D Camera. *IEEE Robot. Autom. Lett.* **2024**, *9*, 651–658. [CrossRef]
4. Gragnaniello, D.; Greco, A.; Saggese, A.; Vento, M.; Vicinanza, A. Benchmarking 2D Multi-Object Detection and Tracking Algorithms in Autonomous Vehicle Driving Scenarios. *Sensors* **2023**, *23*, 4024. [CrossRef]
5. Mendhe, A.; Chaudhari, H.B.; Diwan, A.; Rathod, S.M.; Sharma, A. Object Detection and Tracking for Autonomous Vehicle using AI in CARLA. In Proceedings of the 2022 International Conference on Industry 4.0 Technology (I4Tech), Pune, India, 23–24 September 2022; pp. 1–5. [CrossRef]
6. Xie, D.; Xu, Y.; Wang, R. Obstacle detection and tracking method for autonomous vehicle based on three-dimensional LiDAR. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 172988141983158. [CrossRef]
7. Nguyen, P.A.; Tran, S.T. Tracking customers in crowded retail scenes with Siamese Tracker. In Proceedings of the 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), Ho Chi Minh City, Vietnam, 14–15 October 2020; pp. 1–6. [CrossRef]
8. Lee, J.; Moon, S.; Nam, D.-W.; Lee, J.; Oh, A.R.; Yoo, W. A Study on Sports Player Tracking based on Video using Deep Learning. In Proceedings of the 2020 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Republic of Korea, 21–23 October 2020; pp. 1161–1163. [CrossRef]
9. Ouardirhi, Z.; Mahmoudi, S.A.; Zbakh, M. Enhancing Object Detection in Smart Video Surveillance: A Survey of Occlusion-Handling Approaches. *Electronics* **2024**, *13*, 541. [CrossRef]
10. Azevedo, P.; Santos, V. YOLO-Based Object Detection and Tracking for Autonomous Vehicles Using Edge Devices. In *ROBOT2022: Fifth Iberian Robotics Conference*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 297–308. [CrossRef]
11. Gupta, A.; Anpalagan, A.; Guan, L.; Khwaja, A.S. Deep learning for object detection and scene perception in self-driving cars: Survey, challenges and issues. *Array* **2021**, *10*, 100057. [CrossRef]
12. Moksyakov, A.; Wu, Y.; Gadsden, S.A.; Yawney, J.; AlShabi, M. Object Detection and Tracking with YOLO and the Sliding Innovation Filter. *Sensors* **2024**, *24*, 2107. [CrossRef]
13. Balamurali, M.; Mihankhah, E. SimMining-3D: Altitude-Aware 3D Object Detection in Complex Mining Environments: A Novel Dataset and ROS-Based Automatic Annotation Pipeline. *arXiv* **2023**, arXiv:2312.06113. [CrossRef]
14. Dippal, I.; Hiren, M. Identity Retention of Multiple Objects under Extreme Occlusion Scenarios using Feature Descriptors. *J. Commun. Softw. Syst.* **2018**, *14*, 290–301. [CrossRef]
15. Luo, W.; Xing, J.; Milan, A.; Zhang, X.; Liu, W.; Kim, T.-K. Multiple object tracking: A literature review. *Artif. Intell.* **2021**, *293*, 103448. [CrossRef]
16. Wu, Y.; Wang, Y.; Liao, Y.; Wu, F.; Ye, H.; Li, S. Tracking Transforming Objects: A Benchmark. *arXiv* **2024**, arXiv:2404.18143v1.
17. Gomes, T.; Matias, D.; Campos, A.; Cunha, L.; Roriz, R. A Survey on Ground Segmentation Methods for Automotive LiDAR Sensors. *Sensors* **2023**, *23*, 601. [CrossRef]
18. Deng, W.; Chen, X.; Jiang, J. A Staged Real-Time Ground Segmentation Algorithm of 3D LiDAR Point Cloud. *Electronics* **2024**, *13*, 841. [CrossRef]
19. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2015**, arXiv:1506.02640.
20. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016. ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2016; Volume 9905. [CrossRef]

21. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2015**, arXiv:1506.01497. [CrossRef]
22. Bharat Mahaur, B.; Mishra, K.K. Small-object detection based on YOLOv5 in autonomous driving systems. *Pattern Recognit. Lett.* **2023**, *168*, 115–122. [CrossRef]
23. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
24. Li, K.; Wang, Y.; Hu, Z. Improved YOLOv7 for Small Object Detection Algorithm Based on Attention and Dynamic Convolution. *Appl. Sci.* **2023**, *13*, 9316. [CrossRef]
25. Huang, H.; Wang, B.; Xiao, J.; Zhu, T. Improved small-object detection using YOLOv8: A comparative study. *Appl. Comput. Eng.* **2024**, *41*, 80–88. [CrossRef]
26. Lee, J.; Su, Y. Balancing Privacy and Accuracy: Exploring the Impact of Data Anonymization on Deep Learning Models in Computer Vision. *IEEE Access* **2024**, *12*, 8346–8358. [CrossRef]
27. Liu, Y.; Jiang, X.; Cao, W.; Sun, J.; Gao, F. Detection of Thrombin Based on Fluorescence Energy Transfer Between Semiconducting Polymer Dots and BHQ-Labelled Aptamers. *Sensors* **2018**, *18*, 589. [CrossRef]
28. Noguera, J.M.; Jiménez, J.R. Mobile Volume Rendering: Past, Present and Future. *IEEE Trans. Vis. Comput. Graph.* **2016**, *22*, 1164–1178. [CrossRef]
29. Kwak, K.; Huber, D.F.; Badino, H.; Kanade, T. Extrinsic Calibration of a Single Line Scanning Lidar and a Camera. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 3283–3289.
30. Gunjal, P.R.; Gunjal, B.R.; Shinde, H.A.; Vanam, S.M.; Aher, S.S. Moving Object Tracking Using Kalman Filter. In Proceedings of the 2018 International Conference on Advances in Communication and Computing Technology (ICACCT), Sangamner, India, 8–9 February 2018; pp. 544–547. [CrossRef]
31. Feng, Z. High Speed Moving Target Tracking Algorithm based on Mean Shift for Video Human Motion. *J. Phys. Conf. Ser.* **2021**, *1744*, 042180. [CrossRef]
32. Liu, H.; Wu, C.; Wang, H. Real time object detection using LiDAR and camera fusion for autonomous driving. *Sci. Rep.* **2023**, *13*, 8056. [CrossRef]
33. Kim, T.-L.; Park, T.-H. Camera-LiDAR Fusion Method with Feature Switch Layer for Object Detection Networks. *Sensors* **2022**, *22*, 7163. [CrossRef]
34. Shen, Z.; Liang, H.; Lin, L.; Wang, Z.; Huang, W.; Yu, J. Fast Ground Segmentation for 3D LiDAR Point Cloud Based on Jump-Convolution-Process. *Remote Sens.* **2021**, *13*, 3239. [CrossRef]
35. Paigwar, A.; Erkent, Ö.; González, D.S.; Laugier, C. GndNet: Fast Ground Plane Estimation and Point Cloud Segmentation for Autonomous Vehicles. In Proceedings of the IROS 2020-IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 2150–2156. [CrossRef]
36. Dai, Y.; Lee, K. 3D map building based on extrinsic sensor calibration method and object contour detector with a fully convolutional neural network. *Meas. Control.* **2023**, *56*, 215–227. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Local-Global Representation Enhancement for Multi-View Graph Clustering

Xingwang Zhao [1,*], Zhedong Hou [1] and Jie Wang [2]

[1] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China; hzdsxdx@163.com

[2] School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China; wangjie_reg@163.com

[*] Correspondence: zhaoxw84@163.com

Abstract: In recent years, multi-view graph clustering algorithms based on representations learning have received extensive attention. However, existing algorithms are still limited in two main aspects, first, most algorithms employ graph convolution networks to learn the local representations, but the presence of high-frequency noise in these representations limits the clustering performance. Second, in the process of constructing a global representation based on the local representations, most algorithms focus on the consistency of each view while ignoring complementarity, resulting in lower representation quality. To address the aforementioned issues, a local-global representation enhancement for multi-view graph clustering algorithm is proposed in this paper. First, the low-frequency signals in the local representations are enhanced by a low-pass graph encoder, which yields smoother and more suitable local representations for clustering. Second, by introducing an attention mechanism, the local embedded representations of each view can be weighted and fused to obtain a global representation. Finally, to enhance the quality of the global representation, it is jointly optimized using the neighborhood contrastive loss and reconstruction loss. The final clustering results are obtained by applying the k-means algorithm to the global representation. A wealth of experiments have validated the effectiveness and robustness of the proposed algorithm.

Keywords: multi-view learning; contrastive learning; graph clustering; graph learning; representation learning

Citation: Zhao, X.; Hou, Z.; Wang, J. Local-Global Representation Enhancement for Multi-View Graph Clustering. *Electronics* **2024**, *13*, 1788. https://doi.org/10.3390/electronics13091788

Academic Editor: Cheng Siong Chin

Received: 18 March 2024
Revised: 22 April 2024
Accepted: 30 April 2024
Published: 6 May 2024

1. Introduction

Graph clustering, as a crucial task in graph analysis [1], aims to partition a graph into multiple clusters, ensuring that nodes within the same cluster exhibit similarities in terms of both graph topology and attribute values [2,3]. This process enables the identification of cohesive groups within the graph, facilitating various downstream tasks such as community detection , anomaly detection, and recommendation systems [4–6]. Various graph clustering algorithms have been applied in fields such as social recommendation, link prediction, citation network analysis, protein interaction analysis, and brain network analysis [7,8]. Most existing graph clustering algorithms focus on single-view graph data [9]. However, the graph data in practical applications are typically multi-relational. For instance, in biological networks, the interactions between proteins in some organisms may involve multiple interaction patterns among thousands of protein molecules, with each protein [10] having specific attribute information. In social networks [11], individuals may have different types of social relationships, such as friends, followers, and co-groups, with each individual possessing certain descriptive characteristics. In a transportation planning network, locations are abstracted as nodes, while different types of roads: subways, highways, railways, etc., are abstracted as sides of different views. By analyzing multi-view graph data, it helps to improve the understanding of the complexity of urban transportation systems, and can provide support for improving traffic mobility and safety [12]. In comparison to single-view graph data, multi-view graph data have a multi-layered topological structure and contain richer information, facilitating a more accurate detection of pattern structures within the network. For example, a social networking platform contains rich user behavioral data and interpersonal relationship information. By integrating multi-view graph clustering algorithms to merge user's social connections, interests, and other attribute information, the social networking platform can more accurately discover connections between different groups, enabling more precise social circle recommendation services. By introducing more information about user backgrounds, interpersonal relationships, and other aspects, the platform can make product recommendations more targeted and adaptive. By comprehensively considering user behavioral data and rich interpersonal relationships, it is possible to identify potential fake accounts and fraudulent activities, further ensuring the security of the social networking platform. Therefore, multi-view graph clustering algorithms can leverage social information and user behavior patterns on the platform to provide users with more precise, intelligent, and secure services, thereby enhancing the user experience and stickiness of the social networking platform. However, there are both correlations and heterogeneity within the cluster structures contained in different views, which introduces new challenges to clustering tasks [13,14].

In recent years, the issue of multi-view graph clustering has attracted widespread attention from researchers, leading to the emergence of a series of related algorithms. These algorithms can be categorized into two types: graph clustering based on consensus graph learning and graph clustering based on representation learning. The former category attempts to learn a consensus graph by maximizing the consistency between different views. Subsequently, utilizing a traditional clustering algorithm, it derives the final clustering results [15,16]. However, clustering the consensus graph directly may result in the loss of specific information within each view. The latter category aims to integrate the attribute information of each node with the topological structure of the graph, while maximizing the preservation of information from multiple views. These algorithms map the data into a joint low-dimensional vector representation that can be used for clustering [17,18]. However, most algorithms have several limitations. For example, local representations that are constructed directly using a graph convolution network (GCN) are susceptible to high-frequency noise. Additionally, a global representation constructed from local representations of each view often only focuses on the consistency information of each view, effectively ignoring the complementary information [19].

To overcome these limitations, this paper describes a novel local-global representation enhancement for multi-view graph clustering (LGMGC) algorithm. This algorithm enhances low-frequency signals in the local representations through graph filtering, making them more suitable for clustering tasks. An attention mechanism is employed to allow the global representation to integrate information from various views, thus enhancing the connections between similar nodes in the global embedded representation and improving the clustering results. This enhances the attention towards complementary information within the global representation. Specifically, the graph data are encoded using a combination of low-pass graph filters and a multilayer perceptron (MLP). This encoding process enhances the low-frequency signals present in the local representations. The local representations from each view are then integrated into a global representation using the attention mechanism. The exploration of the topological characteristics of each view is strengthened through the reconstruction of the adjacency matrices, and the introduction of neighborhood contrastive regularization enhances the connectivity between nodes with similar attributes in the global representation, which clarifies the cluster structure. Finally, a traditional clustering algorithm is applied to the enhanced representation to obtain the final clustering results.

In summary, the contributions of this article can be outlined as follows:

- A new multi-view graph clustering algorithm via local-global representation enhancement is proposed. LGMGC enhances the local and global representations to obtain a more suitable representation for clustering.
- A simple and effective graph encoder that enhances the low-frequency signals to obtain a smoother representation is proposed.
- Comprehensive experimentation on three benchmark datasets illustrates the excellent performance of the proposed algorithm in comparison to existing deep graph clustering algorithms.

The paper is structured as follows: In Section 2, a review of related work on multi-view graph clustering is offered. In Section 3, the specific process of the proposed algorithm are thoroughly elucidated. In Section 4, we conducts an evaluation of the proposed algorithm, comparing it to existing algorithms. In Section 5, we conducted ablation experiments to verify the effectiveness of each component. In Section 6, we discussed the experimental results of the paper. In Section 7, we conclude this work and suggest the focus for future work.

2. Related Work

In this section, the related work on neighbor-based recommendation of collaborative filtering and attention mechanism is briefly reviewed.

2.1. Graph Clustering Based on Consensus Graph Learning

This category of algorithms aims to maximize the consistency between different views by learning a consensus graph. The final clustering result is obtained by applying graph partitioning or other spectral graph techniques to the consensus graph. These algorithms mainly consist of three steps: (1) preprocessing of multi-view data; (2) learning a consensus graph to maximize view consistency; and (3) clustering based on the learned consensus graph. Utilizing data information or prior knowledge to guide the learning of consensus graphs constitutes a crucial step. The self-weighted multi-view graph clustering (SwMC) algorithm initially generates a similarity matrix between nodes based on the features of each view, before maximizing the consistency of the similarity matrices across different views using a learning consensus graph approach. This algorithm leverages the Laplacian matrix of the constrained consensus graph to aid in clustering The most crucial step is how to utilize data information or prior knowledge to guide the learning of the consensus graph [20]. The multi-view attributed graph clustering (MvAGC) algorithm employs low-pass filters to smooth multiple views, and then simultaneously learns a consensus graph while generating anchor points [15]. Lin et al. [16] employs a learning consensus graph approach to maximize the consistency of smooth representations across different views. This optimizes the consensus graph structure through contrastive learning, ultimately achieving higher-quality clustering results. In this algorithm, contrastive learning brings similar nodes closer to each other and pushes dissimilar nodes apart. Thus, the consensus graph more accurately reflects the relationships between features in different views, resulting in more precise clustering. Lin et al. [21] employ low-pass filters to achieve smooth representations. Subsequently, they aim to maximize the consistency of these representations across various views while also exploring the consistency of high-order topological structure information within each view. Liu et al. [22] applied standard network embedding methods to process merged graphs or individual layers, without leveraging interlayer interactions, aiming to construct a vector space for information from multiple views. Robust multi-view spectral clustering via low-rank and sparse decomposition(RMSC) incorporate elements of low-rank and sparse decomposition. Initially, its construct a transition probability matrix from each individual view. Subsequently, these matrices are utilized to reconstruct a shared low-rank transition probability matrix, serving as a critical input for the standard Markov chain method used in clustering [23]. Fettal et al. uses a simple linear model to simultaneously accomplish clustering and representation learning [24]. These algorithms demonstrate the effectiveness of learning consensus graphs for analyzing multi-view graph data. The learning of consensus graphs reveals hidden relationships and common features between different views. However, directly partitioning the consensus graph may overlook specific information present in individual views.

2.2. Graph Clustering Based on Representation Learning

This category of algorithms learns a compact representation of nodes from multi-view graph data, and then applies clustering to the compact representation. Such algorithms can generally be divided into three steps: (1) the extraction of a compact representation of multiple views using graph embedding techniques; (2) the imposition of external constraints, such as adding corresponding regularization terms to the loss function, which ensures that the representation is more suitable for clustering; and (3) the application of traditional algorithms to the compact low-dimensional representation to obtain clustering results. The purpose of a graph auto-encoder (GAE) is to reconstitute graph by taking node features as input, compressing them into a low-dimensional representation, and reconstructing the resulting graph, followed by k-means clustering on the low-dimensional representation. While GAE serves as the cornerstone of this approach, its limitation lies in its capability to handle only single-view graph data and its inefficiency when dealing with large-scale graph data [25]. Tang et al. [26] introduces an edge sampling algorithm to enhance the effectiveness and efficiency of inference. The one2multi graph autoencoder for multi-view graph clustering (O2MAC) algorithm reconstructs the remaining views by selecting an informative view from among the multiple available views, thereby learning the node embedding. Simultaneously, the O2MAC algorithm uses self-training clustering objectives to make the representation more suitable for clustering. Finally, clustering is performed based on the learned embeddings [17]. Cai et al. [18] extracts representations of both global and local information using autoencoders, and combines different features based on their importance through adaptive weight learning algorithms. Xia et al. [27] uses graph autoencoders to extract representations of each view, before applying block-diagonal representation constraints to better explore the clustering structure. The learned clustering labels are then used to guide the learning of node representations and coeffcient matrices, which are subsequently used for clustering. Cheng et al. [28] uses a dual-path encoder to capture consistency information across different views. The first path extracts node representations, while the second path employs a consistency embedding encoder to capture the consistency of geometric relationships and probability distributions among different views. Ultimately, The resulting clustering is based on the representations learned by the consistency embedding encoder. The powerful representation learning capability of graph neural networks enables representation learning-based algorithms to explore deeper information within graph data. However, when constructing global representations based on the local embeddings of each view, they often only focus on the consistency information among views, while ignoring complementary information.

3. Proposed Algorithm

In this section, we first present some notation and definitions, and then describe the proposed algorithm in detail.

3.1. Notation and Preliminaries

A multi-view graph can be represented as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}^1, \cdots, \mathcal{E}^M, \mathbf{X}\}$, where $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ is the set of n nodes, and M sets of edges $\{\mathcal{E}^m\}_{m=1}^M$ describe the interaction between nodes in the corresponding M views. $e_{ij}^m \in \mathcal{E}^m$ represents a linkage between nodes i and j in the m-th view. These M interaction types can also be described by adjacency matrices $\{\mathbf{A}_m \in \mathbb{R}^{n\times n}\}_{m=1}^M$, where $\mathbf{A}_{ij}^m = 1$ if $e_{ij}^m \in \mathcal{E}^m$ and $\mathbf{A}_{ij}^m = 0$ otherwise. $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\} \in \mathbb{R}^{n\times d}$ is the node feature matrix, $\mathbf{x}_i$ is the feature vector of v_i, and $\mathbf{D}_m$ represents the degree matrix of the graph under the m-th view. $\widetilde{\mathbf{A}}_m = \mathbf{D}_m^{-\frac{1}{2}}\mathbf{A}_m\mathbf{D}_m^{-\frac{1}{2}}$ and $\mathbf{L}^m = \mathbf{I} - \widetilde{\mathbf{A}}_m$ represent the normalized adjacency matrix and symmetric normalized Laplacian matrix for the m-th view, respectively.

The objective of multi-view graph clustering is to divide the nodes in $\mathcal{G}$ into c clusters $\{S_1, S_2, ..., S_c\}$. The nodes in the same cluster are similar in topology and attributes, while the nodes in different clusters are quite different.

3.2. Framework of Multi-View Graph Clustering via Local-Global Representation Enhancement

The framework of the proposed algorithm, as illustrated in Figure 1, consists of three main modules: local representation generation and enhancement, global representation generation, and global representation enhancement.

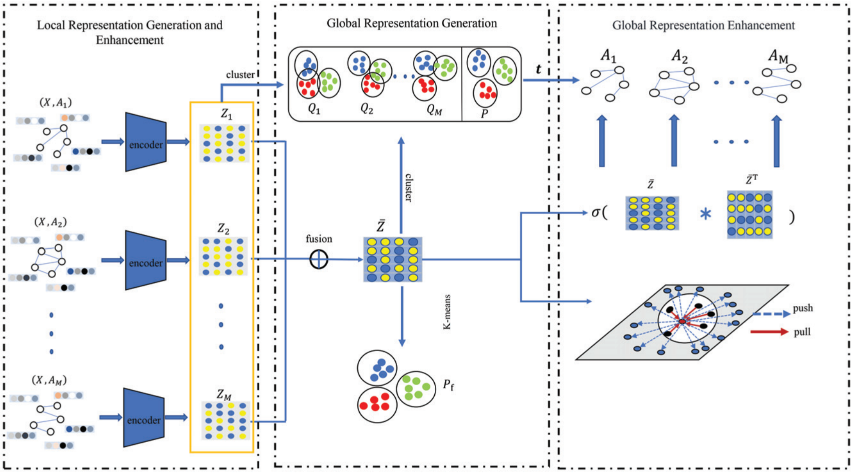


Figure 1. Illustration of LGMGC. In the local representation generation module, low-pass graph filtering combined with MLP is used to enhance the low-frequency signals of each view's representation. In the global representation generation module, an attention mechanism is employed to fuse the representations of various views, resulting in a global representation that captures both consensus information and view-specific information. In the global enhancement module, topological information of each view is explored by reconstructing the adjacency matrix. Additionally, nodes with similar topological structures and properties are pulled together while dissimilar nodes are pushed farther apart via neighborhood contrastive loss.

3.3. Local Representation Generation and Enhancement

The purpose of this module is to generate local representations for each view while enhancing the low-frequency signals within these representations. This process is accomplished through multiple graph encoders consisting of two inputs: $\mathbf{A}_m$ and $\mathbf{X}$.

First of all, we will explain the reasons for choosing low frequency signals. The symmetric regularized Laplace matrix $\mathbf{L}$ can be eigen-decomposed into $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$ where, $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$, $\lambda_i (i = 1, 2, \ldots, n)$ is the eigenvalue, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_n]^{\mathrm{T}}$ is the corresponding orthogonal eigenvector. The eigenvalues can be considered as frequencies, and the corresponding eigenvectors can be considered as Fourier bases. The smoothness of $\mathbf{u}_i$ can be reflected by λ_i.

$$\sum_{a_{j,k\in\mathcal{E}}} a_{jk}[\mathbf{u}_i(j) - \mathbf{u}_i(k)]^2 = \mathbf{u}_i^{\mathrm{T}}\mathbf{L}\mathbf{u}_i = \lambda_i. \tag{1}$$

As can be seen from Equation (1), the smaller the eigenvalue, the smoother the base signal. This means that the attribute components corresponding to smaller eigenvalues in the graph data have less difference between neighboring nodes, that is, the attributes of neighboring nodes are more similar. Therefore, the smaller the eigenvector corresponding to the eigenvalue is more conducive to the formation of cluster structure, which indicates that the graph signal required by the clustering task should contain a larger proportion of low-frequency base signals. Meanwhile, the effectiveness of GCN does not stem from non-linear feature extraction, but from aggregating features from neighboring nodes. The layering of multiple GCNs may lead to intertwining weight matrices, consequently diminishing the quality of representation learning.

Thus, the module is divided into two steps: neighbor aggregation and linear embedding. In the neighbor aggregation step, a low-pass graph filter is used to aggregate neighbor information, which enhances the low-frequency signals in the attributes, resulting in a smoothed representation that is more suitable for downstream clustering tasks. The process of filtering can be described as follows:

$$\mathbf{H}_m = (\mathbf{I} - \widetilde{\mathbf{A}}_m)^k \mathbf{X}, \tag{2}$$

where, $\mathbf{H}_m$ denotes the smoothed representation of the m-th view after filtering, and k denotes the number of layers in the graph filters. In the linear embedding step, $\mathbf{H}_m$ is embedded into a low-dimensional space by the MLP encoders without an activation function, The local embedding representation $\mathbf{Z}_m$ of the m-th view is obtained as follows:

$$\mathbf{Z}_m = f_m(\mathbf{H}_m) = \mathbf{H}_m \mathbf{W}_m + \mathbf{b}_m, \tag{3}$$

where, $\mathbf{W}_m$ and $\mathbf{b}_m$ are the learnable parameters of the encoder for the m-th view. Through the low-pass graph encoder, high-frequency noise in attributes is filtered out, the entanglement of weight matrix is avoided, the quality of representation is enhanced, and the clustering performance is improved (See Section 5.4).

3.4. Global Representation Generation

The objective of this module is to fuse the local representations $\mathbf{Z}_m$ from each view, thus obtaining a global representation $\overline{\mathbf{Z}}$ that incorporates information from all views:

$$\overline{\mathbf{Z}} = \sum_{m=1}^{M} \mathbf{Z}_m. \tag{4}$$

However, the quality of the views in multi-view graph data varies. If only high-quality views are selected for clustering, the global embedding representation would lose information from the remaining views. Conversely, treating all views equally would allow lower-quality views to adversely impact clustering results. To effectively combine complementary information from multi-view graph data and mitigate the influence of lower-quality views on clustering outcomes, distinct weights are assigned to each view through a self-supervised strategy.

In Section 3.5, the clustering structure in the global representation is enhanced by neighborhood contrast loss. Consequently, it can be argued that when the clustering result of the local representation of a view is closer to the clustering result of the global representation, the clustering quality of the view is higher and the view quality is better. During the training process, attention mechanisms are utilized to allocate respective weights to each view based on the similarity between the local and global clustering results for different views.

Specifically, the global representation $\overline{\mathbf{Z}}$ is input into the k-means clustering algorithm to generate pseudo-labels P. Using the same algorithm, predicted labels Q_m are also obtained for each view. These pseudo-labels P serve as the ground truth, while the predicted labels Q_m are considered as the results for calculating the clustering score (such as normalized mutual information), denoted as i.e., $score_m$. The weight w_m assigned to the m-th view is calculated according to the following formula:

$$w_m = \frac{1}{1 - score_m}, \tag{5}$$

By Equation (5), Greater weight is assigned to views that are more similar to the clustering result of the global representation. Then normalize the weights of each view.

$$t_m = \frac{w_m}{\sum_{i=1}^{M} w_i}. \tag{6}$$

Therefore, the fusion representation based on the attention mechanism is constructed as follows:

$$\overline{\mathbf{Z}} = \sum_{m=1}^{M} t_m \mathbf{Z}_m. \tag{7}$$

In contrast to traditional data, the primary information in graph data is embedded within its topology. For example, different views of the same group of nodes contain both consistent parts (consistency information) and distinct parts (highlighting complementary information) in their respective topological relationships. Through the introduction of an attention mechanism into the iterative training involving reconstruction loss and neighborhood contrast loss, the global representation can effectively assimilate information from each view while emphasizing the greater impact of views exhibiting more prominent clustering structures. This approach efficiently balances the complementary insights across varying perspectives, maximizing the advantages of multi-perspective learning.

3.5. Global Representation Enhancement

This module optimizes the global representation $\overline{\mathbf{Z}}$ by exploring the topological and attribute information from multiple views while strengthening the connections between similar nodes in $\overline{\mathbf{Z}}$. This process clarifies the cluster structure in $\overline{\mathbf{Z}}$.

The exploration of topological and attribute information from each view is accomplished through the reconstruction of the adjacency matrix. To comprehensively integrate information of each view into the global representation, $\overline{\mathbf{Z}}$ is used to reconstruct the adjacency matrix $\hat{\mathbf{A}}$:

$$\hat{\mathbf{A}} = \sigma(\overline{\mathbf{Z}}, \overline{\mathbf{Z}}^{\mathrm{T}}), \tag{8}$$

where, σ denotes the sigmoid activation function. The reconstruction loss is then computed as follows to capture the information of the m-th view.

$$L_{rec}^{m} = -\sum \mathbf{A}_{ij}^{m} \ln \hat{\mathbf{A}}_{ij} + (1 - \mathbf{A}_{ij}^{m}) \ln (1 - \hat{\mathbf{A}}_{ij}). \tag{9}$$

Considering the variations in quality among different views, the weights acquired from Equations (5) and (6) are integrated into the loss function. Thus, the overall reconstruction loss function is defined as follows

$$L_{rec} = \sum_{m=1}^{M} t_m L_{rec}^{m}. \tag{10}$$

The optimized global representation effectively captures both the consensus information and individual characteristics from views in the graph data. To enhance the connections between similar nodes and highlight the cluster structure in the global embedding representation, the neighborhood contrastive loss algorithm is introduced.

In terms of attributes and topology, nodes within the same cluster are similar, while nodes between different clusters have significant differences. in order to better highlight the cluster structure in the global representation, contrastive learning can be used to push similar node representations closer together, while pushing others away.

Specifically, for the given node representation $\overline{\mathbf{z}}_i$, the top-K similar nodes to $\overline{\mathbf{z}}_i$ are calculated by KNN algorithm, and these nodes form a positive pair with $\overline{\mathbf{z}}_i$, and the remaining nodes form a negative pair. The neighborhood contrastive loss L_{nbr} is as follows:

$$L_{nbr} = \sum_{i=1}^{n} \sum_{j \in \mathbb{N}_i} - \log \frac{\exp\big(\mathrm{sim}\big(\overline{\mathbf{z}}_i, \overline{\mathbf{z}}_j\big)\big)}{\sum_{p \neq i}^{n} \exp\big(\mathrm{sim}\big(\overline{\mathbf{z}}_i, \overline{\mathbf{z}}_p\big)\big)}, \tag{11}$$

where the pair-wise similarity $\mathrm{sim}(\overline{\mathbf{z}}_{\mathrm{i}}, \overline{\mathbf{z}}_{\mathrm{j}})$ is measured by the cosine similarity. $\mathbb{N}_i$ is a set consisting of the indices of the K samples obtained by applying the KNN to $\overline{\mathbf{z}}_i$.

By minimizing Equation (11), similar samples are pushed closer together while dissimilar samples are pushed father apart. The objective of this minimization is to enhance the discrimination between samples in the global representation.

Combining Equations (10) and (11), our model optimizes the following loss function:

$$L = L_{rec} + \alpha L_{nbr} \tag{12}$$

where, α is a hyper-parameter used for balancing L_{rec} and L_{nbr}. To optimize this objective function, we employ the Adam optimizer. The complete procedures is outlined in Algorithm 1.

Algorithm 1 LGMGC

```
Input:Multi-view graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}^1, \cdots, \mathcal{E}^M, \mathbf{X}\}$, number of clusters $c$, order of graph filtering $k$, number of KNN Samples $K$, hyperparameter $\alpha$, iteration number $epochs$
Output:Clustering results $P_{\mathrm{f}}$
for epoch = 1 to epochs do
    Obtain $\mathbf{Z}_m$ for each view according to Equation (3);
    Obtain $\overline{\mathbf{Z}}$ according to Equation (4) ;
    Obtain $\mathbf{t}_m$ according to Equation (6);
    Update model parameters by minimizing Equation (12);
end for
Obtain $P_{\mathrm{f}}$ by performing k-means clustering on $\overline{\mathbf{Z}}$
```

4. Experiments

Benchmark datasets, baseline algorithms, evaluation metrics, and parameter settings are introduced in this section.

4.1. Datasets

For our experiments, three widely used real-world datasets, namely ACM (http://dl.acm.org (accessed on 17 November 2023)), DBLP (https://dblp.uni-trier.de/ (accessed on 17 November 2023)), and IMDB (https://www.imdb.com/ (accessed on 17 November 2023)), are used to evaluate our algorithm. Detailed statistics are presented in Table 1.

Table 1. Statistical information of the experimental datasets.

Datasets	Nodes	Edges	Features	Clusters
ACM	3025	Co-Subject (29,281) Co-Author (2,210,761)	1830	3
DBLP	4057	Co-Author (11,113) Co-Conference (6,776,335) Co-Term (5,000,495)	334	4
IMDB	4780	Co-Actor (98,010) Co-Director (21,018)	1232	3

ACM: This dataset is generated from information about ACM publications. The nodes in this datasets represent the paper. There are two types of relationships between nodes and each type of relationship corresponds to a view. The nodes features are the elements of a bag-of-words of keywords. According to the research field of the paper, it can be divided into three categories.

DBLP: This dataset contains information that describes the author's network. The nodes in this dataset represent the authors. There are three types of relationships between nodes, each relationship corresponding to a view. The nodes features are elements of a set of keyword words. According to the research field of the author, it can be divided into four categories.

IMDB: This dataset contains information on a movie network. The nodes in this dataset represent the movie. There are two types of relationships between movie, each relationship corresponding to a view. The nodes features are elements of a set of keyword words. According to the theme of the movie, it can be divided into three categories.

4.2. Baseline Algorithms and Evaluation Metrics

In the pursuit of validating the effectiveness of the proposed algorithm, we undertake a comparative analysis against eight baseline algorithms. These algorithms include single-view algorithms GAE [25] , large-scale information network embedding (LINE) [26], traditional multi-view algorithms Principled multilayer network embedding (PMNE) [22], RMSC [23], SwMC [20], multi-view graph clustering based on representation O2MAC [17], and multi-view graph clustering based on consensus graph learning MvAGC [15], and multi-view attributed graph clustering(MAGC) [21], and simultaneous linear multi-view attributed graph representation learning and clustering(LMGEC) [24]. A brief introduction to these algorithms is provided below:

(1) **GAE** is a single view algorithm that uses graph autoencoders to generate embedded representations. this algorithm is applied to each graph view and the best results are reported.
(2) **LINE** is a single view graph clustering algorithm applied to large-scale graph data. this algorithm is applied to each graph view and the best results are reported.
(3) **PMNE** projects multi-view graph in to a representative vector space.
(4) **RMSC** is a multi-view clustering algorithm designed to address noise in input data.
(5) **SwMC** implements clustering multi-view data while learning weights of each view.
(6) **O2MAC** learns node embedding by reconstructing entire view with the most information-rich information view.
(7) **MvAGC** is a multi-view graph clustering algorithm that performs graph filtering to achieve multi-view attributed graph clustering.
(8) **MAGC** is a multi-view graph clustering method using node attributes and exploring higher-order graph structure information.
(9) **LMGEC** uses a simple linear model to simultaneously accomplish clustering and representation learning.

To assess the quality of the clustering results, we employ four metrics: clustering accuracy (ACC), normalized mutual information (NMI), adjusted Rand index (ARI), and clustering F1-score (F1).

4.3. Parameter Settings

The hyper-parameters of LGMGC are set as follows: learning rate $lr = 0.001$, maximum number of iterations epochs = 200, number of layers in MLP layers = 1, output dimension of MLP dimension = 512, and number of graph filtering layers $k = 2$. For ACM DBLP and IMDB, the balancing hyper-parameters $\alpha = 10$. For O2MAC, MvAGC and LMGEC, we use original codes. To ensure fairness in comparison, we use the default hyper-parameters settings reported in the original papers. GAE uses same structure as the encoder in O2MAC. Each set of experiments is run ten times, and the average results are reported.

4.4. Experimental Results of Different Algorithms

In this subsection, we present and analyze the results of our experiments.

To evaluate the effectiveness of our LGMGC in multi-view graph clustering tasks, we compare it with nine baseline algorithms on three datasets. Table 2 lists the results. The best results are highlighted in bold. As can be seen, LGMGC performs the best on ACM and DBLP in terms of ACC, NMI, ARI, and F1. For the accuracy (ACC) metrics, the LGMGC model has shown average improvements of 18.59%, 6.74%, and 10.87% on the ACM, DBLP, and IMDB datasets, respectively, compared to existing models. Single-view algorithms such as GAE and LINE do not perform well with multi-view graph data because they cannot leverage information from additional views. LGMGC significantly outperforms other traditional multi-view clustering algorithms, such as PMNE, RMSC, and SwMC. Although these algorithms consider all views, they fail to explore both attribute and topological information. PMNE and SwMC can only explore topological information, whereas RMSC can only leverage attribute information. In contrast, our algorithm effectively uses both topological and attribute information through the graph encoder. Our algorithm performs significantly better than O2MAC across all three datasets. This is because O2MAC solely considers the optimal view and disregards information from the remaining views. Compared with the consensus graph learning algorithms MvAGC, MAGC and LMGEC, the proposed algorithm achieves better performance. This is the result of MvAGC, MAGC and LMGEC relying on learned consensus graphs for clustering, which may lead to the omission of specific information from each view. In particular, the clustering performance of LGMGC on IMDB is inferior to MAGC. The main reason is that each view edge and attribute of IMDB data set are sparse, and the attribute information and topology information in the representation learned by low-pass filter are less.

Table 2. Clustering results of various algorithms on three datasets.

Algorithms		**GAE**	**LINE**	**PMNE**	**RMSC**	**SwMC**	**O2MAC**	**MvAGC**	**MAGC**	**LMGEC**	**LGMGC**
ACM	ACC	0.8216	0.6479	0.6936	0.6315	0.3831	0.9042	0.8975	0.8806	0.9302	**0.9388**
	NMI	0.4914	0.3941	0.4648	0.3973	0.0838	0.6923	0.6735	0.6180	0.7513	**0.7735**
	ARI	0.5444	0.3433	0.4302	0.3312	0.0187	0.7394	0.7212	0.6808	0.8031	**0.8263**
	F1	0.8225	0.6594	0.6955	0.5746	0.4709	0.9053	0.8986	0.8835	0.9311	**0.9382**
DBLP	ACC	0.8859	0.8689	0.7925	0.8994	0.6538	0.9074	0.9277	0.9282	0.9285	**0.9334**
	NMI	0.6825	0.6676	0.5914	0.7111	0.3760	0.7287	0.7727	0.7768	0.7739	**0.7860**
	ARI	0.7410	0.6988	0.5265	0.7647	0.3800	0.7780	0.8276	0.8267	0.8284	**0.8394**
	F1	0.8743	0.8564	0.7966	0.8248	0.5602	0.9013	0.9225	0.9237	0.9241	**0.9289**
IMDB	ACC	0.4298	0.4268	0.4958	0.2702	0.2617	0.4502	0.5633	**0.6125**	0.5893	0.5998
	NMI	0.0402	0.0031	0.0359	0.0054	0.0056	0.0421	0.0317	**0.1167**	0.0632	0.0913
	ARI	0.0473	−0.0090	0.0366	0.0018	0.0004	0.0564	0.0940	**0.1806**	0.1294	0.1710
	F1	0.4062	0.2870	0.3906	0.3775	0.3714	0.1459	0.3783	0.4551	0.4267	**0.4565**

The best results are highlighted in bold.

5. Ablation Study

5.1. Effect of Multi-View Learning

In order to verify the effectiveness of the multi-view learning of LGMGC, The single view of each datasets is entered separately and the results are report in Table 3. It can be seen that the clustering performance of the individual view is always inferior to the consensus. In addition, the clustering performance of different views in the same dataset varies. This validates the effectiveness of multi-view learning in our algorithm.

Table 3. Clustering results of the proposed algorithm on different views.

Datasets		V1	V2	V3	ALL
ACM	ACC	0.9197	0.7230	-	**0.9388**
	NMI	0.7185	0.5155	-	**0.7735**
	ARI	0.7765	0.4719	-	**0.8263**
	F1	0.9198	0.7101		**0.9382**
DBLP	ACC	0.6621	0.6717	0.9247	**0.9334**
	NMI	0.3743	0.3349	0.7782	**0.7860**
	ARI	0.2649	0.3125	0.8329	**0.8394**
	F1	0.6695	0.6724	0.9197	**0.9289**
IMDB	ACC	0.5730	0.5787	-	**0.5998**
	NMI	0.0640	0.0811	-	**0.0913**
	ARI	0.1196	0.1518	-	**0.1710**
	F1	0.4298	0.4504	-	**0.4565**

The best results are highlighted in bold.

5.2. Effect of Reconstruction Loss

By minimizing the reconstruction loss, our algorithm maximizes the preservation of topological information from each view in the fused embedding. In order to verify the effectiveness of the reconstruction loss in LGMGC, the clustering results without the reconstruction loss are reported in Table 4. It can be seen that the addition of reconstruction loss achieves clustering performance improvements. These results verify that LGMGC benefits from the reconstruction loss.

Table 4. Clustering results without reconstruction loss.

Datasets		LGMGC w/o L_{rec}	LGMGC
ACM	ACC	0.9233	**0.9388**
	NMI	0.7349	**0.7735**
	ARI	0.7854	**0.8263**
	F1	0.9241	**0.9382**
DBLP	ACC	0.7678	**0.9334**
	NMI	0.5124	**0.7860**
	ARI	0.5198	**0.8394**
	F1	0.7432	**0.9289**
IMDB	ACC	0.5852	**0.5998**
	NMI	0.0754	**0.0913**
	ARI	0.1389	**0.1710**
	F1	0.4473	**0.4565**

The best results are highlighted in bold.

5.3. Effect of Neighborhood Contrastive Loss

By minimizing the contrastive loss, our algorithm pulls similar nodes closer and pushes dissimilar nodes further, which could highlight the cluster structure in the global representation.we report the clustering performance without the neighborhood contrastive loss in Table 5. On all datasets, the performance drops sharply without the contrastive loss. LGMGC achieves ACC improvements of 2.7%, 1.0%, and 6.53% on the DBLP, ACM, and IMDB datasets, respectively. Regarding the other metrics, the neighborhood contrastive loss significantly enhances the performance. The experimental results in Table 5 provide strong evidence for the effectiveness of the neighborhood contrastive loss in our proposed model.

Table 5. Clustering results without neighborhood contrastive loss.

Datasets		LGMGC w/o L_{nbr}	LGMGC
ACM	ACC	0.9111	**0.9388**
	NMI	0.7070	**0.7735**
	ARI	0.7552	**0.8263**
	F1	0.9123	**0.9382**
DBLP	ACC	0.9232	**0.9334**
	NMI	0.7774	**0.7860**
	ARI	0.8293	**0.8394**
	F1	0.9185	**0.9289**
IMDB	ACC	0.5345	**0.5998**
	NMI	0.0044	**0.0913**
	ARI	0.0188	**0.1710**
	F1	0.2916	**0.4565**

The best results are highlighted in bold.

5.4. Effect of Graph Encoder

The graph encoder proposed in this paper enhances the low-frequency signals of the local embedding representations in each view, and alleviates the problem of over-smoothing. To validate the effectiveness of the graph encoder, we replaced the original encoder with a GCN and compared the performance of the two models while varying the number of network layers. To ensure a fair comparison, the GCN was applied with the same parameters as the encoder and did not use any activation function. Figure 2 shows the clustering performance of the two models for different numbers of network layers. In most cases, the proposed graph encoder outperforms the GCN. Additionally, as the number of layers increases, the performance of the GCN decreases, while that of the graph encoder remains relatively stable. The reason for this phenomenon may be the trouble caused by redundant weight matrices in GCN.

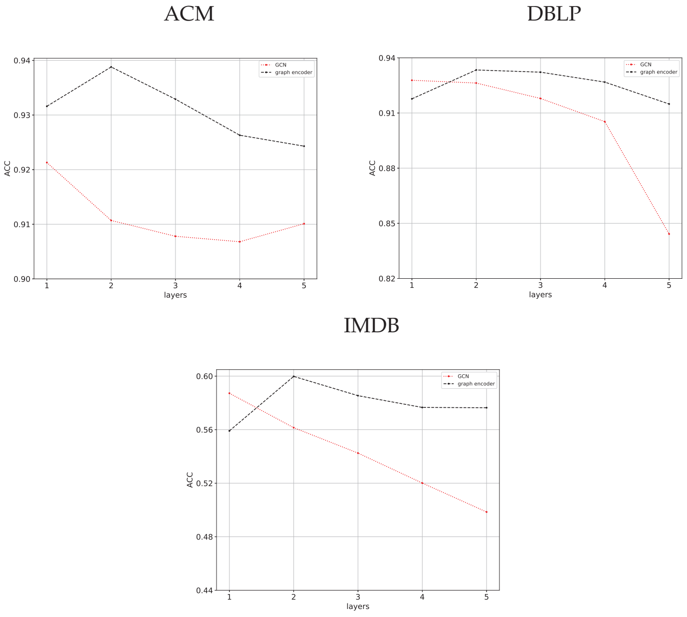


Figure 2. Clustering results of the GCN and the graph encoder on three datasets with different number of network layers.

5.5. Parameter Analysis

We briefly analyze the impact of parameters k and α on the clustering performance to evaluate the stability of LGMGC. The clustering results using parameter settings of $k = \{1, 2, 3, 4, 5\}$ and $\alpha = \{0.1, 1, 10, 100, 1000\}$ are shown in Figure 3. It can be observed that with a fixed k, the algorithm achieves great clustering performance on all three datasets when the balance parameter is set to 10. However, when α is too large, there is a significant decline in clustering performance on the dense DBLP dataset. On the other hand, when the balance parameter is too small, there is a noticeable decrease in clustering performance on the sparse IMDB dataset. When the balance parameter α is fixed, both excessively small and large numbers of filter layers lead to unfavorable clustering results. The reason for this phenomenon is that the small k leads encoder fails to capture higher-order topological information, while an excessive number of filter layers leads to over-smoothing, rendering the nodes indistinguishable.

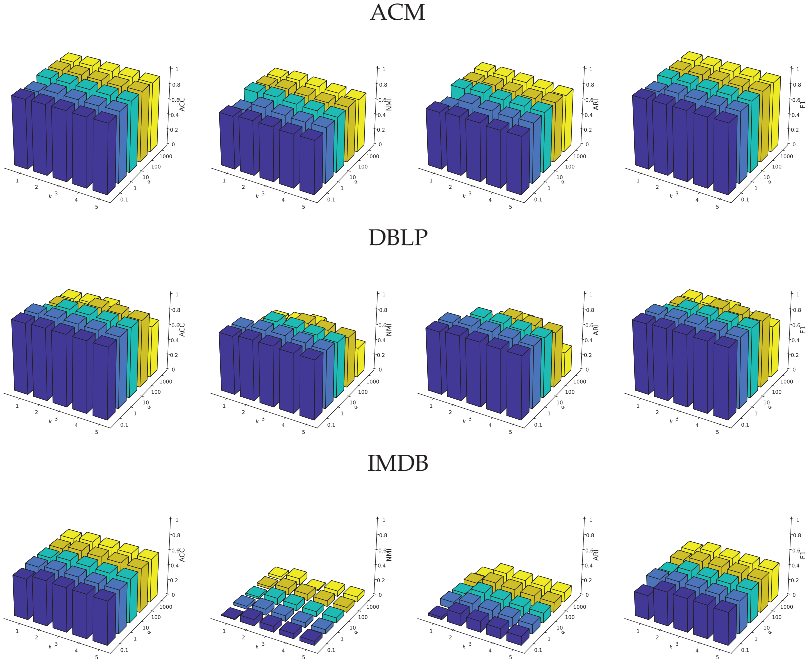


Figure 3. Sensitivity analysis of parameters k and α on three datasets.

6. Discussion

We have developed a method for multi-view graph clustering that enhances the clustering performance by improving both local and global representations. In contrast to previous approaches that utilized traditional GCN with a single graph filter, our algorithm introduces a new encoder. This encoder is a combination of graph filters and MLP, which enables the amplification of local representations while extracting deeper-level information between data points during the construction of local representations. Additionally, a joint loss function was designed to enhance the clustering quality of global representations. Experimental results demonstrate that compared to existing algorithms, our model achieves superior performance in multi-view graph clustering tasks. This indicates the meaningful improvement in clustering effectiveness through the enhancement of both local and global representations. The proposed algorithm plays a crucial role in exploring community structures in multi-view graph data. This research can aid in capturing the complex relationships within and between communities in multi-view graph data from various perspectives. A single data source might not fully reveal the community structure, while combining multiple data sources can provide a more comprehensive and accurate description of the communities. In the analysis of social networks, companies can take into account multiple types of relationships between users to obtain more accurate and comprehensive community structures. For example, in the analysis of social networks, companies can comprehensively consider multiple types of relationships between users to uncover more accurate and comprehensive community structures. Furthermore, during the analysis of citation networks, it becomes possible to better identify relevant patterns and structures that may have been overlooked or difficult to perceive. This contributes to the elucidation of potential research topics, scholarly communities, and other aspects within a specific field. During the analysis of citation networks, it allows for a better identification of potentially overlooked or hard-to-perceive patterns and structures. This enhanced recognition enables researchers to uncover relevant relationships and structures that may have gone unnoticed using traditional analysis methods.

7. Conclusions and Future Work

In this paper, we propose a multi-view graph clustering model, which optimizes the representation from local and global perspectives within a unified framework to develop a more clustering-suitable representation. The proposed algorithm's effectiveness is validated through experiments on real-world multi-view graph data. A potential limitation is the high computational cost on large-scale nodes. One of our hypotheses is that in the contrast loss function, we can greatly reduce the time complexity by selecting representative anchor points instead of all nodes to participate in the training. The methods of choosing a representative anchor point will become the focus of our future work.

Author Contributions: Conceptualization, X.Z.; methodology, X.Z. and Z.H.; software, Z.H.; investigation, J.W.; writing—original draft preparation, writing—review and editing, X.Z.; supervision, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Nos. 62072293 and 62306205), and the 1331 Engineering Project of Shanxi Province, China.

Data Availability Statement: The datasets used for this work can be found ACM (http://dl.acm.org (accessed on 17 November 2023)), DBLP (https://dblp.uni-trier.de/ (accessed on 17 November 2023)), and IMDB (https://www.imdb.com/ (accessed on 17 November 2023)).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, H.; Wang, J.; Duan, R.; Yan, C. DCOM-GNN: A Deep Clustering Optimization Method for Graph Neural Networks. *Knowl.-Based Syst.* **2023**, *279*, 110961. [CrossRef]
2. Hu, D.; Feng, D.; Xie, Y. EGC: A novel event-oriented graph clustering framework for social media text. *Inf. Process. Manag.* **2022**, *59*, 103059. [CrossRef]
3. Yu, J.; Jia, A.L. MLGAL: Multi-level Label Graph Adaptive Learning for node clustering in the attributed graph. *Knowl.-Based Syst.* **2023**, *278*, 110876. [CrossRef]
4. Guo, Y.; Zang, Z.; Gao, H.; Xu, X.; Wang, R.; Liu, L.; Li, J. Unsupervised social event detection via hybrid graph contrastive learning and reinforced incremental clustering. *Knowl.-Based Syst.* **2024**, *284*, 111225. [CrossRef]
5. Guo, H.; Zhou, Z.; Zhao, D.; Gaaloul, W. EGNN: Energy-efficient anomaly detection for IoT multivariate time series data using graph neural network. *Future Gener. Comput. Syst.* **2024**, *151*, 45–56. [CrossRef]
6. Zhao, S.; Zheng, Y.; Li, J.; Zhang, X.; Tang, C.; Tan, Z. Pure kernel graph fusion tensor subspace clustering under non-negative matrix factorization framework. *Inf. Process. Manag.* **2024**, *61*, 103603. [CrossRef]
7. Li, X.; Sun, L.; Ling, M.; Peng, Y. A survey of graph neural network based recommendation in social networks. *Neural Comput.* **2023**, *549*, 126441. [CrossRef]
8. Mohamed, H.A.; Pilutti, D.; James, S.; Del Bue, A.; Pelillo, M.; Vascon, S. Locality-aware subgraphs for inductive link prediction in knowledge graphs. *Pattern Recognit. Lett* **2023**, *167*, 90–97. [CrossRef]
9. Liao, H.; Hu, J.; Li, T.; Du, S.; Peng, B. Deep linear graph attention model for attributed graph clustering. *Knowl.-Based Syst.* **2022**, *246*, 108665. [CrossRef]

10. Berahmand, K.; Nasiri, E.; Pir mohammadiani, R.; Li, Y. Spectral clustering on protein-protein interaction networks via constructing affinity matrix using attributed graph embedding. *Comput. Biol. Med.* **2021**, *138*, 104933. [CrossRef] [PubMed]
11. Xia, Y.; Xu, Q.; Fang, J.; Tang, R.; Du, P. Bipartite graph-based community-to-community matching in local energy market considering socially networked prosumers. *Appl. Energy* **2024**, *353*, 122245. [CrossRef]
12. Hu, Z.; Deng, J.; Han, J.; Yuan, K. Review on application of graph neural network in traffic prediction. *J. Traffic Transp. Eng.* **2023**, *23*, 39–61.
13. Liu, L.; Kang, Z.; Ruan, J.; He, X. Multilayer graph contrastive clustering network. *Inf. Sci* **2022**, *613*, 256–267. [CrossRef]
14. Wang, R.; Li, L.; Tao, X.; Wang, P.; Liu, P. Contrastive and attentive graph learning for multi-view clustering. *Inf. Process. Manag.* **2022**, *59*, 102967. [CrossRef]
15. Lin, Z.; Kang, Z. Graph Filter-based Multi-view Attributed Graph Clustering. In Proceedings of the IJCAI, Virtual, 19–27 August 2021; pp. 2723–2729.
16. Pan, E.; Kang, Z. Multi-view contrastive graph clustering. In Proceedings of the NeurIPS, Virtual, 6–14 December 2021; pp. 2148–2159.
17. Fan, S.; Wang, X.; Shi, C.; Lu, E.; Lin, K.; Wang, B. One2Multi graph autoencoder for multi-view graph clustering. In Proceedings of the WWW'20, Taipei, China, 20–24 April 2020; pp. 3070–3076.
18. Cai, E.; Huang, J.; Huang, B.; Xu, S.; Zhu, J. Grae: Graph recurrent autoencoder for multi-view graph clustering. In Proceedings of the ACAI, Sanya, China, 22–24 December 2021; pp. 1–9.
19. Liang, J.; Liu, X.; Bai, L.; Cao, F.; Wang, D. Incomplete multi-view clustering via local and global co-regularization. *Sci. China Inf. Sci.* **2022**, *65*, 152105. [CrossRef]
20. Nie, F.; Li, J.; Li, X. Self-weighted multiview clustering with multiple graphs. In Proceedings of the 26th IJCAI, Melbourne, Australia, 19–25 August 2017; pp. 2564–2570.
21. Lin, Z.; Kang, Z.; Zhang, L.; Tian, L. Multi-view Attributed Graph Clustering. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 1872–1880. [CrossRef]
22. Liu, W.; Chen, P.; Yeung, S. Principled Multilayer Network Embedding. In Proceedings of the ICDM, New Orleans, LA, USA, 18–21 November 2017; pp. 134–141.
23. Xia, R.; Pan, Y.; Du, L. Robust Multi-view Spectral Clustering via Low-rank and Sparse Decomposition. In Proceedings of the AAAI, Québec City, QC, Canada, 27–31 July 2014; pp. 2149–2155.
24. Fettal, C.; Labiod, L.; Nadif, M. Simultaneous Linear Multi-View Attributed Graph Representation Learning and Clustering. In Proceedings of the WSDM, Singapore, 27 February–3 March 2023; pp. 303–311.
25. Kipf, T.; Welling, M. Variational Graph Auto-Encoders. *arXiv* **2016**, arXiv:1611.07308.
26. Tang, J.; Qu, M.; Wang, M.; Zhang, Y.; Yan, J.; Mei, Q. LINE: Large-scale Information Network Embedding. In Proceedings of the ICONIP, San Diego, CA, USA, 7–9 May 2015; pp. 1067–1077.
27. Xia, W.; Wang, S.; Yang, M.; Gao, Q.; Han, J.; Gao, X. Multi-view graph embedding clustering network: Joint self-supervision and block diagonal representation. *Neural Netw.* **2022**, *145*, 1–9. [CrossRef] [PubMed]
28. Cheng, J.; Wang, Q.; Tao, Z.; Xie, D.; Gao, Q. Multi-view attribute graph convolution networks for clustering. In Proceedings of the IJCAI, Virtual, 19–27 August 2021; pp. 2973–2979.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

A Novel Framework for Risk Warning That Utilizes an Improved Generative Adversarial Network and Categorical Boosting

Yan Peng, Yue Liu, Jie Wang and Xiao Li *

School of Management, Capital Normal University, Beijing 100056, China; pengyan@cnu.edu.cn (Y.P.); 2222902038@cnu.edu.cn (Y.L.); wangjie@cnu.edu.cn (J.W.)
* Correspondence: lixiao@cnu.edu.cn; Tel.: +86-132-6165-0877

Abstract: To address the problems of inadequate training and low precision in prediction models with small-sample-size and incomplete data, a novel SALGAN-CatBoost-SSAGA framework is proposed in this paper. We utilize the standard K-nearest neighbor algorithm to interpolate missing values in incomplete data, and employ EllipticEnvelope to identify outliers. SALGAN, a generative adversarial network with a self-attention mechanism of label awareness, is utilized to generate virtual samples and increase the diversity of the training data for model training. To avoid local optima and improve the accuracy and stability of the standard CatBoost prediction model, an improved Sparrow Search Algorithm (SSA)–Genetic Algorithm (GA) combination is adopted to construct an effective CatBoost-SSAGA model for risk warning, in which the SSAGA is used for the global parameter optimization of CatBoost. A UCI heart disease dataset is used for heart disease risk prediction. The experimental results show the superiority of the proposed model in terms of accuracy, precision, recall, and F1-values, as well as the AUC.

Keywords: small-sample datasets; data augmentation; improved sparrow search algorithm; novel risk warning; GAN

Citation: Peng, Y.; Liu, Y.; Wang, J.; Li, X. A Novel Framework for Risk Warning That Utilizes an Improved Generative Adversarial Network and Categorical Boosting. *Electronics* **2024**, *13*, 1538. https://doi.org/10.3390/electronics13081538

Academic Editor: Ping-Feng Pai

Received: 2 February 2024
Revised: 29 March 2024
Accepted: 10 April 2024
Published: 18 April 2024

1. Introduction

In reality, many application scenarios contain very few labeled samples, and also many datasets are incomplete with missing information. For example, in the field of medical diagnoses, doctors may only obtain data from a few patients, which is particularly common in the diagnoses of rare or emerging diseases. In the financial field, especially in personal credit scoring or fraud detection, it is necessary to use limited-sample data to predict credit risks or identify fraudulent activity. Therefore, developing methods for learning from small and incomplete samples is an urgent need. Model fine-tuning, data augmentation, and transfer learning are the mainstream technologies used to solve such problems [1]. Meanwhile, a significant body of research has demonstrated the potential of machine learning models in risk prediction on small-sample datasets. However, several challenges still need to be addressed.

Fine-tuning and transfer learning methods have problems with model overfitting when the target dataset and the source dataset are dissimilar. Data augmentation approaches may introduce noise or alter features. The learning ability of a single machine learning model varies across different datasets, resulting in inconsistent prediction performance and limited generalization capability [2]. Although deep learning models can achieve promising prediction accuracy, they require a significant amount of data and a complex training process, and are prone to issues such as gradient vanishing or exploding and poor interpretability [3]. On the other hand, ensemble learning has achieved good results in multi-class prediction tasks, and the appropriate selection of hyperparameter tuning methods can improve the prediction accuracy of the model. However, there is still room for system optimization.

Thus, in this paper, we analyze small-size data and construct a risk-warning framework called SALGAN-CatBoost-SSAGA, which consists of two main parts: SALGAN data augmentation and an improved CatBoost prediction model (i.e., CatBoost-SSAGA). The method of the label-aware self-attention mechanism-based generative adversarial network (SALGAN) learns the various implicit correlations and dependencies of different types of labeled data, contributing to the generation of highly realistic data instances. The prediction model is constructed by CatBoost, which is integrated with an enhanced SSAGA for global parameter optimization. We conduct experiments on a small-sample dataset of heart disease, which demonstrates the effectiveness of the proposed model.

The main contributions of this paper are as follows:

(1) We introduce a novel SALGAN-CatBoost-SSAGA framework for small-sample risk warning.
(2) We propose a SALGAN that generates virtual data according to label types, effectively enhancing small-sample data.
(3) We present a hybrid algorithm, the SSAGA, which combines the SSA and GA to optimize the parameters of the standard CatBoost model, which could improve the prediction accuracy of the CatBoost model.
(4) We conduct small-sample prediction experiments using the UCI heart disease dataset, which demonstrates the advantages of the proposed model in terms of its classification accuracy, recall, precision, F1-score, and AUC, indicating its effectiveness in predicting small-sample data.

2. Related Work

2.1. Few-Shot Learning

Few-shot learning [4] aims to construct machine learning models that can solve real-world problems using a limited amount of training data. In few-shot learning, there are typically two main challenges: inter-class variance and intra-class variance [5]. Currently, few-shot learning primarily includes methods based on model fine-tuning, transfer learning, and data augmentation.

The method of model fine-tuning [6] usually involves pretraining the network model on a large dataset, then fixing some parameters and fine-tuning specific parameters of the network model on a small-sample dataset to obtain a fine-tuned model. Transfer learning [7] helps to train reliable decision functions in the target domain by transferring knowledge from auxiliary sources. This approach addresses the learning problem when the sample data in the target domain are either unlabeled or consist of only a limited number of labeled samples. However, since the target sample set and the source sample set may not be similar, the two methods may lead to overfitting problems of the machine learning model on the target sample set.

Data augmentation [8] includes methods based on unlabeled data, feature enhancement and data synthesis. Methods based on unlabeled data involve using large amounts of unlabeled data to expand the original small-sample dataset, such as semi-supervised learning [9] and transductive learning [10]. Feature enhancement involves adding features in the feature space of the original sample to increase the diversity of features for classification. Schwartz et al. [11] proposed the Delta encoder, which synthesizes new samples for unseen categories by observing a small number of samples and uses these synthetic samples to train a classifier. Data synthesis refers to the synthesis of new labeled data for small-sample categories to augment the training data, and a commonly used method for this is the use of generative adversarial networks (GANs) [12]. A GAN is a deep learning model proposed by Ian J. Goodfellow and his colleagues in 2014. It consists of two mutually competitive networks: a generator G and a discriminator D. Y Kataoka et al. [13] reported image generation that leverages the effectiveness of attention mechanisms and the GAN approach. N

Park et al. [14] proposed table-GAN, which uses GANs to synthesize fake tables that are statistically similar to the original tables but do not cause information leakage.

2.2. CatBoost Algorithm

Categorical Boosting (CatBoost) [15] is an enhanced framework of Gradient Boosted Decision Tree (GBDT), which is a commonly used classification algorithm. It is based on a symmetric decision tree as the base learner, and effectively suppresses the gradient bias and prediction bias existing in the gradient decision tree by introducing a rank-boosting strategy. Moreover, CatBoost is characterized by its robust support for categorical variables and exceptional predictive accuracy. Li et al. [16] developed a weather prediction model combining wavelet denoising and CatBoost, which is faster and more accurate than LSTM and Random Forest. PS Kumar et al. [17] developed a CatBoost ensemble technique based on GBDT, specifically for the prediction of early-stage diabetes. Comparative experiments with other machine learning methods have demonstrated that CatBoost excels in various performance metrics. Wang et al. [18] investigated the efficacy of CatBoost in predicting severe hand-foot-and-mouth disease, finding it to have an accuracy rate of 87.6%, higher than other algorithms.

However, despite CatBoost's effectiveness, its extensive hyperparameter space can significantly impact classification results. Therefore, it is crucial to employ parameter optimization algorithms to fine-tune the hyperparameters, enhancing CatBoost's full potential in diverse applications. Cheng et al. [19] used the grid search method with cross-validation to optimize the super parameters of catboost, respectively, and the model showed the highest accuracy in a suspended solids experiment. Jin et al. [20] trained CatBoost, Random Forest, and other models through cyclic training and adjusting the given parameters, and then used the cross-validation method to conduct a grid search for secondary adjustments. Their experimental results show that the prediction effect of CatBoost after two rounds of optimization was significantly higher than that of other models.

2.3. Hyperparameter Optimization Algorithm

The hyperparameters should be determined before the model runs, and they have a relatively important impact on the performance of the model. Currently, there are many optimization methods available, such as the grid search (GS) method [21] and the Bayesian optimization algorithm (BOA) [22]. Some studies also employ swarm intelligence optimization algorithms such as Grey Wolf Optimization (GWO) [23], the Genetic Algorithm (GA) [24], and the Sparrow Search Algorithm (SSA) [25]. However, there are still some deficiencies in these parameter optimization methods, such as the fact that cross-validation and grid search methods do not consider super parameters or only consider a small number of common super parameters; the BOA and GWO do not grasp the global trend of the prediction performance of the model, and are prone to falling into local optimization; the results of the GA are affected by the initial advantages and disadvantages, and cannot eliminate the randomness of the optimization results.

Among them, the SSA is a preferable choice due to the advantages of its simple structure and flexibility, but its optimization ability and convergence speed still need to be improved [26]. Therefore, many studies have focused on optimizing the SSA. Ou et al. [27] improved the SSA by using the good point set method and reducing nonlinear inertia weights to prevent the SSA from falling into local optima. Wang et al. [28] employed a multi-sample learning strategy to assist the SSA in achieving a better optimization capability and convergence speed.

3. Preliminaries

3.1. CatBoost

In the GBDT algorithm, a commonly used method for dealing with categorical features is to replace them with the average value of the category feature label, which can be expressed as Equation (1):

$$\overset{\wedge}{\mathrm{x}}{}_k^i = \frac{\sum\limits_{j=1}^{p-1} [x_{j,k} = x_{i,k}] \bullet Y_i}{\sum\limits_{j=1}^{n} [x_{j,k} = x_{i,k}]} \tag{1}$$

where, x_k^i represents the *i*th category feature value of the *k*th training sample. If a feature has fewer category values, converting it to a numerical value is equivalent to assigning the label value of that record. This scenario commonly leads to overfitting issues.

In response to this, the CatBoost algorithm addresses a specific value within the categorical features. When converting each feature to a numerical type for each sample, the algorithm calculates the average based on the category label preceding the sample, incorporating prior knowledge and weight coefficients. This approach aims to reduce the noise caused by low-frequency features in the categorical features, as shown in Equation (2):

$$\hat{x}{}_k^i = \frac{\sum\limits_{j=1}^{p-1} [x_{\varepsilon_{j,k}} = x_{\varepsilon_{j,k}}] Y_{\varepsilon_j} + \mathrm{ap}}{\sum\limits_{j=1}^{p-1} [x_{\varepsilon_{j,k}} = x_{\varepsilon_{p,k}}] + \mathrm{a}} \tag{2}$$

where, $\hat{x}{}_k^i$ represents the statistical target variable, $x_{\varepsilon_{j,k}}$ denotes the categorical feature, Y_{ε_j} corresponds to the label value of the feature, a denotes the weight coefficient, and p represents the prior term.

3.2. SSA

The SSA is a swarm intelligence optimization algorithm that simulates the behavior of sparrows foraging and avoiding predators. In this algorithm, the sparrow population is divided into two categories: discoverers and followers.

Discoverers are responsible for searching for food and providing information about foraging areas to the entire population. The position of the discoverer is updated as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp(\frac{i}{\alpha \cdot iter_{\max}}), R < S \\ X_{i,j}^t + Q \cdot L, R \geq S \end{cases} \tag{3}$$

where $X_{i,j}^t$ represents the position of the *i*th sparrow in dimension *j* at iteration *t*; α is a random number in the range of (0, 1]; $iter_{\max}$ is the maximum number of iterations, a constant value; $\exp(x)$ denotes the exponential function with base e; and Q is a random number following a normal distribution. R is the alert value, and if R is smaller than the safety value S, it indicates that the sparrow's environment is relatively safe, allowing for extensive foraging exploration. Conversely, if R is larger than S, it indicates that some individuals have detected predators and issued an alarm to move towards a safe zone, ensuring the safety of the population.

The other individuals in the population are followers, who come to forage based on the information provided by the discoverers. The update of their positions is expressed as Equation (4):

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp(\frac{x_w - x_{i,j}^t}{\alpha \cdot iter_{\max}}), i \geq \frac{n}{2} \\ X_b^{t+1} + \left| X_{i,j}^t - X_b^{t+1} \right| \cdot A^T (AA^T)^{-1} \cdot L, otherwise \end{cases} \tag{4}$$

where X_P represents the current optimal position of the discoverer; A is a 1 × d matrix with elements randomly assigned as 1 or −1. The variable i in $i \geq \frac{n}{2}$ represents the quantity of followers within the population, with n indicating the population's size. This condition is commonly referred to as the "hunger judgment" and is employed to assess whether an individual necessitates foraging.

In addition, to effectively avoid predator attacks, the algorithm also introduces an early warning mechanism that selects a certain proportion of individuals as scouts, who are responsible for detecting and warning of potential threats. The positions of the scouts are updated using Equation (5):

$$X_{i,j}^{t+1} = \begin{cases} X_b^t + \beta \left| X_{i,j}^t - X_b^t \right|, f_i > f_g \\ X_{i,j}^t + k\left(\frac{\left|X_{i,j}^t - X_w^t\right|}{(f_i - f_w) + \gamma}\right), f_i = f_g \end{cases} \tag{5}$$

where β denotes the learning rate, which is a normally distributed random number that controls the speed at which the model updates parameters during each iteration. The random variable k is a value within the interval $[-1, 1]$ used to control the direction of sparrow movement and is a small constant employed to prevent division by zero. Finally, f_i represents the value of the objective function at the current position.

4. Model Construction

4.1. Framework

The overall framework structure of the SALGAN-CatBoost-SSAGA framework is shown in Figure 1, and mainly consists of four parts: data cleaning, data augmentation, risk-warning prediction, and model evaluation.

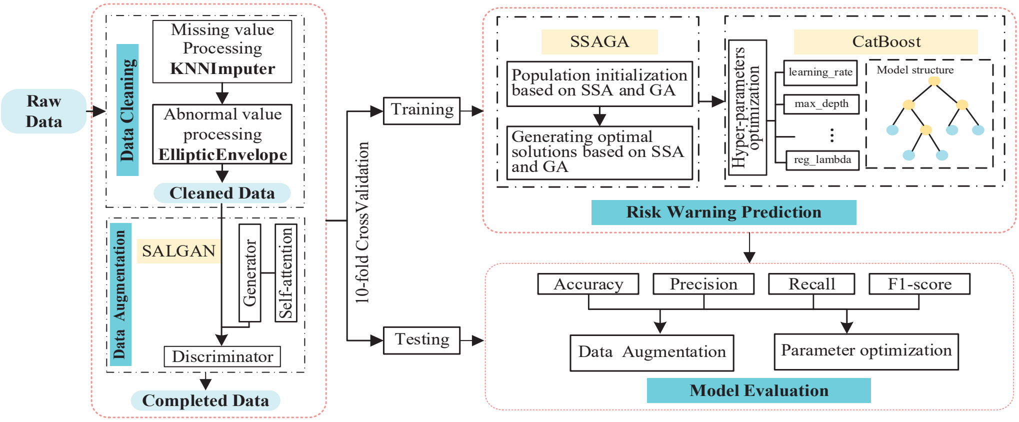


Figure 1. Framework of SALGAN-CatBoost-SSAGA.

The workflow of the SALGAN-CatBoost-SSAGA model is shown as follows:

(1) Data cleaning

For feature incompleteness, the KNNImputer algorithm is utilized to interpolate missing data from the original datasets, while the EllipticEnvelope algorithm is applied to remove outliers, thus completing the datasets.

(2) Data augmentation

In order to address the challenges associated with small-sample learning, this study implements a generative adversarial network that is utilized with a label-aware self-attention mechanism (SALGAN), aiming to generate high-quality synthetic sample data through this methodology.

(3) Risk-warning prediction

We improve the SSA using the GA and apply the improved SSAGA to optimize the hyperparameters of CatBoost, including n_estimators, learning_rate, max_depth, reg_lambda, and subsample. We then train the classifier.

(4) Model evaluation

We propose to test the performance of the evaluation indexes of the new framework, using this as the basis for the evaluation of the SALGAN-CatBoost-SSAGA risk-warning model.

SALGAN-CatBoost-SSAGA consists of two main parts: the SALGAN data augmentation method and the CatBoost-SSAGA prediction model. It is primarily used to address prediction problems in small-sized datasets. Firstly, for data augmentation, the SALGAN combines self-attention with a GAN, and, more importantly, introduces label awareness. The SALGAN not only autonomously adapts to various data features and relationships but also enhances the generator's ability to effectively grasp and replicate data distributions across different labels, thereby generating more realistic and contextually relevant synthetic data samples. By using the SALGAN to generate virtual samples, the diversity of the training data for model training is increased, thus avoiding insufficient learning in the subsequent models. Secondly, to enhance the prediction accuracy of CatBoost, we opted to optimize it using the SSA. However, considering that the SSA may suffer from issues such as a poor quality of randomly generated initial populations and being prone to local optima, we chose to use the GA to further optimize the SSA. By incorporating the GA into the process, a hybrid optimization approach called the SSAGA was formed. The SSAGA not only enhances the global search capability and accelerates the convergence speed, but also increases the possibility of finding the global optimal solution.

4.2. Data Cleaning

Two key steps for data cleaning are employed in this study, aimed at enhancing the completeness and accuracy of the data, thereby ensuring high reliability in subsequent analyses.

(1) Missing value interpolation

We utilize the KNNImputer algorithm, which has demonstrated strong performance in multiple studies, to estimate the values of missing data. The core of this algorithm is to extract the k-closest samples in the dataset, and then use the distribution of these samples to fill in the missing data values. If the missing values are discrete, the plurality of the k-nearest neighbors will be voted to fill them; otherwise, the average of the nearest neighbors will be used to fill them.

(2) Outlier removal

The EllipticEnvelope algorithm is utilized to identify outliers. This algorithm assumes that the normal sample data conform to a multivariate Gaussian distribution, while the abnormal sample data do not follow this distribution. Its objective is to find the smallest ellipse that can cover the majority of the samples and consider the points outside of the ellipse as outliers.

4.3. Data Augmentation Based on the SALGAN

GANs are capable of capturing and learning the complex distribution characteristics of data, including various implicit correlations and dependencies, which aids in generating highly realistic data instances. However, achieving a balance in the learning process between the generator and discriminator can be challenging, leading to model instability and convergence difficulties.

Self-attention mechanisms enhance GANs by focusing on different segments of the data, aiding GANs to better grasp global structures. This results in capturing finer details and patterns in data generation, reducing model collapse issues. The model adapts autonomously based on varying features and relationships within the data.

Considering the distinct characteristics of differently labeled samples, modeling the data distribution for various labels allows the generator to better capture and reflect each label's unique features. This label-aware synthetic data generation approach facilitates the generator's more effective learning and mimicking of data distributions under different labels, yielding more realistic and contextually accurate synthetic samples.

We introduce the SALGAN for data augmentation. Compared to a traditional GAN, the SALGAN not only adapts autonomously to diverse data features and relationships but also enables the generator to more effectively learn and mimic data distributions across different labels, creating more realistic and context-relevant synthetic data samples. The experimental process of the SALGAN is illustrated in Figure 2, wherein the input is an N × M matrix, and the output is an (N + T) × M matrix; N is the number of original data items; M is the number of data items; and T is the number of generated virtual sample data items.

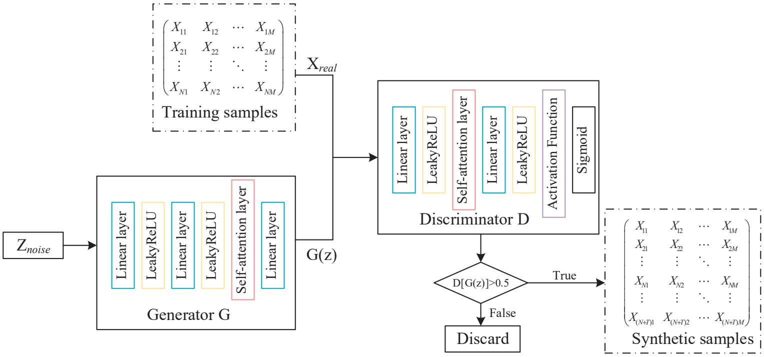


Figure 2. Flowchart of the SALGAN's process.

Data are classified by label types and generated in batches. Random noise Z_{noise}, which is a vector randomly generated from a normal distribution, serves as the input for the generator network. The generator uses Z_{noise} to generate a set of synthetic data. The discriminator receives the real data and fake data generated by the generator, and its task is to distinguish the two sets of data and output a probability value indicating the possibility that the data are real. The results of the loss function calculation are used to correct the back-propagation error and refine the parameters of the two networks. This iterative process continues until the generator and the discriminator reach a balanced state. The generator can create enough convincing data to copy the discriminator, and the discriminator is good at accurately distinguishing real data from false data.

1. Generator G

(1) Fully Connected Layer: We receive the input data and map them to a higher-dimensional hidden layer space, providing the basis for subsequent processing. The first linear layer maps the input size to the hidden layer size, and the second linear layer maps the hidden layer size back onto itself. Each linear layer is followed by the LeakyReLU Activation Function. This configuration introduces nonlinearity, enabling the model to capture more complex data patterns.

(2) Self-Attention Layer: The self-attention layer captures relationships among the input data, as illustrated in Figure 3.

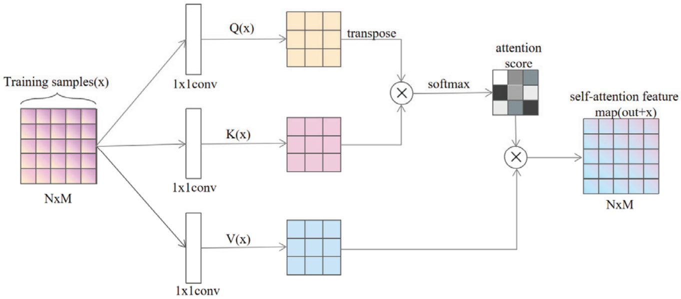


Figure 3. Flowchart of the self-attention layer's process.

First, for the input feature matrix $x \in R^{N*M}$, we perform linear transformation to generate query (Q), key (K), and value (V) matrices:

$$\begin{aligned} Q &= XW^{Q} \\ K &= XW^{K} \\ V &= XW^{V} \end{aligned} \tag{6}$$

where $W^{Q} \in R^{M*d_k}$, $W^{K} \in R^{M*d_k}$, and $W^{V} \in R^{M*d_k}$ are the learnable weight matrices, and d_k and d_v represent the dimensions of the keys and values, respectively.

Second, attention scores are calculated, as shown in Equation (7), by performing matrix multiplication between the transpose of $Q(x)$ and $K(x)$ to calculate attention scores.

$$\text{Attention Score} = QK^{T} \tag{7}$$

Third, weighted value vectors are calculated. Using the softmax function, the attention scores are normalized, as shown in Equation (8), and then the softmaxed attention scores are multiplied by $V(x)$ according to Equation (9) to obtain the weighted values, which represent the comprehensive information of all input features weighted by their corresponding attention scores.

$$\text{Softmax Score} = \text{softmax}\left(\frac{\text{Attention Score}}{\sqrt{d_k}}\right) \tag{8}$$

$$\text{Weighted Value} = \text{Softmax Score} \times V \tag{9}$$

(3) Output Layer: We map the high-dimensional representation of the hidden layer to the output layer. This consists of a linear layer and the Tanh Activation Function for generating the final generated data.

The generated data samples are compared with real data samples and adjust their own weights according to the output of the discriminator. During training, the constant confrontation between the generator and the discriminator enables the generator to produce progressively more realistic and higher-quality data samples.

2. Discriminator D

The discriminator analyzes the input data and effectively distinguishes between the real data X_{real} and generated data G(Z). The output of the discriminator is used for self-adjustment and is fed back to the generator to guide improvements in the generation process. Binary cross-entropy loss (BCELoss) is used as the main loss function of the discriminator, as shown in Equation (10). This loss function makes the discriminator judgement more accurate and generates data closer to the actual data by minimizing the binary cross-entropy loss.

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} y_i \bullet \log(p(y_i)) + (1 - y_i) \bullet \log(1 - p(y_i)) \tag{10}$$

where y represents a binary label, either 0 or 1, and $p(y)$ denotes the probability of the output belonging to a given label. N signifies the number of groups of objects for which the model makes predictions.

4.4. Risk-Warning Model Based on CatBoost-SSAGA

CatBoost is an algorithm that supports various categorical variables, effectively reduces the prediction bias in GBDT, thus reducing the risk of overfitting, and has a high accuracy. However, the processing of categorical features takes a long time and CatBoost has many hyperparameters that need to be tuned. Therefore, this study combines the SSA and GA to construct a hybrid algorithm, named the SSAGA, to optimize the parameters of CatBoost.

4.4.1. SSAGA

The initial quality of the population randomly generated by the SSA is poor. It is easy for the SSA to fall into local optima in large or complex optimization spaces, sometimes even jumping out of the optimization space.

The GA introduces new solutions through crossover and mutation operations, which is conducive to increasing the diversity of the population. It aids in carrying out local searches and evolutionary optimization, which can further refine the solution in the found good region. In addition, the GA is more stable and able to optimize in a complex optimization space.

By combining the SSA and GA, this paper proposes the hybrid SSAGA, which could enhance global search capabilities, accelerate convergence speed, and increase the likelihood of finding global optima. This methodology aims to effectively obtain the optimal hyperparameters for the CatBoost model. The specific workflow of the model is depicted in Figure 4.

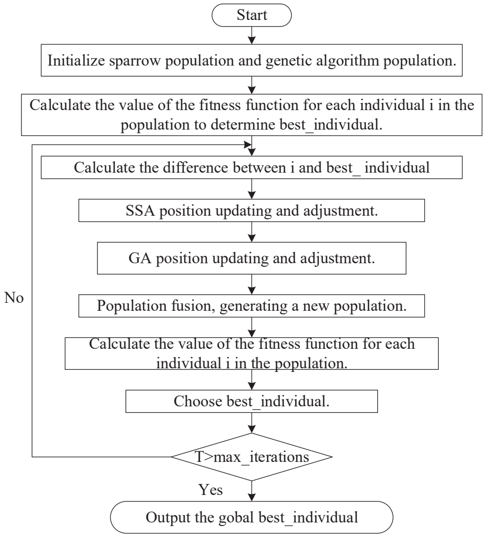

Start
Initialize sparrow population and genetic algorithm population.
Calculate the value of the fitness function for each individual i in the population to determine best_individual.
Calculate the difference between i and best_ individual
SSA position updating and adjustment.
GA position updating and adjustment.
Population fusion, generating a new population.
Calculate the value of the fitness function for each individual i in the population.
Choose best_individual.
No
T>max_iterations
Yes
Output the gobal best_individual

Figure 4. Flowchart of the SSAGA.

The optimization process of the SSAGA is as follows:

Step 1 Initialization: The population size (pop_size), maximum iteration number (max_iterations), and parameter dimension (dimension) are set. The Sparrow Search Algorithm population and Genetic Algorithm population are initialized, and random parameter vectors are generated.

Step 2 Iterative optimization: Within the specified maximum iteration number, the algorithm alternately executes the following steps:

a. Sparrow Search Algorithm Phase: Initially, each sparrow's performance in the population is evaluated using a fitness function, identifying the current optimal- and least-fit individuals. Subsequently, the location of the sparrow population is updated. In each iteration, sparrows adjust their positions based on the current best (best_sparrow) and worst (worst_sparrow) locations. This updating mechanism mimics sparrows' foraging behavior, where some sparrows converge towards the best solution (leader sparrows and followers), while others explore in the opposite direction of the worst solution (scouts). The updated parameter values are constrained within their valid range.

b. Genetic Algorithm Phase: Parental selection is conducted using the select_parents() function, employing a roulette-wheel selection method based on the fitness function, with the selection probability being proportional to the expected fitness. A crossover operation on the selected parental individuals is performed using the crossover() function, where a crossover point is randomly chosen to mix the genes of two parental individuals in a certain proportion. Mutation operation on post-crossover individuals is executed using the mutate() function, introducing random perturbations to certain genes of the individuals. The new individuals obtained from the crossover and mutation are merged with the original population to form a new Genetic Algorithm population.

c. Optimal Individual Update: The Sparrow Search Algorithm and Genetic Algorithm populations are merged, and the optimal individual is selected based on the fitness function, specifically the individual with the lowest fitness function value.

d. Termination Condition Assessment: The iteration process concludes when either the maximum number of iterations is reached or specific stopping criteria are satisfied (e.g., the fitness function value falls below a certain threshold).

Step 3 Output Results: The optimal parameter combination and the fitness function value are returned as the optimization outcomes.

By integrating the two algorithms, the SSAGA continually updates its position and individual evolution during the hybrid process to effectively enhance the optimization, leading to superior hyperparameter configurations.

4.4.2. CatBoost-SSAGA

According to the proposed optimization strategies, the mechanism of CatBoost-SSAGA is shown in Algorithm 1. First, we initialize the parameters of the SSAGA. Secondly, we generate populations for both the SSA and GA, separately, then merge them into a new population. Third, we update the position of the sparrows and the global fitness. Finally, Algorithm 1 returns the best position and its corresponding fitness value, which represent the optimal hyperparameters for CatBoost.

CatBoost-SSAGA involves multiple steps such as parameter optimization, model training, and validation, each of which affects the overall complexity. The following presents a complexity analysis of these steps:

(1) Data Preparation and Preprocessing: The complexity of the data loading and preprocessing is typically O(n × m), where n is the number of samples and m is the number of features.

(2) Parameter Optimization and Model Training

Population Initialization: The complexity of the SSA and GA population initialization is O(pop_size × dimension), where pop_size is the population size, and dimension is the parameter dimension.

Iterative Optimization:

In each iteration, the complexity of updating the sparrow positions and performing the GA operations is also O(pop_size × dimension).

The training complexity of the CatBoost model depends on n_estimators (the number of trees), max_depth (the tree depth), the number of samples, and the number of features. The training complexity of CatBoost can be roughly represented as O(n_estimators × n × m).

Due to the use of ten-fold cross-validation, the computational complexity is further increased as the model needs to be trained and evaluated on each fold, making the complexity approximately O(10 × n_estimators × n × m).

(3) Optimal Solution Selection: The complexity of selecting the optimal solution from the merged population is O(pop_size), because it needs to iterate in the population to find the individual with the highest fitness.

Algorithm 1. CatBoost-SSAGA

```
Input: Population size P, Dimension D, Upper bound ub, Lower bound lb, Maximum iterations $\varepsilon$,
Strategy parameter S.
Output: Best fitness value $f_{Gb}$ and Best position $X_{Gb}$.
    1. Initialize empty lists: X = [], F = []
    2. Generate initial population for SSA and GA:
      a. For SSA (Sparrow Search Algorithm):
         - Use initialize_sparrows function with inputs pop_size = P, dimension = D, lb, and ub to
           create sparrows population
      b. For GA (Genetic Algorithm):
         - Use initialize_ga_population function with the same inputs to create GA population
      c. Combine both populations: X = SSA population + GA population
    3. For each iteration t from 1 to $\varepsilon$:
      a. Calculate decay rate $\epsilon = 1-(t/\varepsilon)$
      b. For each individual I in X:
         - Evaluate fitness using evaluate_fitness function
         - F = CatBoost.fit(X)
      c. Get $X_b$, $f_b$, $X_w$, $f_w$
      d. Update positions of first pdNum individuals in X using SSA strategy:
         - Apply update_sparrow_positions_enhanced influenced by $X_b$ and $X_w$
      e. Update positions of remaining individuals in X using GA strategy:
         - Select parents from X
         - Perform crossover and mutation to generate new offspring
         - Replace corresponding individuals in X with new offspring
      f. Re-evaluate fitness of entire population X
      g. If a better fitness is found (indicating higher accuracy from CatBoost), update $f_{Gb}$ and $X_{Gb}$
      h. End
    4. Return $f_{Gb}$ and $X_{Gb}$
```

4.5. Indicators of Model Evaluation

The proposed risk-warning model was mainly applied to a binary classification problem, wherein the samples were divided into positive and negative classes, and the prediction results were evaluated using a confusion matrix, i.e., Table 1. Accuracy, precision, recall, F1-scores, and the AUC were used as evaluation metrics for the proposed model. Accuracy refers to the percentage of samples correctly predicted by the classifier in the total samples. Recall represents the proportion of correctly predicted positive samples. The F1-score is the harmonic average of precision and recall, which considers both evaluation indicators and reflects the model's robustness. The AUC (area under the curve) measures the area under the ROC (Receiver Operating Characteristic) curve, where a higher AUC indicates a better classification effect of the classifier.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{12}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

$$\mathrm{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{14}$$

Table 1. Confusion matrix.

	Positive	**Negative**
True	True Positive (*TP*)	True Negative (*TN*)
False	False Positive (*FP*)	False Negative (*FN*)

5. Experiments

5.1. Datasets and Preprocessing

1. Datasets

For this study, five independent and usable datasets of heart disease are selected, namely cleveland.data, hungarian.data, switzerland.data, VA-long-beach.data, and basel.data, which were obtained from the UCI Machine Learning Repository [29]. After merging, the dataset consists of 1190 data instances and contains three attributes with significant missing values, hence it is classified as a small and incomplete dataset. The merged dataset consists of a complete set of 14 attributes, of which 13 are used for predicting heart disease as feature attributes, and the remaining 1 is used as a labeled sample. The specific attribute descriptions are shown in Table 2.

2. Preprocessing

After the cleaning process, the experimental dataset is left with 981 samples, each of which contains 14 common attributes, of which 13 are used to predict the characteristic attributes of heart disease and the remaining one is used as a labeled sample. For data augmentation, virtual data are generated using the SALGAN, in accordance with the types of labels. As a result, the final dataset comprises 1981 samples.

3. Model training

Cross-validation effectively reflects the robustness of the model. In the experiments, all of the evaluation metrics in the experiments are based on 10-fold cross-validation and calculated as averages and standard deviations.

CatBoost incorporates numerous hyperparameters, including iterations, learning_rate, depth, reg_lambda, subsample, border_count, and so on. This study focuses on fine-tuning the hyperparameters that significantly impact model performance. The primary parameters we optimized are the learning rate, tree depth, maximum number of trees, and regularization coefficient, among five other parameters. Table 3 lists the default values of these parameters as well as the ranges we used for optimization.

Table 2. Dataset attribute descriptions.

Name	Description
Age	Continuously variable values
Sex	0 = Female
	1 = Male
Cp	0 = Classic angina pectoris
	1 = Atypical angina pectoris
	2 = Non-angina pectoris
	3 = Asymptomatic
Trestbps	Continuously variable values
Chol	Continuously variable values
Fbs	0 = <120mg/d
	1 = >120mg/d
Restecg	0 = Normal
	1 = Existence of a segment exception
	2 = Possible or definite left ventricular hypertrophy
Thalach	Continuously variable values
Exang	0 = No
	1 = Yes
Oldpeak	Continuously variable values
Slope	0 = Up
	1 = Float
	2 = Down
Ca	0
	1
	2
	3
Thal	1 = Normal
	2 = Irreparable
	3 = Reparable
Target	0 = No
	1 = Yes

Table 3. Optimal parameters of CatBoost-SSAGA.

Name	Optimization Scope	SSAGA-CatBoost
learning_rate	[0.001, 0.2]	0.00298
max_depth	[4, 10]	9
n_estimators	[1100, 1500]	1153
reg_lambda	[0.01, 10]	2.62938
subsample	[0.9, 1]	0.98761

5.2. Comparative Experiments

For this section, we conducted three sets of comparative experiments and ablation studies on the UCI heart disease dataset to validate the performance of the proposed SALGAN-CatBoost-SSAGA model.

1. Performance comparison between different algorithms

To verify the effectiveness of the proposed framework, we conducted comparative experiments between five commonly used tree models and SALGAN-CatBoost-SSAGA, including XGBoost version 2.0.3, LightGBM version 3.3.5, and Scikit-learn version 1.0.2. The experimental results in terms of the accuracy, precision, recall, F1-scores, and AUC are shown in Table 4 and Figure 5. The results demonstrate the superior performance of SALGAN-CatBoost-SSAGA over the individual machine learning models. Table 4 indicates that SALGAN-CatBoost-SSAGA performs the best in all metrics, with an accuracy of 90.56%, a precision of 87.79%, a recall of 87.45%, and an F1-score of 87.54%, which are much higher than those of the other models, indicating that the proposed framework performs well. It also shows smaller standard deviations, indicating greater stability. As shown in Figure 5, the area under the curve for SALGAN-CatBoost-SSAGA is the largest, reaching 0.96, which illustrates that it has high accuracy and the best overall classification performance.

Table 4. The algorithms' performances.

Model	Accuracy	Precision	Recall	F1-Score
RF	79.71 ± 0.03	81.29 ± 0.03	79.14 ± 0.05	80.10 ± 0.03
lightGBM	77.78 ± 0.02	79.52 ± 0.03	77.18 ± 0.06	78.16 ± 0.03
xgBoost	77.37 ± 0.02	78.85 ± 0.03	77.17 ± 0.04	77.90 ± 0.03
AdaBoost	71.66 ± 0.03	72.66 ± 0.02	72.65 ± 0.06	72.55 ± 0.03
Decision Tree	71.15 ± 0.03	72.76 ± 0.03	70.87 ± 0.04	71.76 ± 0.03
SALGAN-CatBoost-SSAGA	90.56 ± 0.01	87.79 ± 0.02	87.45 ± 0.03	87.54 ± 0.02

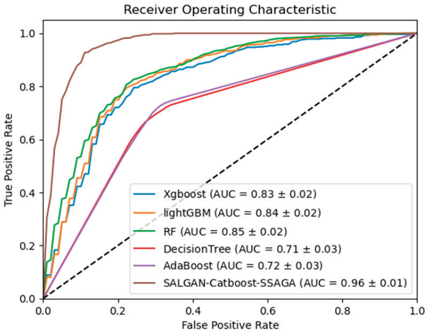


Figure 5. ROC for SALGAN-CatBoost-SSAGA and other models.

2. Impact of data augmentation on prediction results

In order to verify the effectiveness of the data enhancement method, we used the standard CatBoost model to compare the risk predictions of the data before and after data enhancement. The experimental results are shown in Table 5. These experimental results indicate that augmenting the dataset through data enhancement techniques led to a notable improvement in the performance of the CatBoost model across all evaluation metrics. Compared to the performance prior to data augmentation, there was an increase of over 10% in all metrics.

Table 5. Comparison of prediction results based on data augmentation.

Datasets	Accuracy	Precision	Recall	F1-Score
Before	79.57 ± 0.04	79.94 ± 0.05	79.58 ± 0.07	79.50 ± 0.04
After	87.83 ± 0.01	87.87 ± 0.02	87.81 ± 0.02	87.81 ± 0.01

3. Impact of parameter optimization on prediction results

To evaluate the effectiveness of the SSAGA hybrid algorithm, we utilized a dataset augmented through SALGAN data enhancement to compare its performance with that achieved by optimizing the CatBoost model parameters using solely the SSA or GA. The experimental results are presented in Table 6. These results demonstrate that, compared to the individual use of the SSA or GA, the SSAGA showed the best performance across all evaluation metrics, particularly in terms of accuracy.

Table 6. Comparison of prediction results based on the SSAGA.

Algorithm	Accuracy	Precision	Recall	F1-Score
SSA	89.45 ± 0.02	86.57 ± 0.07	85.63 ± 0.09	86.63 ± 0.07
GA	87.58 ± 0.02	84.03 ± 0.08	83.27 ± 0.1	83.87 ± 0.09
SSAGA	90.56 ± 0.01	87.79 ± 0.02	87.45 ± 0.03	87.54 ± 0.02

4. Ablation experiments

In order to evaluate the overall performance of SALGAN-CatBoost-SSAGA and verify the necessity of each module, we conducted ablation experiments on the SALGAN and SSAGA. As shown in Figure 6, SALGAN-CatBoost-SSAGA outperformed the other two models in key performance metrics such as accuracy, precision, recall, and F1-scores. This finding not only highlights the excellent predictive ability of SALGAN-CatBoost-SSAGA but also clearly proves the rationality of our choice of this combined model. It shows the obvious advantages of the combination model in improving the prediction performance compared with a single model, thus verifying the effectiveness of our combination strategy.

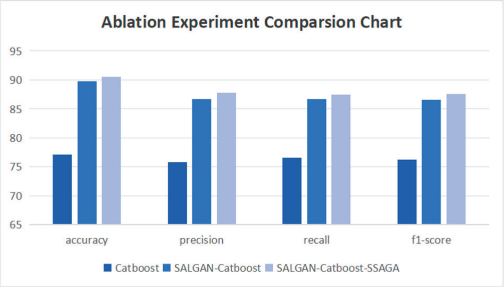


Figure 6. Ablation experiment comparison chart.

6. Conclusions

This paper introduces the SALGAN-CatBoost-SSAGA predictive model, which is designed for small-sample and incomplete datasets. The SALGAN is capable of learning various hidden correlations and dependencies within different types of labeled data, thereby facilitating the generation of highly realistic data instances. This enhancement aids in the model's learning process with sample data. Additionally, in order to find the global optimal parameters of the CatBoost algorithm, we propose the SSAGA, an algorithm that combines the SSA and GA, which helps CatBoost find global optimal parameters more effectively, avoiding local optima and improving the accuracy and stability of the prediction model. The experimental results show that the framework achieves the expected effect in data augmentation and improving prediction accuracy. The performance of the framework is better than other comparison methods in all selected indicators, which proves its feasibility and effectiveness. Therefore, it is very suitable for prediction tasks with small-sample datasets. Future works will focus more on directly incorporating small-sample incomplete datasets from different fields to validate further the generalizability of the model proposed in this study.

Author Contributions: Conceptualization, Y.P. and Y.L.; methodology, Y.P.; software, Y.L.; validation, Y.P., Y.L. and X.L.; resources, X.L.; writing—original draft preparation, Y.L.; writing—review and editing, Y.P., J.W. and X.L.; visualization, Y.L.; supervision, J.W.; project administration, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, Grants number 62172287 and number 62102273.

Data Availability Statement: The original data presented in the study are openly available at http://archive.ics.uci.edu/dataset/45/heart+disease (accessed on 9 April 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhao, K.; Jin, X.; Wang, Y. Survey on few-shot learning. *J. Softw.* **2021**, *32*, 349–369.
2. Ansarullah, S.I.; Kumar, P. A systematic literature review on cardiovascular disorder identification using knowledge mining and machine learning method. *Int. J. Recent Technol. Eng.* **2019**, *7*, 1009–1015.
3. Yekkala, I.; Dixit, S.; Jabbar, M.A. Prediction of heart disease using ensemble learning and Particle Swarm Optimization. In Proceedings of the 2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon), Bengaluru, India, 17–19 August 2017; pp. 691–698.
4. Li, F.F.; Fergus, R.; Perona, P. One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 594–611.
5. Liu, Y.; Long, M.; Cao, Z.; Wang, J. Few-Shot Object Recognition from Machine-Labeled Web Images. *IEEE Trans. Image Process.* **2020**, *29*, 594–611.
6. Wang, X.; Huang, T.E.; Darrell, T.; Gonzalez, J.E.; Yu, F. Frustratingly Simple Few-Shot Object Detection. *arXiv* **2020**. [CrossRef]
7. Wang, J.; Chen, Y. *Introduction to Transfer Learning*; Electronic Industry Press: Beijing, China, 2021.
8. Hu, X.; Chen, S. A survey of few-shot learning based on machine learning. *Intell. Comput. Appl.* **2021**, *11*, 191–195+201.
9. Zhu, X.; Ghahramani, Z.; Lafferty, J.D. Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In Proceedings of the Twentieth International Conference on Machine Learning (ICML 2003), Washington, DC, USA, 21–24 August 2003; pp. 912–919.
10. Gliozzo, J.; Mesiti, M.; Notaro, M.; Petrini, A.; Patak, A.; Puertas-Gallardo, A.; Paccanaro, A.; Valentini, G.; Casiraghi, E. Heterogeneous data integration methods for patient similarity networks. *Brief. Bioinform.* **2022**, *23*, bbac207. [CrossRef]
11. Schwartz, E.; Karlinsky, L.; Shtok, J.; Harary, S.; Marder, M.; Kumar, A.; Feris, R.; Giryes, R.; Bronstein, A. Delta-encoder: An effective sample synthesis method for few-shot object recognition. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 2850–2860.
12. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
13. Kataoka, Y.; Matsubara, T.; Uehara, K. Image generation using generative adversarial networks and attention mechanism. In Proceedings of the 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, Japan, 26–29 June 2016; pp. 1–6.
14. Park, N.; Mohammadi, M.; Gorde, K.; Jajodia, S.; Park, H. Data Synthesis based on Generative Adversarial Networks. *Proc. VLDB Endow.* **2018**, *11*, 1071–1083. [CrossRef]
15. Dorogush, A.V.; Ershov, V.; Gulin, A. CatBoost: Gradient boosting with categorical features support. *arXiv* **2018**, arXiv:1810.11363.
16. Diao, L.; Niu, D.; Zang, Z.; Chen, C. Short-term weather forecast based on wavelet denoising and catboost. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 3760–3764.
17. Kumar, P.S.; Kumari, A.; Mohapatra, S.; Naik, B.; Nayak, J.; Mishra, M. CatBoost ensemble approach for diabetes risk prediction at early stages. In Proceedings of the 2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology (ODICON), Bhubaneswar, India, 8–9 January 2021; pp. 1–6.
18. Wang, B.; Feng, H.; Wang, F.; Qin, X.; Huang, P.; Dang, D.; Zhao, J.; Yi, J. Application of CatBoost model based on machine learning in predicting severe hand-foot-mouth disease. *Chin. J. Infect. Control* **2019**, *18*, 12–16.
19. Chen, D.; Chen, Y.; Feng, X.; Wu, S. Retrieving suspended matter concentration in rivers based on hyperparameter optimized CatBoost algorithm. *J. Geo-Inf. Sci.* **2022**, *24*, 780–791.
20. Jin, C.; Yu, J.; Wang, Q.; Chen, L.J. Prediction of blasting Fragment large block percentage ratio based on ensemble learning CatBoost model. *J. Northeast. Univ. (Nat. Sci.)* **2023**, *44*, 1743–1750.
21. Xu, L.; Guo, C. Predicting Survival rates for gastric cancer based on ensemble learning. *Data Anal. Knowl. Discov.* **2021**, *5*, 86–99.

22. Yang, C.; Liu, L.; Zhang, Y.; Zhu, W.; Zhang, S. Machine learning based on landslide susceptibility assessment with Bayesian optimized the hyper parameters. *Bull. Geol. Sci. Technol.* **2022**, *41*, 228–238.
23. Tikhamarine, Y.; Souag-Gamane, D.; Kisi, O. A new intelligent method for monthly streamflow prediction: Hybrid wavelet support vector regression based on grey wolf optimizer (WSVR-GWO). *Arab. J. Geosci.* **2019**, *12*, 540. [CrossRef]
24. Feng, T.; Peng, Y.; Wang, J. ISGS: A combinatorial model for hysteresis effects. *Acta Electron. Sin.* **2023**, *51*, 2504–2509.
25. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* **2020**, *8*, 22–34. [CrossRef]
26. Meng, K.; Chen, C.; Xin, B. MSSSA: A multi-strategy enhanced sparrow search algorithm for global optimization. *Front. Inf. Technol. Electron. Eng.* **2022**, *23*, 1828–1847. [CrossRef]
27. Ou, Y.; Yu, L.; Yan, A. An Improved Sparrow Search Algorithm for Location Optimization of Logistics Distribution Centers. *J. Circuits Syst. Comput.* **2023**, *32*, 2350150. [CrossRef]
28. Wang, J.; Wang, Z.; Li, J.; Peng, Y. An Interpretable Depression Prediction Model for the Elderly Based on ISSA Optimized LightGBM. *J. Beijing Inst. Technol.* **2023**, *32*, 168–180.
29. Janosi, A.; Steinbrunn, W.; Pfisterer, M.; Detrano, R. Heart Disease. UCI Machine Learning Repository. 1988. Available online: https://archive.ics.uci.edu/dataset/45/heart+disease (accessed on 9 April 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

A Lightweight 6D Pose Estimation Network Based on Improved Atrous Spatial Pyramid Pooling

Fupan Wang [1], Xiaohang Tang [1], Yadong Wu [2,*], Yinfan Wang [1], Huarong Chen [1], Guijuan Wang [1] and Jing Liao [1]

[1] School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 621010, China; wangfp@swust.edu.cn (F.W.); ixllus@mails.swust.edu.cn (X.T.); wangyf@mails.swust.edu.cn (Y.W.); chenhuarong@swust.edu.cn (H.C.); guijuanwang@swust.edu.cn (G.W.); liaojing@swust.edu.cn (J.L.)

[2] School of Computer Science and Engineering, Sichuan University of Science & Engineering, Yibin 644007, China

* Correspondence: wuyadong@suse.edu.cn

Abstract: It is difficult for lightweight neural networks to produce accurate 6DoF pose estimation effects due to their accuracy being affected by scale changes. To solve this problem, we propose a method with good performance and robustness based on previous research. The enhanced PVNet-based method uses depth-wise convolution to build a lightweight network. In addition, coordinate attention and atrous spatial pyramid pooling are used to ensure accuracy and robustness. This method effectively reduces the network size and computational complexity and is a lightweight 6DoF pose estimation method based on monocular RGB images. Experiments on public datasets and self-built datasets show that the average ADD(-S) estimation accuracy and 2D projection index of the improved method are improved. For datasets with large changes in object scale, the estimation accuracy of the average ADD(-S) is greatly improved.

Keywords: 6DoF pose estimation; depth-wise convolution; coordinate attention; atrous spatial pyramid pooling

Citation: Wang, F.; Tang, X.; Wu, Y.; Wang, Y.; Chen, H.; Wang, G.; Liao, J. A Lightweight 6D Pose Estimation Network Based on Improved Atrous Spatial Pyramid Pooling. *Electronics* **2024**, *13*, 1321. https://doi.org/10.3390/electronics13071321

Academic Editor: Ping-Feng Pai

Received: 28 February 2024
Revised: 24 March 2024
Accepted: 29 March 2024
Published: 1 April 2024

1. Introduction

In today's quest for intelligence-driven technology, applications such as intelligent car driving, augmented reality, human–computer interactions, etc., are gradually transitioning from concepts to real life. The realization of all these applications relies heavily on support from relevant technologies. Six-degree-of-freedom (6D) object pose estimation based on computer vision is an important technology in this regard. The key to this technology lies in recovering the 3D translation and 3D rotation information of target objects from images or point cloud data. Accurately and efficiently estimating the 6D information of objects in real-world scenarios is of significant value for enhancing the safety of automated driving, strengthening the immersion of virtual and real interactions, and improving the reliability of robotic operations.

Unlike traditional 6D pose estimation methods that rely on multiple sensors, computer vision-based methods are currently the mainstream approach in research, significantly reducing the complexity and application costs of the entire pose estimation system. Early methods based on template matching involve the construction of a template library of the same target object from images at different angles and distances, calculating the similarity between the real value and template images in the estimation task. However, such methods are computationally complex and time-consuming. Subsequently, algorithms like SIFT [1], FAST [2], and BRIEF [3] have been applied to extract invariant features from images as key points, with pose estimation then being performed by matching key points with real points. However, objects in real environments are often affected by factors such as complex backgrounds, changes in lighting conditions, occlusions, and variations in viewpoints, leading to difficulties in extracting effective key points. With the development of 3D scanning technology, the difficulty of obtaining 3D object models has been reduced. Methods based on aligning models with object images or point clouds have been proposed, which offer high pose estimation accuracy. However, these methods require high model accuracy, depend on depth information, and have poor applicability to non-rigid objects [3].

With the successful application of deep learning methods in various tasks, researchers have begun to explore the use of deep neural networks (DNNs) to solve the 6D pose estimation problem for RGB images. As a pioneering work, PoseNet [4] was the first to adopt a CNN to regress camera pose from a single image, successfully proving that it is possible to achieve end-to-end camera pose regression with just one picture as input [5]. Deep learning methods, such as CNNs, can automatically learn feature representations, making the extracted key points more robust, effectively capturing the structure and semantic information of images and thus enhancing feature discrimination. In particular, in complex scenarios common in object pose estimation, such as cluttered backgrounds, changes in lighting, and object occlusion, deep learning-based pose estimation methods have achieved promising results. Researchers designing conventional 6D pose estimation methods tend to focus more on designing complex networks to improve estimation performance while ignoring the practical deployment challenges arising from the high model complexity and large parameter counts.

To address this issue and meet the needs of typical applications like mobile augmented reality, in this paper, a lightweight, deep learning-based pose estimation method is proposed. By improving the pose estimation network and using RGB images obtained from monocular cameras to estimate the six-degree-of-freedom camera pose, the method achieves robustness and accuracy while meeting the requirements for lightweight deployment. The main contributions of this paper are as follows:

1. To reduce the computational cost of 6D object pose estimation, we propose a lightweight model in which residual depth-wise separable convolution is combined with an improved atrous spatial pyramid pooling (ASPP) method.
2. We introduce a coordinate attention mechanism and address the issue of object scale variation, which was not considered in the original method. Additionally, we incorporate a multi-scale pyramid pooling module. These enhancements effectively reduce the model's parameter and computation complexity while significantly improving the pose estimation accuracy for objects with large-scale variations.
3. The effectiveness of this lightweight model is validated and analyzed through experiments on both publicly available and self-built datasets.

2. Related Works

DNNs learn object poses directly from images or point cloud data, reducing the reliance on depth information typical in traditional methods and thus driving the rapid development of pose estimation methods based on monocular RGB images. Deep learning 6D pose estimation methods can be classified into two categories based on the training strategy: direct strategy and indirect strategy methods.

Deep learning methods based on a direct strategy are also known as single-stage methods. In this approach, depth information is combined and the embedding space of poses is learned directly, or 3D translation and rotation information is regressed directly, through end-to-end learning. Its mapping relationship is simple and it has a fast inference speed. To address the limited accuracy of directly regressing pose estimation from images, the DeepIM model optimizes poses by iteratively training CNNs to match model-rendered images with input images [6]. In scenes with occlusions and clutter, in Posecnn, translation information is estimated by locating objects in the center of the image relative to the camera distance, and object rotation is estimated through regression representation, which is capable of handling symmetric objects [7]. This network can simultaneously perform both object detection and pose estimation tasks. Wang et al. [8] designed the DenseFusion framework for RGB-D datasets, which fully utilizes integrated complementary pixel information and depth information for pose estimation and integrates an end-to-end iterative pose optimization program, achieving real-time inference. YOLO-based methods have an advantage in pose estimation compared to other methods with regard to processing speed and are more advantageous for real-time estimations [9,10]. Single-stage deep learning methods can simultaneously complete object pose estimation in one stage with faster inference speeds, making them more applicable to applications with high real-time requirements. However, they may have slight shortcomings regarding accuracy and robustness.

In deep learning-based pose estimation methods relying on the indirect strategy, 6D pose estimation tasks are primarily completed using single-view RGB images. This method is also known as the two-stage approach, where the entire task flow is divided into two stages: establishing 2D–3D correspondence relationships for key points and recovering pose information using PnP/RANSAC variant algorithms. Compared to direct strategy methods, it has advantages with regard to robustness and accuracy when occlusions, lighting changes, or complex backgrounds are present. Visual occlusion is a common problem in practical applications. To address this, Rad et al. proposed the BB8 method, which uses a convolutional neural network to estimate a pose by locating the spatial positions of eight corner points of the object, which showed good robustness against occlusion and cluttered backgrounds [11]. Zhao et al. [12] further selected key points from eight corner points in space. This was improved upon in PVNet, where sparse key points are discarded as reference points, pixel-to-key point vectors are introduced through a semantic segmentation network, and the estimation accuracy is enhanced in scenarios with occlusion and symmetric objects [13]. Pix2Pose utilizes adversarial generative networks to address similar issues [14]. Chen et al. proposed a method that first calibrates in two dimensions and then estimates poses using 2D–3D correspondence relationships, achieving significant breakthroughs in accuracy [15].

In summary, indirect strategy methods have advantages with regard to robustness and accuracy in complex environments, but they involve more steps and require more computational resources and time.

3. Improved Model Based on Atrous Spatial Pyramid Pooling

To reduce the computational cost of object 6D pose estimation, in this paper, a lightweight model is proposed, combining residual depth-wise separable convolution with an improved ASPP method based on PVNet. Firstly, we utilized residual depth-wise separable convolution layers to construct a lightweight network, thereby reducing the demand for memory and computing resources. Secondly, to prevent the loss of channel and coordinate information caused by multi-layer convolutions, which affects the estimation accuracy, we introduced a coordinate attention mechanism. Addressing the issue of inadequate recognition accuracy in previous methods caused by neglecting object scale changes, we added a multi-scale pyramid pooling module for processing. Finally, a two-stage pixel voting and PnP (Perspective-n-Point) solving task was performed on the decoder's output feature maps. This approach effectively reduced the model's parameter and computational overhead, resulting in a significant improvement in the pose estimation accuracy for objects with large-scale variations. The complete 6D pose estimation network framework is illustrated in Figure 1.

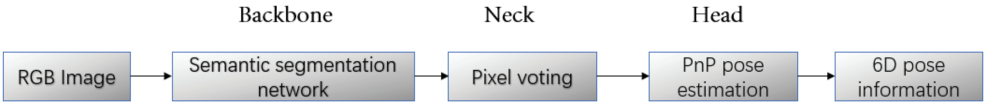


Figure 1. Six-degree-of-freedom pose estimation network framework.

3.1. Backbone Network

The backbone network based on improved ASPP consists of 13 layers of depth-wise separable convolution, a coordinate attention mechanism module, and an ASPP module as the encoder, followed by three upsampling and convolution operations in the decoder. The final output is a target segmentation feature map. The network architecture is illustrated in Figure 2.

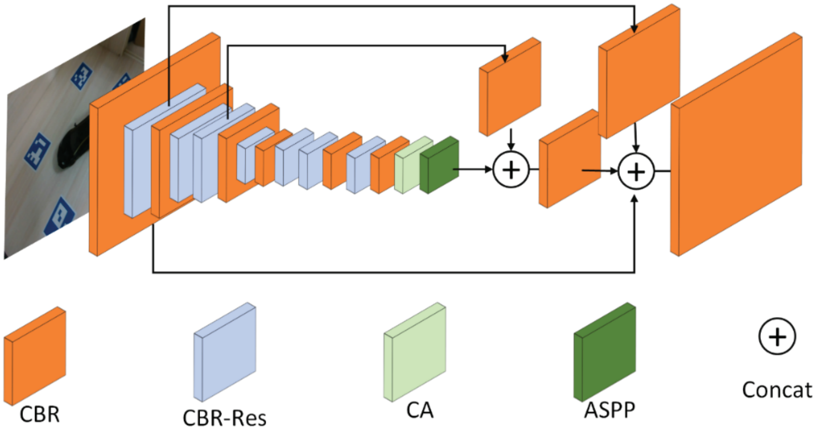


Figure 2. The backbone network based on improved ASPP consists of several components: CBR, representing depth-wise separable convolution layers; CBR-Res, indicating depth-wise separable convolution layers with residual connections; CA, denoting the coordinate attention mechanism module; ASPP, representing the improved ASPP module; and Concat, which involves upsampling and concatenation operations.

3.1.1. Residual Depth-Wise Separable Network

In PVNet, outstanding performance in keypoint detection is achieved by using a pre-trained ResNet model based on a large-scale dataset as the target segmentation network, leading to a significant improvement in the accuracy of 6D pose estimation over previous methods [16]. However, this model and its related improvements have a relatively large volume, making them difficult to deploy on devices with requirements for lightweight applications. Drawing inspiration from the MobileNet series of algorithms [17,18], in this study, a lightweight backbone network model based on depth separable convolution operations was designed, significantly reducing the number of network layers and model parameters.

The depth separable convolution operation can be seen as the process of splitting a standard convolution into a grouped convolution and a pointwise convolution. Compared to standard convolution, which convolves an image with m channels using n sets of corresponding m × 3 × 3 convolution kernels to generate n feature maps, in-depth separable convolution, a two-stage mode, was adopted. First, a 1 × 3 × 3 convolution kernel was applied to each of the m channels, generating m intermediate feature maps, and then n sets of m × 1 × 1 convolution kernels were used to generate n feature maps. Figure 3 illustrates the process of the two convolutions.

Aiming to produce a feature map with C_{oup} channels using a square convolution kernel of size $k \times k$, following a uniform standard where bias parameters with a quantity of 1 are disregarded and each multiply–add operation is treated as one floating-point calculation, for an input image with dimensions of $W \times H$ and C_{inp} channels, we can approximate the parameter and computation complexity of standard convolution as follows:

$$Paramsstd = Coup \times k_2 \times Cinp \tag{1}$$

$$FLOPsstd = Cinp \times k_2 \times Coup \times W \times H \tag{2}$$

An approximate description of the parameter and computation complexity of depth-wise separable convolution is as follows:

$$Paramsdw\&pw = Cinp \times k_2 + 1 \times Cinp \times Coup \tag{3}$$

$$FLOPsdw\&pw = Cinp \times k_2 \times W \times H + 1 \times Cinp \times Coup \times W \times H \tag{4}$$

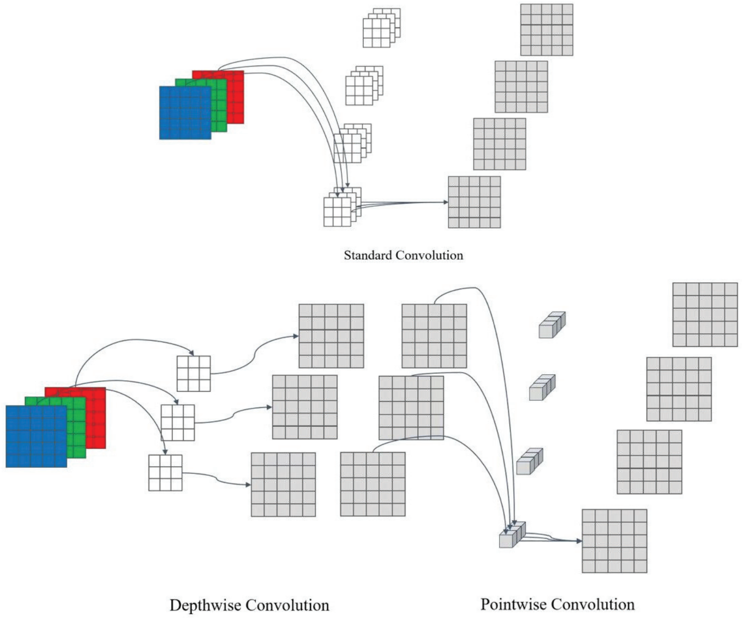


Figure 3. The upper diagram shows the calculation process of standard convolution, while the lower diagram illustrates the computation process of depth-wise separable convolution.

Compared to standard convolution operations, the parameter and computation complexity required for depth-wise separable convolution is $\frac{1}{Coup} + \frac{1}{k^2}$ times that of standard convolution. Particularly in this model, where the output channels of the image ranged from 64 to 1024 and the depth-wise separable convolution kernels were all 3 × 3 in size, there was a significant advantage in reducing the computational overhead compared to standard convolution. Considering that, in the process of semantic segmentation, multiple downsampling and convolution operations on the original image may lead to the loss of low-dimensional information, adding residual connections to convolution layers with the same input and output channels preserves low-dimensional feature information while obtaining high-dimensional semantic information after convolution, which significantly improves the final accuracy of object pose estimation.

3.1.2. Improved Atrous Spatial Pyramid Pooling

In the decoder part of the network, feature extraction was performed through convolution and pooling operations. All output feature maps were compressed into a single feature vector via global average pooling, which ignored the non-uniformity of features across different channels. Additionally, influenced by the number of layers and the size of convolution kernels, lightweight networks often overlook the spatial positional information of features.

The coordinate attention mechanism [19] is an attention method suitable for lightweight networks. It is characterized by a lightweight and plug-and-play nature and can capture inter-channel information while also considering directional and positional information. Additionally, it achieves significant improvement in tasks such as object detection and semantic segmentation, which involve dense predictions. In the coordinate attention mechanism, average pooling and concatenation were performed in both the horizontal and vertical directions and channel information was fused through convolution, encoding spatial information simultaneously. After splitting, multiplication with the input features was performed, emphasizing feature information in spatial positions. The coordinate attention mechanism notably enhanced the performance of dense prediction tasks in lightweight mobile networks. Through experimental analysis, it was decided to place it at the end of the encoder, reducing interference from subsequent convolution operations. Additionally, the reduced number of channels at the end could decrease computational costs.

The application of depth-wise separable convolution significantly reduced the model complexity. However, in computer vision-related tasks, the size of the receptive field largely determines the effectiveness of the model. Increasing the network depth and using larger convolutional kernels are direct and effective methods to enhance the receptive field. However, they inevitably exacerbate model complexity. Through experiments, it has been found that simply increasing the network depth results in limited performance improvements relative to the cost incurred.

Luo et al. [20] proposed the concept of an effective receptive field, demonstrating in their work that the effective receptive field only represents a portion of the entire receptive field. Therefore, a larger receptive field is needed in practice to cover the desired region. Dilated convolution, also known as atrous convolution, is one method to effectively enlarge the receptive field. This is achieved by introducing "holes" into the convolutional kernel, effectively expanding the receptive field without increasing the number of parameters in the kernel.

The ASPP module extends the concept of dilated convolution by introducing multiple dilated convolutions with different rates [21]. It considers feature information from multiple receptive fields, thereby enhancing the perception of objects at different scales. Figure 4 illustrates dilated convolutions with different rates. By employing this method, it is possible to capture object boundaries and detailed information while reducing computational complexity, thereby improving the recognition accuracy.

Building upon ASPP, an attention mechanism module was incorporated to further enhance the feature information at different scales, highlighting key point information. Given the limited overall data volume of single-target training samples, overfitting was likely to occur. Therefore, a dropout layer was applied at the output end during training. During the forward propagation process, random hidden neurons were dropped, reducing the interdependency between neurons and enhancing the network's generalization capability. The mathematical expression for dropout is expressed as follows:

$$y_i = \begin{cases} 0, & dropout(probability = p), \\ \frac{x_i}{1-p}, & otherwise. \end{cases} \tag{5}$$

where y_i and x_i represent the output and input of neuron i, respectively. During training, the output was zeroed out with probability p; during testing, dropout was not applied.

For tasks requiring dynamic object pose estimation, scale variation poses a significant challenge. Compared to previous work, the improved ASPP module exhibits a noticeable robustness enhancement in pose estimation accuracy for datasets with significant scale variations. The improved ASPP module is illustrated in Figure 5.

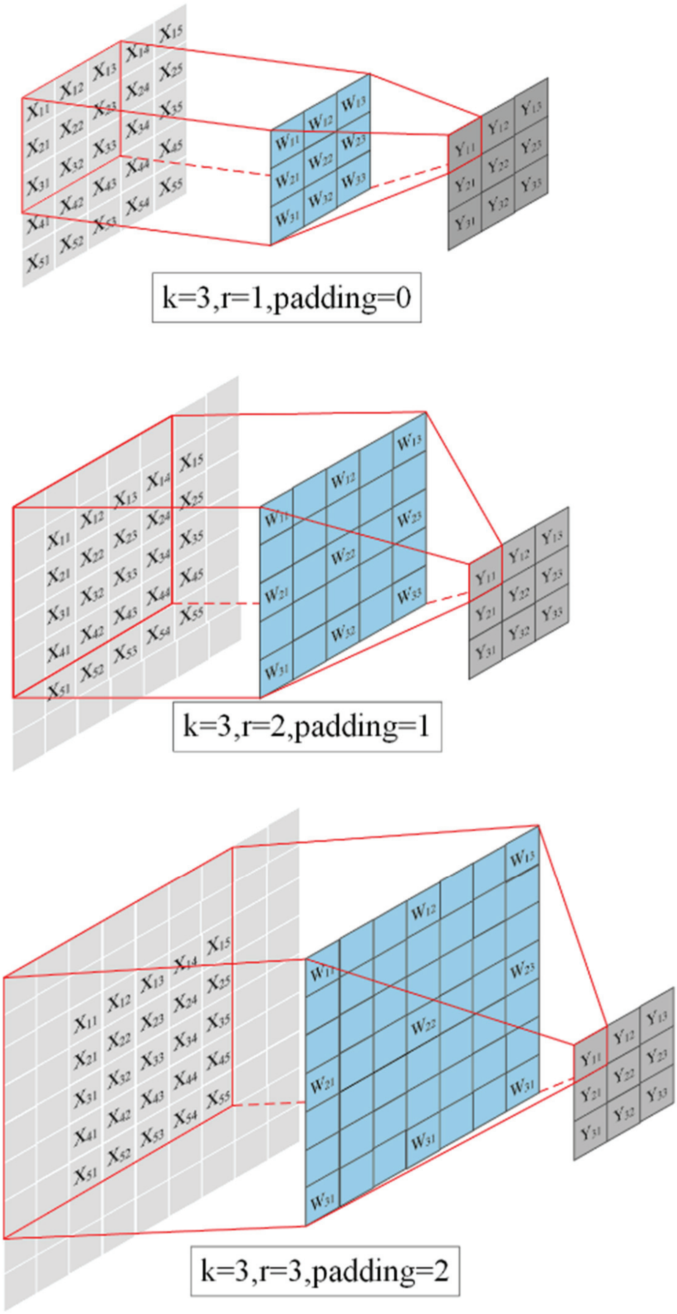


Figure 4. Atrous convolutions with dilation rates r of 1, 2, and 3.

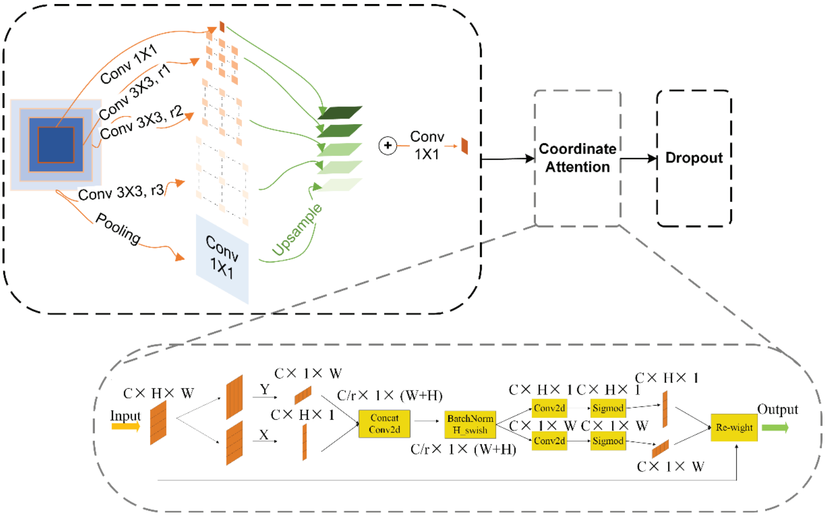


Figure 5. Improved ASPP module.

3.2. Pixel Voting and Pose Estimation

The input RGB image underwent a target segmentation task through a pre-trained backbone network, obtaining the segmented object image as an output. Between the 2D key point x on the segmented image and any pixel point p, there existed a unit vector $vx(p)$, satisfying the relation $vx(p) = \frac{x-p}{||x-p||_2}$. Two pixel points on the image were randomly selected, and the intersection of the unit vectors pointing to the possible positions of the key point was taken as the hypothesis position h_1. This process was repeated n times to generate a set of n hypotheses $\{hi|i = 1, \ldots, n\}$. For $p \in O$ (a pixel point p belonging to the target 0), there existed a corresponding vector $vh(p) = \frac{h_i-p}{||h_i-p||_2}$ for each hypothesis. Whether a hypothesis was assigned depended on whether the dot product $v = vx(p) \cdot vh(p)$ met the threshold θ of the indicator function $1\theta(v)$. This step is referred to as pixel voting. After completing the pixel voting process, the voting score of this set of hypotheses needed to be consistent with the spatial distribution of the true key points. Therefore, the hypothesis point with the highest score was considered the most likely to be the true key point. The scoring function and the calculation formula for the i-th hypothesis are as follows:

$$1\theta(v) = \begin{cases} 1, & if\ v > \theta, \\ 0, & otherwise. \end{cases} \tag{6}$$

$$wi = \sum\nolimits_{p \in O} 1\theta(v) \tag{7}$$

After completing the scoring calculation for n hypotheses, the position information of the hypothesis with the highest score was selected as the position information for the true key point. Through the approximate random consensus sampling method, pixels belonging to the target were randomly selected, thus maintaining the uniformity of the pixel distribution involved in voting and reducing unnecessary score calculations.

The rotation matrix R is a 3 × 3 matrix used to describe the rotational changes of an object in three-dimensional space. It is a common representation method for describing rotations in 3D space. Each column of the matrix typically represented the rotations of the object around the X, Y, and Z axes of the coordinate system, respectively. This method provided a transformation representation from one coordinate system to another, where each column vector represented the direction of a coordinate axis in the rotated coordinate system relative to the original coordinate system.

The translation matrix T was a 3 × 1 vector that described the displacement or shift of an object in three-dimensional space. This vector extended the object's position from its original location to a new location within the coordinate system.

Unlike Euler angles, the rotation matrix representation method does not suffer from gimbal lock issues and provides a continuous and unique representation throughout the entire rotation space. This means that rotation matrices can represent arbitrary rotations without being limited by specific axis orders or rotation sequences, offering a more flexible and stable representation of rotations.

$$R\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{8}$$

$$[R|T]\begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \end{bmatrix} \tag{9}$$

By calculating the coordinates of key points through pixel voting and utilizing the rotation matrix representation method, combined with the 3D object model, the pose information of the object could be estimated.

The data obtained by projecting the known 3D model of the target object into two dimensions and annotating the key points were input into the neural network for training to obtain the spatial mapping from 3D to 2D, enabling the neural network to predict key points. Subsequently, the PnP algorithm or trained network inferred the 3D point location information based on the 2D key points. That is, given an RGB image I and a set of corresponding key point information, consisting of 2D key point information of the image $O = \{Oi|i = 1, \ldots, N\}$ and 3D model point information of the object $M = \{Mi|i = 1, \ldots, N\}$, we aimed to solve the pose relationship $P = [R|T]$.

The method of pixel voting improved the issue of key point selection on objects in 2D images. The selection of actual points on the object in 3D models also impacts the solution results. By employing furthest-point sampling, the 3D model's surface points were sampled as key points, resulting in a uniform key point distribution across the 3D model's surface. Experiments on PVNet have demonstrated that this method effectively reduces interference compared to traditional methods relying on corner points as key points, thereby enhancing the pose estimation accuracy.

3.3. Training Strategy

Monocular RGB images were used as training and testing samples. In the 6D object pose estimation task, typically, 15% of the data were chosen as training samples and 85% were chosen as testing samples. The model was trained for 120 epochs on the entire dataset, with an initial learning rate of 4×10^{-3}. After every 20 epochs, the learning rate was reduced by half.

The output target segmentation image was treated as a binary classification problem, where the task was to determine whether an object was in the foreground (object) or background (interference). The cross-entropy loss function was used as the main loss function for the backbone network. The distance between generated hypothesis points and true key points could intuitively infer the error of the result. This was equivalent to the squared sum of the horizontal and vertical components of the vector difference, $\Delta vk(p) = \widetilde{vk}(p) - vk(p)$, pointing from any pixel to the hypothesis and true points. To address the issue of outlier gradient explosions, smooth $L1$ *Loss* was employed to smooth outliers, and the sum of losses for all key points was used as the loss function for key point prediction.

$$Loss = \sum\nolimits_{k=1}^{K} \Big(s\_L1 \Big(\Delta vk(p)_x + s_{L1(\Delta vk(p)_y)} \tag{10}$$

$$Loss(abs) = \sum_{k=1}^{K}\left(|\Delta vk(p)_x| + \left|\Delta vk(p)_y\right|\right) \tag{11}$$

The same loss function results can be caused by position offsets and scale differences. To reduce interference caused by scale changes, the distance from the hypothesis points to the center point was added as a reference. The absolute value of the sum of $\Delta vk(p)$ in the horizontal and vertical directions, denoted as *Loss(abs)*, was used as a reference; under the same loss condition, the smaller the *Loss(abs)*, the greater the scale change.

4. Experimental Results and Analysis

4.1. Dataset

Commonly used datasets for 6D object pose estimation include LineMOD, LineMod-Occlusion [22], and T-LESS [23]. The LineMOD dataset contains RGB-D image data and corresponding labels for multiple small geometric objects with complex shapes, countering challenges such as weak object texture features, complex background environments, and varying lighting conditions. In this study, we not only conducted pose estimation tests on the LineMOD dataset but also designed new datasets using 3D cameras, including an airplane model dataset and a car model dataset, to determine the accuracy improvements of our method for multi-type, multi-scale object pose estimation. These two datasets also present various challenges, including significant scale changes, weak texture features, blurry images, lighting variations, and incomplete objects.

The assembly of the self-built datasets followed the production process used for the LineMOD dataset. Images of objects were captured from various angles in various poses under different lighting conditions using an RGB-D camera (Realsense2). This process yielded RGB images, depth images, and intrinsic camera parameter data for the objects. Aruco markers in the images were used for registration between pairs of images, and the matrix transformation between two sets of points was calculated. Through iterative registration and optimization, a globally consistent transformation was generated, producing pose change files (4 × 4 homogeneous matrices) for each image relative to the first frame. The depth image data were converted into point clouds and denoised. Meshlab, a mesh processing tool, was used to remove unwanted scene content, as shown in Figure 6. Finally, based on the pose changes and processed mesh, relocalization was performed to compute the 2D projection information of the new mesh in the camera's coordinates. Corresponding pixel masks (Figure 7) and label files representing the true pose information were generated.

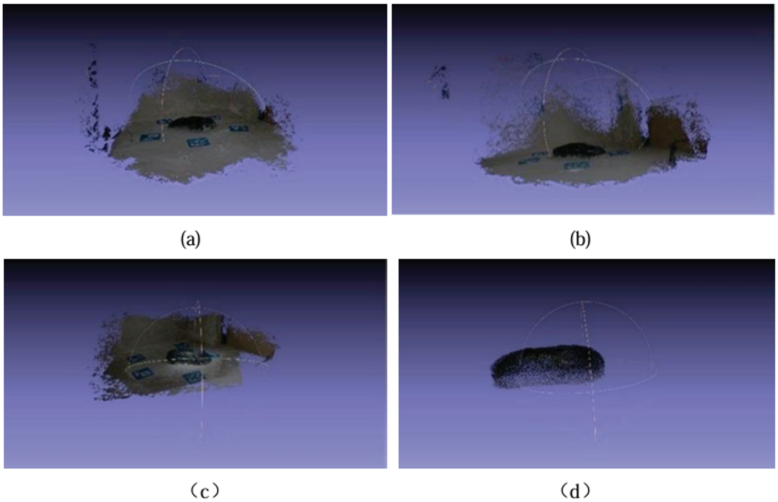


Figure 6. Point clouds were generated from images captured at different angles in various poses under different lighting conditions (**a**–**c**). The processing results (**d**).

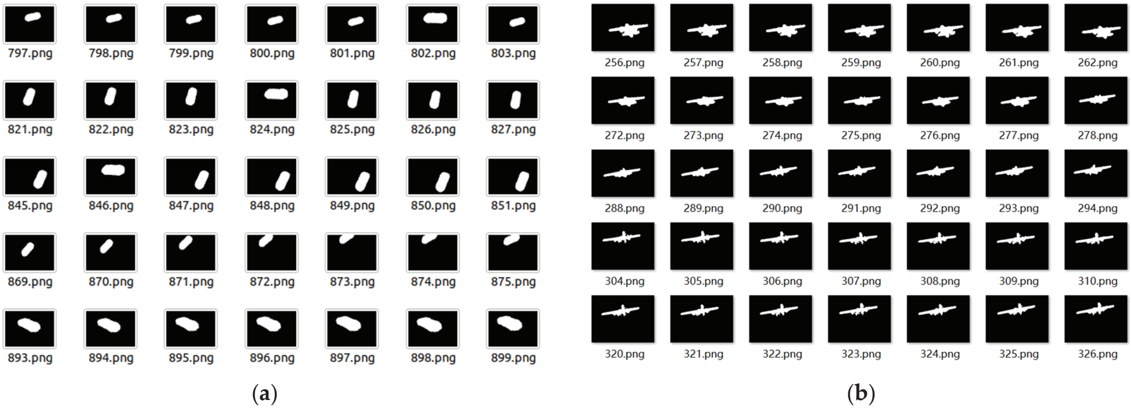


Figure 7. Generated mask images: (**a**) car model, (**b**) airplane model.

In the experiments, RGB images were used for training and testing, while depth images were used to obtain point cloud data to reconstruct 3D models. It should be noted that there may have been some differences in label accuracy between the self-built dataset and the LineMOD dataset, which may have resulted in the 2D projection performance being inferior for the self-built dataset compared to the LineMOD dataset in terms of precision.

4.2. Performance Evaluation Metrics

ADD and ADD-S, respectively, represent the average distance of points and the average distance of the nearest points, and are the most used evaluation metrics for rigid object pose estimation. ADD measures accuracy by computing the average distance between the 3D model points predicted by the pose estimation algorithm and the corresponding annotated model points. ADD-S extends ADD by considering symmetric objects and calculates distances based on the nearest point criterion for symmetric objects. The errors in translation and rotation for ADD(-S) metrics depend on the size and shape of the rigid body, and a threshold of 10% is typically set on the model's diameter. The 2D projection performance metric measures algorithm accuracy by projecting the model vertices of the target object onto the 2D image plane and computing the average error between the 2D projection points predicted by the pose estimation algorithm and the true annotated 2D vertices. In this paper, we primarily use these two metrics to evaluate the performance of our method.

4.3. Experimental Results

The experiments were performed on the Ubuntu 18.04 operating system utilizing the Torch 1.8 deep learning framework with CUDA version 11.1. In terms of hardware, an Intel(R) Xeon(R) Platinum 8255C twelve-core processor was employed as the CPU, while an RTX 3090 GPU was used. The performance of the improved method and baseline method was quantitatively determined on the LineMOD dataset in terms of the ADD(-S) and 2D projection metrics, respectively, as shown in Tables 1 and 2.

In the experiments on the LineMOD dataset, our proposed method outperformed most of the baseline methods in terms of the average ADD(-S) accuracy, with an average improvement of 1.93% compared to PVNet. Particularly, there was a significant improvement in pose recognition for objects with relatively large-scale variations, such as apes, cans, and ducks, with the maximum improvement reaching 13.04%. For the self-built dataset with lower precision and lower general image quality, the recognition accuracy of the proposed method was relatively superior, demonstrating better robustness for practical applications. Similarly, the self-built datasets of airplanes and car models also exhibited relatively large variations in object scale, indicating the excellent performance of our method in pose estimation for objects with significant scale changes. As shown in Table 2, the proposed method outperformed the baseline method in terms of the 2D projection performance accuracy. However, considering the correlation heatmap between the 2D projection performance estimation accuracy and the ADD estimation accuracy in Figure 8, where values closer to 1 indicate a higher positive correlation and values closer to -1 indicate a higher negative correlation, the experimental results suggest that there was no significant correlation between the 2D projection accuracy and the 6D pose estimation accuracy. Therefore, accurate 2D projection points do not necessarily improve the accuracy of 6D pose estimation.

Table 1. The ADD(-S) accuracy of our method and the baseline method for the LineMOD dataset and the self-built dataset. The middle column represents the pose estimation accuracy without refinement processing, while the right column represents the pose estimation accuracy after refinement processing. Datasets labeled with a superscript "+" indicate symmetric object datasets.

	Without Refinement					Refinement	
Method	**Zhao**	**PVNet**	**DPOD [24]**	**SSPE [25]**	**Ours**	**DPOD+**	**RePose [26]**
Ape	41.2	43.62	53.28	52.5	67.05	87.70	79.5
Benchvice	85.7	99.90	95.34	-	98.46	98.50	100.0
Cam	78.9	86.86	90.36	-	88.24	96.10	99.2
Can	85.2	95.47	94.10	99.2	97.65	99.70	99.8
Cat	73.9	79.34	60.38	88.5	85.86	94.70	97.9
Driller	77.0	96.43	97.72	98.8	93.66	98.80	99.0
Duck	42.7	52.58	66.01	68.7	75.24	86.30	80.3
Eggbox $^+$	78.9	99.15	99.72	100.0	99.72	99.90	100
Glue $^+$	72.5	95.66	93.83	98.5	75.86	96.80	98.3
Hole puncher	63.9	81.92	65.83	88.1	79.46	86.90	96.9
Iron	94.4	98.88	99.80	-	95.23	100.0	100.0
Lamp	98.1	99.33	88.11	-	98.85	96.80	99.8
Phone	51.0	92.41	74.24	-	87.90	94.70	98.9
Aircraft	-	71.72	-	-	79.80	-	-
Car	-	83.17	-	-	97.34	-	-
Average	72.6	85.09	82.98	86.8	88.02	95.15	96.1

Table 2. The 2D projection performance accuracy of our method and the baseline method for the LineMOD dataset. Datasets labeled with a superscript "+" indicate symmetric object datasets.

	BB8	**YOLO-6D**	**PVNet**	**Ours**
Ape	95.3	92.10	99.23	100.0
Bench vice	80.0	95.06	99.81	99.52
Cam	80.9	93.24	99.21	99.22
Jar	84.1	97.44	99.90	99.90
Cat	97.0	97.41	99.30	99.60
Driller	74.1	79.41	96.92	98.82
Duck	81.2	94.65	98.02	99.05
Eggbox $^+$	87.9	90.33	99.34	98.87
Glue $^+$	89.0	96.53	98.45	98.85
Hole puncher	90.5	92.86	100.0	99.62
Iron	78.9	82.94	99.18	98.98
Lamp	74.4	76.87	98.27	97.70
Phone	77.6	86.07	99.42	99.52
Average	83.9	90.37	99.00	99.20

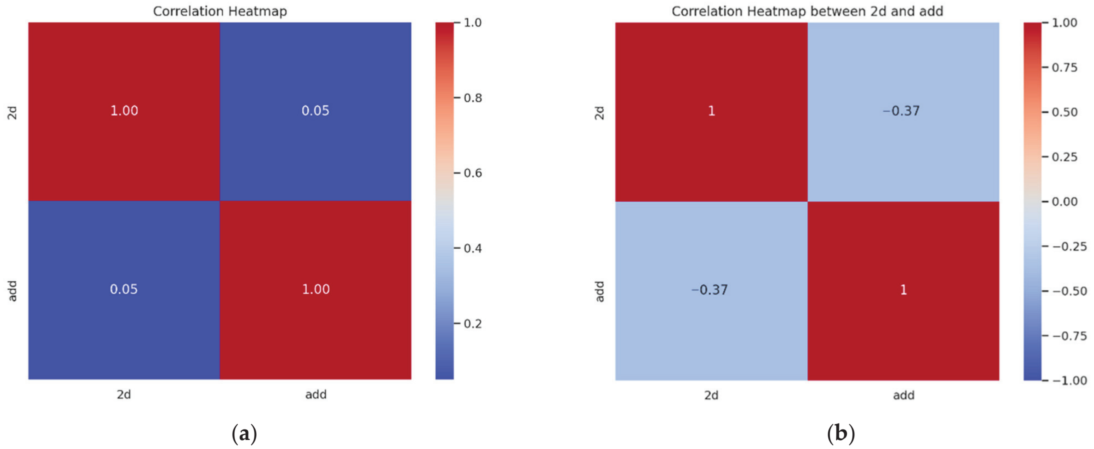


Figure 8. (**a**) Correlation heatmap between the 2D projection performance estimation accuracy and the ADD estimation accuracy using the PVNet method; (**b**) correlation heatmap between the 2D projection performance estimation accuracy and the ADD estimation accuracy using our method.

In Figure 9, we compare the training curves of the baseline method and our method. With the same number of training steps, our method converged faster, had a shorter training time per epoch, exhibited smaller fluctuations when dealing with anomalies, and demonstrated better generalization. Figure 10 illustrates the intuitive visualization results obtained using the improved method for both the LineMOD dataset and our self-built dataset.

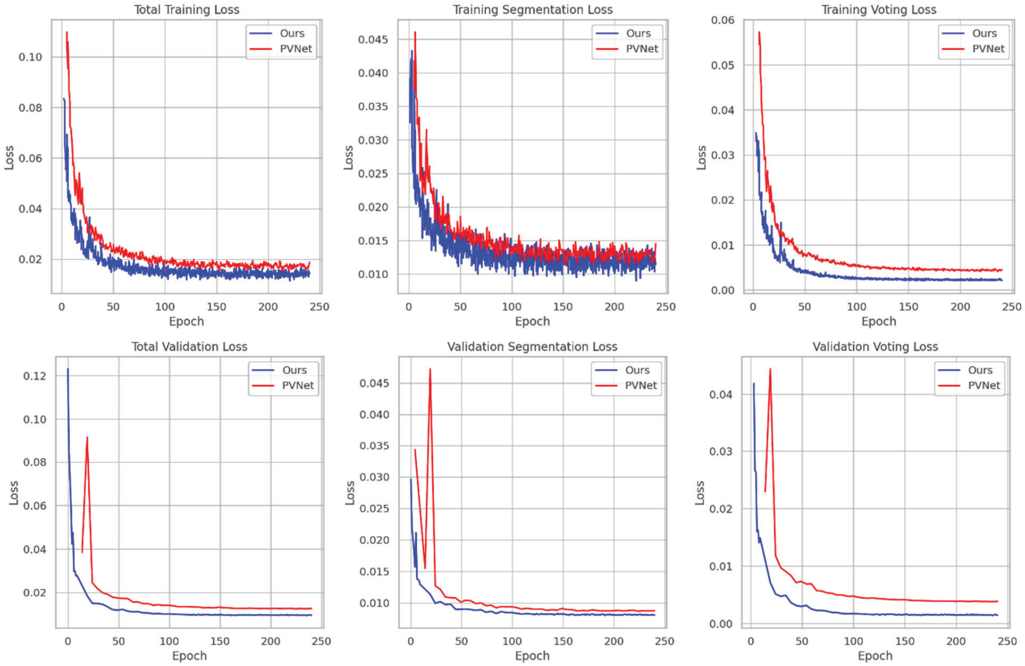


Figure 9. The training loss variation plot for the PVNet method and our proposed method. The orange line represents the PVNet method while the blue line represents our method (based on the self-built car dataset).

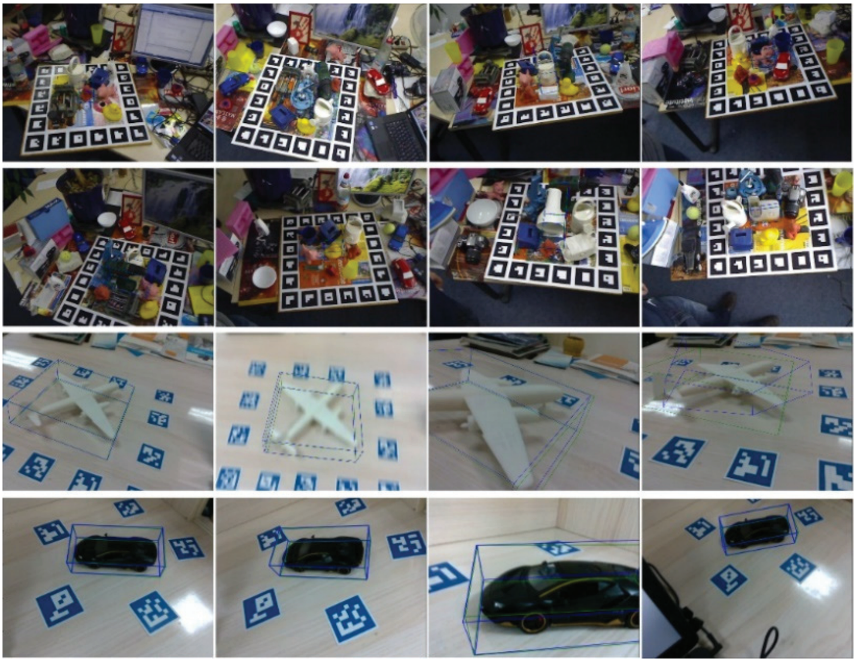


Figure 10. Visualization results of pose estimation using our method on the dataset. The green box outlines indicate the ground truth object pose information while the blue box outlines indicate the estimated object pose information obtained through inference. The top two rows display the pose estimation results of some objects in the LineMOD dataset while the bottom two rows demonstrate the performance on the airplane and car model datasets, including the handling of blur, illumination variations, viewpoint changes, scale variations, and occlusion, as well as instances of erroneous estimations.

4.4. Ablation Experiment

Table 3 shows the performance of the improved networks trained on the convex dataset in LineMOD using different numbers of layers and modules, including the ADD and 2D projection performance, as well as the corresponding network parameters, computational complexity, and GPU inference time.

As the number of network layers increased, the accuracy of object pose estimation improved. However, further deepening the network layers due to the increased number of channels at the end led to a rapid growth in the number of parameters, resulting in less noticeable improvements in estimation accuracy. Maintaining a certain number of network layers and reinforcing spatial and channel information by adding coordinate attention at the end or introducing residual connections within the convolutional network for layers with the same input and output channels could significantly enhance the estimation accuracy without a substantial increase in parameters or computational load. The ASPP module, incorporating an attention mechanism, effectively fused features at multiple scales, further enhancing the accuracy and robustness of pose estimation, particularly positively impacting small-diameter object estimation. Considering the superior performance of our method on the self-built airplane and car model datasets, it can be concluded that this module exhibits good robustness for significant scale variations.

A comparison of the PVNet method with the lightweight model parameters is shown in Table 4. The improved method reduced the number of parameters by 57.69%, decreased the computational load by 31.47%, and reduced the model weight by 57.43%. This makes it more suitable for deployment on mobile devices and embedded systems, thus expanding the application scenarios for object pose estimation. However, in terms of the GPU computing time, the improved method slightly trailed the original method. This is because the split two-step calculation mode of depth-wise separable convolution operations does not fully leverage the benefits of high-performance GPU stream parallel computing, leading to performance waste and a slight decrease in computational speed.

Table 3. DC stands for networks constructed using depth-wise separable convolution, with the number indicating the network layers. CA represents the coordinate attention module, Res indicates convolutional layers with residual connections, and ASPP denotes the ASPP module.

	ADD	2D Projection	Params	Computational Cost	Time
DC10	54.81	96.57	0.7 M	16.3 B	7.03 ms
DC11	54.51	97.16	0.8 M	16.7 B	7.18 ms
DC15	65.69	98.33	2.5 M	21.9 B	9.13 ms
DC10 + CA	63.43	98.33	0.7 M	16.3 B	7.15 ms
DC11 + CA	62.84	98.43	0.8 M	16.7 B	7.31 ms
DC11 + Res	62.06	97.45	0.8 M	16.7 B	7.36 ms
DC13 + Res	68.33	98.63	1.3 M	16.1 B	7.61 ms
DC13 + Res + CA	77.65	99.12	4 M	27.2 B	11.67 ms
DC13 + Res + CA + ASPP	88.24	99.22	5.5 M	49.8 B	14.10 ms

Table 4. Before and after model parameter comparison.

	Params	Computational Cost	Weight Size	GPU Frame Computation Time
PVNet	13.0 M	72.7 B	148 MB	13.34 ms
Ours	5.5 M	49.8 B	63 MB	14.10 ms

5. Conclusions

In this paper, an improved algorithm was proposed aimed at balancing the accuracy and simplicity of object pose estimation networks while enhancing the robustness for scale variations. A lightweight semantic segmentation model using depth-wise separable convolutions for keypoint detection was constructed, integrating a coordinate attention mechanism and an improved ASPP module to effectively fuse multi-scale information and increase receptive fields. As a result, a lightweight object pose estimation network with both accuracy and robustness was obtained. Through experiments on the LineMOD and self-built datasets, this method demonstrated certain improvements in average accuracy, particularly in pose estimation for objects with significant scale variations and low image quality, compared to baseline methods. Moreover, compared to previous methods, in this approach, the number of parameters, computational complexity, and weights were significantly reduced, making it more suitable for deployment on low-performance devices such as mobile platforms. However, this method exhibited poor recognition accuracy for the LineMOD glue dataset. Glue is a small, symmetrical object, causing potential overfitting during training and thus resulting in a decrease in accuracy. On the premise of not increasing the complexity of the model, we believe that a more efficient and suitable 2D–3D key point solution method can play a significant role in further improving the pose estimation accuracy.

Author Contributions: Conceptualization, F.W. and Y.W. (Yadong Wu); methodology, F.W and X.T.; software, X.T. and Y.W. (Yinfan Wang); validation, H.C., G.W., and J.L.; writing—original draft preparation, F.W.; writing—review and editing, Y.W. (Yinfan Wang); visualization, X.T.; supervision, Y.W. (Yadong Wu). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61872304.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author due to privacy.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
2. Viswanathan, D.G. Features from accelerated segment test (fast). In Proceedings of the 10th Workshop on Image Analysis for Multimedia Interactive Services, London, UK, 6–8 May 2009; pp. 6–8.
3. Mohammad, S.; Morris, T. Binary robust independent elementary feature features for texture segmentation. *Adv. Sci. Lett.* **2017**, *23*, 5178–5182. [CrossRef]
4. Hinterstoisser, S.; Lepetit, V.; Ilic, S.; Holzer, S.; Bradski, G.; Konolige, K. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In *Asian Conference on Computer Vision, Computer Vision—ACCV 2012 Proceedings of the11th Asian Conference on Computer Vision, Daejeon, Korea, 5–9 November 2012*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 548–562.
5. Kendall, A.; Grimes, M.; Cipolla, R. Posenet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2938–2946.
6. Li, Y.; Wang, G.; Ji, X.; Xiang, Y.; Fox, D. Deepim: Deep iterative matching for 6d pose estimation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 683–698.
7. Xiang, Y.; Schmidt, T.; Narayanan, V.; Fox, D. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In Proceedings of the Robotics: Science and Systems (RSS), Pittsburgh, PA, USA, 26–30 June 2018.
8. Wang, C.; Xu, D.; Zhu, Y.; Martin-Martin, R.; Lu, C.; Li, F.-F.; Savarese, S. Densefusion: 6d object pose estimation by iterative dense fusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3343–3352.
9. Sundermeyer, M.; Marton, Z.-C.; Durner, M.; Brucker, M. Implicit 3d orientation learning for 6d object detection from rgb images. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 699–715.
10. Wang, G.; Manhardt, F.; Tombari, F.; Ji, X. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 16611–16621.
11. Rad, M.; Lepetit, V. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3828–3836.
12. Zhao, Z.; Peng, G.; Wang, H.; Fang, H.-S.; Li, C.; Lu, C. Estimating 6D pose from localizing designated surface keypoints. *arXiv* **2018**, arXiv:1812.01387.
13. Peng, S.; Liu, Y.; Huang, Q.; Zhou, X.; Bao, H. Pvnet: Pixel-wise voting network for 6dof pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4561–4570.
14. Park, K.; Patten, T.; Vincze, M. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Repulic of Korea, 7 October–2 November 2019; pp. 7668–7677.
15. Chen, B.; Chin, T.J.; Klimavicius, M. Occlusion-robust object pose estimation with holistic representation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 2929–2939.
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 7–30 June 2016; pp. 770–778.
17. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
18. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
19. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
20. Luo, W.; Li, Y.; Urtasun, R.; Zeme, R. Understanding the effective receptive field in deep convolutional neural networks. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016; p. 29.

21. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef] [PubMed]
22. Brachmann, E.; Krull, A.; Michel, F.; Gumhold, S.; Shotton, J.; Rother, C. Learning 6D object pose estimation using 3D object coordinates. In *European Conference on Computer Vision, Computer Vision—ECCV 2014, Proceedings of the13th European Conference, Zurich, Switzerland, 6–12 September 2014*; Springer: Cham, Switzerland, 2014; pp. 536–551.
23. Hodan, T.; Haluza, P.; Obdrzalek, S.; Matas, J.; Lourakis, M.; Zabulis, X. T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-Less Objects. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; IEEE: Piscataway, NJ,USA, 2017. [CrossRef]
24. Zakharov, S.; Shugurov, I.; Ilic, S. Dpod: 6d pose object detector and refiner. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Repulic of Korea, 7 October–2 November 2019; pp. 1941–1950.
25. Gupta, A.; Medhi, J.; Chattopadhyay, A.; Gupta, V. End-to-end differentiable 6DoF object pose estimation with local and global constraints. *arXiv* **2020**, arXiv:2011.11078.
26. Iwase, S.; Liu, X.; Khirodkar, R.; Yokota, R.; Kitani, K.M. Repose: Fast 6d object pose refinement via deep texture rendering. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 3303–3312.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

Article

Deep Neural Network Confidence Calibration from Stochastic Weight Averaging

Zongjing Cao [1], Yan Li [1], Dong-Ho Kim [2] and Byeong-Seok Shin [1,*]

[1] Department of Electrical and Computer Engineering, Inha University, Incheon 22212, Republic of Korea; zjcao@inha.edu (Z.C.); leeyeon@inha.ac.kr (Y.L.)

[2] Global School of Media, Soongsil University, Seoul 06978, Republic of Korea; dkim@ssu.ac.kr

* Correspondence: bsshin@inha.ac.kr; Tel.: +82-32-860-7452

Abstract: Overconfidence in deep neural networks (DNN) reduces the model's generalization performance and increases its risk. The deep ensemble method improves model robustness and generalization of the model by combining prediction results from multiple DNNs. However, training multiple DNNs for model averaging is a time-consuming and resource-intensive process. Moreover, combining multiple base learners (also called inducers) is hard to master, and any wrong choice may result in lower prediction accuracy than from a single inducer. We propose an approximation method for deep ensembles that can obtain ensembles of multiple DNNs without any additional costs. Specifically, multiple local optimal parameters generated during the training phase are sampled and saved by using an intelligent strategy. We use cycle learning rates starting at 75% of the training process and save the weights associated with the minimum learning rate in every iteration. Saved sets of the multiple model parameters are used as weights for a new model to perform forward propagation during the testing phase. Experiments on benchmarks of two different modalities, static images and dynamic videos, show that our method not only reduces the calibration error of the model but also improves the accuracy of the model.

Keywords: confidence calibration; deep ensemble learning; stochastic weight averaging

Citation: Cao, Z.; Li, Y.; Kim, D.-H.; Shin, B.-S. Deep Neural Network Confidence Calibration from Stochastic Weight Averaging. *Electronics* **2024**, *13*, 503. https://doi.org/10.3390/electronics13030503

Academic Editors: Wentao Li, Huiyan Zhang, Tao Zhan and Chao Zhang

Received: 21 December 2023
Revised: 19 January 2024
Accepted: 23 January 2024
Published: 25 January 2024

1. Introduction

During training, a deep neural network (DNN) learns the output probability, which indicates the DNN's confidence in the results. Some recent work has found that the confidence of a DNN is not consistent with its accuracy [1–3]. These works point out that DNNs suffer from overconfidence. An overconfident DNN gives high confidence in wrong predictions. The problem of overconfidence poses a huge challenge to the deployment of DNNs in real-world applications. For example, in health care, criminal justice, and autonomous driving applications, we expect models to have a certain degree of confidence in their predictions in order to make more informed decisions. Confidence calibration not only enhances the model's ability to generalize and minimizes potential risks but also greatly aids in its interpretability [4,5].

Confidence calibration is the process of adjusting the predicted probabilities of a model to better reflect the true likelihood of its predictions being correct [2,6]. More formally, a completely calibrated classification model is one in which the probability of the predicted outcome $\hat{Y}$ being equal to the actual outcome Y is defined as $\mathbb{P}(\hat{Y} = Y \mid \hat{P} = p) = p$, where p falls within the range of $[0,1]$, and $\hat{P}$ is the model's associated confidence. It is expected that $\hat{P}$ will be calibrated, indicating that it accurately reflects an actual probability. An accurately-calibrated classifier is a probabilistic classifier that can be directly interpreted in terms of confidence score through its predicted probability output. To illustrate, a precisely calibrated (binary) classifier that yields 100 samples with a confidence score of 0.6 for every prediction indicates that 60 samples will be accurately classified. Confidence calibration refers to a model's capacity to accurately assign probabilities to its predictions [7].

In recent years, multiple techniques have been introduced to generate predictive confidence scores through calibration [2,8,9], including post-processing, Bayesian neural networks, and deep ensemble methods. Temperature scaling is a post-processing calibration method proposed by Guo et al. [2]. The main principle revolves around the utilization of a singular scalar parameter, $T > 0$, denoting the temperature, to modify the logit score prior to the implementation of the softmax function. The method cannot effectively handle out-of-distribution data because T is computed on the validation set. The idea behind the Bayesian neural network approach is to infer the probability distribution of the DNN parameters. This distribution is used to sample the parameters for single forward propagation, resulting in random predictions that are influenced by diverse model weights. However, precise Bayesian inference is computationally difficult for neural networks and incurs extremely expensive computational and memory costs [3].

Deep ensemble learning [10] combines the predictions of several base estimators to reduce the variance of predictions and reduce generalization errors. The concept of ensembling is based on the idea that a group of models can work together to enhance the strengths and minimize the weaknesses of individual base learners. Deep ensembles were originally proposed and discussed to improve the prediction performance of DNNs. In [11], the authors, through the experimental analysis of several regression and classification tasks, showed that averaging the predictions of ensemble models can also be used to derive useful uncertainty estimates. Moreover, in [12], deep ensembles were shown to be state-of-the-art for the domain shift (or out-of-distribution) setting. Compared to single-model methods [2], the computational costs and memory consumption of the deep ensemble approach are significantly higher. Moreover, the additional computations increase linearly with the number of base learners. Some interesting methods, such as distillation, sub-ensembles, and batch ensembles, have been proposed to solve these problems. However, these approaches necessitate substantial changes to the training process and remain costly in regards to both time and computational resources. To overcome these challenges, work in [13] presented an approximate method to implement an ensemble of models without increasing the training cost. During the training phase, multiple *snapshots* of the model are periodically saved, and the predictions from the multiple *snapshots* are averaged during the testing phase. Instead of training M models from scratch, the snapshot ensembles in [13] were created by changing the learning rate to allow the optimizer to reach the local minimum M times during the optimization process. In [14], the study demonstrated that simple curves connect the optima of complex loss functions, with consistent training and test accuracy. Based on this geometric finding, they proposed a new approximate ensembling procedure called fast geometric ensembling (FGE). The FGE algorithm uncovers various networks by taking small steps in the weight space while remaining in a low test error region. FGE allows training of highly effective ensembles in the same amount of time it takes to train a single model. However, these single-model multiple-weight methods were initially proposed to improve the accuracy of the DNN but with little attention paid to confidence in the output. It remains unclear whether these methods are effective in reducing confidence errors.

To fill this gap, we propose a confidence calibration method based on stochastic weight averaging. We achieve this by training a single DNN to converge to multiple local minima on the loss surface and to sample and save the model parameters by using a smart strategy. At a high level, the concept of averaging the stochastic gradient descent (SGD) iterations has been around for several decades in the field of convex optimization [9,15]. In convex optimization, researchers have primarily focused on optimizing convergence rates by implementing averaged SGD. In deep learning, the use of averaged SGD results in a smoother trajectory for SGD iterations but yields minimal differences in performance. By contrast, we are more concerned in this work with the effect of the method on the calibration error. Multiple locally optimal weights generated in the training phase are specially sampled, and then forward propagation is computed as new parameters for the base learner during the testing phase. We force the optimizer to explore a variety of models rather than converge on just one solution by using a modified learning rate strategy. We use a smart strategy to select relatively more meaningful weights from a large number of candidate weights. To summarize, instead of designing multiple sets of DNNs, the method simply trains a single model to obtain well-calibrated confidence output. We tested our method on two benchmarks with varying modalities, including static images and dynamic videos. Our experimental findings indicate that our method effectively minimizes calibration error and enhances the model's precision. The code is publicly available at github.com/zjcao/swaCal (accessed on 22 January 2024).

The main contributions of our work are summarized as follows.

(1) We propose an alternate ensemble learning approach to improve the quality of neural network uncertainty measures to overcome overconfidence without incurring additional computational costs.

(2) We evaluate our approach using two benchmarks with different modalities: static images and dynamic videos. The results of our experiments demonstrate that our approach successfully reduces calibration error and enhances the model's accuracy.

The remainder of this paper is structured as follows. Section 2 discusses related studies on deep ensemble learning and confidence calibration. Section 3 describes our proposed approach in detail. The experiments and results are presented in Section 4. Section 5 is the conclusion and suggests further studies.

2. Related Work

2.1. Confidence Calibration of a Deep Neural Network

Confidence calibration is a sub-task of open-set recognition that aims to improve the accuracy of confidence scores for DNN output. DNN output is the probability during an inference process that indicates the model's confidence in the result. A precisely calibrated confidence can represent the probability that the predicted label is correct. Although DNNs have obtained good prediction accuracy in a variety of visual tasks, recent studies have found that DNNs suffer from overconfidence [2,7,8]. For a classification task, data scientists usually use softmax output (predicted probability) as the true probability of correctness in the predicted category. This might have been reasonable for traditional network models in the past, but it is not applicable to the DNNs of today. In [16], the authors found that passing a point estimate through the softmax function produced a high probability. After that, Guo et al. [2] found the same problem and demonstrated through a series of ablation experiments that model depth and width, batch normalization, and weight decay have strong effects on the confidence calibration of DNNs. The process of calibrating a classifier involves creating a calibrator that translates the classifier's output into a calibrated probability ranging from 0 to 1. The calibrator attempts to predict the conditional probability of the event, $p(y_i = 1/f_i)$ based on the classifier's output, f_i, for a specific sample.

The confidence calibration methods can be classified into three types depending on the approach: (a) regularization methods during the training phase, (b) post-processing methods after the training phase, and (c) DNN-based uncertainty estimation methods. Regularization-based confidence calibration methods are performed by changing the objective function or by augmenting the training dataset during the DNN training process. Label smoothing, data augmentation, and objective function modification are three commonly used methods for confidence calibration based on regularization [17–19]. On the other hand, various methods have been developed in the last decade to recalibrate model confidence through post-processing steps. Temperature scaling, proposed in [2], is a simple and effective technique for recalibrating the prediction probability of modern neural networks. The core algorithm used for temperature scaling rescales the logit value, $f_i(x)$, by using a single scale parameter, T (temperature), before passing it to the softmax function: $softmax(f_i/T)$. T is extracted by minimizing the negative log-likelihood on the validation datasets after the model training is complete. Although temperature scaling is an effective way to quantify the total predictive uncertainty of calibrated probabilities, it cannot capture the uncertainty caused by out-of-distribution data [12]. By decreasing model uncertainty, the confidence prediction of a DNN can be better calibrated. The rationale for this is that the remaining uncertainty in the predicted data more accurately reflects the true uncertainty in the prediction. Bayesian and ensemble methods are two methods that are used to estimate model uncertainty. The key idea of the Bayesian method is to infer the probability distribution over the model parameters. The prior distribution of the neural network parameters is specified, and then the posterior distribution of the parameters is calculated using the training data. Finally, the uncertainty of the model is predicted according to Bayesian theory. The main challenge for Bayesian deep learning is to specify meaningful conjugate priors for the parameters. Moreover, exact Bayesian inference is often computationally difficult for neural networks with a large number of parameters. Therefore, approximate Bayesian inference techniques, such as variational inference, Laplace approximation, and Markov chain Monte Carlo (MC), are usually used to calculate posterior probabilities [1,6]. Although the Bayesian neural network can estimate the uncertainty of the prediction, the inference phase procedure requires substantial modification due to the inclusion of Bayes' law. In addition, specifying meaningful priors for Bayesian neural networks is a big challenge [3,4].

2.2. Deep Ensemble Learning

The deep ensemble learning approach merges the benefits of deep learning and ensemble learning, resulting in a final model with improved generalization abilities [20]. The main idea behind deep ensemble learning is that by combining several models, the deficiency of a single base learner may be compensated for by other base learners so the overall predictions of the ensemble are better than a single base learner. The ensemble method for deep learning is roughly divided into two steps: (1) training different models (the training phase) and (2) merging prediction results (the inference phase). In the training phase, multiple models can be obtained using different model architectures, training data, and training strategies. In the inference phase, the same input can be provided to the model for the prediction. Finally, the prediction of each model is combined according to a certain strategy to obtain the final prediction. The deep ensemble method can usually be one of two types: the averaging method and the boosting method [5,6]. The core idea of the averaging method is to build several estimators independently and then average their predictions, as is performed in the bagging method and random forest. Boosting methods such as AdaBoost and gradient tree boosting create the base estimators sequentially to reduce the bias of the combined models. Boosting and random forest are classical machine learning ensemble techniques and complementary methods [21].

In deep learning, dropout is designed as a regularization technique for neural networks to avoid overfitting, which can also be interpreted as an ensemble of multiple models. In [16], the authors showed that MC dropout can be used to quantify the uncertainty of the model. However, in [11], the authors found that in various datasets and tasks of regression and classification, deep ensemble methods were superior to MC dropout in quantifying uncertainty. Furthermore, the deep ensemble method has been shown to be robust in quantifying uncertainty for data beyond distribution [12,22]. However, the computational costs and memory consumption of the ensemble method are significantly higher than in other methods. Therefore, how to effectively reduce the computational workload and memory consumption has become a new research topic in the area of ensemble methods. Pruning methods [23] reduce the complexity of the ensemble by pruning members and reducing redundancy among members. Other approaches, such as batch ensembles and sub-ensembles, attempt to reduce computational cost and memory usage by sharing portions among individual members. We propose an approximate ensemble learning approach that is simple to implement, requires very little hyperparameter tuning, and achieves comparable performance.

3. Confidence Calibration

3.1. Approximate Deep Ensemble Learning

Confidence calibration is the degree to which the uncertainty of the prediction matches the true underlying uncertainty in the data. The softmax probability score is commonly employed by scientists as a confidence metric for the predictions in image classification tasks. Studies have shown that DNN prediction scores can be either overly confident or lacking in confidence. Deep ensemble learning methods reduce prediction variance and improve robustness and generalization by combining prediction results from multiple DNNs constructed using a specific learning strategy. The main principle behind the ensemble approach is to construct several independent estimators and then average their predictions to yield the final prediction. Due to its reduced variance, the combined estimator achieves better results than any single base estimator. Formally, given sample x, the ensemble prediction, $p(y|x)$, is estimated by averaging the predictions of all the models, which can be expressed as $p(y|x) = \frac{1}{M}\sum_{i=1}^{M} p_i(y|x)$, where $p_i(y|x)$ is the predicted output of the i-th base estimator, and m is the number of models in deep ensemble learning. It has been shown that a DNN ensemble can improve the robustness and accuracy of the system compared to using individual networks. However, the computational burden of training multiple DNNs for the ensemble is considerable. Mastering the combinations of ensemble models can be challenging, and an incorrect selection could lead to reduced prediction accuracy compared to using a single model.

On the other hand, in [11], the authors showed that the utilization of an ensemble method for averaging DNN predictions can provide reliable estimates of uncertainty. In addition, the authors in [12] pointed out that a deep ensemble not only performs well in quantifying the uncertainty of a model but also has good robustness against out-of-distribution data. However, little attention has been paid to whether the confidence in the output predictions of ensemble models is representative of the true probability. The ensemble method requires more training time and computations compared to other confidence calibration methods, such as Bayesian neural networks and the post-calibration method.

We propose using the approximate ensemble method to calibrate the confidence of DNNs. We utilize a cyclical learning rate to collect models that are spatially close to each other but that produce diverse predictions. Instead of individually training multiple base estimators, our method performs a single supervised training session to obtain well-calibrated confidence scores. Assume that the training data, D, has N pairs of independent and identically distributed samples, denoted $D = \{x_n, y_n\}_{n=1}^{N}$, where $x \in \mathbb{R}^d$ represents the d-dimensional features, while y is the label for a classification task and is one of K classes (i.e., $y \in \{1, \ldots, K\}$). Given the input data, x, our task is to use the DNN to model the probabilistic predictive distribution of the labels, $p_\theta(y|x)$, where θ is the weight parameters of the DNN. During the training phase, we utilize the training set D to conduct standard supervised learning on the model via the proper scoring rules (which is further explained in later sections). The parameters of DNNs are often several orders of magnitude greater than the training data points. That is, they include a large possible function space that may be very close to the data generation function. Thus, there are multiple low-loss valleys (local optimums) during the whole learning period, all corresponding to good but different functions (here, we call them candidate functions). These candidate functions represent varying assumptions used to determine the underlying fundamental function. The more candidate functions in the ensemble, the more likely it is to represent the truth, thus, constructing a more robust model. Figure 1 shows the schematic of the loss landscape based on SGD optimization. We can see that several local optima appear throughout the training process. Note that the valley where the loss finally arrives is not considered the global optimum either.

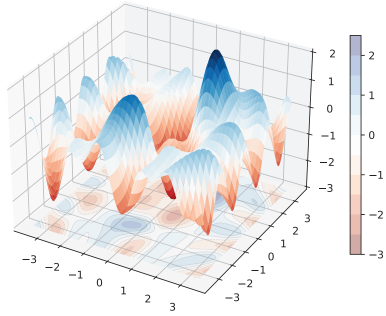


Figure 1. Illustration of the 3D loss landscape with SGD optimization of the DNN during the training phase.

The weights of the model are updated several times in each iteration cycle. The sampling and saving of these weights are intuitively important to our method. In general, four metrics are counted in each iteration cycle: training accuracy, training loss, validation accuracy, and validation loss. Instead of saving the model using the validation accuracy, we are using a cyclic learning rate for the last 25% training times and saving the network weights corresponding to the lowest value of the learning rate in each cycle. That is, when we save the model we only care whether the current cycle learning rate is at its lowest. Afterwards, we perform stochastic weighted averaging on the multiple sets of weights obtained.

3.2. Stochastic Weighted Averaging

During the training stage, the multiple local optimal weights generated during the training process are sampled and simulated as new parameters of the base learner for inference. Since the loss trajectories and weight values are different, this leads the ensemble model to make diverse predictions. After the training stage, we save m model weights, $\theta_1, \theta_2, \ldots, \theta_m$ each of which are used in the final ensemble. Figure 2 illustrates the data flow of fusion of our approach. In the inference phase, the given input is fed to multiple base learners to predict output. The parameters of the multiple base learners are sampled from the training process by a smart sampling strategy. The method allows for the training of highly effective ensembles in the same amount of time it takes to train a single DNN.

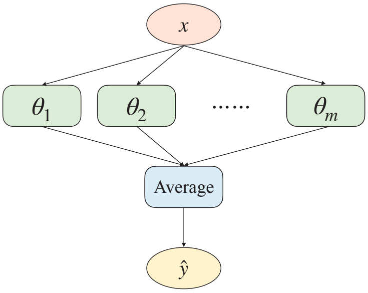


Figure 2. The data flow of fusion during the testing phase of our method, where $\theta_1, \theta_2, \ldots, \theta_m$ indicate the set of m base estimators, and $\hat{y}$ represents the output of base estimator, θ_m, on sample x.

In this work, we treated the ensemble as a uniformly weighted mixture model and combined the predictions as follows:

$$p(y|x) = \frac{1}{m}\sum_{i=1}^{m} p_{\theta_i}(y|x, \theta_i), \tag{1}$$

where m denotes the number of DNNs in the ensemble, and θ_i indicates the parameter of the DNN. For classification tasks, this corresponds to averaging the predicted probabilities. The difference is that we output the logit values, $p(y|x)$, of multiple base learners after weighted averaging; that is, only a set of confidence scores is obtained. The final prediction result of x is then output according to this score. We hypothesized that weighted averaging of multiple confidence scores could effectively reduce variance and produce well-calibrated outputs, and the following experiments proved our hypothesis.

4. Experiment Results

We evaluated our approach using two benchmark datasets with different modalities: static images and dynamic videos. In both cases, we followed standard training, validation, and testing protocols. We evaluated the quality of the confidence score estimation and the accuracy of the prediction. We show across two different datasets that our method improves confidence scores without reducing classification error.

4.1. Evaluation Calibration Quality

Before showing experiments to recalibrate the classifier, the metrics for evaluating the effectiveness of the calibration of the classifier need to be presented. Proper scoring rules measure the quality of predictive uncertainty. A scoring rule assigns a numerical score to a predictive distribution, $p(y|x)$, rewarding better-calibrated predictions over worse ones. Negative log-likelihood (NLL) is a popular and proper scoring rule for multi-class classification tasks when measuring the accuracy of predicted probabilities. Given probabilistic model $p(y|x)$ and n samples, NLL is defined as follows:

$$\mathcal{L} = -\sum_{i=1}^{n} \log(p(y_i|x_i)). \tag{2}$$

In the field of deep learning, NLL is also known as cross-entropy loss. In this work, we use NLL as a training criterion.

To quantify the quality of the given model's confidence calibration, we use the following evaluation metrics: expected calibration error (ECE), maximum calibration error (MCE), and root mean square calibration error (RMSCE) [24]. ECE measures the correspondence between the probability and the accuracy of the prediction. It is computed as the average gap between within-bin accuracy, and within-bin predicted probability for m bins and can be expressed as follows:

$$\mathrm{ECE} = \sum_{i}^{N} b_i \|(p_i - c_i)\| \tag{3}$$

where b_i indicates the fraction of data points in bin i, p_i presents the average accuracy in bin i, and c_i indicates the average confidence in bin i.

In contrast to evaluation metrics, reliability diagrams are a visual representation of the quality of the model's confidence calibration. It plots the true frequency of a classifier's correctly classified labels against the predicted probability. Note that if a DNN is perfectly calibrated, the diagram should plot the identity function.

4.2. Application 1: Gesture Recognition Task

Computers can interpret human gestures as commands through gesture recognition, which is a form of perceptual computing user interface. The main objective of gesture recognition is to categorize a gesture video clip into a specific action group. Gesture recognition technology is highly applicable in various industries, including robot control, autonomous driving, and virtual reality. The gesture recognition model should not only correctly understand the command of the gesture but should also have a certain confidence in the prediction. Unlike static images, the uncertainty of the deep learning model on dynamic video is usually more difficult to capture. To evaluate the effectiveness of our method more comprehensively, we first tested it on a video-based gesture recognition dataset. A precisely calibrated gesture recognition system functions as a probabilistic classifier, allowing for the direct interpretation of the predicted probability output as a confidence score. This probability gives a certain level of confidence in the prediction.

The publicly available Jester dataset [25] was used to asses the proposed method. It contains 148,092 videos of people making standard hand gestures in front of a laptop or webcam. We split the dataset into two categories (a closed set and an open set) and proceeded to create smaller sets by randomly selecting data from the closed set at a ratio of approximately 1:4. There are 20 class gestures within the closed mini-datasets, with a split of 8:1:1 for training, validation, and testing. The training, validation, and testing sets contain 22,000, 2400, and 2240 gesture samples, respectively. We trained, validated, and tested all models on the closed Jester mini-dataset in this study.

We utilized the PyTorch 2.1 deep learning framework to implement the proposed network. The training and validation were conducted on a server containing four NVIDIA GeForce RTX 3090 GPUs. In our study, we employed a 3D ResNet-18 model that was initially trained on Kinetics-400. We then proceeded to fine-tune this model using the Jester training set. We utilized SGD with a momentum of 0.9 as the optimizer and conducted training on the model for 25 epochs with a batch size of 16. Following the settings from [15], during the first 75% of training, we adopted the standard decaying learning rate strategy, followed by a consistent and high learning rate for the remaining 25% (as shown in Figure 3). The use of a modified learning rate scheme is expected to keep the optimizer bouncing around the optimum, exploring different models rather than simply converging to a single solution. Our model obtained 90.02% accuracy on the training set at the 25th epoch, with a corresponding cross-entropy loss of approximately 0.28. At the same time, the validation set achieved its best rate of 86.35% at the 21st epoch.

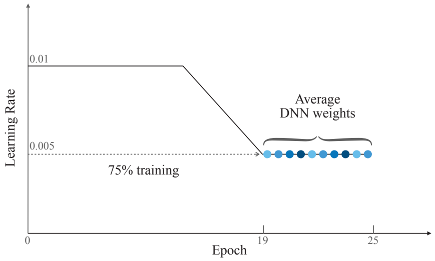


Figure 3. Illustration of the learning rate schedule. During the initial 75% of training, a standard decaying schedule is employed, followed by a high constant value for the remaining 25%. The dots of different colors represent the weights of the model in different training epochs.

In addition to evaluating the expected calibration error using ECM, MCE, and RMSCE, we also report the accuracy and average confidence in the model's classifications. The results obtained from the Jester test dataset are summarized in Table 1. We evaluated our method on multiple scoring functions, including SGD (baseline), MC-Dropout [16], and Logits-Scaling [26]. On the Jester test set, our method reduced the ECE error from 3.76% to 1.16%. Meanwhile, we observed that our method not only significantly reduced ECE and MCE but also improved the prediction accuracy of the model to a certain extent. For example, classification accuracy increased from 83.89% to 84.44%. The experiment report further illustrates the effectiveness of our method in improving the quality of the confidence estimation from DNNs in gesture recognition tasks.

Table 1. Summary of classification accuracy, average confidence, RMSCE, MCE, and ECE, compared to other methods, as obtained from the Jester test set. All values are percentages; (↑) indicates that higher values are desirable, (↓) means lower values are better, and bold indicates the best results.

Method	Jester Test Set				
	Acc. (↑)	Conf. (↑)	RMSCE (↓)	MCE (↓)	ECE (↓)
Baseline (SGD)	83.89	87.65	8.73	4.53	3.76
MC-Dropout	84.21	89.91	3.70	8.34	4.83
Logits-Scaling	83.32	87.21	5.19	4.66	4.24
Ours	**84.44**	**85.09**	3.71	**2.49**	**1.16**

We used visualization to gain an intuitive understanding of the calibration of the predicted probabilities. The reliability diagram illustrates the degree of calibration in the probabilistic predictions of a classifier. The reliability diagram displays the average predicted probability on the x-axis for each bin, while the y-axis reflects the fraction of positive samples, indicating the proportion of samples with positive categories in each bin. The diagram should represent the identity function if the model is accurately calibrated. If the diagonal is not perfect, it suggests the model was not calibrated correctly. The comparison diagrams in Figure 4 show that after using our method, the gap was effectively reduced, and the reliability diagram is closer to an identity function. The upper charts in the figure are visualizations of the average confidence and accuracy, and below them are reliability diagrams.

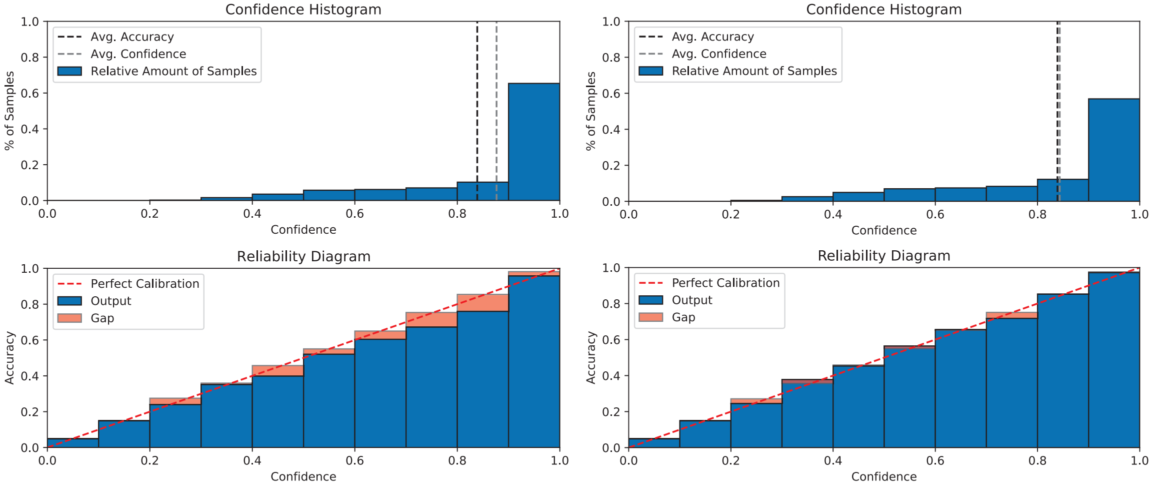


Figure 4. Comparison of reliability diagram and confidence histogram on the Jester test set. The upper charts are visualizations of average confidence and accuracy, and below them are reliability diagrams from (**left**) the SGD optimization method and (**right**) from our method.

4.3. Application 2: Image Classification Task

To evaluate the performance of the model in different tasks, we tested our proposed method in an image classification task using the CINIC-10 datasets [27]. They were compiled by combining CIFAR-10 with images selected and downsampled from the ImageNet database. The CINIC-10 datasets had 270,000 images and were divided into three groups for training, validation, and testing. In each subset, there were 10 categories, each with 90,000 images. We used three architectures that were pre-trained on ImageNet for a backbone network (ResNet-50 [28], Wide-ResNet-50 [29], and VGG-16 [30]), and we then fine-tuned them on the CINIC-10 training set using the transfer learning approach. The experimental setup and training strategy are consistent with those presented in Section 4.2.

Comparisons of classification accuracy, average confidence, RMSCE, MCE, and ECE of the different architectures and methods on the CINIC-10 test set are summarized in Table 2. We evaluated our method using the same scoring function as in Section 4.2, including: SGD (baseline), MC-Dropout [16], and Logits-Scaling [26]. We observed that the VGG16-based architecture yielded the best confidence calibration, with improved classification accuracy. The other two architectures significantly improved accuracy by about 1.13% on average, compared with the SGD optimization approach. Comparative results demonstrate that our approach decreases the model's calibration error while also increasing its accuracy.

Table 2. Summary of classification accuracy, average confidence, RMSCE, MCE, and ECE compared to other methods, as obtained from the CINIC-10 test set, where (↑) indicates higher values are desirable, (↓) means lower values are better, and bold indicates the best result.

Architecture	**Method**	**CINIC-10 Test Set**				
		Acc. (↑)	**Conf. (↑)**	**RMSCE (↓)**	**MCE (↓)**	**ECE (↓)**
ResNet-50	Baseline (SGD)	72.92	75.93	5.22	3.39	3.01
	MC-Dropout	73.03	76.28	5.70	3.62	3.25
	Logits-Scaling	72.46	75.66	6.02	3.59	3.21
	Ours	**73.63**	**75.78**	**4.38**	**2.80**	**2.36**
Wide-ResNet-50	Baseline (SGD)	73.06	77.01	7.48	4.43	3.97
	MC-Dropout	72.82	77.05	7.85	4.83	4.31
	Logits-Scaling	69.01	72.04	5.67	3.57	3.07
	Ours	**74.62**	**76.38**	**7.42**	**3.62**	**2.52**
VGG-16	Baseline (SGD)	78.07	82.69	8.95	5.23	4.62
	MC Dropout	78.97	85.23	20.01	7.04	6.29
	Logits-Scaling	76.08	78.42	5.79	2.56	2.37
	Ours	**80.51**	**80.11**	**3.56**	**0.51**	**0.39**

The MC-dropout method can be viewed as an approximation of Bayesian neural networks. The method requires modification of the original network architecture and in the testing phase. It is also necessary to perform multiple forward propagation and average multiple predictions during the testing phase. In contrast, our method can be regarded as a trainable method. There is no need to modify the model structure, and only one forward propagation is required in the test phase, which greatly reduces the inference time. The Logits-scaling method requires recalculating a hyperparameter T based on the validation set after training and using T to modify the output of softmax during inference on new inputs. An obvious shortcoming of this method is that it is very dependent on data sets and has relatively less generality.

To visualize the calibration effect of our method on the CINIC-10 dataset, we also visualized the calibration curves. The reliability diagram before and after confidence calibration is shown in Figure 5. The red dashed line indicates the best calibration, where the output confidence precisely reflects the accuracy. In the confidence histogram (upper left), we observe a large gap between the mean confidence and the confidence; while in the upper right graph (our method), we can see that this gap is greatly reduced. Comparing the left and right reliability diagrams, we can also visualize that our approach is closer to the red dashed line. These results are consistent with what we observed in application 1 in Section 4.2.

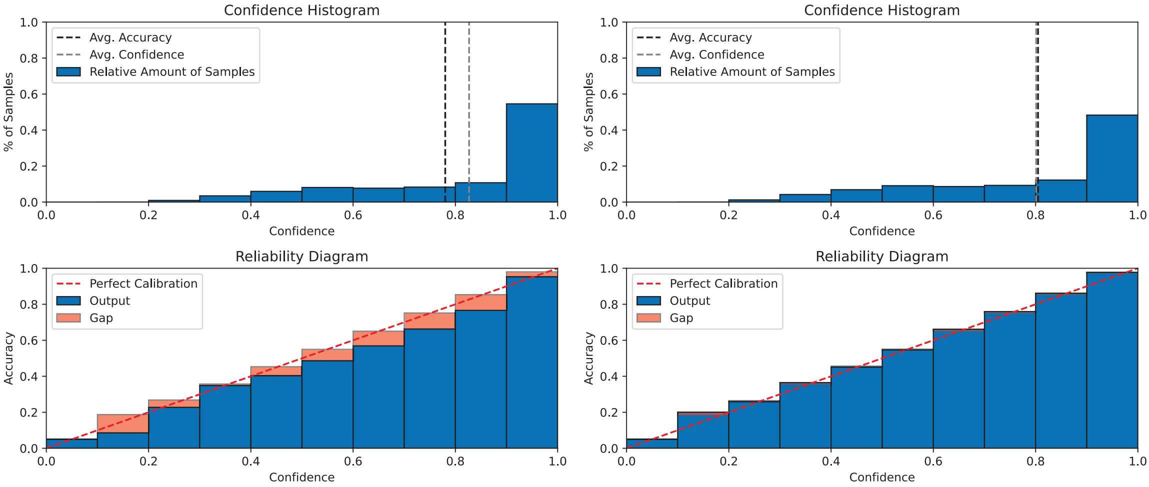


Figure 5. Comparison of reliability diagram and confidence histogram from VGG-16 trained on the CINIC-10 test set. The upper charts are visualizations of average confidence and accuracy, and below them are reliability diagrams from (**left**) the SGD optimization method and (**right**) from our method.

5. Discussion and Future Work

We believe that there are three important factors that make our proposed approach work. First, our method uses a modified learning rate schedule. This allows the optimizer to continuously explore the optimal weights for high performance rather than just reaching or limiting itself to a single solution. For example, we use a standard decaying learning rate strategy for the first 75% of iterations, and a cyclical learning rate for the remaining 25% of iterations. Second, we choose to save the corresponding model weights when the learning rate decays to its lowest value rather then saving model weights according to the validation accuracy. Note that we are using a cyclical learning rate, so the learning rate will decay to the lowest value multiple times; that is, we obtain multiple sets of model weight parameters. The third is to average the model weight parameters traversed by the optimizer. The key idea behind this is to leverage deep learning's unique training objectives flatness [14] to improve generalization and reduce overconfidence.

DNNs are usually learned through a stochastic training algorithm, which means the DNN is sensitive to the training data and may learn a different set of weights at the end of the training session, resulting in different predictions. The goal of machine learning is to develop methods and algorithms to learn from the data; that is, extract the *residing* information from the data. In fact, learning parameters from data is an inverse problem; we need to infer the cause (parameters) from the result (observed data). In this work, we proposed an approximate deep ensemble method to calibrate the confidence of DNNs. The ambiguity in target y for a given x was captured by obtaining a probabilistic model with appropriate scoring rules. In addition, the approximate combination of ensembles was used to predict the output for x in an attempt to capture model uncertainty. Through experiments on two benchmark datasets for image classification and gesture recognition, we demonstrated that our method obtained well-calibrated confidence estimates. Removing correlations from individual network predictions to promote ensemble diversity and further improve performance is left to future work. It would be meaningful to refine multiple ensemble models into a simpler single model to obtain a model with good confidence calibration.

Author Contributions: Conceptualization, Z.C.; methodology, Z.C. and Y.L.; software, Z.C.; validation, Z.C., Y.L. and D.-H.K.; investigation, Z.C., Y.L. and D.-H.K.; original draft preparation, Z.C.; writing—review and editing, Z.C., Y.L., D.-H.K. and B.-S.S.; supervision, Y.L., D.-H.K. and B.-S.S.; project administration, B.-S.S.; funding acquisition, B.-S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by National Research Foundation of Korea (NRF) grants funded by the Korean government (no. NRF-022R1A2B5B01001553 and no. NRF-022R1A4A1033549) and by an Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-00155915, Artificial Intelligence Convergence Innovation Human Resources Development [Inha University]).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare they have no conflicts of interest related to the publication of this paper.

References

1. Jiang, X.; Deng, X. Knowledge reverse distillation based confidence calibration for deep neural networks. *Neural Process. Lett.* **2023**, *55*, 345–360. [CrossRef]
2. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning, ICML'17, Sydney, NSW, Australia, 6–11 August 2017; pp. 1321–1330.
3. Gawlikowski, J.; Tassi, C.R.N.; Ali, M.; Lee, J.; Humt, M.; Feng, J.; Kruspe, A.; Triebel, R.; Jung, P.; Roscher, R.; et al. A survey of uncertainty in deep neural networks. *Artif. Intell. Rev.* **2023**, *56*, 1513–1589. [CrossRef]
4. Abdar, M.; Pourpanah, F.; Hussain, S.; Rezazadegan, D.; Liu, L.; Ghavamzadeh, M.; Fieguth, P.; Cao, X.; Khosravi, A.; Acharya, U.R.; et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Inf. Fusion.* **2021**, *76*, 243–297. [CrossRef]
5. Jospin, L.V.; Laga, H.; Boussaid, F.; Buntine, W.; Bennamoun, M. Hands-on Bayesian neural networks—A tutorial for deep learning users. *IEEE Comput. Intell. Mag.* **2022**, *17*, 29–48. [CrossRef]
6. Wang, H.; Yeung, D.Y. A survey on Bayesian deep learning. *ACM Comput. Surv.* **2020**, *53*, 1–37. [CrossRef]
7. Munir, M.A.; Khan, M.H.; Khan, S.; Khan, F.S. Bridging Precision and Confidence: A Train-Time Loss for Calibrating Object Detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Los Alamitos, CA, USA, 28 June 2023; pp. 11474–11483.
8. Lee, J.; Park, S. A Study on the Calibrated Confidence of Text Classification Using a Variational Bayes. *Appl. Sci.* **2022**, *12*, 9007. [CrossRef]
9. Psaros, A.F.; Meng, X.; Zou, Z.; Guo, L.; Karniadakis, G.E. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *J. Comput. Phys.* **2023**, *477*, 111902. [CrossRef]
10. Ganaie, M.A.; Hu, M.; Malik, A.; Tanveer, M.; Suganthan, P. Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105151. [CrossRef]
11. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–12.
12. Ovadia, Y.; Fertig, E.; Ren, J.; Nado, Z.; Sculley, D.; Nowozin, S.; Dillon, J.; Lakshminarayanan, B.; Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–12.
13. Huang, G.; Li, Y.; Pleiss, G.; Liu, Z.; Hopcroft, J.E.; Weinberger, K.Q. Snapshot Ensembles: Train 1, get M for free. *arXiv* **2017**, arXiv:1704.00109.
14. Garipov, T.; Izmailov, P.; Podoprikhin, D.; Vetrov, D.P.; Wilson, A.G. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 1–10.
15. Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.P.; Wilson, A.G. Averaging Weights Leads to Wider Optima and Better Generalization. In Proceedings of the Conference on Uncertainty in Artificial Intelligence, Monterey, CA, USA, 6–10 August 2018.
16. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1050–1059.
17. Mller, R.; Kornblith, S.; Hinton, G.E. When does label smoothing help? *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–10.
18. Rahaman, R. Uncertainty quantification and deep ensembles. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 20063–20075.
19. Patel, K.; Beluch, W.; Zhang, D.; Pfeiffer, M.; Yang, B. On-manifold adversarial data augmentation improves uncertainty calibration. In Proceedings of the 25th International Conference on Pattern Recognition, Milan, Italy, 10–15 January 2021; pp. 8029–8036.
20. Yang, Y.; Lv, H.; Chen, N. A survey on ensemble learning under the era of deep learning. *Artif. Intell. Rev.* **2023**, *56*, 5545–5589. [CrossRef]
21. Mienye, I.D.; Sun, Y. A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access* **2022**, *10*, 99129–99149. [CrossRef]

22. Mahajan, P.; Uddin, S.; Hajati, F.; Moni, M.A. Ensemble Learning for Disease Prediction: A Review. *Healthcare* **2023**, *11*, 1808. [CrossRef]
23. Guo, H.; Liu, H.; Li, R.; Wu, C.; Guo, Y.; Xu, M. Margin diversity based ordering ensemble pruning. *Neurocomputing* **2018**, *275*, 237–246. [CrossRef]
24. Fernando, K.R.M.; Tsokos, C.P. Dynamically weighted balanced loss: Class imbalanced learning and confidence calibration of deep neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 2940–2951. [CrossRef]
25. Materzynska, J.; Berger, G.; Bax, I.; Memisevic, R. The Jester Dataset: A Large-Scale Video Dataset of Human Gestures. In Proceedings of the 2019 International Conference on Computer Vision Workshop, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2874–2882.
26. Wei, H.; Xie, R.; Cheng, H.; Feng, L.; An, B.; Li, Y. Mitigating neural network overconfidence with logit normalization. In Proceedings of the International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; pp. 23631–23644.
27. Darlow, L.N.; Crowley, E.J.; Antoniou, A.; Storkey, A.J. Cinic-10 is not imagenet or cifar-10. *arXiv* **2018**, arXiv:1810.03505.
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
29. Zagoruyko, S.; Komodakis, N. Wide residual networks. *arXiv* **2016**, arXiv:1605.07146.
30. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

Article

Visual Analysis Method for Traffic Trajectory with Dynamic Topic Movement Patterns Based on the Improved Markov Decision Process

Huarong Chen [1], Yadong Wu [2,*], Huaquan Tang [3], Jing Lei [3], Guijuan Wang [1], Weixin Zhao [4], Jing Liao [1], Fupan Wang [1] and Zhong Wang [1]

[1] School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 621010, China; chenhuarong@swust.edu.cn (H.C.); guijuanwang@swust.edu.cn (G.W.); liaojing@swust.edu.cn (J.L.); wangfp@swust.edu.cn (F.W.); zhongw@mails.swust.edu.cn (Z.W.)
[2] School of Computer Science and Engineering, Sichuan University of Science & Engineering, Yibin 644007, China
[3] Technical Center, Mianyang Xinchen Engine Co., Ltd., Mianyang 621000, China; huaquan.tang@xce.com.cn (H.T.); jing.lei@xce.com.cn (J.L.)
[4] College of Computer Science, Sichuan University, Chengdu 610065, China; zhaoweixin@stu.scu.edu.cn
* Correspondence: wuyadong@suse.edu.cn

Abstract: The visual analysis of trajectory topics is helpful for mining potential trajectory patterns, but the traditional visual analysis method ignores the evolution of the temporal coherence of the topic. In this paper, a novel visual analysis method for dynamic topic analysis of traffic trajectory is proposed, which is used to explore and analyze the traffic trajectory topic and evolution. Firstly, the spatial information is integrated into trajectory words, calculating the dynamic trajectory topic model based on dynamic analysis modeling and, consequently, correlating the evolution of the trajectory topic between adjacent time slices. Secondly, in the trajectory topic, a representative trajectory sequence is generated to overcome the problem of the trajectory topic model not considering the word order, based on the improved Markov Decision Process. Subsequently, a set of meaningful visual codes is designed to analyze the trajectory topic and its evolution through the parallel window visual model from a spatial-temporal perspective. Finally, a case evaluation shows that the proposed method is effective in analyzing potential trajectory movement patterns.

Keywords: trajectory topic model; Markov Decision Process; visual analysis

Citation: Chen, H.; Wu, Y.; Tang, H.; Lei, J.; Wang, G.; Zhao, W.; Liao, J.; Wang, F.; Wang, Z. Visual Analysis Method for Traffic Trajectory with Dynamic Topic Movement Patterns Based on the Improved Markov Decision Process. *Electronics* **2024**, *13*, 467. https://doi.org/10.3390/electronics13030467

Academic Editor: Dah-Jye Lee

Received: 29 December 2023
Revised: 17 January 2024
Accepted: 20 January 2024
Published: 23 January 2024

1. Introduction

Visual exploration of traffic trajectory data is constrained by the limited capacity for visual information and the lack of interactive guidance. Traffic trajectories record the historical travel of vehicles, with visual information and potential features at spatial-temporal levels. Exploring traffic trajectories through visualization can intuitively obtain visual information about the trajectories and mine the potential movement patterns. However, drawing the traffic trajectory directly on a map will be limited by the information capacity and will cause problems such as visual occlusion. The most intuitive method to help the visual exploration of traffic trajectory is to reduce the visual density by computing the trajectory features. Liu et al. [1] effectively reduce the visual density of Origin to Destinatin (OD) flow by dividing and exploring urban functional areas. Liu et al. [2] employ a tensor decomposition algorithm to segment multi-dimensional spatiotemporal data and reduce the number of visualization elements on the same screen. Zhou et al. [3] propose a visual exploration system to reduce the visual clutter and strengthen the relevance of OD flows. Deng et al. [4] provide an efficient visual presentation design of temporal events by abstracting the workflow of temporal and spatial interaction association analysis. Andrienko et al. [5] designed a method to abstract the characteristics of spatio-temporal

OD mobility to reduce the intersection and occlusion of the trajectory flow graph. Unfortunately, these methods calculate trajectory patterns at the data feature level and do not provide enough semantic guidance for the analyst. In addition, analysts usually need to manually select and iterate different motion data slices to search for hidden patterns in trajectory data, making it difficult to deeply analyze potential trajectory movement patterns from a spatial-temporal perspective. We address these issues and propose a set of meaningful visual encodings. This visualization model establishes a parallel window model. The model presents the trajectory topic characteristics from the spatial and temporal perspectives, which is helpful in reducing visual interference and providing guidance for visual interactive exploration.

The dynamic topic information in the trajectory topic model reflects the evolution of the topic. A topic model, which provides topic guidance for trajectory analysis by applying the topic model to trajectory data analysis, is a kind of statistical model. Most of the existing methods that apply topic models to trajectory modeling are based on Latent Dirichlet Allocation (LDA) [6], Non-negative Matrix Factorization (NMF) [7], and other unsupervised topic algorithms, focusing on optimizing topic extraction strategies. Specifically, Chu et al. [8] introduce the LDA to infer hidden patterns of moving taxi populations. Liu et al. [9] transform the traffic data into a document library to capture latent traffic semantic patterns based on the NMF model. Liu et al. [10] employ the bigram topic model to analyze textualized trajectories. Tao et al. [11] introduce a hierarchical topic model H-NMF to extract multi-granularity traffic topics to capture mobility patterns. Traffic trajectory is dynamic, and the corresponding trajectory topic constantly evolves. However, the trajectory topic analysis based on the above modeling methods only considers the trajectory spatial information. It does not consider the temporal dynamic evolution of the trajectory topic, resulting in the loss of trajectory topic evolution information.

Trajectory features are described by trajectory semantics [12]. Trajectory order is an essential part of the semantics of trajectory topics. The topic model describes the topic content by counting word frequency but does not consider the ordinal relationship between words. Traffic trajectories are sequentially connected trajectory segments with a sequential relationship. Trajectory sequence information is necessary for trajectory topic modeling. However, applying the topic model to the topic modeling of traffic trajectory is challenging and needs to overcome the influence of the topic model not containing word sequence information on the interpretability of trajectory topics. In summary, there needs to be more trajectory order semantics and topic evolution information in existing work on trajectory topics. This implies the need for visual encodings that provide sufficient semantic guidance to analyze trajectory topic evolution.

To solve these problems, a visual analysis method is proposed to analyze the dynamic topics of trajectories. This method considers the evolution information of trajectory topics and overcomes the problem of missing word sequence information in probabilistic models. Firstly, the dynamic topic model (DTM) [13] for a trajectory is established to analyze the trajectory topic evolution fitted by parameter distribution. Secondly, by calculating the thematic entropy, we can quantify the informational content of the trajectory topics, which indicates meaningful thematic time segments, thereby reducing the time required for the manual repetitive selection of data slices. Subsequently, we employ Markov chains to represent trajectory words and address the issue of topic models not retaining word order information. Based on the Markov chains, we utilize an improved Markov Decision Process (MDP) to generate trajectory sequences that represent the topics, which aids in analyzing the topic content. Moreover, we introduce a meaningful visual encoding scheme, presenting trajectory topic visualizations separately through parallel views from both abstract maps and thematic information perspectives. Lastly, a case study based on a real dataset was conducted to demonstrate the efficacy of the methodology proposed in this paper in exploring dynamic trajectory topics. In summary, the primary contributions of this work are as follows:

- A trajectory dynamic topic model is proposed. This model considers the topic evolution and is used to process traffic trajectory data. This model helps to obtain trajectory dynamic topics and capture their evolution. The topic entropy is calculated to represent the volume of information in trajectory topics and is utilized to analyze the evolution of the topic.
- The improved Markov Decision Process is proposed to generate representative trajectory sequences of a topic. This method aims to overcome the problem of probabilistic models which often lose word sequence information.
- The visual models, featured with the parallel and regular arrangement, are designed to help users explore the trajectory dynamic topics and related indicators.
- Experimental studies based on the publicly available dataset demonstrate the effectiveness of this method in trajectory dynamic topic exploration.

2. Related Works

2.1. Trajectory Topic Modeling

The basic idea of topic modeling is to vectorize a given corpus by term frequency or inverse document frequency. Topic modeling decomposes a document-term matrix into document-topic and topic-word subsets, subsequently optimizing these subsets using probabilistic or decomposition techniques. The clustering of implicit semantic structures in documents obtained through topic modeling can be applied to the discovery of trajectory semantic structures and trajectory topics. Trajectory topic modeling aims to describe motion patterns by calculating sub-trajectory frequency information, thereby further extracting the content of trajectory topics. Commonly used topic models include LDA [6], NMF [7], and others.

As common topic computing methods, LDA and NMF are widely used in topic modeling. LDA computes the posterior distribution of the data based on Bayesian algorithms, whereas NMF produces deterministic outcomes from matrix decomposition, not probabilistic values. Current efforts in topic modeling largely revolve around modifications or advancements of the LDA and NMF methodologies. For instance, Liao et al. [14] established a trajectory topic model based on LDA to unearth latent driving patterns and analyze a driver's travel intentions. Huang et al. [15] employed LDA to explore hidden ship mobility patterns from the trajectory document corpus. In contrast, Liu et al. [2] applied tensor decomposition to multi-dimensional spatiotemporal data to extract regional patterns. Addressing the challenge of selecting an optimal number of topics, Liu et al. [9] proposed an interactive trajectory topic modeling method based on a semi-supervised non-negative matrix factorization (SS-NMF) for trajectory data. Similarly, Tao et al. [11] presented a hierarchical topic analysis approach to balance the analyses of varying granularities of movement patterns. Notably, NMF's performance degrades with more significant numbers of topics, whereas LDA exhibits more robust modeling results.

Beyond using LDA and NMF for trajectory topic modeling, Liu et al. [10] adopted the Biterm Topic Model (BTM) [16] to model trajectory movement patterns, leveraging the Apriori algorithm for sub-trajectory extraction and using sub-trajectory sequences to indicate directionality. Originating for short-text analysis, BTM differs from LDA in that its word generation is corpus-wide. Without a document hierarchy, BTM primarily retrieves word probabilities related to topics, making it challenging to obtain document-specific topic probabilities. However, different trajectory topics are implied in trajectory datasets. Analyzing trajectory data deeply and obtaining specific topic probabilities using BTM is difficult.

These topic modeling methods, including LDA, NMF, and BTM, calculate the trajectory topic within the time interval and do not calculate the trajectory topic evolution between time slices at the model level. Dynamic topic modeling adds a time factor, and the topic content changes with time. Corresponding Dynamic topic versions exist for NMF and LDA, namely, Dynamic NMF [7] and DTM [13]. Dynamic NMF [17] uses the state space to represent the NMF time dependence. The dynamic NMF first divides the corpus into time slices and then runs the NMF algorithm on the time window to obtain the topic of each time slice. After finding topics in each time slice, the dynamic NMF classifies semantically similar topics as dynamic topics. DTM is based on LDA but uses variational methods to connect topic variations in different time slices. Compared with Dynamic NMF, DTM has more continuity between adjacent time slices.

A significant limitation of these methods is their focus on trajectory topics within specific time intervals without accounting for temporal evolution at the model level. Dynamic topic modeling introduces time as a variable, allowing topic content to evolve. NMF and LDA have their dynamic counterparts, specifically Dynamic NMF and DTM. Dynamic NMF represents NMF's temporal dependencies using state spaces, segments the corpus into temporal slices, and then runs the NMF algorithm over these time windows. After identifying topics within each time slice, semantically similar topics are categorized as dynamic topics. On the other hand, DTM, grounded in LDA, employs variational methods to link topic variations across distinct temporal slices, offering enhanced continuity compared to Dynamic NMF. Yao et al. [18], leveraging non-trajectory data from social media, developed a DTM for spatiotemporal sequence analysis. Specifically, they achieved dynamic topic analysis by covering adjacent temporal slices with radii from social media locational points. This approach bypasses trajectory point connections and computes trajectory topics at the individual point level.

In general, there is an issue of weak continuity in trajectory topics across slices when establishing a trajectory topic model. To address this problem, this paper introduces DTM into the analysis of dynamic trajectory topics for the first time, continuously tracking the evolution of trajectory topics through DTM. Additionally, by improving the Monte Carlo method to represent transition probabilities between trajectory segments, it constructs trajectory segment sequence information, providing more details for describing the evolution of trajectory topics.

2.2. Visual Analysis of Trajectory Topic

Visual analysis is often used to explore the inherent features and patterns encoded in trajectory data, because of the connection between humans and computers provided by interactive visual interfaces [19,20]. Topic visualization primarily revolves around the interrelationships among topic sets, encompassing the temporal evolution of topics and the static relations between them. The Theme River [21], an early visualization approach, is well-suited for analyzing topic evolution. While the fluid visual characteristics of the Theme River facilitate capturing change trends over time, its river-like curve metaphor poses challenges for quantitative analysis of topic attributes. In addition, Sankey diagrams are another standard visualization tool. They use rectangles to quantify topic evolution but can become crowded and complex when there is a multitude of topics. To overcome the above problems, the parallel matrix heatmap is employed for quantitative analysis of topic evolution, enabling the identification of trends in topic evolution. The visual feature of its neat arrangement allows for a clear and intuitive interpretation of visual information.

Methods for trajectory topic visual analysis merge geographical and visualization views, offering multi-level spatial information for trajectory topic pattern analysis. He et al. [22] incorporated LDA into trajectory topics to depict spatio-temporal topic variations, harnessing visual filters to mine spatio-temporal text data. Similarly, Chu et al. [8] employed LDA to dissect vast amounts of taxi trajectory data within cities, unveiling concealed knowledge through visualization. Similarly, Gao et al. [23] leveraged LDA to extract topics related to road cyclists' behavior and movement semantics, adopting clustering and interactive visualization techniques to identify further clusters of similar cyclists and road segments within the city. By visualizing and retrieving taxi trajectory data, AL-Douhuki et al. [24] provide a semantic management approach for taxi trajectory data. Ali et al. [25] proposed a topic modeling approach grounded in LDA based on ontology, extracting topics and features by retrieving traffic content from social networks. Gao et al. [23] used LDA to extract the topics of cyclist behavior semantics and moving semantics on roads. Then, they implemented multiple interactive visual interfaces to facilitate the interpretation for analysts. Yan et al. [26] utilized a BTM to unearth latent topics within trajectories and subsequently analyze these topics via visualization.

Concerning trajectory points, Zhou et al. [27] and Liu et al. [1] implemented sampling strategies to reduce visual clutter, facilitating the discovery of trajectory patterns. Recognizing the large-scale nature of trajectory data, Zhou et al. [3] designed visual representations of Origin-Destination (OD) flows, iteratively probing OD movement patterns. Andrienko et al. [5] aggregated data to identify spatial similarities in trajectories and abstracted spatio-temporal features to delve into trajectory flows. Wang et al. [28] introduced a multi-scale geographical area data visualization method predicated on spatial attribute associations, enhancing visual perception of both global and detailed features. While these methodologies provide visual optimization at the trajectory point and line levels, investing considerable effort in visually exploring trajectory features, they fall short in contrasting spatial trajectory topic information and topic features between temporal slices.

In summary, to effectively explore the evolution of trajectory topics from different perspectives of trajectory points, trajectory lines, and trajectory topics using visual components, this paper proposes a multi-entity cascade visual analysis method under parallel views. This method aims to further reveal the potential patterns of dynamic trajectory topics by presenting the topic features through parallel abstract graphs and parallel heat matrices containing time series information. The parallel view overcomes the challenges of quantitatively analyzing topic river graphs and dealing with the complexity of crowded Sankey diagrams by organizing visual elements in a parallel layout and cascading the analysis of topic semantics. This study studies the construction and characteristics of dynamic spatio-temporal trajectory graphs to comprehend the evolution of dynamic trajectory topics.

3. Methods

3.1. Overview

One of the challenges in trajectory topic modeling and analysis is effectively integrating both temporal and spatial information into the topic computation. To address these intricacies, our study employed three specific strategies.

First, concerning temporal information, we utilized trajectory segments partitioned based on specific time intervals to model dynamic trajectory topics. Given that the starting and ending points of trajectory segments are fixed, there is no ambiguity in trajectory words, meaning each trajectory word has a singular interpretation. Furthermore, the MDP generates trajectory sequences within the collection of trajectory segments representing a trajectory topic, ensuring that spatially discrete trajectory segments are interconnected. By leveraging these three strategies, we can efficiently compute trajectory topics imbued with both temporal and spatial dimensions. In the following sections of this chapter, we will delve deeper into these methodologies, providing a comprehensive exposition of their intricacies and applications.

3.2. Trajectory Dynamic Topic Modeling

Topic modeling employs statistical methods to discern abstract topics within documents. When applied to trajectory information, it facilitates the extraction of abstract trajectory topics. The DTM, in contrast to conventional topic models such as LDA and NMF, incorporates temporal sequence information, making it particularly well-suited to the temporal variability inherent in trajectories. A study closely aligned with our approach is [9], where the authors employed NMF to topic-model trajectories across different time periods. However, such an approach lacks temporal coherence in trajectory topics. Compared to NMF, DTM offers superior coherence and yields more robust and flexible fitting results.

DTM treats the posterior distribution of the model parameters in the current time window as the conditional distribution for the parameters in the subsequent time window.

Building upon the foundation of LDA by integrating a temporal dimension, DTM becomes especially adept at analyzing topic evolution. However, being a probabilistic model, DTM overlooks word order during its computational process. Such an oversight can be problematic, given that word order mirrors the sequential nature of trajectory segments. Our study addresses this limitation through an enhanced MDP, which we will delve into comprehensively in Section 3.3.

The trajectory DTM describes the trajectory motion pattern with time. In the DTM of traffic trajectory, we take the trajectory segment as a word and integrate the trajectory spatial information into the word. The trajectory data is divided according to time slices, with the latter moment evolving from the previous moment. Since the Dirichlet distribution is not suitable for use in time series models, as shown in Equation (1), a Gaussian noise evolved state space model is used to connect the natural parameter β about the trajectory topic to sample the word distribution $\Phi(Z(d_n))$ about the trajectory topic $Z(d_n)$. In Equation (2), the natural parameter α about the trajectory topic ratio is also connected using the Gaussian noise evolution state-space model to sample the trajectory topic distribution $\Theta(d)$ of the generated trajectory document d. In the DTM model, the prior probabilities α for document topics and the distribution β for topics are dynamically iterated.

$$\beta_{t+1,k} \Big| \beta_{t,k} \sim N\Big(\beta_{t,k}, \sigma^2 I\Big) \tag{1}$$

$$\alpha_{t+1,k} \Big| \alpha_{t,k} \sim N\Big(\alpha_{t,k}, \delta^2 I\Big) \tag{2}$$

Equations (1) and (2) link the natural parameters β_t and α_t for each topic, where β_t and α_t are the Gaussian noise natural parameters for time slice t and topic k. σ and δ are the variance parameters. After linking both the trajectory topic and its proportion distribution, we sequentially bind the collection of trajectory topic models. The corpus generation process for the trajectory DTM is as follows: Within time slice t, a possible word (trajectory segment) distribution β for trajectory topic k is first sampled. Then, a potential topic distribution $\alpha(t)$ for each document (a collection of trajectory segments) is sampled. For each document within time slice t, the topic distribution η for document d is obtained. For every word in the document during time slice t, the possible topic $z(t, d, n)$ for the nth word in document d is sampled, and the corresponding word $w(t, d, n)$ is generated. The number of topics is determined using an algorithm that identifies the optimal topic count based on perplexity. This generation process's graphical model is illustrated in Figure 1. Both α and β at time t are generated based on α and β at time $t-1$. Using time dynamics, the kth topic at slice t smoothly evolves from the kth topic at slice $t-1$. When the horizontal arrow is removed, the graphical model is reduced to a set of separate topic models.

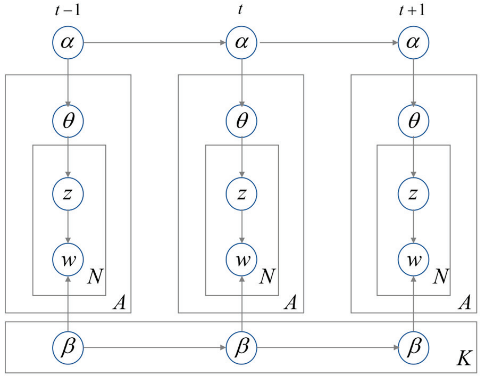


Figure 1. The generation process of the trajectory dynamic topic model includes three time slices [15].

The goal of the trajectory DTM is to calculate the posterior distribution. The objective of the variational method is to optimize the free parameters of the distribution over latent variables, making the distribution closely approximate the true posterior in terms of Kullback–Liebler (KL) divergence and subsequently substituting it for the genuine posterior. Within DTM, a variational approach based on the variational Kalman filter approximation is utilized to compute the posterior distribution [13]. Entropy measures the average amount of information expressed in every allocation of a random variable, depicting the system's level of disorder. We posit that when the system exhibits a low level of disorder, the trajectory topic is relatively stable. Conversely, with high disorder levels, the trajectory topics are relatively diverse. We compute the information entropy concerning topic probabilities within each time slice to provide information quantity across varying time slices, assisting in the subsequent selection of representative time slices for further analysis. The topic entropy at time t can be expressed as in Equation (3), where p represents the topic probability distribution at moment t.

$$H(t) = -\sum\nolimits_{t=1}^{K} [p(t) \log p(t)] \tag{3}$$

3.3. Trajectory Dynamic Topic Generating

3.3.1. Sub-Trajectory Modeling

The trajectory dynamic topic consists of sub-trajectory sets that adhere to Markovian properties. Extracting representative trajectory sequences from these sub-trajectory sets embodies the trajectory topic. Trajectory segments illustrate transitions between states. Following a particular policy, the MDP can generate trajectory sequences within the dynamic topic sub-trajectory sets that adhere to the Markovian properties, representing the trajectory topic. A Markov chain depicts the random process of transitioning from one state to another and is a mathematical construct composed of a set of random variables. Sub-trajectory sets describe inter-regional transitions, fundamentally representing state transitions. We can characterize the transition relationships within these sub-trajectory sets using Markov chains. The trajectory topic is composed of the sub-trajectory set $E = \{e_1, e_2, \cdots, e_n\}$. The set of areas for trajectory transitions form the state space $S = \{s_1, s_2, \cdots s_n\}$. A sub-trajectory x_i, transitioning from region s_i to another region x_j, exhibits frequency characteristics. According to Markovian properties, the probability of moving from the current state to the next is only dependent on the current state, and not on prior locations, adhering to the following Equation (4):

$$\Pr(X_{n+1} = x | X_1 = x, X_2 = x_2, \cdots, X_n = x_n) = \Pr(X_{n+1} = x | X_n = x) \tag{4}$$

where $\{X_1, X_2, \cdots, X_n\}$ represents the random state sequence comprised of state space S. $P_r(X_{n+1} = x | X_n = x)$ represents the probability of the current state X_n transitioning to the next state X_{n+1}. Given that the random process of the trajectory topic is a mathematical construct regarding the set of random variables of the sub-trajectory combinations, we can derive representative trajectories within the trajectory topic by constructing a Markov chain related to the sub-trajectory topic.

The Markov chain uses a probabilistic automaton to display transition probabilities between states, resulting in the transition matrix P. This matrix is a series of directed graphs, with edges representing the probability of transitioning from S_n to S_{n+1}. Each state in the state space is included once as a row and a column, with the matrix indicating the probability of moving from a row state to a column state and the probabilities of each row summing up to 1. The steady-state distribution of the Markov chain, regarding the sub-trajectory set, aids in generating consistent trajectory sequences. Within the sub-trajectory set, the time-invariant Markov chain $\{X_n, n \geq 0\}$ of the transition matrix P has its initial state space S_i, conforming to the probability distribution $\varnothing = \{\varnothing_0, \varnothing_1, \ldots, \varnothing_n\}$, satisfying the matrix Equation (5):

$$\varnothing = \varnothing P \tag{5}$$

where $\varnothing$ represents the steady-state distribution of the Markov chain, meaning a time-invariant Markov chain is wholly described by its initial state space S_i and its transition matrix P. We simulate the trajectory topic's initial state probability distribution $\varnothing_i$ and transition matrix against matrix Equation (5). As illustrated in Figure 2, the initial state of the sub-trajectories within the topic satisfies the stationary distribution, which can make the Markov process of the trajectory topic a stationary random process. As shown in Figure 2, the vertical axis is the probability of sub-trajectory state, and the horizontal axis is the number of iterations. Different colored line segments represent different trajectory segments. After approximately 60 simulation steps, a steady state is achieved under the action of the transition matrix P, with the distribution beginning to converge to a stable distribution at a certain point. Hence, the initial state of the sub-trajectory collection concerning the trajectory topic complies with the steady-state distribution. This implies that no matter the initial state of the trajectory, the equilibrium distribution will always be the same, achieving a stable initialization. The MDP concerning the sub-trajectory becomes a stationary stochastic process, meaning its finite-dimensional distributions are translation-invariant. This can further generate stable Markov sequences that represent the trajectory topic.

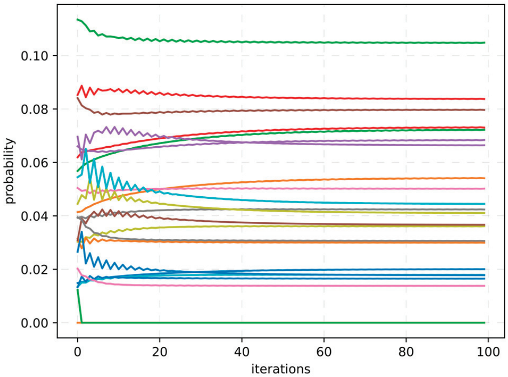


Figure 2. The trend of transition probability of sub-trajectories within a trajectory topic as the number of iterations increases. Lines represent trajectory topics, distinguished by different colors.

3.3.2. Trajectory Sequence Generating

The MDP elucidates the decision-making aspect of state transitions within Markov sequences. Within the sub-trajectory sets of the trajectory topic, transitions between sub-trajectories adhere to Markovian properties. Representative trajectory sequences within the trajectory topic can be generated using the MDP. This sequentially decided representative Markov sequence can serve as a succinct representation of the trajectory topic.

Formally, the MDP concerning the sub-trajectory set can be represented by the tuple $< S, A, R, P, \pi >$. The state $S = \{s_{1,} s_2, \cdots, s_n\}$ signifies the set of regions for trajectory transitions. The action space $A = \{a_1, a_2, \cdots, a_n\}$ depicts transitions between states, essentially forming the starting point S_i and endpoint S_j of a sub-trajectory. The reward R reflects the reward achievable when transitioning from one state to another within the action space. As illustrated by Equation (6):

$$R = (1 + k)P - kL_p \tag{6}$$

the reward value R is composed of two components, specifically the transition probability P and the penalty term Φ. The penalty term Φ is composed of hyperparameters k and L_p.

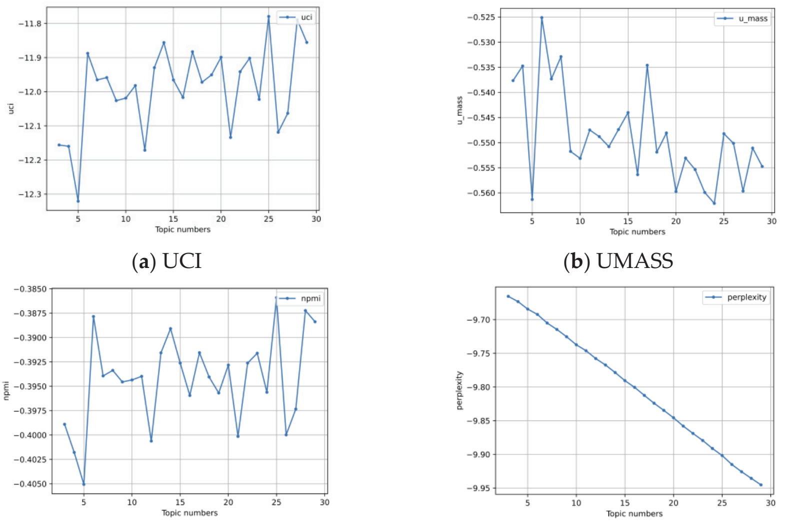
Our aspiration is for the target sub-trajectory segment to maintain a relatively consistent speed, fluctuating within a certain speed range. As presented in Equation (7):

$$L_p = |V - V_{bm}| \times \left(\sum_{i=0}^{N} v_i - v_{bm_i}\right)^{-1} \tag{7}$$

the objective of establishing the penalty term L_p is to curtail trajectory segments with substantial speed deviations. Consequently, we obtain the time slices of the trajectory dynamic topic and the preset number of trajectory segments N to compute the baseline speed V_{bm}. Subsequently, trajectory segments with significant deviations are penalized. The transition probability $P : S \times A \times S \to [0, 1]$ expresses the probability of an action transitioning from one state to another. The essence of the policy π is to maximize the reward for each action. Specifically, π commences from a stable steady-state initial condition. During each trajectory transition, it randomly selects actions based on the transition probability, conducts finite dynamic lookups, and eventually attains the Markov sequence representing the trajectory topic.

3.4. Data Set and Hyperparameter Settings

We chose taxi trajectory data from Chengdu as our experimental dataset for our study. This dataset comprises approximately 14,000 taxi-formatted trajectories commonly used for trajectory research. The dataset includes vehicle ID, latitude, longitude, passengers, date, and time attributes from 3 August 2014, to 30 August 2014 [29]. Before running the trajectory dynamic topic modeling, we calculated, on an hourly basis, the hyperparameters suitable for the DTM on this data. We employed perplexity and coherence to guide the acquisition of the optimal topic number. A lower perplexity indicates better topic performance. Coherence computation measures the consistency of words constituting a topic. A higher coherence score means topic words reinforce each other, enhancing the interpretability of the topic. Figure 3 describes the consistency score results of the experimental dataset. After evaluating the performance of all measurement metrics, we settled on 17 as the number of topics to be computed.



(**a**) UCI

(**b**) UMASS

(**c**) NPMI

(**d**) perplexity

Figure 3. The coherence scores and perplexity are calculated which describe topic performance. Topic coherence measures the coherence of words within a topic; higher coherence is preferable. Indicators describing topic coherence include UCI [30], UMASS [31], and NPMI [32]. Lower perplexity indicates a better topic model.

4. Visual Design

From a spatiotemporal perspective, we aim to dissect the composition and evolution of trajectory topics. The visual analysis of dynamic trajectory topics presents challenges. First, the dynamism of trajectory topics requires an intuitive representation through visualization, showcasing sequential changes across time and space. Secondly, given that dynamic trajectory topics are expressed as a series of trajectory word sets, a need arises to envisage the three-dimensional sequential evolution information linking time intervals, trajectory topics, and topic words.

We introduce the Parallel Viewport Visualization method for dynamic trajectory topic analysis to address these challenges. This technique provides a clear depiction of the dynamic shifts in trajectory topics through its parallel temporal visualization module and abstract map correlation module. This section delves deeper into the visualization technique for analyzing dynamic trajectory topic models.

4.1. Trajectory Dynamic Topic Time Evolution Analysis

The evolution of trajectory topics necessitates a compelling portrayal of their progression over time, offering a visual glance of the topic components. The information regarding the evolution of dynamic trajectory topics is embedded across multiple temporal slices, spanning various topics, each bearing its temporal evolution of topic words. This calls for an efficient visualization encompassing the three-dimensional data related to time, topic, and topic words. We calculated the topic distribution for each time slice and the evolution of specific topics between different time slices. We further assessed the information content of time slices and topics using entropy calculations. As illustrated in Figure 4, trajectory dynamic topic calculation consists of three parts. First, the original trajectory data is processed into a time-slice document composed of trajectory words, and then the trajectory dynamic topic is obtained by DTM and improved by MDP. Finally, the spatial and temporal evolution information of the trajectory dynamic topic is analyzed through the map view, parallel abstract map, and parallel matrix heatmap.

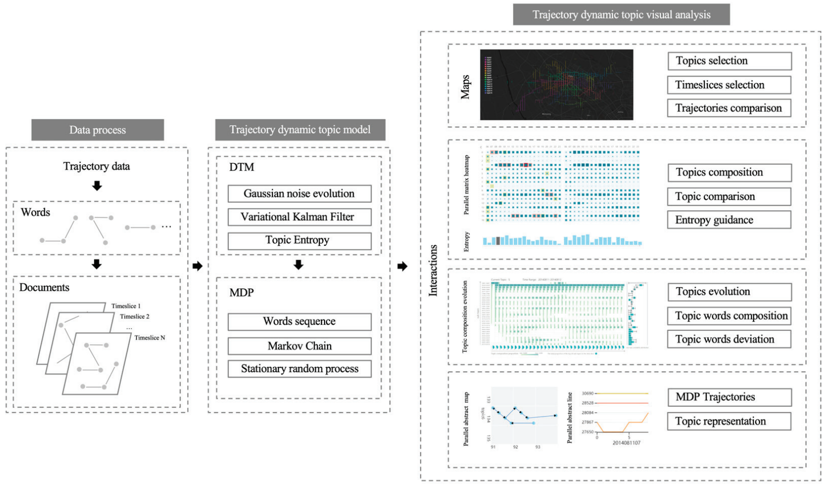
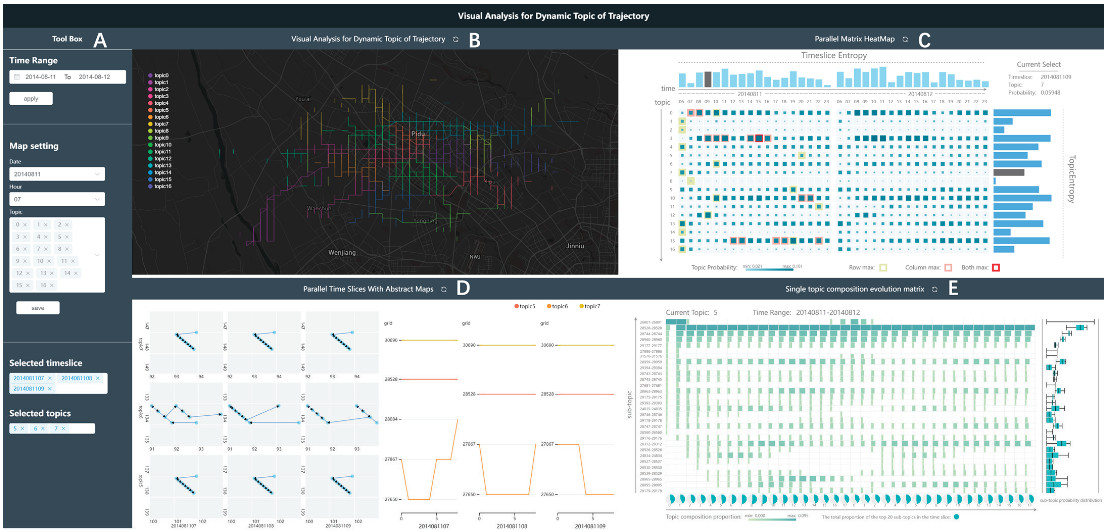


Figure 4. Calculation pipeline of trajectory dynamic topic. It represents the data analysis process from processing trajectory data and establishing a dynamic trajectory model to visual analysis of the dynamic trajectory model. Starting with data processing, the dataset is partitioned based on time slices, forming trajectory documents composed of trajectory segments. Subsequently, the DTM is utilized to establish a dynamic trajectory topic model. MDP is employed to generate representative trajectory sequences. Finally, combining maps and parallel views, spatial and temporal visual analysis is conducted to explore the evolution of trajectory topics.

While topic river plots are a conventional tool to visualize topic transitions, their curve estimations can introduce visual uncertainties. Thus, it becomes essential to sidestep visual biases stemming from visual encoding. To overcome these problems, we adopted the parallel matrix heatmap on cascade analysis. Comprising two parallel heatmaps and associated components, it facilitates efficient visual searches of dense three-dimensional information within a visually organized layout.

Figure 5C's topic evolution heatmap and Figure 5E's topic word heatmap form the parallel matrix heatmap, offering a concise and intuitive representation of time-topic and time-topic word information. The hierarchical construction of the time-topic and time-topic word cascade analysis allows for interaction across different hierarchical levels. In Figure 5C's heatmap, the x-axis denotes time, and the y-axis represents topics, with rectangle sizes encoding the composition of trajectory topics for corresponding time slices. To optimize exploration, we outlined rectangles with the highest probabilities in each row and column to swiftly pinpoint representative sub-topics. To reduce user interaction time when selecting the target trajectory's time slice, we computed the entropy of topics for each time slice, aiding users in choosing meaningful topic time slices, and displayed entropy values using bar charts on the right side of the heatmap. Similarly, the entropy of each topic was computed for target topic selection and was represented by using bar charts above the heatmap.



Tool Box

Time Range: 2014-08-11 To 2014-08-12

Map setting

Date: 20140811

Hour: 07

Topic: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16

Selected timeslice: 2014081107, 2014081108, 2014081109

Selected topics: 5, 6, 7

Current Select

Timeslice: 2014081109

Topic: 7

Probability: 0.05948

Current Topic: 5

Time Range: 20140811-20140812

Figure 5. The developed trajectory dynamic topic visualization system includes visual components: toolbox, trajectory topic map, topic evolution heatmap, parallel abstract map, and topic word heatmap. These visual components correspond to (**A**–**E**) respectively. The system shows the dynamic topic of trajectory from 11–12 August 2014, in one-hour time slices.

Within the dynamic trajectory topic, topic words constituting the topic would evolve over time. As depicted in Figure 5E's topic word heatmap, the evolution of the topic components is described by the changing proportions of topic words. Specifically, the heatmap elements were shifted from the center-expanding rectangles to left-aligned, right-expanding rectangles to facilitate clearer vertical comparisons among topic words within the same category. Typically, the x-axis of the topic word heatmap represents time, and the y-axis represents the topic words constituting the topic. For this study, we chose to display the top 35 topic words based on probability rankings, with their respective proportions among all topic words represented by a pie chart below. One can select topics of interest from the topic evolution heatmap to update the topic word heatmap, thus achieving cascade analysis concerning time, topics, and topic words.

4.2. Spatial Evolution Analysis of Trajectory Dynamic Topic

Exploring trajectory topics through maps provides a spatial perspective on the features inherent to such trajectories. A trajectory topic is not just a solitary line but also a complex mosaic comprising numerous sub-trajectory segments. It is much like deciphering the narrative of a tale that unfolds across different periods in time, spotting the changes in topics between time slices, and comparing the alterations within these sub-trajectory segments. In our quest to extract the essence of these trajectory topics, we have crafted parallel abstract maps. Just imagine running the MDP on sub-trajectory segments within time slices. This allows us to gauge transition probabilities, eventually leading us to a trajectory sequence that truly captures the spirit of the overarching trajectory topic. This spatial data is about geographical coordinates and the sequence in which they are arranged. Abstract maps convey the trajectory topics by depicting the spatial relative positions of trajectory points. Utilizing parallel windows in abstract maps allows for presenting the evolutionary information of trajectory topics from both temporal and spatial dimensions. This offers an intuitive comparison of representative trajectory sequences across different periods and topics.

As illustrated in Figure 5D's parallel abstract map, the horizontal axis represents time, while the vertical axis signifies topics. Users can select their areas of interest in the topic evolution heatmap to update the parallel abstract map, facilitating an analysis of trajectory topic evolution. This multi-instance comparison of spatial information effectively highlights the differences among trajectory sequences. Furthermore, to counteract the loss of spatial sequence information due to the overlapping trajectories on the map, we utilize a topic trajectory segment line chart to represent the trajectory topic sequence details. This chart conveys the order and composition of trajectory segments related to specific topics, offering insights into the relationship between trajectory sequences and their corresponding topics. Additionally, users can select topics of interest within the parallel abstract map to refresh the actual map, delving deeper into the detailed information of the trajectory topics.

5. Case Study

In the process of modeling transportation trajectory data, we adopted a trajectory DTM that considers topic evolution. This facilitates the extraction of dynamic trajectory topics and their subsequent evolutions. The target audience for this dynamic trajectory topic analysis includes those keen on understanding regional vehicle traffic patterns, such as urban planners and optimization specialists.

To demonstrate the efficacy of our methodology in the domain of dynamic trajectory topic analysis, we conducted a case study based on taxi data from Chengdu. This data spans 28 days, yielding 5,184,637 trajectory sequences after data cleaning. The granularity of analysis is correlated with the number of topics; the greater the number of topics, the finer the details within each topic. For this case, we defined 17 topics, with a minimum of 6 iterations for LDA, and set α to 0.01 to compute the dynamic trajectory topics.

5.1. Trajectory Topic Evolution Analysis

There are noticeable differences in trajectory topics within the same time slice. A single time frame encompasses multiple trajectory topics, each exhibiting varying intensities. Figure 5B illustrates the trajectory topic map at 7 a.m. on 11 August 2014, where most topics exhibit localized distributions. Having discerned the spatial distribution of trajectory topics, it is imperative to delve deeper into topic evolution over time.

5.1.1. Spatial Evolution of Trajectory Topics

There are significant differences between the trajectory segments that form trajectory topics at different time slices in the map. Before comparing the spatial evolution characteristics of the trajectory topics, we need to determine the target time slice. In Figure 5C, the topic evolution heatmap offers a vertical perspective, revealing different topic probabilities within identical time slices. The entropy for time slices 07 and 23 on 11 August 2014, is relatively low, signifying minimal differentiation among the topic probabilities and, consequently, a smaller information content. In contrast, the entropy for time slices 11 and 15 on the same date is markedly higher, with apparent disparities in topic probabilities, translating to more extensive information content.

The composition and location of trajectory segments within different trajectory themes vary across time slices. Based on the discussion above about the entropy of time slices, we select the 07 time slice to examine the spatial evolution of trajectory topics. As depicted in Figure 6, a comparative analysis of topics 3 and 12 over consecutive time frames demonstrates an uptrend in trajectory segment count with increasing topic probability.

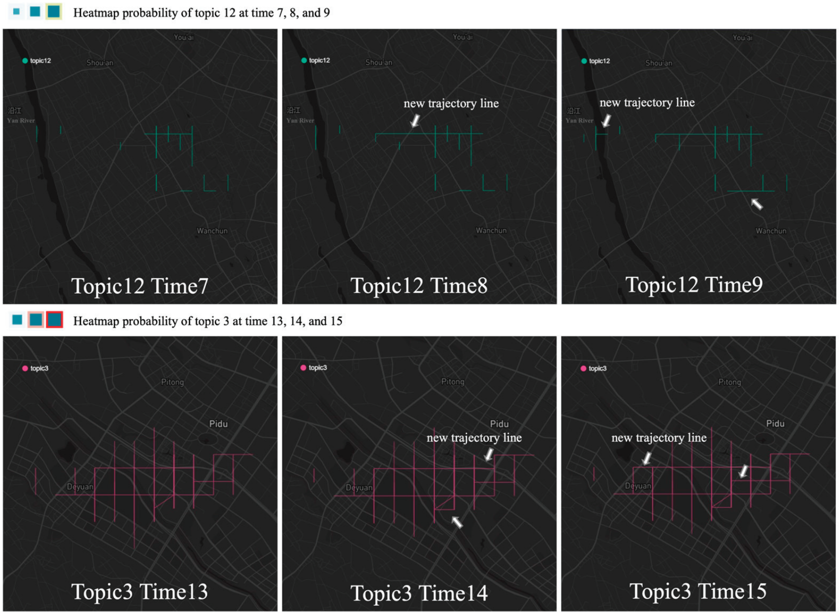


Figure 6. Trajectory topics are spatial and evolve over time. The trajectory segments that constitute a trajectory topic are subject to change. The evolution of trajectories for topic 3 and topic 12 on the map across three time slices is displayed. The white arrow indicates the new trajectory line. Trajectory topic 12 adds the middle trajectory line at time 8 and the left and lower right trajectory lines at time 9. Trajectory topic 3 adds the next volume and the right trajectory line at time 14 and adds the left, middle, and right trajectory lines at time 15.

5.1.2. Trajectories Topic Time Evolution

The information entropy of trajectory topics shows distinct differences across different time slices. One can examine the evolution of specific topics by observing Figure 5C's heatmap from a horizontal perspective. The bar chart on the right of Figure 5C shows the information entropy of topic evolution. Topic entropy is used to indicate the amount of trajectory topic information. Topic 8 can be seen to have the smallest information entropy because topic 8 has the smallest probability and little evolutionary difference. Topics 3 and 15 have high entropy due to the significant change in topic probability during the evolution of these topics. Further, we can analyze the trajectory topic evolution at the topic word level. Figure 7 shows the composition evolution of trajectory words between trajectory topic 8 and trajectory topic 10 from "20140811" to "20140812". The topic entropy of trajectory topic 8 is smaller than that of trajectory topic 10. That is, the change degree of topic words of trajectory topic 8 is smaller than that of trajectory topic 10. This also indicates that the number of trajectory words required to form trajectory topic 8 is smaller than those required for trajectory topic 10.

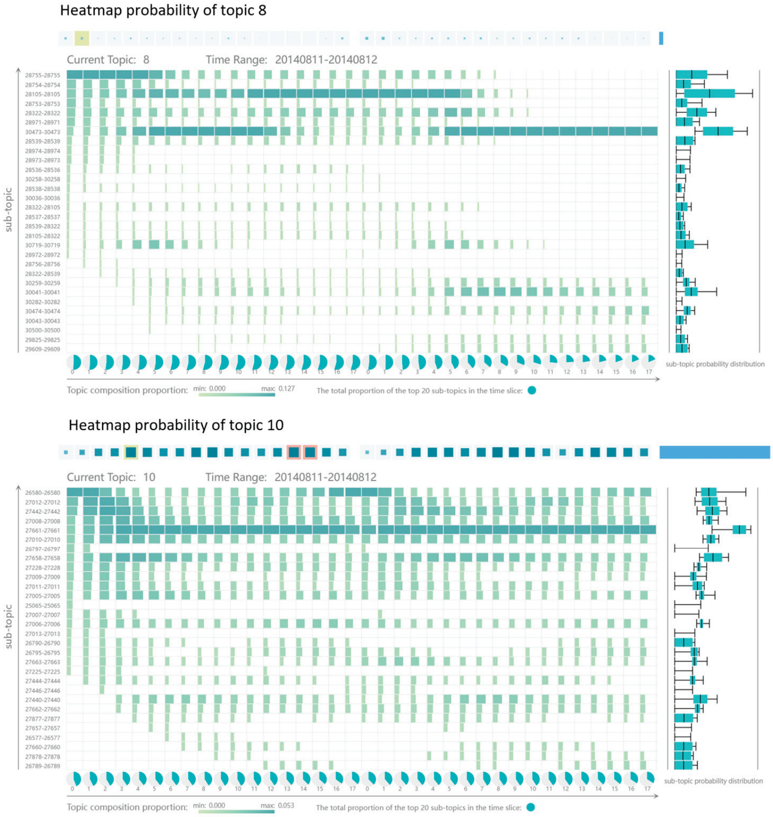


Figure 7. Different trajectory topics and their evolution exhibit significant differences. We compared the composition evolution of trajectory words, using the matrix heatmaps of topic words for trajectories 8 and 10. The figure captures the top 35 topic words that comprise the topic model and shows the composition proportion of the trajectory topic concerning the topic words (trajectory segment). The upper part of the topic word evolution matrix is the probability and topic entropy of the topic in different time slices. The pie chart on the lower side of the topic term evolution matrix gives the highest probability of 35 topic terms and the proportion of the probability sum of all topic terms. On the right side of the topic term evolution matrix is a box graph about the probability of the topic term, which reflects the distribution of the topic term over time.

5.2. Trajectory Topic Sequence Analysis

The trajectory sequence reflects the movement patterns within the trajectory topics. The trajectory sequence is generated through the improved MDP, incorporating the spatial information of trajectory words, and represents the most representative trajectory sequences within the trajectory topics. In computing trajectory topics, the trajectory sequences have been simplified by partitioning the map into grids. Concentrating trajectories within these grids facilitates the generation of trajectory sequences. This method transforms trajectory sequences into movements within a grid and between grids. We believe that movements within a grid are significant because, in the original data entries, a vehicle might not travel a great distance, and not in all records does a vehicle's movement span across grids. Figure 8 displays the representative trajectory topics generated through an enhanced MDP, presenting an overview of the topics using an abstract map approach. Due to scenarios in the computational data where vehicles move within a single grid, sequences of movements within a single grid arise, as illustrated in Figure 8d,e. This reflects real situations present in the trajectory data. Additionally, trajectory sequences generated over time in topics 6 and 12 can supplement the trajectory direction information on the map, indicating the most probable movement patterns under that time's trajectory topic.

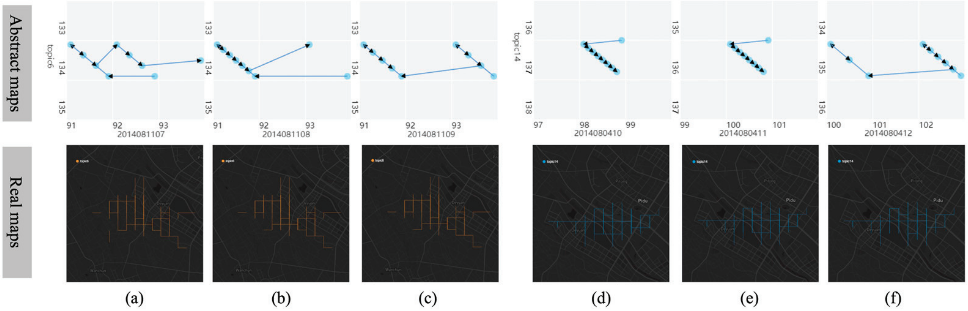


Figure 8. Analysis of trajectory topic sequence through abstract map and real map. The first row is the abstract map generated by improving the MDP, and the second row is the real map corresponding to the time and topic. (**a**–**f**) represent the combinations for abstract map and real map at different time slices.

6. Discussion

The core of the dynamic trajectory topic model is the evolution of trajectory topics. Compared to previous works on trajectory topic computation [2,9,11,14,18], this study introduces a method for the temporal evolution of topics. This method facilitates dynamic trajectory topic analysis by connecting the model parameters to the topic models of adjacent windows. It is more suited to real-world scenarios than matrix factorization-based trajectory topic analysis methods such as NMF.

The order of topic words is an essential component of the semantic structure in dynamic trajectory topics. Moreover, while the topic model is a probabilistic model regarding the composition of topic words, previous works did not consider the relationships between topic words, that is, the sequence of trajectory segments. In this study, an enhanced MDP, grounded on the probabilities of topic words, is introduced for generating representative trajectory sequences under specific trajectory topics. Our approach preserved the original data characteristics without imposing intricate feature constraints on the applied trajectory dataset. We hope this method can inspire similar research to focus on the inherent features of trajectory data and the continuity of trajectory information in the real world. However, there are certain limitations to our study. During the computation of dynamic trajectory topics, due to computational constraints inherent to the LDA model, a significant number of computational resources is required when fitting the LDA. Furthermore, there is a need to explore new trajectory point aggregation methods to compress the original trajectory point data for more efficient sequences.

7. Conclusions

A dynamic topic analysis approach for traffic trajectory data was proposed in response to the issue of existing trajectory visualization analysis methods overlooking the temporal evolution of trajectory topics. This facilitates the visual exploration and analysis of traffic trajectory topics and their evolution. Initially, the DTM is employed to characterize the dynamic topics of trajectories, seamlessly integrating both temporal and spatial information into the DTM. Subsequently, we propose an enhanced Markov decision process to generate topic trajectory sequences, thereby incorporating trajectory sequence information into the trajectory topics. Following this, visual views designed based on parallel windows are introduced for exploring the evolution of trajectory topics. The objective is to convey information about dynamic trajectory topics through concise and intuitive visualization components. Ultimately, the efficacy of our methodology is illustrated through a case study focusing on the evolution of trajectory topics.

Since the trajectory dynamic topic model algorithm is based on a variational method to fit a posteriori distribution, there is room to enhance its computational efficiency. In the future, we plan to improve the efficiency of the dynamic trajectory topic model algorithm by adopting distributed learning algorithms, such as federated learning. Moreover, future research could explore the patterns and relationships of dynamic trajectory topics within and outside grids under various grid partitioning methods and investigate efficient and compact visualizations of spatiotemporal trajectory features.

Author Contributions: Conceptualization, H.C., Y.W. and W.Z.; software, H.T., J.L. (Jing Lei) and Z.W.; data preparation, G.W. and J.L. (Jing Liao); evaluation, H.C. and F.W.; Manuscript, H.C., G.W. and W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61872304.

Data Availability Statement: Data available in online website [29].

Conflicts of Interest: Authors Huaquan Tang and Jing Lei are employed by the company Mianyang Xinchen Engine Co., Ltd, Mianyang, Sichuan, China. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

Latent Dirichlet Allocation (LDA). Non-negative Matrix Factorization (NMF). Biterm Topic Model (BTM). dynamic topic model (DTM). Markov Decision Process (MDP).

References

1. Liu, L.; Zhang, H.; Liu, J.; Liu, S.; Chen, W.; Man, J. Visual exploration of urban functional zones based on augmented nonnegative tensor factorization. *J. Vis.* **2021**, *24*, 331–347. [CrossRef]
2. Liu, D.; Xu, P.; Ren, L. TPFlow: Progressive Partition and Multidimensional Pattern Extraction for Large-Scale Spatio-Temporal Data Analysis. *IEEE Trans. Vis. Comput. Graph.* **2019**, *25*, 1–11. [CrossRef] [PubMed]
3. Zhou, Z.; Meng, L.; Tang, C.; Zhao, Y.; Guo, Z.; Hu, M.; Chen, W. Visual Abstraction of Large Scale Geospatial Origin-Destination Movement Data. *IEEE Trans. Vis. Comput. Graph.* **2019**, *25*, 43–53. [CrossRef] [PubMed]
4. Deng, Z.; Weng, D.; Liang, Y.; Bao, J.; Zheng, Y.; Schreck, T.; Xu, M.; Wu, Y. Visual cascade analytics of large-scale spatiotemporal data. *IEEE Trans. Vis. Comput. Graph.* **2021**, *28*, 2486–2499. [CrossRef] [PubMed]
5. Andrienko, G.; Andrienko, N.; Fuchs, G.; Wood, J. Revealing Patterns and Trends of Mass Mobility Through Spatial and Temporal Abstraction of Origin-Destination Movement Data. *IEEE Trans. Vis. Comput. Graph.* **2017**, *23*, 2120–2136. [CrossRef] [PubMed]
6. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
7. Lee, D.; Seung, H.S. Algorithms for non-negative matrix factorization. *Adv. Neural Inf. Process. Syst.* **2000**, *13*, 535–541.
8. Chu, D.; Sheets, D.A.; Zhao, Y.; Wu, Y.; Yang, J.; Zheng, M.; Chen, G. Visualizing Hidden Themes of Taxi Movement with Semantic Transformation. In Proceedings of the 2014 IEEE Pacific Visualization Symposium, Yokohama, Japan, 4–7 March 2014; pp. 137–144.
9. Liu, L.; Zhan, H.; Liu, J.; Man, J. Visual analysis of traffic data via spatio-temporal graphs and interactive topic modeling. *J. Vis.* **2019**, *22*, 141–160. [CrossRef]
10. Liu, H.; Jin, S.; Yan, Y.; Tao, Y.; Lin, H. Visual analytics of taxi trajectory data via topical sub-trajectories. *Vis. Inform.* **2019**, *3*, 140–149. [CrossRef]
11. Tao, Y.; Tang, Y. Progressive visual analysis of traffic data based on hierarchical topic refinement and detail analysis. *J. Vis.* **2023**, *26*, 367–384. [CrossRef]
12. Chen, C.; Liu, Q.; Wang, X.; Liao, C.; Zhang, D. semi-Traj2Graph Identifying Fine-Grained Driving Style With GPS Trajectory Data via Multi-Task Learning. *IEEE Trans. Big Data* **2021**, *8*, 1550–1565. [CrossRef]
13. Blei, D.M.; Lafferty, J.D. Dynamic topic models. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 113–120.

14. Liao, L.; Wu, J.; Zou, F.; Pan, J.; Li, T. Trajectory topic modelling to characterize driving behaviors with GPS-based trajectory data. *J. Internet Technol.* **2018**, *19*, 815–824.
15. Huang, L.; Wen, Y.; Guo, W.; Zhu, X.; Zhou, C.; Zhang, F.; Zhu, M. Mobility pattern analysis of ship trajectories based on semantic transformation and topic model. *Ocean Eng.* **2020**, *201*, 107092. [CrossRef]
16. Wallach, H.M. Topic modeling: Beyond bag-of-words. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006.
17. Mohammadiha, N.; Smaragdis, P.; Panahandeh, G.; Doclo, S. A state-space approach to dynamic nonnegative matrix factorization. *IEEE Trans. Signal Process.* **2014**, *63*, 949–959. [CrossRef]
18. Yao, F.; Wang, Y. Tracking urban geo-topics based on dynamic topic model. *Comput. Environ. Urban Syst.* **2020**, *79*, 101419. [CrossRef]
19. Chen, W.; Huang, Z.; Wu, F.; Zhu, M.; Guan, H.; Maciejewski, R. VAUD: A visual analysis approach for exploring spatio-temporal urban data. *IEEE Trans. Vis. Comput. Graph.* **2017**, *24*, 2636–2648. [CrossRef]
20. Liao, C.; Chen, C.; Zhang, Z.; Xie, H. Understanding and visualizing passengers' travel behaviours: A device-free sensing way leveraging taxi trajectory data. *Pers. Ubiquitous Comput.* **2019**, *26*, 491–503. [CrossRef]
21. Havre, S.L.; Hetzler, E.G.; Nowell, L.T. ThemeRiver: Visualizing theme changes over time. In Proceedings of the IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings, Salt Lake City, UT, USA, 9–10 October 2000; pp. 115–123.
22. He, J.; Chen, C. Spatiotemporal analytics of topic trajectory. In Proceedings of the 9th International Symposium on Visual Information Communication and Interaction, Dallas, TX, USA, 24–26 September 2016; pp. 112–116.
23. Gao, X.; Liao, C.; Chen, C.; Li, R. Visual Exploration of Cycling Semantics with GPS Trajectory Data. *Appl. Sci.* **2023**, *13*, 2748. [CrossRef]
24. Al-Dohuki, S.; Wu, Y.; Kamw, F.; Yang, J.; Li, X.; Zhao, Y.; Ye, X.; Chen, W.; Ma, C.; Wang, F. Semantictraj: A new approach to interacting with massive taxi trajectories. *IEEE Trans. Vis. Comput. Graph.* **2016**, *23*, 11–20. [CrossRef]
25. Ali, F.; Kwak, D.; Khan, P.; El-Sappagh, S.; Ali, A.; Ullah, S.; Kim, K.H.; Kwak, K.-S. Transportation sentiment analysis using word embedding and ontology-based topic modeling. *Knowl.-Based Syst.* **2019**, *174*, 27–42. [CrossRef]
26. Yan, Y. Visual Analytics Based on Topic Models. Ph.D. Thesis, Zhejiang University, Hangzhou, China, 2019.
27. Zhou, Z.; Zhang, X.; Yang, Z.; Chen, Y.; Liu, Y.; Wen, J.; Chen, B.; Zhao, Y.; Chen, W. Visual Abstraction of Geographical Point Data with Spatial Autocorrelations. In Proceedings of the 2020 IEEE Conference on Visual Analytics Science and Technology (VAST), Salt Lake City, UT, USA, 25–30 October 2020; pp. 60–71.
28. Wang, H.; Ni, Y.; Sun, L.; Chen, Y.; Xu, T.; Chen, X.; Su, W.; Zhou, Z. Hierarchical visualization of geographical areal data with spatial attribute association. *Vis. Inform.* **2021**, *5*, 82–91. [CrossRef]
29. Chengdu Taxi GPS Data. Available online: https://www.pkbigdata.com/common/zhzgbCmptDetails.html (accessed on 21 January 2024).
30. Newman, D.; Lau, J.H.; Grieser, K.; Baldwin, T. Automatic evaluation of topic coherence. In Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, CA, USA, 2–4 June 2010; pp. 100–108.
31. Mimno, D.; Wallach, H.; Talley, E.; Leenders, M.; McCallum, A. Optimizing semantic coherence in topic models. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, UK, 27–31 July 2011; pp. 262–272.
32. Aletras, N.; Stevenson, M. Evaluating topic coherence using distributional semantics. In Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers, Potsdam, Germany, 19–22 March 2013; pp. 13–22.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

RoCS: Knowledge Graph Embedding Based on Joint Cosine Similarity

Lifeng Wang [1], Juan Luo [1], Shiqiao Deng [1] and Xiuyuan Guo [2,*]

[1] School of Computer Science and Electronic Engineering, Hunan University, Changsha 410000, China; rick@hnu.edu.cn (L.W.); juanluo@hnu.edu.cn (J.L.); shiqiaodeng@hnu.edu.cn (S.D.)

[2] School of Journalism and Communication, Hunan University, Changsha 410000, China

* Correspondence: guoxiuyuan@hnu.edu.cn

Abstract: Knowledge graphs usually have many missing links, and predicting the relationships between entities has become a hot research topic in recent years. Knowledge graph embedding research maps entities and relations to a low-dimensional continuous space representation to predict links between entities. The present research shows that the key to the knowledge graph embedding approach is the design of scoring functions. According to the scoring function, knowledge graph embedding methods can be classified into dot product models and distance models. We find that the triple scores obtained using the dot product model or the distance model were unbounded, which leads to large variance. In this paper, we propose RotatE Cosine Similarity (RoCS), a method to compute the joint cosine similarity of complex vectors as a scoring function to make the triple scores bounded. Our approach combines the rotational properties of the complex vector embedding model RotatE to model complex relational patterns. The experimental results demonstrate that the newly introduced RoCS yields substantial enhancements compared to RotatE across various knowledge graph benchmarks, improving up to 4.0% in hits at 1 (Hits@1) on WN18RR and improving up to 3.3% in Hits@1 on FB15K-237. Meanwhile, our method achieves some new state-of-the-art (SOTA), including Hits@3 of 95.6%, Hits@10 of 96.4% on WN18, and mean reciprocal rank (MRR) of 48.9% and Hits@1 of 44.5% on WN18RR.

Keywords: complex vectors; embedding; joint cosine similarity; knowledge graphs; scoring function; unbounded

Citation: Wang, L.; Luo, J.; Deng, S.; Guo, X. RoCS: Knowledge Graph Embedding Based on Joint Cosine Similarity. *Electronics* **2024**, *13*, 147. https://doi.org/10.3390/electronics13010147

Academic Editor: Silvia Liberata Ullo

Received: 19 November 2023
Revised: 19 December 2023
Accepted: 27 December 2023
Published: 28 December 2023

1. Introduction

Theknowledge graph is composed of many fact triples (*head entity*, *relation*, *tail entity*), in the directed graph, the source and target nodes correspond to the head and tail entities, respectively, while the relations are depicted as edges [1,2]. In recent years, knowledge graphs (KGs) have found applications across a broad spectrum of real-world scenarios, including intelligent question answering [3], personalized recommendation [4,5], natural language processing [6], and object detection [7,8]. However, real-world knowledge graphs including WordNet [9], Freebase [10], or Yago [11] are usually incomplete. In recent years, predicting missing links through knowledge graph embedding (KGE) has gained substantial attention as a pivotal research area in achieving knowledge graph completion [2].

KGE transforms the entities and relations within the knowledge graph into low-dimensional continuous space representations. Each fact triple (*head entity*, *relation*, *tail entity*) is represented as (h, r, t). If entities and relations are represented using d-dimensional real vectors, $h, r, t \in \mathbb{R}^d$. To evaluate the performance of the entity and the relation representation, the KGE approach evaluates the credibility of the triples by designing a scoring function. The optimization objective of KGE is geared towards ensuring that elevated scores are assigned to positive triples, while negative triples receive lower scores. Presently, prevailing KGE methods can be classified into dot product models and distance models based on the structure of the scoring function. The dot product model is used as a triple scoring function by calculating the dot product between head entity embedding, relation embedding, and tail entity embedding. Examples of such methods include DistMult [12], HoLE [13], ComplEx [14], QuatE [15], which capture semantic information through pairwise feature interactions between potential factors. The distance model uses L_1 or L_2 distance as the scoring function. Among them, the translation model TransE [16] and the complex embedding model RotatE [17] can be classified into this category. In the distance model, the process involves either adding the head entity embedding to the relation embedding or computing the Hadamard product to obtain a vector close to the tail entity embedding. Subsequently, the distance between the two vectors is calculated. Such methods as TransH [18], TransR [19], TorusE [20] and GIE [21] utilize translation invariance to preserve the original semantic relationships.

We observe that the scoring function's range remains unbounded, irrespective of whether it belongs to the dot product model or the distance model. This unrestricted range raises the potential for increased variance. The score range of the dot product model is from negative infinity to positive infinity, and the score range of the distance model is from 0 to positive infinity. Given that the score range of triples is unbounded, the sensitivity of triple scores to variations in both the embeddings of entities and relations results in substantial model variance. To resolve this issue, a straightforward approach is to normalize [22] both the embeddings of entities and relations. The range of triple scores is guaranteed to be bounded by eliminating the difference in numerical values between each feature. However, such approach is affected by the dimensionality of the embedding, and the score range varies for embeddings of different dimensions. As a result, to obtain a fixed bounded range, we adopt the use of cosine similarity as a scoring function. Firstly, the cosine similarity is used as a normalization mechanism, independent of the embedding dimension, and its score is fixed in the range of -1 to 1. Secondly, cosine similarity stands out as a widely employed semantic similarity measure, commonly used to assess the similarity between document vectors [23–25]. Smaller angles between similar vectors aid in distinguishing the encoded information of various types of entity embeddings.

To achieve this goal, we propose RoCS, a KGE based on joint cosine similarity. Cosine similarity is chosen for its bounded range, dimensionality independence, and effectiveness in capturing semantic relationships. This measure ensures numerical stability during training and adapts seamlessly to varying dimensions of embeddings. The rotation embedding model RotatE [17] is a stronger baseline for reasoning about three important relational patterns in knowledge graphs, i.e., symmetric/antisymmetric, inverse and composition. RotatE uses L_2 distance as the score function to score an unbounded range, while we consider the use of cosine similarity as the score function to ensure that the score range is bounded. However, directly calculating the cosine similarity result for two complex vectors can be intricate, while we need a real number result to score the triplet. To address this challenge, we present a joint cosine similarity calculation method as the complex vector cosine similarity, as shown in Figure 1. Specific, we merge the real and imaginary aspects of the complex vector into a novel joint vector, and subsequently compute the cosine similarity of this joint vector. It can be found that the joint cosine similarity does not change the range of the calculated results while reflecting the overall similarity between the two complex vectors. We evaluate the performance of our method on FB15K [16], FB15K-237 [26], WN18 [16], and WN18RR [27] datasets for the link prediction task. Experimental results show that our method outperforms the current state-of-the-art complex vector embedding models ComplEx [14] and RotatE [17] on all evaluation metrics for all datasets. Furthermore, the proposed method RoCS is highly competitive with the current state-of-the-art methods [15,28,29]. We also explore various techniques for computing the cosine similarity of complex vectors, and through experiments, we validate the superiority of our proposed joint method over other approaches.

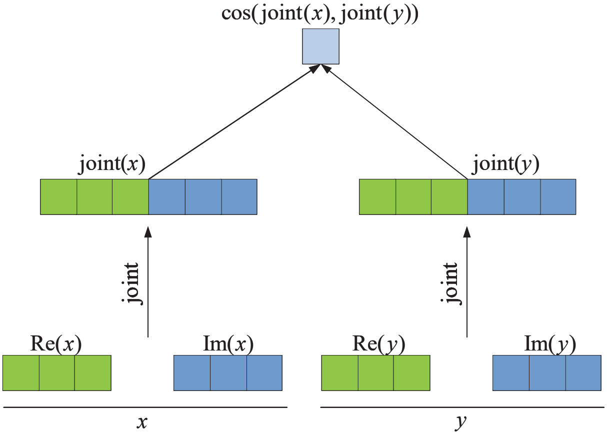


Figure 1. A demonstration of the process of calculating the joint cosine similarity of the complex vectors.

In summary, the primary contributions of our work are as follows:

- We propose a joint cosine similarity method to calculate the complex vector similarity as a scoring function.
- Our approach combines the rotational properties of the complex vector model RotatE to reason about a variety of important relational patterns.
- We have experimentally verified that the proposed RoCS provides a significant improvement over RotatE and achieves results close to or even higher than the current state-of-the-art.

2. Related Work

KGE predicts missing links by mapping symbolic representations of entities and relations into vector or matrix representations. Most KGE methods [30] are considered to utilize triples as learning resources, deriving the semantics of entities and relations from graph structures. Preserving original semantic relations through scoring function design has become a key research focus in recent years [1,2]. Based on the scoring function's structure, the majority of the work can be categorized into dot product models and distance models.

The dot product model takes the form of dot product operations on the head entity, relation, and tail entity. Semantic information is captured through pairwise feature interactions between potential factors. The earliest work is the RESCAL [31], which uses a matrix to represent the relation $r \in \mathbb{R}^{d\times d}$ and a vector to represent the entities $h, t \in \mathbb{R}^d$. To reduce the relation embedding parameters, DistMult [12] constrains the relation matrix to be a diagonal matrix and uses a vector to represent the relation $r \in \mathbb{R}^d$. Since DistMult is overly simple and can only infer symmetric relations. HolE [13] utilizes the cyclic correlation dot product operation to infer anti-symmetric relations. ComplEx [14] applies a complex space to encode entities and relations, utilizing the complex conjugate property to model anti-symmetric relations. To further facilitate feature interaction, QuatE [15] suggests the use of quaternion spaces to represent entities and relations. In addition, there are neural network models including ConvE [27], InteractE [32], graph neural networks [33,34] and tensor decomposition models Tucker [28], LowFER [29] can also be regarded as dot product models.

Distance models utilize relations to translate or rotate the head entity and subsequently calculate the distance to the tail entity as the scoring function. In the case of TransE [16], the relationship is a translation originating from the head entity and extending to the tail entity. Guided by the principle of translation invariance, the sum of the head entity embedding and the relation embedding is expected to be close to the distance between the tail entity embeddings. Consequently, TransE uses the L_1 or L_2 distances as a scoring function. Since the TransE model cannot handle N-to-N relationships, TransH [18] presents a hyperplane representation that maps entities to relationship specifications. TransR [19] consider simplifying the space specified by the hyperplane for relationships. The complex embedding model RotatE [17] has been recently proposed, which uses a complex space to represent entities and relations. RotatE utilizes Euler's formula to represent the relationship as a rotational operation between the head entity and the tail entity. By leveraging the rotation property, RotatE deduces various essential relation patterns [17].

Nevertheless, whether using the dot product model or distance model, the triple scores remain unbounded. Substantial score disparities between positive and negative samples amplify variance and diminish the model's generalization capability. In contrast to prior approaches, we propose the method of computing the joint cosine similarity of complex vectors as a scoring function to constrain the bounded triple scoring range. Moreover, we propose a KGE method utilizing joint cosine similarity. Our work combines the RotaE rotation property of the complex vector embedding model to model a variety of different relational patterns. Table 1 summarizes our approach with other related work. The normalization effect of cosine similarity can reduce the variance and prevent gradient vanishing [35]. Moreover, cosine similarity finds extensive application in natural language processing for assessing the similarity of words, sentences, and document vectors [23–25]. The angle between similar vectors should be smaller, which can also help to distinguish different types of entities. In short, the main motivations behind these models include (1) using cosine similarity can make the triple scores bounded and reduce the variance, (2) distinguishing the embedding information of various entity types, and (3) reflecting the difference in direction between vectors.

Table 1. Comparison of our approach with several representative knowledge graph embedding models in the representation space, score range.

Models	Scoring Function	Representation Space	Score Range
TransE [16]	$-\lVert h+r-t \rVert_{1/2}$	$h,r,t \in \mathbb{R}^d$	Unbounded
DistMult [12]	$\langle h,r,t \rangle$	$h,r,t \in \mathbb{R}^d$	Unbounded
ComplEx [14]	$\mathrm{Re}(\langle h,r,\bar{t} \rangle)$	$h,r,t \in \mathbb{C}^d$	Unbounded
ConvE [27]	$f(\mathrm{vec}(f([h;r] * \omega))\mathrm{W})t$	$h,r,t \in \mathbb{R}^{d'}$	Unbounded
RotatE [17]	$-\lVert h \circ r - t \rVert$	$h,r,t \in \mathbb{C}^d$	Unbounded
QuatE [15]	$h \otimes r^{\triangleleft} \cdot t$	$h,r,t \in \mathbb{H}^d$	Unbounded
LowFER [29]	$\left(S^k \, \mathrm{diag}(U^T h) V^T r\right)^T t$	$h,r,t \in \mathbb{R}^{n \times d}$	Unbounded
RoCS (ours)	$\cos_{\mathrm{joint}}(h \circ r, t)$	$h,r,t \in \mathbb{C}^d$	Bounded

3. RoCS

In this section, we introduce the RoCS method. Initially, we present the novel concept of joint cosine similarity for complex vectors, followed by the introduction of the scoring function derived from this joint cosine similarity. Subsequently, we outline the training methodology and conclude with a detailed discussion of the proposed approach.

3.1. Joint Cosine Similarity of Complex Vectors

Given two complex vectors $x, y \in \mathbb{C}^d$, the definition of cosine similarity is given by the dot product and the vector length. As per the cosine similarity definition, the formula for complex vector cosine similarity calculation is as follows:

$$\cos(x,y) = \frac{x \cdot y}{|x||y|} = \frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum_{i=1}^{d} x_i \bar{x}_i}\sqrt{\sum_{i=1}^{d} y_i \bar{y}_i}}, \tag{1}$$

where $x_i, y_i \in \mathbb{C}$ denotes each elementary component of the complex vector x, y and $\bar{x}_i$ denotes the conjugate complex number of x. Since each complex number includes both real and imaginary components, the dot product part of Equation (1) is calculated as follows,

$$\begin{aligned} x \cdot y = \quad & \textstyle\sum_{i=1}^{d} (\mathrm{Re}(x_i)\,\mathrm{Re}(y_i) - \mathrm{Im}(x_i)\,\mathrm{Im}(y_i)) \\ & \quad +j(\mathrm{Re}(x_i)\,\mathrm{Im}(y_i) + \mathrm{Im}(x_i)\,\mathrm{Re}(y_i)), \end{aligned} \tag{2}$$

where $j^2 = -1$ denotes the complex number sign.

The dot product between the complex vectors results in a complex number. Since the score of the triplet requires a real number to evaluate the training. Thus, if the above complex vector cosine similarity formula is used directly to calculate the triplet score will not work. To overcome this shortcoming, ComplEx [14] proposes calculating the complex dot product without considering the imaginary part of the score. Although considering only the real part of the score results can achieve positive performance. However, it is inaccurate to consider only the real part scores because having the same real part scores does not necessarily mean having the same imaginary part scores.

In contrast to previous work, we introduce a joint cosine similarity calculation method so that both the real part and the imaginary part scores results are considered. First, the complex vector's real and imaginary components are treated as a combined pair of vectors. Then, the combined vector cosine similarity is calculated. Figure 1 illustrates the joint cosine similarity calculation process. The formula for calculating the joint cosine similarity of complex vectors is as follows,

$$\begin{aligned} & \cos_{\mathrm{joint}}(x, y) = \cos(\mathrm{joint}(x), \mathrm{joint}(y)) \\ & = \frac{\mathrm{Re}(x)\cdot\mathrm{Re}(y)+\mathrm{Im}(x)\cdot\mathrm{Im}(y)}{|x|\cdot|y|} \\ & = \frac{\sum_{i=1}^{d}\mathrm{Re}(x_i)\,\mathrm{Re}(y_i)+\sum_{i=1}^{d}\mathrm{Im}(x_i)\,\mathrm{Im}(y_i)}{\sqrt{\sum_{i=1}^{d} x_i\bar{x}_l}\sqrt{\sum_{i=1}^{d} y_i\bar{y}_i}}, \end{aligned} \tag{3}$$

where joint(x) represents the joint of the real part vector of x with the imaginary part vector. Our method preserves the original real and imaginary components by converting a d-dimensional complex vector into a 2d-dimensional real vector before calculating the cosine similarity. Therefore, it can more accurately reflect the degree of similarity of complex vectors.

3.2. Scoring Function Based on Joint Cosine Similarity

In this part, we introduce the RoCS scoring function, which is founded on the concept of joint cosine similarity. We combine the rotational properties of the complex vector embedding model RotatE [17] to model a variety of important relational patterns. RotatE uses a complex vector to represent the head entity h, the relation r, and the tail entity t, i.e., $h, r, t \in \mathbb{C}^d$. RotatE shares similarities with TransE [16], the process involves rotating the head entity's embedding through the relational embedding and subsequently computing the distance to the tail entity as a scoring function. For each element within the embedding vector, the rotation model expects $t_i = h_i r_i$, where $h_i, r_i, t_i \in \mathbb{C}, |r_i| = 1$. With Euler's formula, we can obtain $e^{j\theta_{t,i}} = e^{j\theta_{h,i}} e^{j\theta_{r,i}}$, i.e., $\theta_{t,i} = \theta_{h,i} + \theta_{r,i}$, where j denotes the complex symbol and θ denotes the corresponding complex space phase. It can be found that the rotation model rotates the head entity by Euler angles.

Given that the distance model anticipates the head entity embedding to be in closer proximity to the tail entity following translation or rotation through the relational embedding. RotatE expects the distance between the two complex vectors to be minimized. We believe that there are several problems. First, the higher triple score and smaller distances conflict with each other, which makes the rotation mode have to solve the problem by multiplying the scores by a negative numbers transformation. Second, the range of the triple score is calculated using distance as the score function is unbounded. The significant contrast in scores between positive and negative samples amplifies variance and diminishes the model's generalization capability. Third, the distance model is easily influenced by the embedded dimensionality. The enlargement of the embedding dimension leads to an expanded range in triple scores, which is detrimental to effective model training. For these reasons, We suggest employing joint cosine similarity as the scoring function, computed as follows,

$$S(h,r,t) = \cos_{\text{joint}}(h \circ r, t), \tag{4}$$

where $\circ$ is the Hadamard (or element-wise) product and joint(t) denotes the joint vector for computing t, as shown in Figure 1.

The cosine similarity, as a normalization method, calculates the score range from -1 to 1. This ensures that the score of the triples is bounded and reduces the variance. Simultaneously, regardless of whether the embedding dimension increases or decreases, it does not impact the alteration of the score range, providing advantages for model training. Furthermore, the transformation of the distance model is addressed by utilizing cosine similarity as a scoring function, where a higher score signifies greater similarity. In addition, our proposed joint cosine similarity calculation method considers both real and imaginary components, and the scores are more accurate. While the rotation model RotatE reflects the same degree of the complex vectors by distance, our model RoCS reflects the similarity of the two complex vectors by the phase difference. Figure 2 shows the difference between our method RoCS and the rotation model RotatE.

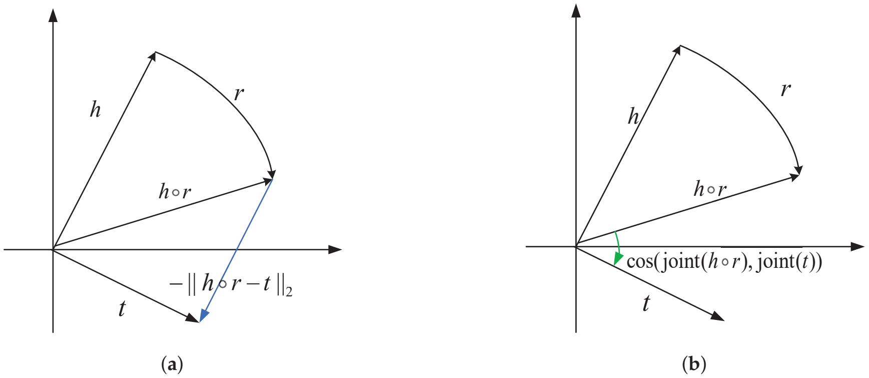


Figure 2. The rotation model calculates the distance between vectors as the score function (left ((**a**) RotatE)), while our approach RoCS uses joint cosine similarity as the score function (right ((**b**) RoCS)).

3.3. Training

Knowledge graph embedding training steps mainly include generating negative samples and designing loss functions. There are uniform sampling, Bernoulli sampling, and self-adversarial negative sampling [17] for generating negative samples. For a fair comparison with the RotatE rotation model, we employ the self-adversarial negative sampling method to generate negative samples. Higher sampling weights are assigned to negative samples with elevated scores, ensuring that these generated negatives contribute more substantial training information. The sampling probability for negative samples in the self-adversarial negative sampling is defined as follows,

$$p\left(h'_j, r, t'_j \mid \{(h_i, r_i, t_i)\}\right) = \frac{\exp \alpha S\left(h'_j, r, t'_j\right)}{\sum_i \exp \alpha S\left(h'_i, r, t'_i\right)}, \tag{5}$$

where α is the sampling hyperparameter.

Loss function design is generally related to the scoring function. The logistic regression loss function [12,14] is typically chosen for the dot product model, and the rank loss [16,19] is typically chosen for the distance model selection [1]. Since the rotation model uses a self-adversarial negative sampling loss function [17], our method also uses the same loss function for a fair comparison. Therefore, the loss function of RoCS is calculated as follows,

$$L = -\log(\sigma(\gamma - \mu S(h,r,t))) - \sum_{i=1}^{n} p\big(h'_i, r, t'_i\big) \log\big(\sigma\big(\mu S\big(h'_i, r, t'_i\big) - \gamma\big)\big), \tag{6}$$

where γ is the fixed margin hyperparameter, μ is the score function scaling hyperparameter. To extend the scoring range of the triplet, We introduce a scaling hyperparameter μ to fine-tune the scoring function, and experimental results demonstrate this helps model training.

3.4. Discussion

In this section, we will first discuss the ability of our method to infer three important patterns of the relations. Then we will discuss the connection of our method with existing methods.

3.4.1. Infer Patterns of the Relations

Knowledge graphs have various relationship patterns, among which the three most common relationship patterns are symmetric/antisymmetric, inverse, and combinatorial. Hence, for precise prediction of missing links in the knowledge graph, the designed score function requires inferring the above three relationship patterns. RotatE [17] expects the tail entity to be equivalent to the head entity after relational rotation, i.e., $t = h \circ r$. All the above relational patterns can be inferred using the rotation property. Specifically, if r is a symmetric relation then $r \circ r = 1$, and if r is an antisymmetric relation then $r \circ r \neq 1$. If r_1 and r_2 are inverse relations, then it is sufficient to satisfy $r_1 = r_2^{-1}$. If r_1 is a combined relationship of r_2, r_3, then $r_1 = r_2 \circ r_3$. The rotation model uses distance as the score function, while our method uses cosine similarity. Since using cosine similarity does not change the rotation property, our method can also infer all the above relationship patterns.

3.4.2. Connection with Existing Methods

Our method can be naturally extended to all distance models. Since both translational and rotational models use relations to translate or rotate the head entity, expecting the rotated head entity to be equivalent to the tail entity, i.e., expecting $t = h + r$ or $t = h \circ r$. Therefore, it is straightforward to use joint cosine similarity instead of distance as the scoring function. After normalizing the entity embedding and relational embedding vectors, the rotation model RotatE [17] is approximately equivalent [36] to our method as follows,

$$\|h_{nor} \circ r_{nor} - t_{nor}\|_2 \approx \sqrt{2\big(1 - \cos_{\text{joint}}(h_{nor} \circ r_{nor}, t_{nor})\big)}, \tag{7}$$

where $h_{nor}, r_{nor}, t_{nor} \in \mathbb{C}^d$, x_{nor} denotes the normalization of the vector x normalization result, i.e., $x_{nor} = \frac{x}{\|x\|_2}$. It is noticed that our method RoCS is approximately equivalent to the normalized RotatE model score function after normalization. When using the real number space to represent entities and relations, the joint cosine similarity is calculated in an exactly equivalent way to the cosine similarity [36]. Therefore, the following equivalence exists between RoCS and the scoring function of the translational model TransE [16] when only the real part is considered,

$$\|h_{nor} + r_{nor} - t_{nor}\|_2 = \sqrt{2(1 - \cos(h_{nor} + r_{nor}, t_{nor})}, \tag{8}$$

where $h_{nor}, r_{nor}, t_{nor} \in \mathbb{R}^d$. In summary, RoCS can be naturally extended to the distance model. We evaluates the similarity based on direction, while the distance model evaluates vector similarity based on distance, as shown in Figure 2.

4. Experiment

In this section, we outline our experimental setup and report the corresponding results. We begin by detailing the experimental configuration. Next, we evaluate the efficacy of our approach compared to the baseline rotation model. Subsequently, we present the results of our method in comparison to state-of-the-art approaches. Finally, we analyze the performance of our model utilizing both the dot product and distance models in the complex space. Additionally, we conduct ablation studies by comparing different methods of computing complex vector cosine similarity.

4.1. Experimental Setup

4.1.1. Datasets

We selected four standard datasets for the link prediction task including FB15k [16], FB15k-237 [26], WN18 [16], and WN18RR [27] to evaluate our proposed method. FB15K is a subset of the real-world knowledge graph Freebase [10], containing 14,951 entities and 1345 relations. FB15k-237 is a subset of FB15K after removing the inverse relations, containing 14,541 entities with 237 relations. WN18 is a subset of WN18 [9], a knowledge graph constructed by vocabulary, containing 40,943 entities and 18 relations. WN18RR is a subset of WN18, with inverse relations removed, containing 40,943 entities and 11 relations. Among them, FB15k-237 and WN18RR primarily exhibit symmetric/anti-symmetric and combinatorial relationship patterns. The principal relationship patterns in FB15K and WN18 involve symmetry/anti-symmetry and inverse patterns. The statistical information of these knowledge graphs is summarized in Table 2.

Table 2. Statistics of datasets.

Dataset	**#Entity**	**#Relation**	**#Train**	**#Valid**	**#Test**
FB15K	14,951	1345	483,142	50,000	59,071
FB15K-237	14,541	237	272,115	17,535	20,466
WN18	40,943	18	141,442	5000	5000
WN18RR	40,943	11	86,835	3034	3134

4.1.2. Evaluation Criterion

We use the score function designed by Equation (4) to score the test triple (h, r, t) with all candidate triples, and calculate the test triple ranking based on the scoring results. All candidate triples are generated by substituting either the head entity or the tail entity of the test triple, i.e., (h, r, t') or (h', r, t). We follow filters setting [16] that all candidate triples are excluded from appearing in the training set, validation set, and test set. Similar to the previous work [28,29], we choose MRR, Hits@*k* ($k \in \{1, 3, 10\}$) as the ranking evaluation metric. MRR represents the average inverse ranking of the test triples, while Hits@*k* indicates the proportion of test triples within the top *k* rankings.

4.1.3. Baselines

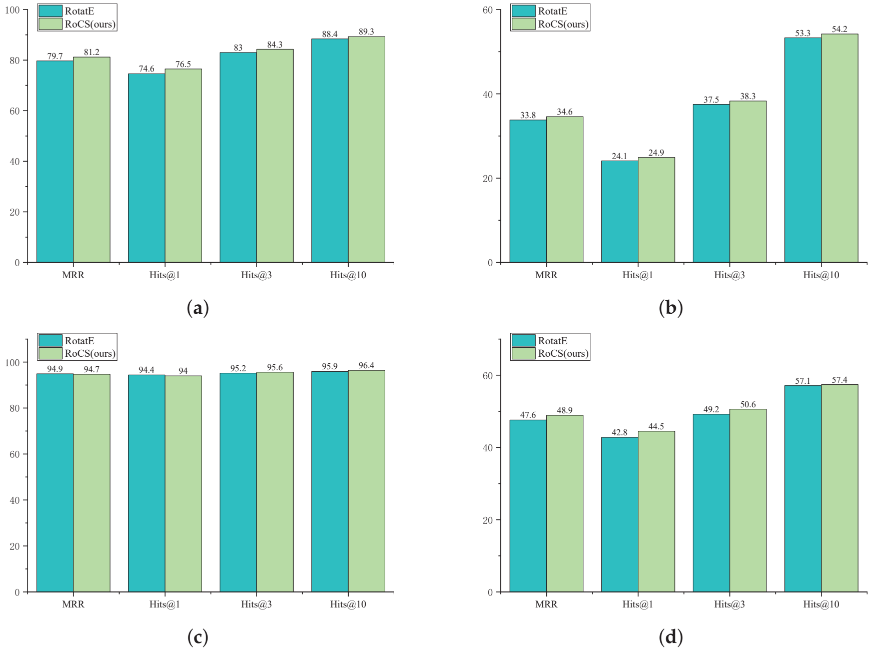
We select some representative baselines from the dot product model and the distance model respectively for comparison. For the dot product model we report DistMult [12], ComplEx [14], ConvE [27], SimplE [37], QuatE [15], TuckER [28] and LowFER [29]. For the distance model we report TransE [16], TorusE [20] and RotatE [17]. Among them, QuatE, TuckER, and LowFER are reported as the latest SOTA models.

4.1.4. Experimental Details

We implement our proposed model based on the Pytorch [38] deep learning framework and train it on an NVIDIA Tesla P100 GPU. We use Adam [39] as the trainer and Equation (6) defines the loss function. We use grid search to determine the hyperparameters, selecting the optimal ones based on evaluating MRR metrics on the validation set. The hyperparameter search range is as follows: fixed margin $\gamma \in \{1, 3, 6, 10, 12, 15, 20, 30\}$, and scaling hyperparameter $\mu \in \{1, 5, 10, 15, 20, 25, 30\ , 50\}$. In addition, the embedding dimension we set the following $d \in \{100, 200,\ 500, 1000\}$, and learning rate $lr \in \{0.001, 0.0001, 0.00003, 0.00001\}$. The initial parameter settings follow the rotation model settings [17].

4.2. Compare RotatE

We first evaluated the performance of our RoCS method alongside the original rotation model RotatE [17]. Figure 3 illustrates the performance of our method RoCS with RotatE on the FB15K, FB15K-237, WN18, and WN18RR datasets. Figure 3a shows that our method improves 1.9% in MRR, 2.5% in Hits@1, 2.6% in Hits@3, and 1.0% in Hits@10 compared to RotatE on the FB15K dataset. Figure 3b shows that our method improves 2.4% in MRR, 3.3% in Hits@1, 2.1% in Hits@3, and 1.7% in Hits@10 compared to RotatE on the FB15K-237 dataset. Figure 3c shows that our method improves -0.2% in MRR, -0.1% in Hits@1, 0.4% in Hits@3, and 0.5% in Hits@10 compared to RotatE on the WN18 dataset. Figure 3d shows that our method improves 2.7% in MRR, 4.0% in Hits@1, 2.8% in Hits@3, and 0.5% in Hits@10 compared to RotatE on the WN18RR dataset. Overall, our method RoCS shows a significant improvement compared to RotatE. It shows that using joint cosine similarity as a scoring function to constrain the bounded triple scores can reduce the variance and improve the model generalization.



(**a**) (**b**) (**c**) (**d**)

Figure 3. Comparison of the link prediction results of our method RoCS with the original rotation model RotatE [17] on FB15K, FB15K-237, WN18 and WN18RR. (**a**) FB15K; (**b**) FB15K-237; (**c**) WN18; (**d**) WN18RR.

4.3. Comparison with Current SOTA Models

The main results of the link predictions are shown in Tables 3 and 4. Table 3 indicates that our RoCS method achieves new benchmarks, achieving a SOTA performance, including Hits@3 of 95.6%, Hits@10 of 96.4% on WN18, and MRR of 48.9%, Hits@1 of 44.5% on WN18RR. Table 4 illustrates that our method RoCS is highly competitive in the FB15K dataset, with MRR, Hits@1, and Hits@3 only below the SOTA model LowFER [29]. And, our method also achieved a top 3 ranking on WN18RR. In short, neither SOTA models TuckER [28], LowFER nor QuatE [15] models can achieve excellent performance in all datasets. TuckER and LowFER perform poorly in the WN18RR dataset, and QuatE models achieve lower performance in FB15K MRR, Hits@1, and Hits@3, while our method RoCS achieves competitive results in all datasets. This indicates that our approach is highly competitive with the current leading knowledge graph embedding models.

Table 3. Link predictions in WN18 and WN18RR results.State-of-the-art (SOTA) results are shown in bold, and top 3 ranking results are underlined. QuatE [15] reports QuatE^3 results, and LowFER [29] reports LowFER-k^* results.

Models	**WN18**				**WN8RR**			
	MRR	**Hits@1**	**Hits@3**	**Hits@10**	**MRR**	**Hits@1**	**Hits@3**	**Hits@10**
TransE [16]	49.5	11.3	88.8	94.3	22.6	-	-	50.1
DistMult [12]	79.7	-	-	94.6	43.0	39.0	44.0	49.0
ComplEx [14]	94.3	93.5	94.6	95.6	46.0	39.0	43.0	48.0
ConvE [27]	94.2	93.9	94.4	94.7	-	-	-	-
SimplE [37]	94.7	94.3	95.0	95.4	-	-	-	-
TorusE [20]	94.1	93.6	94.5	94.7	44.0	41.0	46.0	51.0
RotatE [17]	94.9	94.4	95.2	95.9	47.6	42.8	49.2	57.1
QuatE [15]	95.0	94.5	95.4	95.9	48.8	43.8	**50.8**	**58.2**
TuckER [28]	**95.3**	**94.9**	95.5	95.8	47.0	44.3	48.2	52.6
LowFER [29]	95.0	94.6	95.2	95.8	46.5	43.4	47.9	52.6
RoCS (ours)	94.7	94.0	**95.6**	**96.4**	**48.9**	**44.5**	50.6	57.4

Table 4. Link predictions are shown in FB15K and FB15K-237 results.

Models	**FB15K**				**FB15K-237**			
	MRR	**Hits@1**	**Hits@3**	**Hits@10**	**MRR**	**Hits@1**	**Hits@3**	**Hits@10**
TransE [16]	46.3	29.7	57.8	74.9	29.4	-	-	46.5
DistMult [12]	65.4	54.6	73.3	72.8	24.1	15.5	26.3	41.9
ComplEx [14]	69.2	59.9	75.9	84.0	32.5	23.7	25.6	50.1
ConvE [27]	74.5	67.0	80.1	87.3	-	-	-	-
SimplE [37]	72.7	66.0	77.3	83.8	-	-	-	-
TorusE [20]	73.3	67.4	77.1	83.2	24.7	15.8	27.5	42.8
RotatE [17]	79.7	74.6	83.0	88.4	33.8	24.1	37.5	53.3
QuatE [15]	78.2	71.1	83.5	**90.0**	34.8	24.8	38.2	**55.0**
TuckER [28]	79.5	74.1	83.3	89.2	35.8	**26.6**	39.3	54.4
LowFER [29]	**82.4**	**78.2**	**85.2**	89.7	**35.9**	**26.6**	**39.6**	54.4
RoCS (ours)	81.2	76.5	84.3	89.3	34.6	24.9	38.3	54.2

4.4. Comparing Complex Vector Embeddings

To further investigate the effectiveness of cosine similarity as a score function, we investigated using the same representation space to compare our method with the dot product model ComplEx [14], and the distance model RotatE [17]. For a fair comparison, we additionally train the ComplEx model utilizing a self-adversarial negative sampling loss function [17]. As shown in Table 5, RoCS is significantly outperformed by ComplEx and RotatE. This indicates that using the joint cosine similarity as a scoring function for the complex vector embedding model is better than the dot product model and the distance model.

Table 5. A comparison of the complex vector embedding models ComplEx, RotatE and RoCS (ours), where ComplEx also uses a self-adversarial negative sampling loss function for fair comparison.

Models	**WN18**		**WN8RR**		**FB15K**		**FB15K-237**	
	MRR	**Hits@10**	**MRR**	**Hits@10**	**MRR**	**Hits@10**	**MRR**	**Hits@10**
ComplEx$_{adv}$	89.2	95.4	47.0	55.7	78.0	89.0	32.1	50.9
RotatE	**94.9**	95.9	47.6	57.1	79.7	88.4	33.8	53.3
RoCS(ours)	94.7	**96.4**	**48.9**	**57.4**	**81.2**	**89.3**	**34.6**	**54.2**

4.5. Ablation Study

In Section 3.2, we propose to use the joint cosine similarity as a scoring function to improve the model's performance. To verify the validity of the joint cosine similarity, we compare it with other methods for calculating the cosine similarity of complex vectors. The first method is similar to ComplEx [14], which considers only the real part of the scoring function and is formulated as follows,

$$S_1(h,r,t) = \mathrm{Re}(\cos(h \circ r, t)), \tag{9}$$

where $h, r, t \in \mathbb{C}^d$, Re denotes the real part. The second approach combines the cosine similarity of real vectors and the cosine similarity of imaginary vectors separately. The calculation is as follows,

$$S_2(h,r,t) = \cos(\mathrm{Re}(h \circ r), \mathrm{Re}(t)) + \cos(\mathrm{Im}(h \circ r), \mathrm{Im}(t)), \tag{10}$$

where Im denotes the imaginary part.

We compare the way of calculating complex vector cosine similarity in Equation (9) and Equation (10) to prove that our proposed joint cosine similarity is effective. As shown in Table 6, our proposed method for calculating the joint cosine similarity of complex vectors achieves the best results. The first method is not accurate enough for triplet scoring because the imaginary part of the triplet score is discarded. The second method ignores the connection between the real and imaginary parts, and thus also obtains poorer results. In contrast, our method achieves excellent performance in both cases. Therefore, this proves that our calculation method can give a more accurate scoring result by considering both the real part and the imaginary part.

Table 6. The results of using Equation (9) (RoReCS), Equation (10) (RoAddCS) as score functions on WN18, WN18RR, FB15K, FB15K-237.

Models	WN18		WN8RR		FB15K		FB15K-237	
	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10
RoReCS	63.7	90.7	36.7	44.5	49.1	65.9	23.1	33.4
RoAddCS	92.1	95.2	46.0	52.1	68.5	79.0	28.0	44.9
RoCS	**94.7**	**96.4**	**48.9**	**57.4**	**81.2**	**89.3**	**34.6**	**54.2**

5. Conclusions and Future Work

In this paper, we first propose a method to compute the joint cosine similarity of complex vectors. Then, a knowledge graph embedding model based on joint cosine similarity is proposed, named RoCS. Specifically, the proposed RoCS uses joint cosine similarity as a scoring function to constrain the triple score range to be bounded, thus reducing the variance of the model and improving model performance. Meanwhile, our method combines the rotational properties of the RotatE can reason about a variety of important relational patterns. Our experimental results indicate a significant improvement over the original RotatE model, achieving performance levels that closely rival or even surpass the latest advancements in the field. In the future, we plan to further consider extending the joint cosine similarity to other representation learning problems.

Author Contributions: Conceptualization, L.W., S.D. and X.G.; Methodology, L.W., J.L. and S.D.; Writing—original draft, L.W.; Supervision, J.L. and X.G. All authors have read and agreed to the published version of the manuscript

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62372163, Natural Science Foundation of Chongqin under Grant ZE20210011, Key scientific and technological research and development plan of Hunan Province under Grant 2022NK2046.

Data Availability Statement: The data presented in this study are available in the article.

Acknowledgments: We thank the authors of RotatE for the Knowledge Graph embedding open-source code. Our code is based on their open-source code for improvement.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [CrossRef]
2. Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; Philip, S.Y. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 494–514. [CrossRef] [PubMed]
3. Huang, X.; Zhang, J.; Li, D.; Li, P. Knowledge graph embedding based question answering. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, Australia, 11–15 February 2019; pp. 105–113. [CrossRef]
4. Zhang, F.; Yuan, N.J.; Lian, D.; Xie, X.; Ma, W.Y. Collaborative knowledge base embedding for recommender systems. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 13–17 August 2016; pp. 353–362. [CrossRef]
5. Wang, H.; Zhang, F.; Xie, X.; Guo, M. DKN: Deep knowledge-aware network for news recommendation. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1835–1844. [CrossRef]
6. Yang, B.; Mitchell, T. Leveraging Knowledge Bases in LSTMs for Improving Machine Reading. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017. [CrossRef]
7. Qin, Z.; Cen, C.; Jie, W.; Gee, T.S.; Chandrasekhar, V.R.; Peng, Z.; Zeng, Z. Knowledge-graph based multi-target deep-learning models for train anomaly detection. In Proceedings of the 2018 International Conference on Intelligent Rail Transportation (ICIRT), Singapore, 12–14 December 2018; pp. 1–5. [CrossRef]
8. Chen, C.; Li, K.; Zou, X.; Cheng, Z.; Wei, W.; Tian, Q.; Zeng, Z. Hierarchical Semantic Graph Reasoning for Train Component Detection. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 4502–4514. [CrossRef] [PubMed]
9. Miller, G.A. WordNet: A lexical database for English. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]
10. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, BC, Canada, 9–12 June 2008; pp. 1247–1250. [CrossRef]
11. Suchanek, F.M.; Kasneci, G.; Weikum, G. Yago: A core of semantic knowledge. In Proceedings of the 16th International Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 697–706. [CrossRef]
12. Yang, B.; tau Yih, W.; He, X.; Gao, J.; Deng, L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015; Conference Track Proceedings.
13. Nickel, M.; Rosasco, L.; Poggio, T. Holographic embeddings of knowledge graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30. Available online: https://dl.acm.org/doi/10.5555/3016100.3016172 (accessed on 18 December 2023).
14. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex embeddings for simple link prediction. In Proceedings of the International Conference on Machine Learning (ICML), New York, NY, USA, 19–24 June 2016. Available online: https://dl.acm.org/doi/10.5555/3045390.3045609 (accessed on 18 December 2023).
15. Zhang, S.; Tay, Y.; Yao, L.; Liu, Q. Quaternion knowledge graph embeddings. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019; pp. 2735–2745. [CrossRef]
16. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2787–2795. Available online: https://dl.acm.org/doi/10.5555/2999792.2999923 (accessed on 18 December 2023).
17. Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
18. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014; Volume 28. Available online: https://dl.acm.org/doi/10.5555/2893873.2894046 (accessed on 18 December 2023).
19. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29.
20. Ebisu, T.; Ichise, R. Toruse: Knowledge graph embedding on a lie group. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
21. Cao, Z.; Xu, Q.; Yang, Z.; Cao, X.; Huang, Q. Geometry interaction knowledge graph embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2022; Volume 36, pp. 5521–5529.

22. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 6–11 July 2015; pp. 448–456. Available online: https://dl.acm.org/doi/10.5555/3045118.3045167 (accessed on 18 December 2023).
23. Singhal, A. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.* **2001**, *24*, 35–43.
24. Dai, A.M.; Olah, C.; Le, Q.V. Document Embedding with Paragraph Vectors. *arXiv* **2015**, arXiv:1507.07998.
25. Thongtan, T.; Phienthrakul, T. Sentiment classification using document embeddings trained with cosine similarity. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, Florence, Italy, 28 July–2 August 2019; pp. 407–414. [CrossRef]
26. Toutanova, K.; Chen, D. Observed versus latent features for knowledge base and text inference. In Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, Beijing, China, 26–31 July 2015; pp. 57–66. [CrossRef]
27. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2d knowledge graph embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
28. Balažević, I.; Allen, C.; Hospedales, T.M. Tucker: Tensor factorization for knowledge graph completion. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019. [CrossRef]
29. Amin, S.; Varanasi, S.; Dunfield, K.A.; Neumann, G. LowFER: Low-rank Bilinear Pooling for Link Prediction. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 257–268.
30. Jin, L.; Yao, Z.; Chen, M.; Chen, H.; Zhang, W. A Comprehensive Study on Knowledge Graph Embedding over Relational Patterns Based on Rule Learning. In Proceedings of the International Semantic Web Conference, Athens, Greece, 6–10 November 2023; Springer: Cham, Switzerlnd, 2023; pp. 290–308.
31. Nickel, M.; Tresp, V.; Kriegel, H.P. A three-way model for collective learning on multi-relational data. In Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA, 28 June–2 July 2011; Volume 11, pp. 809–816. Available online: https://dl.acm.org/doi/10.5555/3104482.3104584 (accessed on 18 December 2023).
32. Vashishth, S.; Sanyal, S.; Nitin, V.; Agrawal, N.; Talukdar, P. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 3009–3016.
33. Chen, C.; Li, K.; Wei, W.; Zhou, J.T.; Zeng, Z. Hierarchical graph neural networks for few-shot learning. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 240–252. [CrossRef]
34. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In Proceedings of the European Semantic Web Conference, Heraklion, Crete, Greece, 3–7 June 2018; Springer: Cham, Switzerland, 2018; pp. 593–607. [CrossRef]
35. Luo, C.; Zhan, J.; Xue, X.; Wang, L.; Ren, R.; Yang, Q. Cosine normalization: Using cosine similarity instead of dot product in neural networks. In Proceedings of the International Conference on Artificial Neural Networks, Rhodes, Greece, 4–7 October 2018; Springer: Cham, Switzerland, 2018; pp. 382–391. [CrossRef]
36. Kryszkiewicz, M. The cosine similarity in terms of the euclidean distance. In *Encyclopedia of Business Analytics and Optimization*; IGI Global: Hershey, PA, USA, 2014; pp. 2498–2508. [CrossRef]
37. Kazemi, S.M.; Poole, D. Simple embedding for link prediction in knowledge graphs. In Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, Montreal, QC, Canada, 3–8 December 2018; pp. 4284–4295.
38. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019.
39. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Improved A-Star Path Planning Algorithm in Obstacle Avoidance for the Fixed-Wing Aircraft

Jing Li [1,*], Chaopeng Yu [2], Ze Zhang [3], Zimao Sheng [1], Zhongping Yan [2], Xiaodong Wu [4], Wei Zhou [1], Yang Xie [1] and Jun Huang [1]

[1] School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an 710072, China; shengzimao@mail.nwpu.edu.cn (Z.S.); wdyx@mail.nwpu.edu.cn (W.Z.); xieyang1@mail.nwpu.edu.cn (Y.X.); hj1395147870@mail.nwpu.edu.cn (J.H.)

[2] Aviation Industry Corporation of China Leihua Electronic Technology Research Institute, Wuxi 214000, China; yuchaopeng@avic.com (C.Y.); yanzp003@avic.com (Z.Y.)

[3] AVIC Flight Automatic Control Research Institute, Xi'an 710076, China; zz6002@mail.nwpu.edu.cn

[4] Xi'an Microelectronic Technology Institute, Xi'an 710065, China; wuxiaodong@mail.nwpu.edu.cn

* Correspondence: lijingjing_ren@163.com; Tel.: +86-136-5911-0734

Abstract: The flight management system is a basic component of avionics for modern airliners. However, the airborne flight management system needs to be improved and relies on imports; path planning is the key to the flight management system. Based on the classical A* algorithm, this paper proposes an improved A* path planning algorithm, which solves the problem of low planning efficiency and following a non-smooth path. In order to solve the problem of the large amount of data calculation and long planning time of the classical A* algorithm, a new data structure called a "value table" is designed to replace the open table and close table of the classical A* algorithm to improve the retrieval efficiency, and the Heap sort algorithm is used to optimize the efficiency of node sorting. Aiming at the problem that the flight trajectory is hard to follow, the trajectory smoothing optimization algorithm combined with turning angle limit is proposed. The gray value in the digital map is added to the A* algorithm, and the calculation methods of gray cost, cumulative cost, and estimated cost are improved, which can better meet the constraints of obstacle avoidance. Through the comparative simulation verification of the algorithm, the improved A* algorithm can significantly reduce the path planning time to 1% compared to the classical A* algorithm; it can be seen that the proposed algorithm improves the efficiency of path planning and the smoother planned path, which has obvious advantages compared to the classical A* algorithm.

Keywords: path planning; value table; efficiency of path planning; trajectory smoothing optimization

Citation: Li, J.; Yu, C.; Zhang, Z.; Sheng, Z.; Yan, Z.; Wu, X.; Zhou, W.; Xie, Y.; Huang, J. Improved A-Star Path Planning Algorithm in Obstacle Avoidance for the Fixed-Wing Aircraft. *Electronics* **2023**, *12*, 5047. https://doi.org/10.3390/electronics12245047

Academic Editors: Carlos Tavares Calafate and Christos J. Bouras

Received: 10 October 2023
Revised: 28 November 2023
Accepted: 12 December 2023
Published: 18 December 2023

1. Introduction

The onboard flight management system (FMS) is a professional computer system that can automate various flight tasks and reduce manual workload. Modern civil aircraft crews no longer carry flight engineers or navigators. However, the FMS for general civil aviation aircraft is dependent on the introduction from abroad [1]. FMS is a basic component of airborne avionics, which can realize the automation of various flight tasks. Its main function is to position the aircraft, make flight plans, optimize routes, guide aircraft flight, and reduce the working pressure of crew.

The civil aviation aircraft is prone to environmental collisions and crashes in the case of low visibility in the air, and emergency landing in mountainous terrain. The China International Airlines Flight 129 crash was due to low visibility in the air; its scheduled route error caused the passenger plane to crash into the mountain, and 129 people were unfortunately killed. Different from the traffic warning and obstacle avoidance system (TACS), airborne obstacle avoidance system (ACSA), and near-Earth warning system (GPWS) [2], FMS performs path planning when facing obstacles that may appear in advance, reducing the accident rate of civil aircraft in complex environments such as mountain flight and multi-aircraft flight. An important performance index of FMS is its planning and obstacle avoidance ability when flying in complex terrain environments.

Path planning is one of the core functions of FMS. The autonomous auxiliary path planning for civil aircraft in emergency flight will significantly help the aircraft to guide the emergency landing in bad weather, complex terrain, and sudden accidents, and decrease the incidence of major air crashes.

The path planning algorithm includes two parts: a map preprocessing algorithm and pathfinding algorithm [3]. Digital map processing needs to select the corresponding processing method according to the different elevation data and task requirements. For maps with different accuracy, it is necessary to process the best digital map suitable for the pathfinding algorithm. At present, there are many pathfinding algorithms for path planning. The A* algorithm is one of the most representative heuristic algorithms [4]. Its success rate and superiority of algorithm results are incomparable to other algorithms. However, there are still many directions for optimization of the algorithm. The computational complexity of the algorithm is related to the accuracy of the map. The higher the accuracy of the map, the greater the amount of calculation of the planning algorithm caused by the surge in data volume. The large amount of calculation also causes excessive space memory occupation, and the large amount of data will lead to problems such as program collapse in engineering. The classical A* algorithm cannot add the aircraft performance limit to the track, the planned track helicopter is difficult to follow, and the degree of engineering is low; at present, the application of the A* algorithm only stays in the calculation of an unselectable single track, and cannot be reasonably adjusted according to the task situation.

Many scholars have conducted a lot of research on how to improve the efficiency of the map preprocessing algorithm and pathfinding algorithm. The key step of path planning for robots, including UAVs, is to accurately process map information [5] and reach the target without collision [6]. Jaishankar et al. [7] proposed a distance change method, through which the digital elevation can be represented by grayscale image, and the path planning can be carried out on this basis. Meng H [8] first smoothed and optimized the data in the digital map from four directions, then processed the digital map into the smallest threat surface, and then sought the optimal path on the smallest threat surface. The algorithms lack the interpolation calculation of the appropriate accuracy of the map first, which may result in the situation that the resolution of the elevation data is not enough to support the pathfinding algorithm, or the resolution is too high to cause the data to be too large and the efficiency of the algorithm to be reduced.

The path planning algorithm not only requires that the planned flight path is feasible, but also requires its optimality in some specific criteria, such as calculation time and trajectory length [9]. The calculation time mainly includes map processing time and path planning time, which will be mainly used as the evaluation criteria for different algorithms in this paper. The A* algorithm is a famous algorithm in the field of path planning, which is suitable for the static environment exploration of complex obstacle topographic map [10]. However, the classical A* algorithm is not satisfactory in terms of computational time [11], which seriously hinders the deployment of the A* algorithm and its application in the actual aircraft navigation system. On this basis, many studies have proposed methods to improve the computational performance. Sudhakara et al. [12] proposed an improved A* algorithm to increase the number of turns to plan the path of the robot in a position environment with obstacles. Pal et al. [13] proposed an improved A* algorithm based on capacity consumption to reduce the energy consumption caused by stopping and turning.

In addition to considering the calculation length and trajectory time, the performance requirements of fixed-wing aircraft should be met when planning the path. ElHalawany et al. [14] proposed an improved A* algorithm considering its own size to avoid sharp turns in the path planning of mobile robots, which is necessary in practical applications. Based on the traditional algorithm, this paper adds the constraints of fixed-wing aircraft performance, so that the planned trajectory is easy to follow. In order to improve the performance of fixed-wing aircraft, Durán-Delfín et al. [15] established a mathematical model of fixed-wing convertible vertical take-off and landing aircraft to achieve two flight states along the trajectory. The controller has good performance and can provide sufficient maneuverability. This research will greatly improve the performance of fixed-wing aircraft in the future.

At present, the application of the A* algorithm only stays on the calculation of an unselected single trajectory, considering the minimization of multi-objectives such as path length and altitude [16]. However, for fixed-wing aircraft, the requirements for different costs are different under different flight conditions. Ducho et al. [11] modified the A* algorithm and optimized the algorithm based on the complexity of the environment, so that the algorithm can be applied to various scenarios. Aiming at this problem, this paper optimizes the weights of different costs in the cost function.

The main innovations and contributions of this paper are as follows: (1) Aiming at the problems of large memory occupation and high map accuracy requirements in the path planning of fixed-wing passenger aircraft, the digital map is combined with the requirement of the step size of the demand point to adjust the map accuracy, and the elevation digital map is processed into a grayscale map in combination with the aircraft climbing angle limit, so as to reduce the memory occupation of the map, thereby reducing the amount of calculation and improving the efficiency of path planning; (2) Aiming at the problem that the planned global trajectory is hard to follow by fixed-wing aircraft, this paper considers the flight constraints of fixed-wing aircraft flying, and processes the track into a followable trajectory that satisfies the turning angle of the aircraft through trajectory smoothing optimization; (3) Aiming at the problem of single trajectory in the traditional A* global path planning algorithm, this paper normalizes the cost parameters in the A* algorithm and opens the setting port. At the same time, different cost weight parameters are set for planning. The mode and the proportion of each cost parameter can be dynamically adjusted according to the task situation to find the optimal path that meets the task requirements and aircraft performance constraints; (4) Aiming at the problem of low efficiency of the classical A* algorithm in array structure sorting, the Heap sorting method is adopted to improve the sorting efficiency, and a new data structure called a "value table" is designed to optimize the search efficiency compared to the open table and close table of the classical A* algorithm, which reduces the complexity of the sorting algorithm and satisfies the real-time requirements of the planning algorithm.

2. Path Planning Problem and Modeling

2.1. Path Planning Problem for Fixed-Wing Aircraft

The path planning problem of fixed-wing aircraft refers to the specific path planning requirements $M(m,h,p)$, from the initial point $B_{g,t}$ to the target point $E_{g,t}$; the sets of optimal motion trajectory points $x_{g,t}$, $T_{B,E}$ can be calculated by the pathfinding algorithm $\phi(B_{g,t}\ , E_{g,t})$, which can be described as follows:

$$T_{B,E} = \{ x_{g,t} \in M(m,h,p) | x_{g,t} = \phi(B_{g,t}, E_{g,t}) \} \tag{1}$$

Among them, the mission target requirements M_m, fixed-wing aircraft performance requirements M_h, algorithm performance requirements M_p.

In the case of complex mountainous areas, the primary task of global trajectory planning is to ensure the safe flight of the aircraft, and the aircraft can successfully avoid all obstacles; secondly, the planned trajectory of the aircraft should ensure that the maximum pitch angle constraint, the maximum turning angle constraint, and aircraft's followability are satisfied. At the same time, the planning algorithm should also ensure a certain timeliness.

2.2. Path Planning Optimization Model

2.2.1. Optimized Objective Goal

The path planning problem can be regarded as a kind of constrained optimization problem. Figure 1 is a top view of the track, where we use hollow dots to represent the track points, with the symbol p_i. l_i is the cost of flight distance between each track point. The large black dots represent obstacles in the map. For the altitude cost in the track, its expression is similar to flight distance cost.

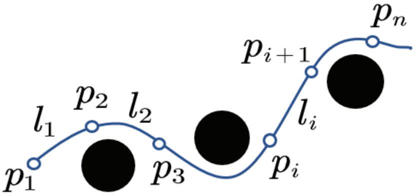


Figure 1. Schematic diagram of flight distance cost.

Objective function commonly used in path planning [17,18] can be described as (2).

$$\min_{\chi} J = \int_{t_0}^{t_f} (\omega_1 c_t^2 + \omega_2 h^2 + \omega_3 f_T) dt \tag{2}$$

where c_t, h, and f_T represent the cost of track length, flight altitude, and threat, respectively. $\omega_1, \omega_2, \omega_3$ are the cost factors, χ indicates the flight trajectory from t_0 to t_f. The optimization goal of path planning studied in this article is to design an optimal path under obstacle avoidance conditions, so as to minimize the cumulative distance cost and altitude cost of the entire flight process. Without involving threat costs, Equation (2) is modified to obtain Formula (3) [19]:

$$\min_{\chi} \sum_{i=1}^{n} (\omega_1 l_i^2 + \omega_2 h_i^2) \tag{3}$$

In the formula, the flight distance l_i from the track point p_i to the point p_{i+1} is expressed, which can be viewed in Figure 1. By reducing the flight distance, the fuel cost of the aircraft can be shortened; h_i indicates the altitude cost between track points p_i and p_{i+1}. ω_1, ω_2 represents the weight of each cost, which is generally valued according to task requirements.

2.2.2. Maximum Pitch Angle Constraint

In order to ensure the fastest flight to the target point, the trajectory must be able to meet the constraints of the maximum pitch angle in the longitudinal maneuver of the aircraft, which requires that there can be no trajectory beyond the maximum pitch angle limit between the two trajectory points in the planned trajectory, as shown in Figure 2.

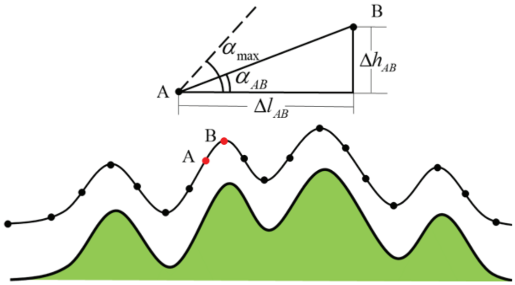


Figure 2. Maximum pitch angle constraint.

Among them, α_{max} presents the maximum pitch angle, and α_{AB} represents the pitch angle from track point A to track point B. It can be obtained by the relative height difference Δh_{AB}, relative horizontal distance Δl_{AB} from track point A to track point B.

$$\alpha_{AB} = \arctan\left(\frac{\Delta h_{AB}}{\Delta l_{AB}}\right),\ |\alpha_{AB}| \leq \alpha_{\max} \tag{4}$$

2.2.3. Maximum Bend Angle Constraint

Aiming at the problem of aircraft flight safety when flying in mountainous terrain, the maximum bend angle of lateral maneuver needs to be constrained. In order to change the course when the aircraft maintains a certain forward flight speed, according to the requirements of turning speed and turning radius, it is necessary to ensure that the turning angle of the track meets the constraint of the maximum turning angle when the distance between the track points is certain, as shown in Figure 3.

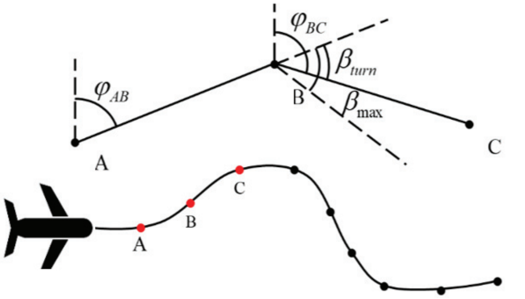


Figure 3. Maximum turning angle constraint.

In Figure 3, the heading at track point A is represented by φ_{AB}, the heading at track point B is represented by φ_{BC}, and the heading angle that needs to be changed from track point A to track point B is represented by β_{turn}; that is the turning angle at track point B.

$$\beta_{turn} = \varphi_{AB} - \varphi_{BC},\ |\beta_{turn}| \leq \beta_{\max} \tag{5}$$

2.2.4. Minimum Terrain Clearance Constraint

The distance between the aircraft and the ground should always be greater than the minimum flight height from the ground, so as not to affect the flight safety due to ground buildings, trees, and so on. Therefore, the height difference Δh_A between the planned track point height and the ground should meet the requirement as follows:

$$\Delta h_A \geq \Delta h_{\min} \tag{6}$$

3. Preprocessing of Map Information

Global path planning is the planning of the aircraft based on the starting point and terrain information before take-off, which requires the terrain data information between the take-off and the end point before take-off, and this information needs to be preprocessed in the three-dimensional space model. The preprocessing algorithm process is shown in Figure 4.

The digital elevation is extracted from the original digital map, and the accuracy information contained in the digital elevation is calculated by using the boundary information and the number of grids. The accuracy requirements of the required digital map are determined by the airborne information storage space and the time requirements for the trajectory planning calculation. The higher the accuracy of the digital map, the greater the storage space required, the longer the calculation time of the track planning, the denser the calculated track points, and the higher the accuracy. The original digital elevation is processed according to the selected map accuracy. If the original digital elevation resolution is too high or too low, it needs to be interpolated to change the resolution of the digital elevation map [18].

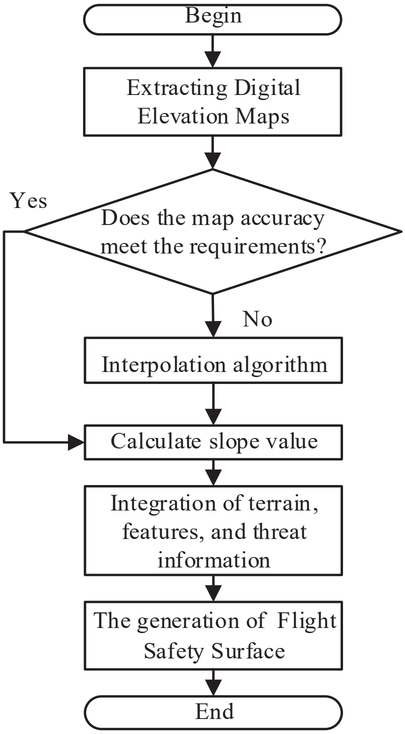

Begin
Extracting Digital Elevation Maps
Does the map accuracy meet the requirements?
Yes
No
Interpolation algorithm
Calculate slope value
Integration of terrain, features, and threat information
The generation of Flight Safety Surface
End

Figure 4. Flow chart of digital map preprocessing algorithm.

In this paper, by reading the high-resolution three-dimensional Digital Elevation Map, the accuracy of the map is adjusted by changing the resolution of the Digital Elevation Model (DEM) through a relatively smooth interpolation algorithm, and the slope value is calculated by the difference algorithm. Combined with the process of slope smoothing and graying, the flight safe surface satisfying the performance constraints such as the climbing rate and the shortest track length of the fixed-wing aircraft could be generated.

3.1. Generation of DEM

The commonly used digital terrain model (DEM) is a mathematical model that describes the parameters such as ground fluctuation and terrain height, and projects the height data of different positions to the data set of the corresponding position on the map in the form of regular gridding or other forms. The model formula is expressed as follows.

$$V_i = (g_i, t_i, h_i), i = 1, 2, \cdots, n \tag{7}$$

In the above formula, g_i represents the longitude corresponding to the point, t_i represents the latitude corresponding to the point, and h_i represents the height corresponding to the point. In order to construct a three-dimensional space with mountainous terrain, a mountain section was added to the DEM. Merge the original elevation model with mountain data to construct the final spatial 3D model, as shown in the following equation.

$$\begin{aligned} &V_i = \{g_i, t_i, \max[h_i, h_{mt}(g_i, t_i)]\}, i = 1, 2, \cdots, n \\ &h_{mt}(g_i, t_i) = z_0 + \sum_{j=1}^{n} h_j \exp\left[\begin{array}{l} -\frac{1}{a^2}\left(\left\lfloor (t_i - t_s)C_t + \frac{1}{2} \right\rfloor - \left\lfloor (t_0 - t_s)C_t + \frac{1}{2} \right\rfloor \right)^2 \\ -\frac{1}{b^2}\left(\left\lfloor (g_i - g_s)C_g + \frac{1}{2} \right\rfloor - \left\lfloor (g_0 - g_s)C_g + \frac{1}{2} \right\rfloor \right)^2 \end{array} \right] \end{aligned} \tag{8}$$

Among them, the longitude, latitude, and height values g_i, t_i, h_i, respectively, of every point in the topographic map are represented; the height of the highest point of the mountain is represented by z_0, the slope setting value of the mountain in the x axial direction is represented by a, and the slope setting value of the mountain in the y axial direction is represented by b. g_0, t_0 is the longitude and latitude of the highest point of the mountain, respectively; g_s, t_s indicates the minimum longitude of the map and the minimum latitude of the map, respectively; C_g, C_t represents the amount of data in the longitude direction and the amount of data in the latitude direction, respectively.

3.2. Adjust Map Resolution

Due to the resolution difference caused by the data source of the elevation digital model, the resolution of the elevation data may be insufficient to support the pathfinding algorithm, or the high resolution may lead to excessive data volume and low algorithm efficiency. Therefore, it is necessary to perform linear interpolation on the elevation digital model containing the mountain model to improve or reduce the map accuracy and meet the requirements of different modal pathfinding task algorithms [20–22]. The commonly used DEM linear interpolation algorithms are the bilinear interpolation algorithm, bicubic Hermite interpolation algorithm, and two-dimensional cubic convolution interpolation algorithm [23,24].

The interpolation is performed by the above three algorithms, and the graphical comparison results by using the original elevation data are shown in Figures A1 and A2 and Table A1 in Appendix B. It can be seen that the mean and variance of the differences between the two-dimensional cubic convolution interpolation and the original elevation are the smallest, and the covariance and correlation coefficient between the two-dimensional cubic convolution interpolation and the original elevation are the largest, after the resolution is reduced, indicating that the two-dimensional cubic convolution interpolation [25] has the highest correlation with the original elevation and the best restoration effect. Therefore, the 2D cubic convolution interpolation algorithm is selected as the interpolation algorithm to adjust the DEM resolution [26].

3.3. Generation of Flight Safety Surface

In order to make the planned track match the performance of the fixed-wing aircraft and avoid collision between the aircraft and the mountain obstacles during the landing process, it is necessary to explore the DEM slope calculation method in combination with the pitch angle limit of the fixed-wing aircraft. The slope of DEM is a description of the steepness of the terrain in three-dimensional space. The mathematical model of the slope description is shown as follows:

$$S = \arctan\sqrt{\varphi_g^2(h) + \varphi_t^2(h)} \tag{9}$$

Among them, $\varphi_g(h)$ and $\varphi_t(h)$ are the difference algorithms in the direction of g and t. The commonly used numerical analysis methods for slope calculation on DEM mainly include simple difference, second-order difference, third-order inverse distance square weight difference, third-order inverse distance weight difference, third-order unweighted difference, and frame difference. The corresponding $\varphi_g(h)$ and $\varphi_t(h)$ in the different algorithms above are shown in Table A2 in the Appendix B. In order to obtain a better slope estimation effect, we use the third-order inverse distance square weight difference algorithm to describe $\varphi_g(h)$ and $\varphi_t(h)$.

Since the slope calculation result is the slope value represented by the radian, in order to facilitate the cost calculation in the pathfinding algorithm and shorten the storage of the digital map, this paper uses the angle to represent the slope value combined with the pitch angle limit of the aircraft, and converts the angle value into an 8-bit unsigned integer in the range of $\left[0,\ 2^8 - 1\right]$, which is shown as follows.

$$G_{g,t} = \left(2^8 - 1\right)\left(\frac{\frac{\pi}{180}S_{g,t} - \alpha_{\min}}{\alpha_{\max} - \alpha_{\min}}\right) \tag{10}$$

Among them, the maximum and minimum values of the pitch angle of the civil aircraft specified for the task requirements are represented by $\alpha_{\max}$, α_{min}, respectively, and the general minimum value defaults to 0. $G_{g,t}$ can be used as a gray value to store the digital map as gray map data related to the performance of civil aircraft, which supports the cost calculation of the improved A* path planning algorithm.

4. Improved A* Path Planning Algorithm

On the premise of flying close to the ground, if the fixed-wing aircraft can fly along the track with gentle terrain, it can maintain a high speed and the task execution time will be shorter. In order to obtain a smoother and more efficient flight trajectory, an improved A* algorithm based on terrain slope is designed.

The A* algorithm in the path planning algorithm can quickly find the optimal solution and obtain the shortest path. It is undoubtedly the best algorithm for global path planning in mountainous terrain, but it still has some shortcomings. Aiming at the defects of the classical A* algorithm and the target requirements of real-time global path planning, the optimization steps are shown in Figure 5.

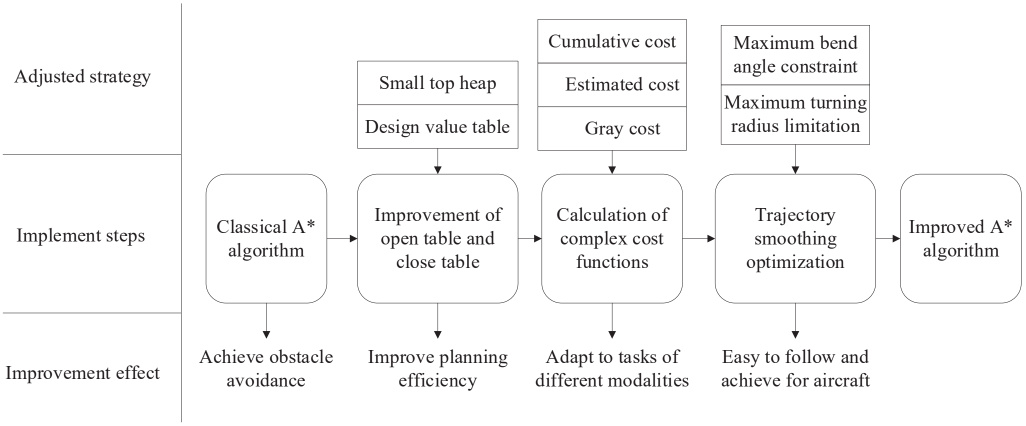

Adjusted strategy
Implement steps
Improvement effect
Small top heap
Design value table
Cumulative cost
Estimated cost
Gray cost
Maximum bend angle constraint
Maximum turning radius limitation
Classical A* algorithm
Improvement of open table and close table
Calculation of complex cost functions
Trajectory smoothing optimization
Improved A* algorithm
Achieve obstacle avoidance
Improve planning efficiency
Adapt to tasks of different modalities
Easy to follow and achieve for aircraft

Figure 5. Optimization steps of global path planning algorithm.

In Figure 5, based on the classical A* algorithm, the data storage and extraction structure in open and close tables is optimized to improve the efficiency of the algorithm. The second part is about the optimization and adjustment of the cost function. The terrain slope parameters are fused into the cost function of the A* algorithm, and different cost functions are calculated according to different requirements to obtain the planned tracks under different task modes. Finally, considering the limitation of aircraft performance, the track is post-processed to generate a three-dimensional safe track after smoothing the height and turning angle, so that the track is easy to follow.

4.1. Cost Function Optimization

Compared with the classical A* algorithm, the cost function F_n is adjusted by Equation (11).

$$F_n(G_n, H_n, I_n) = \omega_G G_n + \omega_H H_n + \omega_I I_n \tag{11}$$

Among them, G_n is the sum of the cost from the starting point $B_{g,t}$ to the current point x_n, the cost that needs to be spent from the current point to the end point is expressed as H_n, the cost of the newly added gray value is expressed as I_n, ω_H is the weight of the estimated distance cost, ω_I is the weight of the slope cost, and ω_G is the weight of the distance cost that has been spent. The calculation method of G_n can be adjusted by the grid distance in different directions, and the distance calculation method of H_n can be adjusted according to the actual model of DEM. The calculation process of the optimized cost function is shown in Algorithm A1.

4.1.1. Cumulative Cost

The grid in the DEM is not a standard rectangle, and the grid length deformation after the Gaussian model projection is worse. Therefore, different weights need to be added to the distance in the latitude and longitude directions. However, due to the different weights of different longitudes and latitudes, there will be a large amount of calculation. Therefore, in order to take the calculation accuracy and calculation efficiency into account, the deformation within the same longitude and latitude is regarded as the same, so the distance calculation is adjusted to the following formula:

$$\begin{cases} \delta_g = \frac{L_t}{N}\cos([t]) \\ \delta_t = \frac{L_t}{N} \end{cases} \tag{12}$$

Among them, L_t is the actual distance of a latitude range, N refers to the number of grids per unit latitude or unit longitude range, $[t]$ is the latitude value rounded, and δ_g and δ_t are the actual distance of a single grid in the longitude and latitude directions.

According to the distance calculation formula and the extended node method of the eight neighborhoods in the A* algorithm, the calculation formula of the cumulative cost G_n can be described as following formula:

$$G_n = G_{n-1} + \frac{L_t}{N}\begin{cases} \cos([t]) \quad n_j = 4,8 \\ 1 \qquad n_j = 2,6 \\ \sqrt{\cos^2([t]) + 1} \quad n_j = 1,3,5,7 \end{cases} \tag{13}$$

Among them, n_j is the eight neighborhoods index of the parent node relative to the current node, and G_{n-1} is the cumulative cost of the previous node. The different positions of the previous node in the eight neighborhoods will change the cost from the previous node to the current node. Since L_t, $[t]$, and N are constant values, the cumulative cost will also be a constant value in the same latitude map with the same resolution. The cumulative cost change generated by a single expansion will change due to the difference in the position j of the extended node relative to the current node.

4.1.2. Estimated Cost

The estimated cost H_n is the cost of estimating the current point to the target point, which can be calculated by the Manhattan distance algorithm with modified latitude and longitude difference.

$$\begin{cases} H_n = \Delta g_{n,E} + \Delta t_{n,E} \\ \Delta g_{n,E} = \frac{\cos([t])N_t}{N_g}|g_n - g_E| \\ \Delta t_{n,E} = |t_n - t_E| \end{cases} \tag{14}$$

Among them, the distance length in the direction g and the error value in the direction t will also be calculated into the distance difference $\Delta g_{n,E}$ in the g direction, so as to synchronize with G_n to eliminate the influence of Gaussian projection on the grid distance deformation.

4.1.3. Gray Cost

The gray cost I_n is the cost of the influence of the terrain slope on the flight of the aircraft. Because it is difficult for the aircraft to climb at a high speed when flying near the ground, it is necessary to limit the terrain slope to find a safe and fast trajectory that can satisfy the pitch angle limit of the aircraft at a certain speed. Taking the gray cost as a part of the cost function in the pathfinding algorithm, the calculation method is the same as the method of gray value as follows:

$$I_n = \left(2^8 - 1\right)\left(\frac{\frac{\pi}{180}S_{g,t} - \alpha_{\min}}{\alpha_{\max} - \alpha_{\min}}\right) \tag{15}$$

4.2. Optimization of Open Table and Close Table

During the execution of the A* algorithm, it is necessary to continuously add selected nodes to the close table, and continuously insert new nodes, delete root nodes, and modify existing nodes in the open table. The classical A* algorithm uses an array structure, and all points that may be traversed are placed in an array of the same open table, and sorted according to different costs.

The purpose of sorting the open table is to always be able to locate the minimum cost point, and to facilitate the insertion of new nodes, modify existing nodes, and delete the minimum point for operation. Therefore, the data structure of the close table and the open table greatly affects the traversal and search efficiency of the nodes.

This paper explores the efficient array sorting method and proposes the following improvement schemes:

(1) An improved data structure "value table" is designed, which combines the open table and the close table. It avoids the heuristic search operation on the array matrix before sorting the open table in the classical A* algorithm, which must have to judge whether the points in the eight fields have appeared in the open table or the close table.

(2) Using "Small Top Heap" to efficiently sort the nodes in the value table when performing operations such as inserting new nodes, deleting root nodes, and modifying nodes, to ensure that the root node in each extracted sequence is the minimum value, and the cost function gradually increases from the root node to the child node. The time complexity is $O(nlogn)$ and the space complexity is $O(1)$, which is lower than other sorting algorithms.

The following will describe the details of the value table designed to improve the search efficiency and the operation details of inserting new nodes, deleting root nodes, and modifying nodes in the value table by using the Heap sort order to improve the sorting efficiency.

4.2.1. Value Table

The information of the points stored in the traditional open table is two-dimensional coordinates. The new value table is stacked by rows and then the minimum points are stacked by columns. This sorting method can be stored for different rows, and only the column coordinates of the corresponding rows need to be stored. In the final value table, only the column coordinate index, the proxy value, and the parent node index that the original open table should store are retained. At the same time, because the parent node of the current point must be a point in the eight-neighborhood, the index of the parent node can replace the original two-dimensional coordinate index by the serial number of the eight-neighborhood. The close table stores two-dimensional coordinates and parent nodes, which are repeated with the open table, so the open table and the close table are merged. Because the open table does not calculate the nodes in the close table into the array when sorting, the value in the close table is set to a null value.

The "value table" proposed in this article combines the open table and the close table, and its data structure is shown in Figure 6.

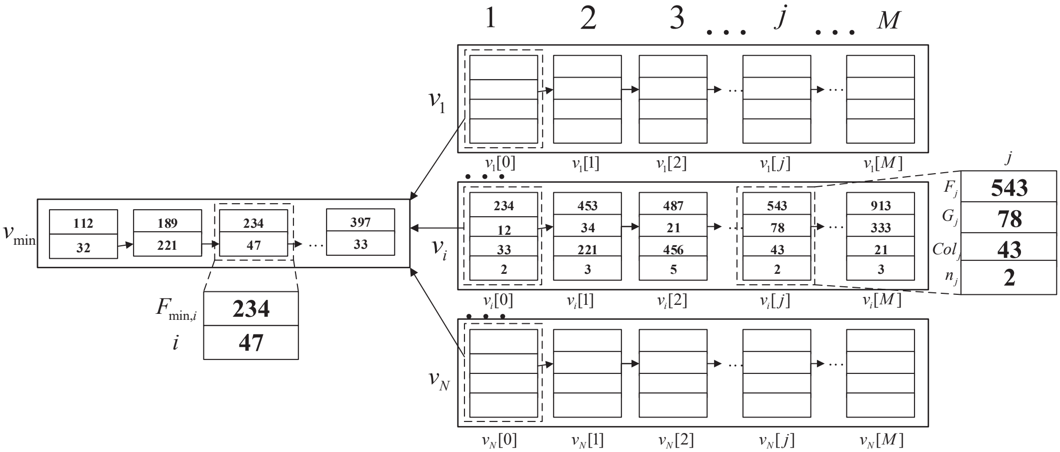


Figure 6. Value table structure.

The value table contains N arrays of v_i rows, and a row array of minimum cost $v_{\min}$. Row array v_i contains cost value F_j, cumulative cost G_j, the column index col_j of current point j, and the index n_j of the eight neighborhoods parent node. The minimum cost array $v_{\min}$ contains the cost value $F_{\min,i}$ minimum value in v_i, and its row index i, among which N is the row numbers of DEM, and M is the column number of DEM.

When it comes to the situation that the parameter (F_i, G_i, n_i) of node (g_i, t_i) needs to be updated, the column index j in row array v_i should be retrieved at first where index j can perfectly match t_i. The row array v_i will be automatically sorted in ascending order by the Heap sorting method. Specific details will be shown in Algorithm A2.

Table $v_{\min}$ stores the cost values and row index of every root node $v_i[0]$ for all $(v_1, v_2, \ldots v_N)$. When the root node value $v_i[0]$ of v_i is modified, the $v_{\min}[k]$ matched to the index i ($v_{\min[k]}.i = i$) and $v_{\min[k]}.F_{\min,i}$ will also be modified later. The row array $v_{\min}$ will be automatically sorted by $F_{\min,i}$ ascending order by the Heap sorting method. Specific details will be shown in Algorithm A3.

According to the continuously backtracking operation of index $v_{\min}[0].i$ in $(v_1[0].n, v_2[0].n, \ldots v_N[0].n)$ after getting the final $v_{\min}$, the row index g_i of the target point will be reached. The column t_i of the target point in $(v_1[0].col, v_2[0].col, \ldots v_N[0].col)$ will be reached by continuously backtracking operation at the same time. Specific details will be shown in Algorithm A4.

The parent node of the current node must belong to one of the eight neighbor nodes of the current node in the improved A* algorithm, based on which the information of the parent node can be gotten by the eight neighbors index relative to the current node, for which the backtracking operation is feasible.

4.2.2. Heap Sorting Method

(1) Inserting new nodes

The diagram of inserting new nodes is shown in Figure 7. Firstly, the new node is placed at the end of the array (node 9 in the graph), and then the size of the new node and the parent node are compared (node 4 and node 9 in the graph). If the new node is smaller than the parent node, the new node is exchanged with the parent node. Repeat the comparison and exchange until the parent node is less than the new node. The process of inserting a new node is the process by which the node continues to rise from the end of the binary tree.

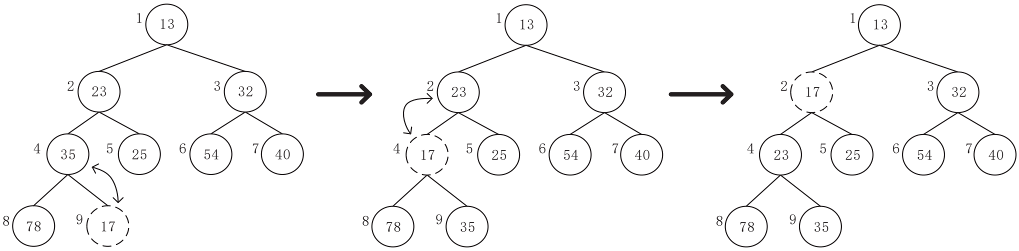


Figure 7. Inserting a new node.

(2) Deleting root nodes

The schematic diagram of deleting root nodes is shown in Figure 8. Firstly, the root node and the end node are exchanged (node 1 and node 9 in Figure 8). At this time, the original root node (node 9) can be deleted, and the new root node is the original end node. The new root node is compared with the child node and exchanged with the smaller node in the child node; that is, node 1 and node 2 are exchanged in the graph until the child node is larger than this node. The process of deleting new nodes is the process of continuous sinking down of nodes from the root nodes of the binary tree.

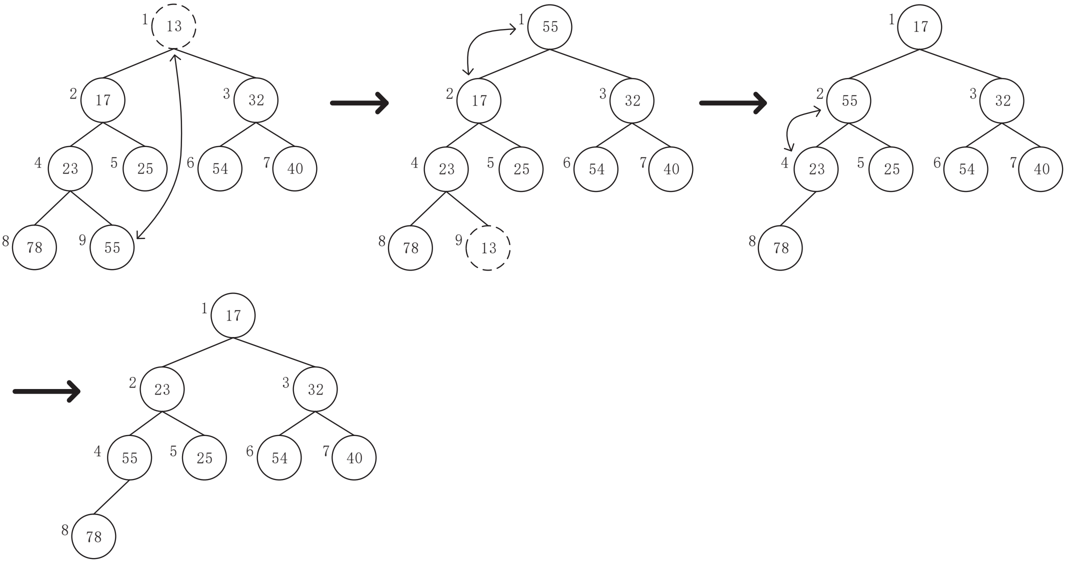


Figure 8. Deleting root nodes.

(3) Modifying nodes

The schematic diagram of modifying the node is shown in Figure 9. The value of the node is modified to the changed value. Owing to the reason that the modification of the node in the value table reduces the node value, the process of modifying the node is the process of the node floating up, which is the same as the process of inserting new nodes.

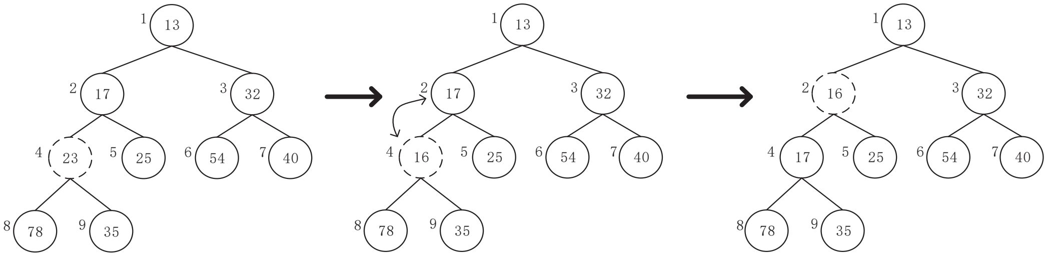


Figure 9. Modifying nodes.

4.3. Trajectory Smoothing Optimization

While the coordinates of the eight neighborhoods are used in the path planning process with constant angle of path, the turning radius of the aircraft in the actual flight is limited. For the convenience of calculation, the maximum turning angle between the three waypoints is limited on the basis of the step size of about 200 m, so that the route can meet the performance requirements of the aircraft. Aiming at the processing of the turning angle, this paper designs the vertical line method to adjust the trajectory, as shown in Figure 10.

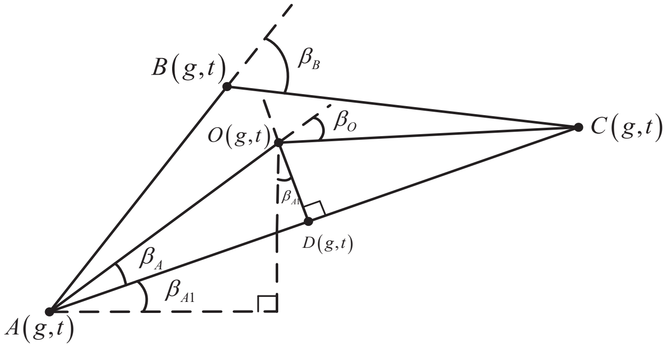


Figure 10. The schematic diagram of the vertical line method.

For the continuous track points A, B, and C, the turning angle at the track point B is β_B. When $\beta_B > \beta_{\max}$, it is necessary to find a point $O(g,t)$ so that β_O can meet the constraint of $\beta_O \leq \beta_{\max}$. In order to make the step size between the two track points as consistent as possible, this paper is designed to find point O on the vertical line of AC, and its corresponding relationship is:

$$\beta_O = 2\beta_A = 2\arctan\left(\frac{OD}{\frac{1}{2}AC}\right) = \beta_{\max} \tag{16}$$

Through the above equation, the quantitative relationship between OD and the coordinates of A and C can be obtained as follows:

$$OD = AD\tan(\beta_A) = \frac{\sqrt{(g_C - g_A)^2 + (t_C - t_A)^2}}{2}\tan\left(\frac{\beta_{\max}}{2}\right) \tag{17}$$

The coordinate O can be obtained according to the midpoint D between coordinate A and the midpoint of coordinate C, line segment OD, and the coordinate axis angle β_{A1} as follows:

$$\begin{cases} \beta_{A1} = \arctan\left(\left|\frac{g_C - g_A}{t_C - t_A}\right|\right) \\ g_D = \frac{g_C + g_A}{2} \\ t_D = \frac{t_C + t_A}{2} \end{cases} \Rightarrow \begin{cases} g_O = g_D + OD\frac{g_B - g_D}{|g_B - g_D|}\sin\beta_{A1} \\ t_O = t_D + OD\frac{t_B - t_D}{|t_B - t_D|}\cos\beta_{A1} \end{cases} \tag{18}$$

The trajectory after optimization is smoother; it can better meet the constraint of the turning angle of the aircraft. The pseudo-code of the improved A* algorithm is described in Algorithm A5.

5. Results

5.1. Experimental Environment

The hardware environment of the laboratory uses an 8-core,16-thread Intel I3-10th-2.4 GHZ main frequency processor, 4G running memory, and GeForce GTX 750 graphics card.

5.2. Experimental Parameters

The test parameters designed for specific global path planning tasks are shown in Table 1. The default starting point and the target point are located in the DEM range of N36E109~N37E110, the DEM resolution is 90 m, the number of grids in the longitude direction and latitude direction is 1201, the projection is Gaussian projection, and the coordinate system is the WGS-84 coordinate system. The terrain top view rendering map and three-dimensional map are shown in Figure 11.

Table 1. Algorithm simulation default parameter table.

Parameter	Value	Parameter	Value
Longitude of starting point	109.790833°	Longitude of target point	109.371667°
Latitude of starting point	36.718333°	Latitude of target point	36.091667°
Gray image minimum slope	0°	Minimum path segment length	200 m
Gray image maximum slope	30°	Interpolation algorithm	2D cubic convolution
Minimum terrain clearance altitude	800 m	Difference algorithm	Third-order inverse distance Squared weight difference
Maximum terrain clearance altitude	12,600 m	Distance type	Manhattan distance
Minimum pitch angle	0°	Cost weight of G	0.6
Maximum pitch angle	10°	Cost weight of H	0.2
Maximum turning angle	10°	Cost weight of I	0.2

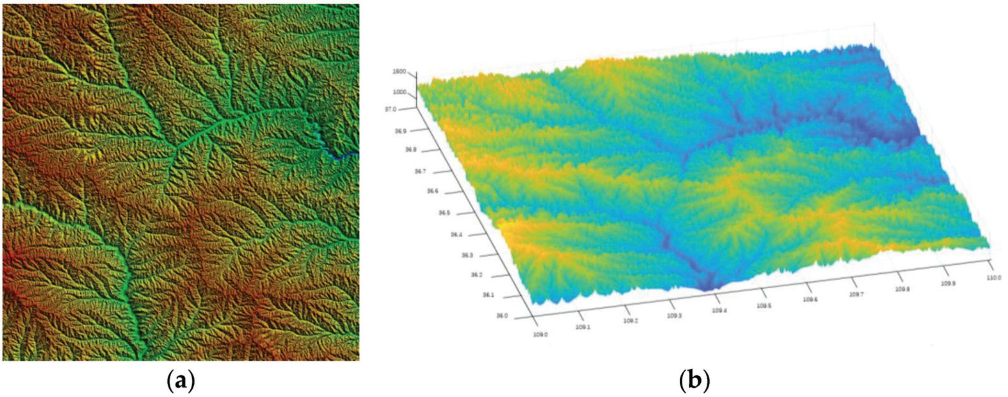


Figure 11. Comparison maps obtained before and after preprocessing of map information. (**a**) Top view rendering of original terrain; (**b**) DEM after map information preprocessing.

5.3. Experimental Results and Analysis

In this paper, the classical A* algorithm and the improved A* algorithm are run separately, the planning parameters of different algorithms are counted, and the classical algorithms and the improved algorithms are compared from the aspects of planning effect and efficiency as Table 2.

Table 2. Comparison simulation data of algorithm before and after improvement.

Simulation Data	Classical A* Algorithm	Improved A* Algorithm
Map processing time/s	0.0669	0.4220
Path planning time/s	412.6251	4.4212
Length of trajectory/m	93,620	93,225

Through the comparison data of the simulation results in Table 2, it can be seen that the improved A* algorithm has increased the map processing time compared with the traditional algorithm, which is due to the increase in map resolution adjustment and slope calculation in the early stage. Such time consumption is necessary, because the processing of the map facilitates the subsequent pathfinding algorithm, and the length of trajectory becomes smaller. It can be seen that in the results, the pathfinding time of the improved algorithm is 1% of the traditional algorithm, and the time is greatly shortened. Finally, the length of the track planned by the improved algorithm is also shorter than that of the traditional algorithm. Although the shortened length is not much for the whole track, it also saves the time to reach the target point to a certain extent.

Combined with the planning path in the schematic diagram of the algorithm simulation performance results between the classical algorithm and the improved algorithm in Figure 12, the point line is the classical A* algorithm, and the solid line is the improved A* algorithm. It can be seen that the path planned by the improved A* algorithm has less steering. In a valley with complex terrain, the path can also be adjusted according to the change in valley terrain to satisfy the constraint of obstacle avoidance.

When the aircraft makes an emergency landing in a complex mountain flight, the flight trajectory after trajectory smoothing optimization shows a smoother landing route with shorter track distance and landing time, as shown in Figure 13. Therefore, the improved A* algorithm improves a good solution for the emergency handling of fixed-wing aircraft in the event of bad weather conditions.

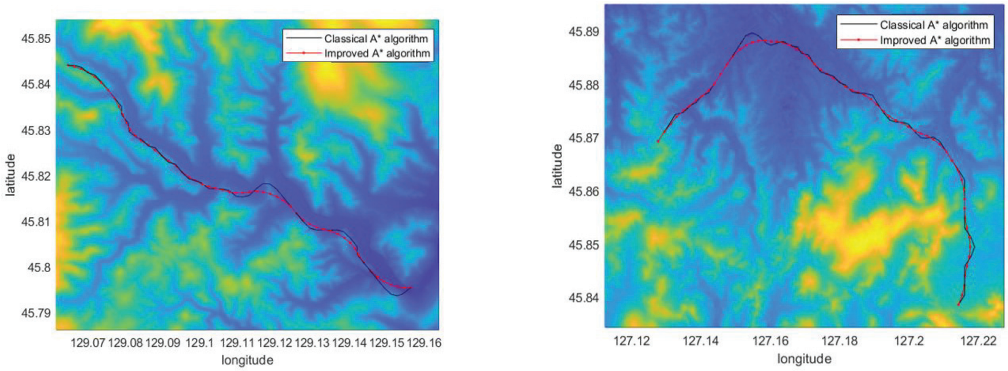


Figure 12. *Cont.*

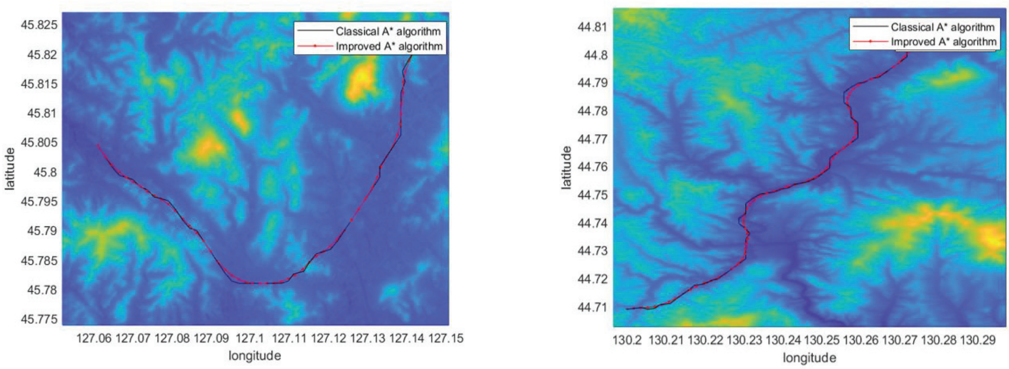


Figure 12. The comparison of the simulation results between the classical A* algorithm and the improved A* algorithm.

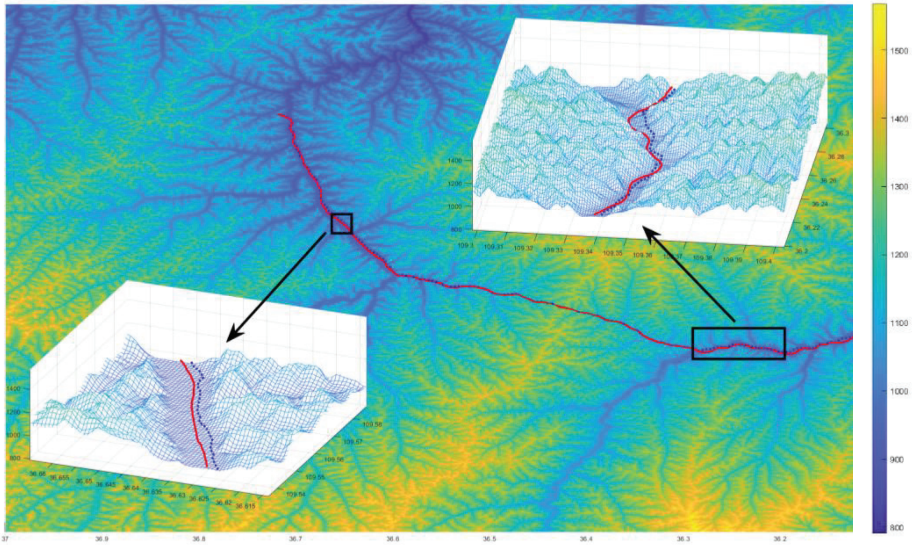


Figure 13. Comparison of aircraft landing trajectory after trajectory smoothing optimization. The blue point line in the figure is the curve before the trajectory optimization, and the red line is the curve after the trajectory optimization.

5.4. Comparative Analysis

In the previous study, the interpolation algorithm was introduced and the 2D cubic convolution interpolation selected as the best interpolation algorithm by comparing the effect of interpolation maps. In order to verify the superiority of 2D cubic convolution interpolation, this section will simulate different interpolation algorithms. The DEM obtained by different algorithms is used to find the path, and the parameters such as the pathfinding time and the path length of the path planned by different algorithms are compared. For the task requirements of different minimum track segment lengths, this paper designs four digital maps with different resolutions to meet the requirements of interpolation algorithms under different task situations.

The minimum track length of the contrast experiment design of the interpolation algorithm is 30 m, 45 m, 200 m, and 500 m for the four groups of algorithm simulation. The simulation results are shown in Table 3.

Table 3. Comparison simulation data of different minimum track length and interpolation algorithm.

Minimum Track Length	Simulation Data	Bilinear	Bicubic Hermite	2D Cubic Convolution
30 m	Map processing time/s	51.15	84.83	38.04
	Path planning time/s	1614.14	1618.77	1859.43
	Length of trajectory/m	97,509	98,486	98,486
45 m	Map processing time/s	29.54	53.12	17.51
	Path planning time/s	831.19	784.20	742.94
	Length of trajectory/m	97,029	97,095	97,095
200 m	Map processing time/s	1.47	2.52	0.61
	Path planning time/s	7.06	7.80	7.91
	Length of trajectory/m	93,879	93,241	93,241
500 m	Map processing time/s	0.20	0.31	0.08
	Path planning time/s	0.64	0.61	0.60
	Length of trajectory/m	83,156	84,039	84,039

Through the setting of different minimum track segment lengths, it can be seen that when the gray image obtained by the interpolation algorithm is used for pathfinding, the map processing time of the two-dimensional cubic convolution interpolation algorithm is the smallest, followed by the two-dimensional cubic convolution interpolation, and the map processing time required for the bicubic Hermite interpolation is the longest. The three interpolation algorithms have little difference in pathfinding time under different resolutions, and most of the pathfinding time of the bicubic Hermite interpolation is relatively small; the effect of bilinear interpolation will become relatively poor as the resolution decreases, while the bicubic Hermite interpolation and the two-dimensional cubic convolution interpolation are consistent with the track length data.

By longitudinally comparing the simulation parameters obtained by the same interpolation algorithm with different resolutions, it can be seen that the higher the resolution, the less time spent on the map processing and pathfinding algorithm of the response. When the pilot actually flies, the resolution can be adjusted as needed to improve the efficiency of the algorithm. The 2D cubic convolution method can also obtain a trajectory with higher accuracy.

In the previous section, by comparing the slope calculation effect and calculation complexity of different algorithms, the third-order inverse distance square weight difference method is selected as the slope calculation method of track planning. In order to further determine its superiority, this section will simulate and test the DEM obtained by different difference algorithms, and compare the pathfinding effect of different difference algorithms. The difference algorithm simulation data comparison is shown in Table 4.

Table 4. Comparison simulation data of different difference algorithms.

Difference Algorithm	Map Processing Time/s	Path Planning Time/s	Length of Trajectory/m
Simple difference	0.40	4.18	93,314
Second-order difference	0.39	4.92	93,319
Third-order inverse distance square weight difference	0.41	4.84	93,241
Third-order inverse distance weight difference	0.40	4.67	94,238
Third-order unweighted difference	0.39	4.86	94,183
Frame difference	0.40	4.54	94,352

It can be seen from the simulation data that the map processing times and pathfinding times of different differential algorithms are less different, so the selection of different difference algorithms has little effect on the final performance.

In the previous section, by comparing the computational efficiency and accuracy of different distance calculation methods, the Manhattan distance is selected as the distance calculation method in the path planning. Because the latitude and longitude distances in the grid are different, the latitude and longitude distance difference are combined with the Manhattan distance.

In this section, in order to compare the effects of different distance calculation methods for path planning, we randomly generate 1000 groups of starting points and target points in the same map. We define the minimum pathfinding time (MPFT) to measure the number of minimum pathfinding algorithms for an algorithm in 1000 path plans as shown in Equation (19).

$$MPFT_i = \sum_{k=1}^{1000} 1[i == \underset{1 \le j \le 4}{\operatorname{argmin}}\{t_j^{(k)}\}], i = 1, 2, 3, 4 \tag{19}$$

where $t_1^{(k)}, t_2^{(k)}, t_3^{(k)}, t_4^{(k)}$ represent the path planning time under the Euclidean distance, Manhattan distance, Diagonal distance, and Chebyshev distance, respectively, in k-th simulation. $1(x)$ is an indicative function when x is true; its value is 1, otherwise it is 0. Comparison simulation data of different distance calculation methods in $MPFT$ are shown in Table 5.

Table 5. Comparison simulation data of different distance calculation methods in MPFT.

Euclidean Distance	Manhattan Distance	Diagonal Distance	Chebyshev Distance
128	613	19	240

According to the statistical results, it can be seen that among the 1000 groups of random tracks, the Manhattan distance has the shortest pathfinding time of 613 times. Therefore, it is undoubtedly the best distance calculation method in the pathfinding algorithm.

6. Discussion

The path planning method based on the improved A* algorithm proposed in this paper has significant advantages compared with the classical A* algorithm. Take the original A* algorithm (including A* [4], LPA* [27], Weighted A* [28,29], etc.) and Hybrid A* algorithm [27] as an example; their characteristics are shown in Table 6.

Table 6. Comparison of different A* algorithms.

Algorithm	Processed Map Format	Sorting Algorithm of Node Data	Storage Structure of Node Data	Smooth Optimization Method of Trajectory
Original A*	Digital raster Graphics (DRG)	Insertion sorting or other sorting methods	Open table and close table	No optimization
Hybrid A*				Consider kinematic corner constraints, using Dubbins curves, or Reeds Shepp curves for trajectory smoothing
Improved A*	DEM after preprocessing of map	Small Top Heap sorting	Value table	Consider turning angle constraint real-time midline optimization

Through the preprocessing operation of resolution adjustment and interpolation of DEM map information, a "value table" is used to store open table node data, and the Small Top Heap structure is used to delete, add, modify, and sort nodes, which greatly reduces the calculation time. The turning angle of the two-dimensional trajectory point calculated by the A* algorithm is smoothed to ensure the planned trajectory point meets the requirements of the aircraft turning angle.

The simulation examples show that the proposed improved A* algorithm can meet the requirements of short calculation time, good smoothness of calculation trajectory, and high security. However, the current trajectory planning algorithm does not consider the underlying dynamic model of the aircraft and the requirements of control performance. In future research, it is planned to incorporate the influence of aircraft speed into the algorithm; that is, the maximum and minimum flight speed effects of the aircraft are considered in the process of planning the trajectory, so that the planned trajectory can meet the flight speed constraints of the aircraft. In addition, the weight distribution of the generation value in the improved A* algorithm and the efficient real-time re-planning obstacle avoidance in the dynamic environment are also problems worthy of further study.

7. Conclusions

This article takes the FMS system of fixed-wing civil aircraft as the background for efficient emergency landing obstacle avoidance and optimal trajectory planning in complex mountainous terrain. It mainly focuses on model generation and preprocessing in global path planning, as well as global path planning algorithms.

This article uses methods such as adjusting map resolution, calculating terrain slope, and generating safe flight surface to preprocess map information, and generates a flyable DEM grayscale map. This solves the problem of excessive data volume in the three-dimensional spatial model of the trajectory planning algorithm, and based on this, generates the grayscale cost of the path planning algorithm. A global path planning algorithm based on the improved A* algorithm combined with grayscale cost is proposed, the effectiveness of the final experimental results of the algorithm is analyzed, and the key interpolation and heuristic operator calculation methods that affect the algorithm are compared and analyzed.

The overall work of the paper is as follows:

A safe flight surface generation algorithm combining pitch angle constraint is proposed to meet the requirements of pathfinding algorithms for digital maps. Select DEM as the data model for the three-dimensional spatial model, and perform two-dimensional cubic convolutional interpolation to address the resolution issue of the digital map, resulting in a digital map that can meet the storage requirements of the pathfinding algorithm and the safety of flight surface information. To address the issue of large amounts of elevation data that are difficult to calculate, a third-order inverse distance squared difference method is used to calculate the terrain slope and generate a grayscale image.

A global path planning algorithm based on the improved A* algorithm is proposed to solve the problems of low planning efficiency and difficulty in following the trajectory. To solve the problems of large data computation and long planning time in the classical A* algorithm, value table and Small Top Heap methods are used to improve and optimize the sorting algorithm and data structure; a trajectory smoothing optimization algorithm combined with turning angle constraints is proposed to address the problem of difficult track following. And comparative analysis of key methods is conducted on the optimized algorithm.

The experimental results show that, compared to the original classic A* algorithm, the improved A* algorithm can significantly reduce the pathfinding time of the flight path, and the planned flight path is smoother and easier to follow, which meets well the requirements of efficient obstacle avoidance and emergency landing in complex mountainous terrain.

Author Contributions: Conceptualization, J.L.; methodology, Z.S. and Z.Z.; software, Z.Z. and W.Z.; validation, W.Z., Y.X. and J.H.; formal analysis, Z.Y.; investigation, J.L. and X.W.; resources, X.W.; data curation, C.Y.; writing original draft preparation, Z.Z.; writing review and editing, Z.S.; supervision, J.L.; project administration, C.Y.; funding acquisition, C.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China, grant number 2021YFB1600603. The APC was funded by the National Key Research and Development Program of China.

Data Availability Statement: The data are not publicly available due to intellectual property rights of the code, privacy, and national geographic information security considerations. Some important algorithm effect comparison information can be downloaded at: https://postimg.cc/gallery/dLb25DK, accessed on 9 October 2023.

Acknowledgments: Thanks for Jing Li's guidance and support from her laboratory students.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A

Algorithm A1: Cost Function

```
Input : DEM matrix $G_{g,t}$, $(g_0, t_0)$, start point $(g_E, t_E)$, target point $(g_i, t_i)$,
weight of the cost $\omega_G$, $\omega_H$, $\omega_I$
Output : Cost value $F_i$
Calculation of gray cost $I_i = I_i(G_{g_i,t_i})$
Calculation of estimated cost $H_i = H_i((g_i, t_i), (g_E, t_E))$
Calculation of cumulative cost $G_i = G_i(G_{g_{n_i},t_{n_i}}, (g_i, t_i), (g_{n_i}, t_{n_i}))$
Calculation of cost value $F_i = \omega_G G_i + \omega_H H_i + \omega_I I_i$
```

Algorithm A2: Modify v

```
Input: node $(g_i, t_i)$; cost value and cumulative cost of current node $(F_i, G_i)$; column of current
node $Col_i$; parent node index $n_i$ in eight neighbors relative to current node; value table
$v : (v_1, v_2, \ldots v_N)$; $M$
Output: $v_{g_i}$ after being modified, value table $v : (v_1, v_2, \ldots v_N)$
for $j = 1 : M$ do
    if $v_{g_i}[j].Col == t_i$ do
        if $v_{g_i}[j].F > F_i$ do
            $v_{g_i}[j].F = F_i$
            $v_{g_i}[j].G = G_i$
            $v_{g_i}[j].n = n_i$
        endif
        continue
    endif
endfor
$v_{g_i}$ is sorted $v_{g_i} \leftarrow \mathrm{sort}(v_{g_i}, F)$ in ascending order through $F$ by Heap sorting method
```

Algorithm A3: Modify $v_{\min}$

```
Input: Value table $(v_1, v_2, \ldots v_N)$; before modified.
Output: $v_{\min}$ after modified
$num = 0$
for $i = 1 : N$ do
if $v_i[0] \neq 0$ do
$k =$ find $(v_{min}.i = i)$
if $k$ does not exist
$num = num + 1$
$v_{\min[num]}.F_{\min,i} = v_i[0].F$
Using Small Top Heap sort to float $v_{\min[k]}.F_{\min,i}$ es.
elseif $k$ exist do
if $v_i[0].F < v_{\min[k]}.F_{\min,i}$
$v_{\min[k]}.F_{\min,i} = v_i[0].F$
Small Top Heap sort to float $v_{\min[k]}.F_{\min,i}$ es.
endif
endif
endif
endfor
```

Algorithm A4: Backtracking

Input: Value table $(v_1, v_2, \ldots v_N, v_{\min})$; start point (g_0, t_0); target point (g_E, t_E)
Output: Trajectory point sets *Path* from start point to target point

```
$g_p = v_{\min}[0].i, t_p = v_{g_p}[0].col, Path = [g_p, t_p]$
while $(g_p, t_p) \neq (g_E, t_E)$ do
    $n_p \leftarrow v_{g_p}[0].n$
    $g_p = n(n_p)$
    $t_p \leftarrow v_{g_p}[0].col$
    $Path \leftarrow Path \cup [g_p, t_p]$
Endwhile
```

Algorithm A5: Improved A* algorithm

Input: DEM matrix $G_{g,t}$; number of rows N; number of columns; value table $(v_1, v_2, \ldots v_N, v_{\min})$; start point (g_0, t_0); target point (g_E, t_E); maximum turning angle $\beta_{\max}$; cost weight $\omega_G, \omega_H, \omega_I$; final planning trajectory point sets $Path = []$
Output: Trajectory point sets *Path*

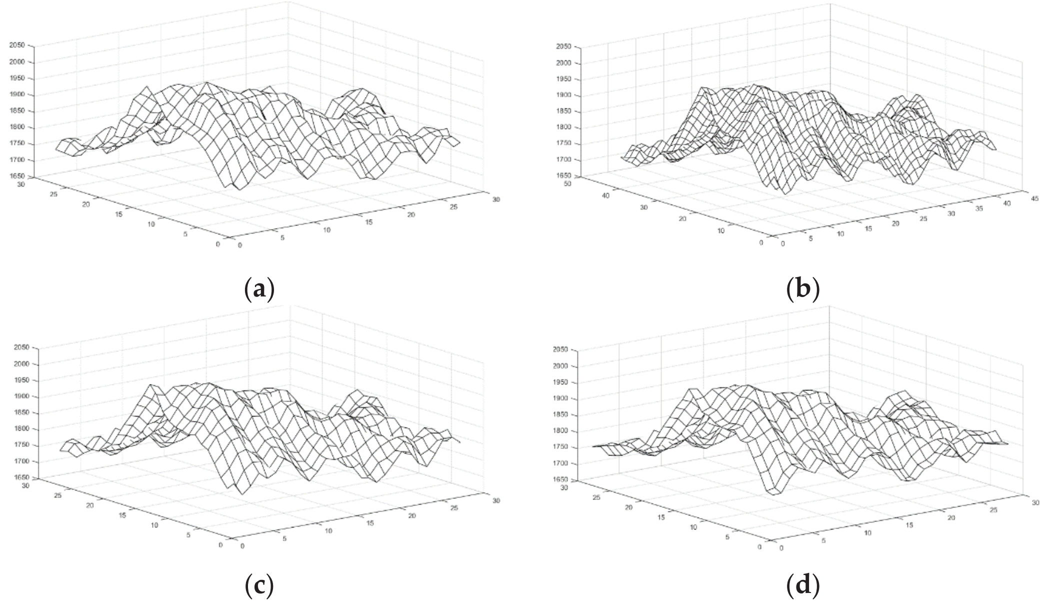
```
Initialize the value table $(v_1, v_2, \ldots v_N, v_{\min})$, current point $p \leftarrow (g_0, t_0)$, puts $v_i$ all the
nodes of $j$ into $F_j \leftarrow \mathrm{Inf}, G_j \leftarrow 0, col_j \leftarrow j, n_j \leftarrow \mathrm{Null}$, Small Top Heap $p\_arr \leftarrow [p]$ (That
means $p\_arr[0] = \arg\min_{(g_p,t_p)} p\_arr[(g_p, t_p)]$).
while $v_{g_E}[0].col \neq t_E$ do
  for $(g_p, t_p) \in p\_arr[0]$ do
    Getting eight neighbor points $\{(g_1, t_1), (g_2, t_2), \ldots (g_8, t_8)\}$ of $(g_p, t_p)$
    Insert $\{(g_1, t_1), (g_2, t_2), \ldots (g_8, t_8)\}$ into $p\_arr$
Delete $(g_p, t_p)$ from $p\_arr$
    for $i = 1:8$ do
      if $v_{g_i}[j].F == \mathrm{Inf}, v_{g_i}[j].col = t_i$ do
 $F_i = \mathrm{CostFunction}(G_{g,t}, (g_i, t_i), (g_0, t_0), (g_E, t_E), \omega_G, \omega_H, \omega_I)$
          $v_{g_i} \leftarrow \mathrm{ModifyV}((g_i, t_i), v_{g_i}, (F_i, G_i, Col_i, n((g_p, t_p))))$
      endif
    endfor
  endfor
  $v_{\min} \leftarrow \mathrm{ModifyVmin}((v_1, v_2, \ldots v_N, v_{\min}))$
endwhile
$Path = \mathrm{Backtracking}((g_0, t_0), (g_E, t_E), (v_1, v_2, \ldots v_N, v_{\min}))$
Trajectory smoothing optimization for $Path$ using maximum turning angle $\beta_{\max}$
```

Appendix B

For the three DEM interpolation algorithms: Bilinear interpolation, Bicubic Hermite interpolation, and 2D cubic convolution interpolation, this paper selects the original DEM data with a resolution of 90 M between 40.4583° N~40.6667° N and 113.3333° E~113.5417° E, and then performs different interpolation algorithms. The processing results are statistically calculated to better obtain the most suitable interpolation algorithm. The processing results are shown in Figures A1 and A2.

In order to compare more quantitatively compared to graphical comparison results, the results of the three algorithms are statistically compared with the original elevation, and the effects of different interpolation algorithms are compared as shown in Table A1.

The corresponding $\varphi_g(h)$ and $\varphi_t(h)$ in different difference algorithms such as simple difference, second-order difference, third-order inverse distance square weight difference, third-order inverse distance weight difference, third-order unweighted difference, and frame difference are shown in Table A2.



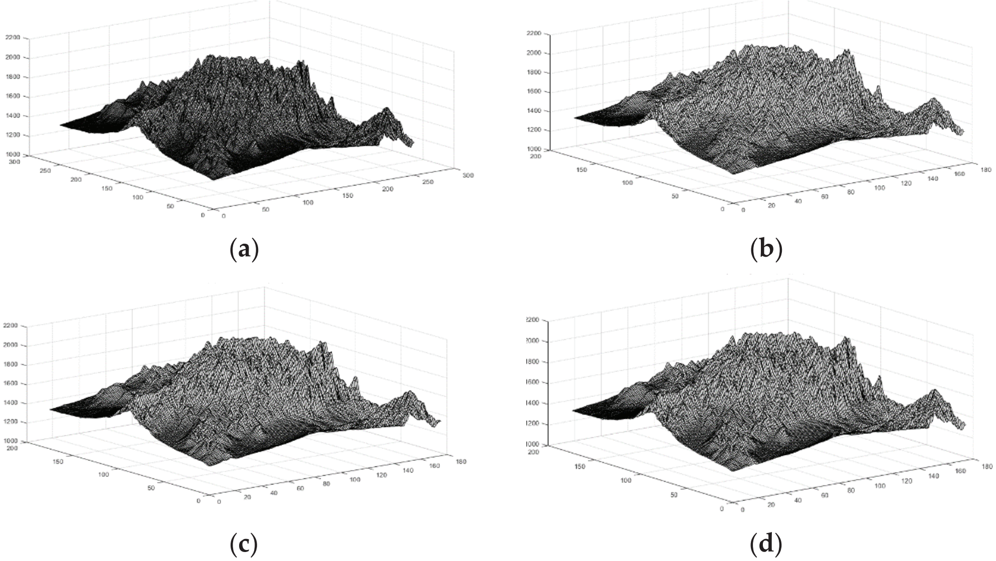
(**a**)

(**b**)

(**c**)

(**d**)

Figure A1. The refinement effect comparison of interpolation algorithm. (**a**) The original digital elevation map; (**b**) Bilinear interpolation; (**c**) Bicubic Hermite interpolation; (**d**) 2D cubic convolution interpolation. (The *z*-axis represents the height in meters).



(**a**)

(**b**)

(**c**)

(**d**)

Figure A2. Interpolation effect diagram with more refined resolution compared to Figure A1. The refinement effect comparison of interpolation algorithm. (**a**) The original digital elevation map; (**b**) Bilinear interpolation; (**c**) Bicubic Hermite interpolation; (**d**) 2D cubic convolution interpolation. (The *z*-axis represents the height in meters).

Table A1. Data comparison analysis table of interpolation algorithm.

Interpolation Algorithm	Mean (Difference)	Variance (Difference)	Covariance	Correlation Coefficient
Bilinear	−3.3265	496.3242	6729.2543	0.9646
Bicubic Hermite	−3.0499	519.5576	6700.7678	0.9628
2D cubic convolution	−3.0025	434.3359	6746.2220	0.9693

Table A2. Comparison table of $\varphi_g(h)$ and $\varphi_t(h)$ in simple difference, second-order difference, third-order inverse distance square weight difference, third-order inverse distance weight difference, third-order unweighted difference, and frame difference algorithms.

Algorithms	$\boldsymbol{\varphi_g(h)}$	$\boldsymbol{\varphi_t(h)}$
Simple difference	$\frac{h_0-h_4}{\delta_g}$	$\frac{h_0-h_2}{\delta_t}$
Second-order difference	$\frac{h_4-h_8}{2\delta_g}$	$\frac{h_2-h_6}{2\delta_t}$
Third-order inverse distance square weight difference	$\frac{h_1-h_3+2(h_8-h_4)+h_7-h_5}{8\delta_g}$	$\frac{h_5-h_3+2(h_6-h_2)+h_7-h_1}{8\delta_t}$
Third-order inverse distance weight difference	$\frac{h_1-h_3+\sqrt{2}(h_8-h_4)+h_7-h_5}{\left(4+2\sqrt{2}\right)\delta_g}$	$\frac{h_5-h_3+\sqrt{2}(h_6-h_2)+h_7-h_1}{\left(4+2\sqrt{2}\right)\delta_t}$
Third-order unweighted difference	$\frac{h_1-h_3+h_8-h_4+h_7-h_5}{6\delta_g}$	$\frac{h_5-h_3+h_6-h_2+h_7-h_1}{6\delta_t}$
Frame difference	$\frac{h_1-h_3+h_7-h_5}{4\delta_g}$	$\frac{h_5-h_3+h_7-h_1}{4\delta_t}$

The variable $h_i, i = 1, 2 \ldots 8$ is the eight neighborhoods elevation of the current point elevation value h_0, and δ_g, δ_t represents the unit minimum distance between the adjacent grid center points in different directions. The order, position, and symbol of the eight neighborhoods are shown in Figure A3.

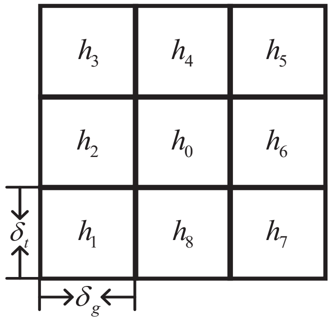


Figure A3. Elevation eight neighborhood diagram.

In order to compare different difference algorithms, this paper selects the original DEM with a resolution of 30 m between 27.776994° N~28.167096° N and 91.696574° E~92.149718° E, as shown in Figure A4. The original elevation model is processed by simple difference, second-order difference, third-order inverse distance square weight difference, third-order inverse distance weight difference, third-order unweighted difference, and frame difference. The processing effect of the difference algorithm is shown in Figure A5. It can be seen that the grayscale images obtained by the third-order inverse distance weight difference, the third-order unweighted difference, and the border difference are better, and the ridges and valleys can be well distinguished by the grayscale and form a continuous path.

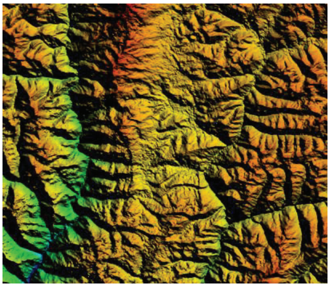


Figure A4. Original digital elevation topographic map.

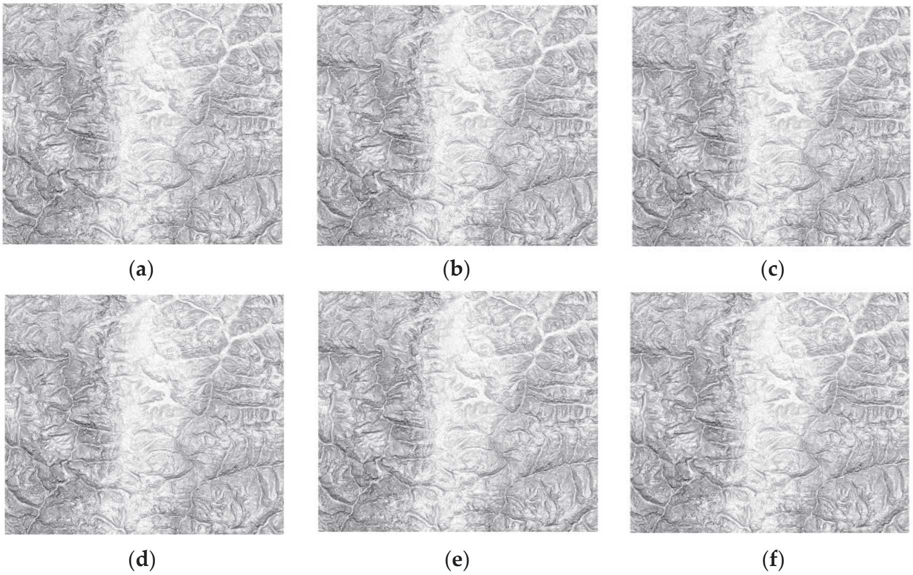


Figure A5. Different differential difference algorithms to deal with the effect diagram. (**a**) Simple difference; (**b**) Second-order difference; (**c**) Third-order inverse distance square weight difference; (**d**) Third-order inverse distance weight difference; (**e**) Third-order unweighted difference; (**f**) Frame difference.

References

1. Lv, K.N. *Trajectory Optimization and Control Technology of Civil Aviation Vehicle*; Nanjing University of Aeronautics and Astronautics: Nanjing, China, 2017; Volume 1, pp. 15–96.
2. Zhou, Q.H. Development of airborne traffic collision avoidance and terrain collision avoidance systems. *Avion. Technol.* **1997**, *3*, 45–49.
3. Hu, Z.; Shen, C.L. Flight path planning based on digital map preprocessing. *J. Nanjing Univ. Aeronaut. Astronaut.* **2002**, *4*, 382–385.
4. Hart, P.E.; Nilsson, N.J.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
5. Luo, R.C.; Lai, C.C. Enriched indoor map construction based on multisensor fusion approach for intelligent service robot. *IEEE Trans. Ind. Electron.* **2012**, *59*, 3135–3145. [CrossRef]
6. Raja, P.; Pugazhenthi, S. Optimal path planning of mobile robots: A review. *Int. J. Phys. Sci.* **2012**, *7*, 1314–1320. [CrossRef]
7. Jaishankar, S.; Pralhad, R.N. 3D off-line path planning for aerial vehicle using distance transform technique. *Procedia Comput. Sci.* **2011**, *4*, 1306–1315. [CrossRef]
8. Meng, H.; Xin, G. UAV route planning based on the genetic simulated annealing algorithm. In Proceedings of the 2010 IEEE International Conference on Mechatronics and Automation, Xi'an, China, 4–7 August 2010; pp. 788–793.
9. LaValle, S.M. *Planning Algorithms*; Cambridge University Press: New York, NY, USA, 2006.
10. Cui, S.G.; Wang, H.; Yang, L. A Simulation Study of A-star Algorithm for Robot Path Planning. In Proceedings of the 16th International Conference on Mechatronics Technology, Tianjin, China, 16 October 2012; pp. 506–510.

11. Duchoň, F.; Babinec, A.; Kajan, M.; Beňo, P.; Florek, M.; Fico, T.; Jurišica, L. Path planning with modified a star algorithm for a mobile robot. *Procedia Eng.* **2014**, *96*, 59–69. [CrossRef]
12. Sudhakara, P.; Ganapathy, V. Trajectory planning of a mobile robot using enhanced A-star algorithm. *Indian J. Sci. Technol.* **2016**, *9*, 1–10. [CrossRef]
13. Pal, A.; Tiwari, R.; Shukla, A. Modified A* algorithm for mobile robot path planning. *Soft Comput. Tech. Vis. Sci.* **2012**, *395*, 183–193.
14. ElHalawany, B.M.; Abdel-Kader, H.M.; TagEldeen, A.; Elsayed, A.E.; Nossair, Z.B. Modified A* algorithm for safer mobile robot navigation. In Proceedings of the 2013 5th International Conference on Modelling, Identification and Control (ICMIC), Cairo, Egypt, 31 August 2013–2 September 2013; pp. 74–78.
15. Durán-Delfín, J.E.; García-Beltrán, C.D.; Guerrero-Sánchez, M.E.; Valencia-Palomo, G.; Hernández-González, O. Modeling and Passivity-Based Control for a convertible fixed-wing VTOL. *Appl. Math. Comput.* **2024**, *461*, 128298. [CrossRef]
16. Jeddisaravi, K.; Alitappeh, R.J.; Guimarães, F.G. Multi-objective mobile robot path planning based on a search. In Proceedings of the 2016 6th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 13–14 October 2016; pp. 7–12.
17. Asseo, S.J. Terrain following/terrain avoidance path optimization using the method of steepest descent. In Proceedings of the IEEE 1988 National Aerospace and Electronics Conference, Dayton, OH, USA, 23–27 May 1988; pp. 1128–1136.
18. Zhang, X.; Hu, X.; Xie, G. Research of the Digital Map Disposing Technology in Route Planning. *Fire Control Command Control* **2012**, *37*, 4. (In Chinese) [CrossRef]
19. Li, X. *Studying on the Integrative Algorithm of the TF/TA Optimal Trajectory Planning*; Northwestern Polytechnical University: Xi'an, China, 2003; Volume 3, pp. 45–100.
20. Kienzle, S. The effect of DEM raster resolution on first order, second order and compound terrain derivatives. *Trans. Gis* **2010**, *8*, 83–111. [CrossRef]
21. Koenderink, J.J. The structure of images. *Biol. Cybern.* **1984**, *50*, 363–370. [CrossRef] [PubMed]
22. Li, X.; Orchard, M.T. New edge-directed interpolation. *IEEE Trans. Image Process.* **2001**, *3*, 36–50.
23. Skidmore, A.K. A comparison of techniques for calculating gradient and aspect from a gridded digital elevation model. *Int. J. Geogr. Inf. Sci.* **1989**, *3*, 32–45. [CrossRef]
24. Shi, W.Z.; Li, Q.Q.; Zhu, C.Q. Estimating the propagation error of DEM from higher-order interpolation algorithms. *Int. J. Remote Sens.* **2005**, *26*, 3069–3084. [CrossRef]
25. Zhang, Z.; Shen, D.; Tang, X.L. Review of route planning for combat aircraft penetration. *Aero Weapon.* **2022**, *29*, 11–19.
26. Zhao, M.Q.; Kang, T.T.; Wang, Q. Research on slope algorithm applicability based on 1:50000 digital elevation model. In Proceedings of the 2011 International Conference on Ecological Protection of Lakes-Wetlands-Watershed and Application of 3S Technology (EPLWW3S 2011 V2), Nanchang, China, 25 June 2011.
27. Kurzer, K. Path Planning in Unstructured Environments: A Real-time Hybrid A* Implementation for Fast and Deterministic Path Generation for the KTH Research Concept Vehicle. Master's Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2016. [CrossRef]
28. Koenig, S.; Likhachev, M.; Furcy, D. Lifelong Planning A. *Artif. Intell.* **2004**, *155*, 93–146. [CrossRef]
29. Rüdiger, E.; Drechsler, R. Weighted A* search—Unifying view and application. *Artif. Intell.* **2009**, *173*, 1310–1342. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Joint Overlapping Event Extraction Model via Role Pre-Judgment with Trigger and Context Embeddings

Qian Chen [1], Kehan Yang [1], Xin Guo [1,*], Suge Wang [1,2], Jian Liao [1] and Jianxing Zheng [1,2]

[1] School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China; chenqian@sxu.edu.cn (Q.C.); 202222407057@email.sxu.edu.cn (K.Y.); wsg@sxu.edu.cn (S.W.); liaoj@sxu.edu.cn (J.L.); jxzheng@sxu.edu.cn (J.Z.)

[2] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan 030006, China

* Correspondence: guoxinjsj@sxu.edu.cn

Abstract: The objective of event extraction is to recognize event triggers and event categories within unstructured text and produce structured event arguments. However, there is a common phenomenon of triggers and arguments of different event types in a sentence that may be the same word elements, which poses new challenges to this task. In this article, a joint learning framework for overlapping event extraction (ROPEE) is proposed. In this framework, a role pre-judgment module is devised prior to argument extraction. It conducts role pre-judgment by leveraging the correlation between event types and roles, as well as trigger embeddings. Experiments on the FewFC show that the proposed model outperforms other baseline models in terms of Trigger Classification, Argument Identification, and Argument Classification by 0.4%, 0.9%, and 0.6%. In scenarios of trigger overlap and argument overlap, the proposed model outperforms the baseline models in terms of Argument Identification and Argument Classification by 0.9%, 1.2%, 0.7%, and 0.6%, respectively, indicating the effectiveness of ROPEE in solving overlapping events.

Keywords: overlapping event extraction; trigger overlap; argument overlap; joint learning; role pre-judgment

Citation: Chen, Q.; Yang, K.; Guo, X.; Wang, S.; Liao, J.; Zheng, J. Joint Overlapping Event Extraction Model via Role Pre-Judgment with Trigger and Context Embeddings. *Electronics* **2023**, *12*, 4688. https://doi.org/10.3390/electronics12224688

Academic Editor: Arkaitz Zubiaga

Received: 29 October 2023
Revised: 16 November 2023
Accepted: 16 November 2023
Published: 18 November 2023

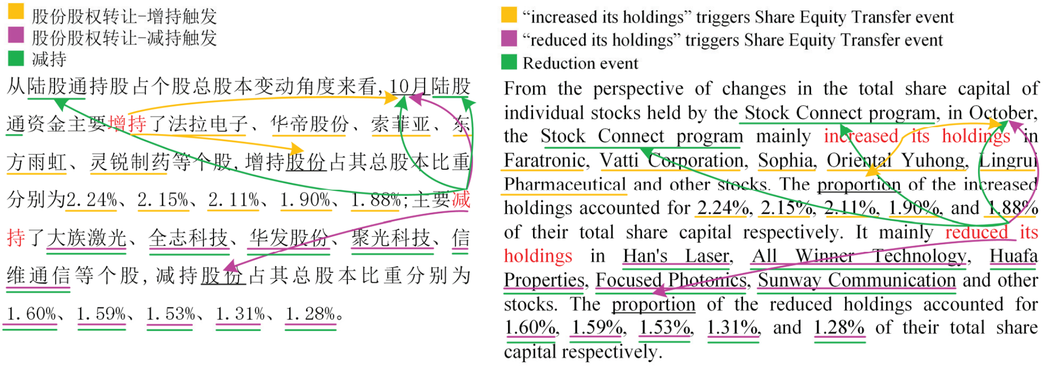
1. Introduction

Event extraction (EE) is a challenging task in natural language understanding that plays a crucial role [1–4]. EE is dedicated to extracting event information occurring in the real world from text, classifying them into predefined event types, identifying the trigger and event participants, etc. [5]. Event extraction can be widely used in information retrieval and summarization, knowledge graph construction [6], intelligence analysis [7], and other fields. EE usually includes four subtasks, i.e., trigger identification (TI), Trigger Classification (TC), Argument Identification (AI), and Argument Classification (AC) [8–10].

The objective of event extraction is to identify events and arguments from the text. The event extraction task is formally defined as follows:

Given an input sentence $x = \{x_1, x_2, \ldots, x_N\}$ consisting of N words, an event-type collection ($\mathcal{C}$), and an argument role collection ($\mathcal{R}$). Let $\mathcal{E}_x$ represent the gold set. $\mathcal{E}_x$ includes all event types ($\mathcal{C}_x \subseteq \mathcal{C}$) in the sentence, all triggers ($\mathcal{T}_{x,c}$) in each event type ($c \in \mathcal{C}_x$), and different argument roles under each event type. $a_r \in \mathcal{A}_x$ is the argument corresponding to $r \in \mathcal{R}$.

However, a sentence may contain multiple events, and the arguments and triggers of these events have complex overlapping phenomena. We summarize this into three situations: (1) Different event types may be triggered by the same trigger. The trigger "减持" (reduced its holdings) marked in red in Figure 1 triggers both the shareholding reduction event and the share equity transfer event. (2) The same argument plays different roles in different event types. In Figure 1, "大族激光" (Han's Laser) plays the role of *obj* in the “股份股权转让” (share equity transfer) event, and also plays the *target-company* role in the share equity transfer, which is triggered by “减持” (reduced its holdings). (3) The same argument plays the same role in different events of the same event type. In Figure 1, the “股份股权转让” (share equity transfer) event appears twice in the sentence, both occurring in “10月” (October). In FewFC (a Chinese financial event extraction dataset) [11], about 13.5% of the sentences have trigger overlap problems, and 21.7% of the sentences have argument overlap problems. The same event type appears repeatedly in 8.3% of the sentences.


🟨 股份股权转让-增持触发
🟪 股份股权转让-减持触发
🟩 减持

从陆股通持股占个股总股本变动角度来看，10月陆股通资金主要增持了法拉电子、华帝股份、索菲亚、东方雨虹、灵锐制药等个股，增持股份占其总股本比重分别为2.24%、2.15%、2.11%、1.90%、1.88%；主要减持了大族激光、全志科技、华发股份、聚光科技、信维通信等个股，减持股份占其总股本比重分别为1.60%、1.59%、1.53%、1.31%、1.28%。

🟨 “increased its holdings” triggers Share Equity Transfer event
🟪 “reduced its holdings” triggers Share Equity Transfer event
🟩 Reduction event

From the perspective of changes in the total share capital of individual stocks held by the Stock Connect program, in October, the Stock Connect program mainly increased its holdings in Faratronic, Vatti Corporation, Sophia, Oriental Yuhong, Lingrui Pharmaceutical and other stocks. The proportion of the increased holdings accounted for 2.24%, 2.15%, 2.11%, 1.90%, and 1.88% of their total share capital respectively. It mainly reduced its holdings in Han's Laser, All Winner Technology, Huafa Properties, Focused Photonics, Sunway Communication and other stocks. The proportion of the reduced holdings accounted for 1.60%, 1.59%, 1.53%, 1.31%, and 1.28% of their total share capital respectively.

Figure 1. Examples of three overlapping phenomena.

Most previous studies partially addressed overlapping issues and did not cover all the above situations. In 2019, Yang et al. [12] utilized a staged pipeline approach for event trigger and argument extraction. Nevertheless, this method overlooked the challenge of trigger overlap. In 2020, Xu et al. [13] used a joint extraction framework to solve the role overlapping problem. Xu defined event relationship triples to represent the relationship between triggers, arguments, and roles, thereby converting the argument classification problem into a relationship extraction problem. In 2020, Huang et al. [14] proposed using a hierarchical knowledge structure graph containing conceptual and semantic reasoning paths to represent knowledge. They employed GEANet to encode intricate knowledge, addressing the issue of trigger extraction in nested structures within the biomedical domain [14]. In 2022, Zhang et al. [15] designed a two-stage pipeline model in which the trigger is identified using a sequence annotation approach, and overlapping arguments are identified through multiple sets of role binary classification networks. In 2023, Yang et al. [16] used a multi-task learning model to extract entity relationships and events, in which a multi-label classification method is used to settle the overlapping role problem shared by these two tasks.

The contributions of this paper are summarized as follows:

(1) A role pre-judgment module is proposed to predict roles based on the correspondence between event types and roles, text embeddings, and trigger embeddings, which can significantly improve the recall rate of each subtask and provide a basis for extracting overlapping arguments.

(2) ROPEE adopts a joint learning framework, and the designed loss function includes the losses of four modules, event-type detection, trigger extraction, role pre-judgment, and argument extraction, so as to effectively learn the interactive relationship between modules during training. Thus, error propagation issues can be alleviated in the prediction stage.

(3) ROPEE outperforms the baseline model by 0.4%, 0.9%, and 0.6% in terms of F1 over TC, AI, and AC on the FewFC dataset. For sentences with overlapping triggers, ROPEE outperforms the baseline model by 0.9% and 1.2% in terms of F1 over AI and AC, respectively. In the case of overlapping arguments, ROPEE demonstrates

superior performance compared to the baseline model, with improvements of 0.7% and 0.6%. This highlights the effectiveness of our suggested approach in managing overlapping occurrences of event phenomena.

The remainder of this paper is organized as follows: In Section 2, related studies are given. In Section 3, the details of the ROPEE model are introduced. Comparative experiments are performed and experimental results are analyzed in Section 4. Section 5 concludes this work.

2. Related Studies

Event extraction is one of the most challenging tasks in information extraction research [17]. Existing paradigms related to event extraction include pipeline methods and joint learning methods [18].

The pipeline-based method handles these four subtasks of EE separately. Each subtask has its own objective function and loss. In 2015, Chen et al. [8] developed a dynamic multi-pooling convolutional neural network (DMCNN). This network utilizes a dynamic multi-pooling layer based on event triggers and arguments to retain essential information, combining both the sentence-level and lexical-level details from the raw text without the need for extensive preprocessing. Most deep learning supervised methods for event extraction require lots of labeled data for training. Annotating large amounts of data is very laborious and hard to get. To gain more insights from limited training data, Yang et al. [12] combined the extraction model and event generation method in 2019 and improved the performance of the argument extractor through a weighted loss function based on various role importance. The above two methods cannot explicitly model the semantics between events and roles, nor can they capture the interaction between them. In 2020, Li et al. [19] devised a multi-stage QA framework to represent event extraction as reading comprehension issues and captured the dynamic connection between each subtask by integrating previous answers into questions. The generative event extraction model proposed by Paolini et al. [20] in 2021 solves the encoding problem of label semantics and other weak supervision signals in a pipeline manner and can improve the performance in few-sample scenarios. Since the loss function in the pipeline-based method is calculated after the argument extraction, error propagation problems may occur.

Joint learning methods integrate the loss in both the trigger extraction stage and argument extraction stage into the final loss function, treating triggers and arguments equally, and the two can mutually promote each other's extraction effects [18]. In 2021, Sheng et al. [9] first covered all event overlap issues through a unified framework with a cascading decoder to perform TC, TI, and argument extraction in sequence, and F1 reached 71.4% on the FewFC dataset. In order to further extract inter-word relationships in overlapping sentences in parallel, Cao et al. [10] proposed a single-stage framework based on inter-word relationships by jointly extracting the intra-word and cross-word pair relationships of triggers and arguments. The above two methods focus on the event extraction task itself and do not introduce additional information or other tasks of joint information extraction. In 2022, Hsu et al. [21] converted event extraction into a conditional generation problem, and extracted triggers and arguments end-to-end through additional prompts and weak supervision information. In 2022, Van Nguyen et al. [22] used an edge weighting mechanism to learn the dependency graph between task instances and jointly complete the information extraction task. In addition to introducing additional prompt information in document-level event extraction, remote dependencies can also be used to improve extraction performance. In 2023, Liu et al. [23] proposed a chain reasoning paradigm for document-level event argument extraction, which represented argument queries by constructing first-order logic rules and T-Norm fuzzy logic, which is used for end-to-end learning. We propose a joint overlapping event extraction model ROPEE for the event overlapping phenomenon. It uses the correspondence between event types and roles and trigger embeddings to predict roles, which not only effectively alleviates error propagation, but also further improves the accuracy of event extraction.

3. ROPEE Model

The overall framework of ROPEE is illustrated in Figure 2. ROPEE includes four modules: event detection, trigger identification, role pre-judgment, and argument extraction. Specifically, type detection predicts potential event types and extracts overlapping triggers by calculating the similarity between sentence representations and event-type embeddings. The role pre-judgment module comprehensively considers text embeddings and trigger embeddings, pre-judgment roles based on the correspondence table between event types and roles, which can assist in the extraction of overlapping arguments. Trigger extraction and argument extraction are based on text representation that incorporates specific event types of information and specific role information, and binary classifiers are adopted to predict the starting and ending positions of triggers or arguments. To minimize error propagation, all modules are jointly learned during training.

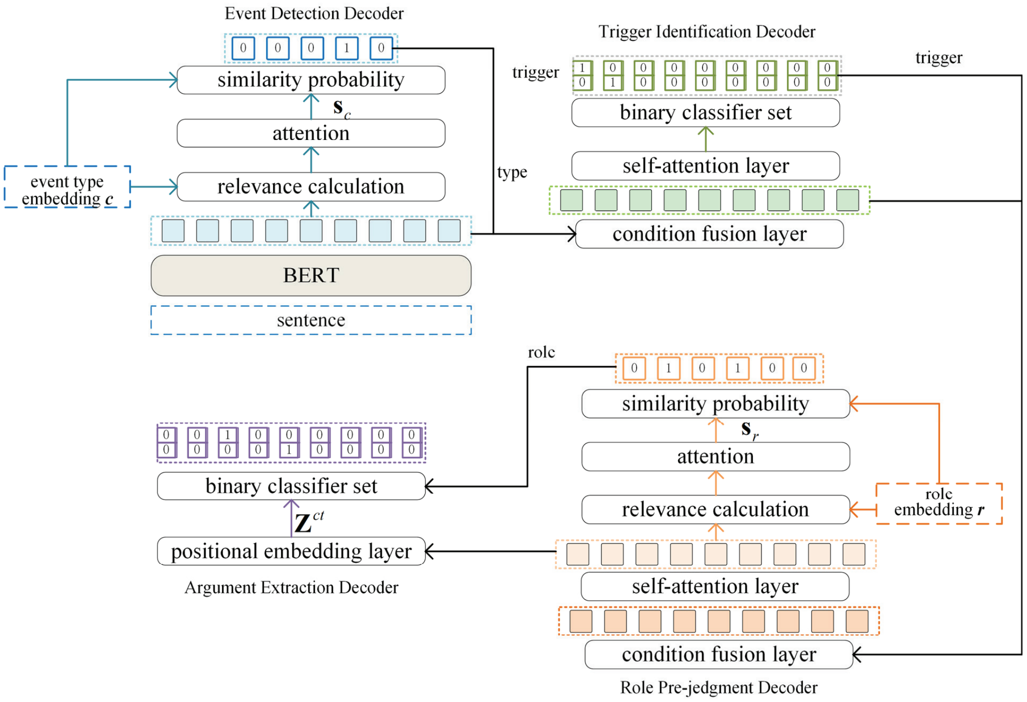


Figure 2. The overall framework of ROPEE.

3.1. *Encoder*

BERT [24] is utilized as an encoder. Sentence $x = \{x_1, x_2, \ldots, x_N\}$ treats each Chinese character (x_i) as a token and is fed into the *bert-base-Chinese* module. The embedding of the sentence is obtained by $\boldsymbol{H}$ = BERT $(x_1, x_2, \ldots, x_N) = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N\} \in \mathbb{R}^{N\times d}$, where d is the dimension of the embeddings.

3.2. *Event Detection Decoder*

The event detection decoder is shown in the upper left corner of Figure 2, which is used to predict potential event types in sentences by calculating the correlation between sentence representations that imply type features and event-type embeddings. Specifically, event-type embeddings can be denoted by a randomly initialized matrix ($\boldsymbol{C} \in \mathbb{R}^{|\mathcal{C}|\times d}$). We apply *rel* to calculate the relevance between each token embedding ($\boldsymbol{h}_i$) and the potential event type ($c \in \boldsymbol{C}$), see Equation (1), and then a sentence representation ($\boldsymbol{s}_c$) adaptive to the event type is obtained, see Equation (2). Thus, the similarity probability between $\boldsymbol{s_c}$ and $\boldsymbol{c}$ is generated using a normalization σ operation for each event, see Equation (3).

$$rel(\boldsymbol{c}, \boldsymbol{h}_i) = \boldsymbol{v}^{\mathrm{T}} \times tanh\left(\boldsymbol{W}^1_{rel}[\boldsymbol{c}; \boldsymbol{h}_i; |\boldsymbol{c} - \boldsymbol{h}_i|; \boldsymbol{c} \odot \boldsymbol{h}_i]\right) \tag{1}$$

$$\boldsymbol{s}_c = \sum_{i=1}^{N} \frac{exp(rel(\boldsymbol{c}, \boldsymbol{h}_i))}{\sum_{j=1}^{N} exp\left(rel\left(\boldsymbol{c}, \boldsymbol{h}_j\right)\right)} \boldsymbol{h}_i \tag{2}$$

$$\hat{c} = \sigma(rel(\boldsymbol{c}, \boldsymbol{s}_c)) \tag{3}$$

where $\boldsymbol{W}^1_{rel} \in \mathbb{R}^{4d \times 4d}$ and $\boldsymbol{v} \in \mathbb{R}^{4d \times 1}$ are parameters of relevance calculation. $|\cdot|$ denotes the element-wise subtraction operation. $\odot$ denotes the element-wise multiplication. $[\cdot;\cdot]$ represents the concatenating operation. σ represents the sigmoid function. Types satisfying $\hat{c} > \xi_1$ are selected as potential event types, where ξ_1 is a threshold hyperparameter between 0 and 1. All potential event types hidden in sentence x constitute the set of event types $\mathcal{C}_x$. The decoder can learn the parameters $\theta_{td} = \{\boldsymbol{W}^1_{rel}, \boldsymbol{v}, \boldsymbol{C}\}$.

3.3. Trigger Identification Decoder

A large number of experiments demonstrate that trigger information can enhance the ability of argument extraction. The trigger identification decoder is used to identify triggers according to a specific event type ($c \in \mathcal{C}_x$). The decoder includes a conditional layer normalization (CLN) [25], a self-attention layer [26], and a binary trigger tagging classifier pair.

CLN fuses the two features and filters out unnecessary information. Here, the event-type information is encoded into the token representation, and the event-typed token representation ($\boldsymbol{g}^c_i$) is obtained:

$$\boldsymbol{g}^c_i = \mathrm{CLN}(\boldsymbol{c}, \boldsymbol{h}_i) = \gamma_c \odot \left(\frac{\boldsymbol{h}_i - \mu}{\sigma}\right) + \beta_c \tag{4}$$

where type embedding $\boldsymbol{c}$ is used as the condition for $\gamma_c = \boldsymbol{W}_\gamma \boldsymbol{c} + \boldsymbol{b}_\gamma$ and $\beta_c = \boldsymbol{W}_\beta \boldsymbol{c} + \boldsymbol{b}_\beta$ in CLN. $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}$ are regarded as the average and deviation of $\boldsymbol{h}_i$:

$$\mu = \frac{1}{d}\sum_{k=1}^{d} h_{ik}, \quad \sigma = \sqrt{\frac{1}{d}\sum_{k=1}^{d}(h_{ik} - \mu)^2} \tag{5}$$

where h_{ik} represents the k-th dimension of $\boldsymbol{h}_i$.

In order to fully consider the contextual connection in the sentence, a self-attention layer [26] is used on the event-typed token representation:

$$\boldsymbol{Z}^c = \mathrm{SelfAttention}(\boldsymbol{G}^c) \tag{6}$$

where $\boldsymbol{G}^c = \left\{\boldsymbol{g}^c_1, \boldsymbol{g}^c_2, \ldots, \boldsymbol{g}^c_N\right\}$, $\boldsymbol{G}^c \in \mathbb{R}^{N \times d}$.

For each token, the binary classifier pair can mark the beginning and end position of a trigger span:

$$\hat{t}^{sc}_i = p\left(t_s \middle| x_i, c\right) = \sigma\left(\boldsymbol{w}^{\mathrm{T}}_{t_s} \boldsymbol{z}^c_i + b_{t_s}\right)\ \hat{t}^{ec}_i = p\left(t_e \middle| x_i, c\right) = \sigma\left(\boldsymbol{w}^{T}_{t_e} \boldsymbol{z}^c_i + b_{t_e}\right) \tag{7}$$

where $\boldsymbol{z}^c_i$ stands for the i-th token embedding in $\boldsymbol{Z}^c$. We select the token satisfying $\hat{t}^{sc}_i > \xi_2$ as the start position, and the one satisfying $\hat{t}^{ec}_i > \xi_3$ as the end. ξ_2 and ξ_3 are threshold hyperparameters. To acquire trigger t, each starting position is enumerated and the nearest subsequent ending position is searched in the sentence. A token span from the start position to the end constitutes a complete trigger. The corresponding triggers are extracted at different stages according to the potential event type. Thus, the trigger overlapping problem can be solved naturally. The set $\mathcal{T}_{c,x}$ contains all predicted triggers (t) under event type (c) in sentence (x). θ_{te} is used to denote all parameters in the trigger identification decoder module.

3.4. Role Pre-Judgment Decoder

Since not all roles appear in a sentence under a specific event type, we designed a role pre-judgment decoder. Based on the predicted event type, it predicts the roles appearing in the sentence based on the corresponding list of event types and roles, providing a basis for extracting overlapping arguments. The decoder consists of three parts: the conditional fusion layer, the self-attention layer, and a role similarity detection function.

In order to obtain richer semantic information, we use CLN to fully integrate trigger embeddings and token representation with event-type knowledge to obtain a new token representation ($\boldsymbol{g}_i^{ct}$), see Equation (8). Here, trigger embedding $\boldsymbol{t}$ is calculated by the average pooling of token embeddings in the trigger span.

$$\boldsymbol{g}_i^{ct} = \mathrm{CLN}(\boldsymbol{g}_i^{c}, \boldsymbol{t}) \tag{8}$$

The self-attention layer then reinforces the contextual relationships and the sentence representation $\boldsymbol{Z}^{ct\prime}$ is obtained:

$$\boldsymbol{Z}^{ct\prime} = \mathrm{SelfAttention}\left(\boldsymbol{G}^{ct}\right) \tag{9}$$

where $\boldsymbol{G}^{ct} = \left\{\boldsymbol{g}_1^{ct}, \boldsymbol{g}_2^{ct}, \ldots, \boldsymbol{g}_N^{ct}\right\}$, $\boldsymbol{G}^{ct} \in \mathbb{R}^{N\times d}$.

The role similarity detection function predicts potential event-type-specific roles in sentences by calculating the correlation between role embeddings and sentence representations fused with role feature information. Specifically, a randomly initialized matrix $\boldsymbol{R}\in \mathbb{R}^{|\mathcal{R}|\times d}$ is used as role embeddings. We apply *rel* to calculate the relevance between each token embedding ($\boldsymbol{z}_i^{ct\prime}$) and the potential role ($\boldsymbol{r}\in\boldsymbol{R}$), see Equation (10), and then a sentence representation ($\boldsymbol{s}_r$) adaptive to the role is obtained, see Equation (11). Thus, the similarity probability between $\boldsymbol{s}_r$ and $\boldsymbol{r}$ is generated, see Equation (12), based on which normalization operation is performed to obtain the predicted probabilities of all roles under a specific event type.

$$rel\left(\boldsymbol{r}, \boldsymbol{z}_i^{ct\prime}\right) = \boldsymbol{v}^{\mathrm{T}} \times tanh\left(\boldsymbol{W}_{rel}^{2}\left[\boldsymbol{r}; \boldsymbol{z}_i^{ct\prime}; \left|\boldsymbol{r} - \boldsymbol{z}_i^{ct\prime}\right|; \boldsymbol{r} \odot \boldsymbol{z}_i^{ct\prime}\right]\right) \tag{10}$$

$$\boldsymbol{s}_r = \sum_{i=1}^{N} \frac{exp\left(rel\left(\boldsymbol{r}, \boldsymbol{z}_i^{ct\prime}\right)\right)}{\sum_{j=1}^{N} exp\left(rel\left(\boldsymbol{r}, \boldsymbol{z}_j^{ct\prime}\right)\right)} \boldsymbol{z}_i^{ct\prime} \tag{11}$$

$$\hat{r}^{ct} = p(r|x_i, c, t) = \sigma(rel(\boldsymbol{r}, \boldsymbol{s}_r)) \tag{12}$$

Select the role that satisfies $\hat{r}^{ct} > \xi_4$ as the potential role type, and ξ_4 is the threshold. The role type set $\mathcal{R}_{t,c,x}$ contains all potential roles whose trigger is t under event type c in sentence x. θ_{re} is used to denote all parameters in the role pre-judgment decoder module.

3.5. Argument Extraction Decoder

An argument extraction decoder is composed of a positional embedding layer (PEL) and role-aware binary classifier pairs for argument tagging.

The relative position of a token to the trigger in the text is beneficial for argument extraction [9,27]. Here, relative position embeddings [8] imply the relative distance information between the current token and the trigger boundary token. Relative positional embeddings are incorporated into the sentence representation ($\boldsymbol{Z}^{ct\prime}$) using a concatenation operation:

$$\boldsymbol{Z}^{ct} = \left[\boldsymbol{Z}^{ct\prime}; \boldsymbol{P}\right] \tag{13}$$

where $\boldsymbol{P} \in \mathbb{R}^{N \times d_p}$ is the relative position embeddings, and d_p is the dimension of position embeddings.

For each token, a binary classifier sequence is employed to mark the boundary position of an argument under $r \in \mathcal{R}_{t,c,x}$:

$$\begin{aligned} \hat{a}_i^{sctr} &= p\left(a_r^s \middle| x_i, c, t, r\right) = \hat{r}^{ct}\sigma\left(w_{r_s}^T z_i^{ct} + b_{r_s}\right) \\ \hat{a}_i^{ectr} &= p\left(a_r^e \middle| x_i, c, t, r\right) = \hat{r}^{ct}\sigma\left(w_{r_e}^T z_i^{ct} + b_{r_e}\right) \end{aligned} \tag{14}$$

where z_i^{ct} represents the i-th token in $\boldsymbol{Z}^{ct}$. For each role (r), select the token that satisfies $\hat{a}_i^{sctr} > \xi_5$ as the starting position and the one that satisfies $\hat{a}_i^{ectr} > \xi_6$ as the end. $\xi_5, \xi_6 \in [0,\ 1]$ are thresholds. In order to extract the boundary of argument (a_r) with role (r), all starting positions are enumerated and the nearest subsequent ending position is searched in the sentence. Tokens between the starting and the ending position constitute a complete argument. In this way, only arguments of a specific role (r) under a specific trigger (t) and specific event type (c) in a sentence are extracted at each prediction stage. Thereby, the argument overlapping problem can be solved naturally. All candidate arguments (a_r) form a set $\mathcal{A}_{r,t,c,x}$, and θ_{ae} denotes the set of all parameters of the PEL and argument classifier.

3.6. Model Training

The loss of four modules is integrated during the training process, so the total loss function is designed as follows:

$$\mathrm{Loss}_{\mathrm{all}} = -\sum_{x \in \mathcal{D}} [\sum_{c \in \mathcal{C}_x} logp_{\theta_1}(c|x) + \sum_{t \in \mathcal{T}_{x,c}} logp_{\theta_2}(t|x,c) + \sum_{r \in \mathcal{R}_{x,c,t}} logp_{\theta_3}(r|x,c,t) + \sum_{a_r \in \mathcal{A}_{x,c,t,r}} logp_{\theta_4}(a_r|x,c,t,r)] \tag{15}$$

where $\Theta \triangleq \{\theta_1, \theta_2, \theta_3, \theta_4\}$. The first two subtasks, $p_{\theta_1}(c|x)$ and $p_{\theta_2}(t|x,c)$, are adopted from [9]. We decomposed the argument extraction loss to $p_{\theta_3}(r|x,c,t)$ and $p_{\theta_4}(a_r|x,c,t,r)$ and formulated it as:

$$\begin{gathered} p_{\theta_1}(c|x) = (\hat{c})^{\overline{c}}(1-\hat{c})^{(1-\overline{c})} \\ p_{\theta_2}(t|x,c) = \prod_{z \in \{s,e\}} \prod_{i=1}^{N} \left(\hat{t}_i^{zc}\right)^{\overline{t}_i^{zc}} \left(1-\hat{t}_i^{zc}\right)^{(1-\overline{t}_i^{zc})} \\ p_{\theta_3}(r|x,c,t) = \left(\hat{r}^{ct}\right)^{\overline{r}^{ct}} \left(1-\hat{r}^{ct}\right)^{(1-\overline{r}^{ct})} \\ p_{\theta_4}(a_r|x,c,t,r) = \prod_{z \in \{s,e\}} \prod_{i=1}^{N} \left(\hat{a}_i^{zctr}\right)^{\overline{a}_i^{zctr}} \left(1-\hat{a}_i^{zctr}\right)^{(1-\overline{a}_i^{zctr})} \end{gathered} \tag{16}$$

where $\hat{c}$, $\hat{t}_i^{sc}$, $\hat{t}_i^{ec}$, $\hat{r}^{ct}$, $\hat{a}_i^{sctr}$, and $\hat{a}_i^{ectr}$ are the predicted probabilities of the event type, starting and ending positions of triggers, role types, and starting and ending positions of arguments, respectively, which can be calculated according to Formulas (3), (7), (12), and (14). $\overline{c}$, $\overline{t}_i^{sc}$, $\overline{t}_i^{ec}$, $\overline{r}^{ct}$, $\overline{a}_i^{sctr}$, and $\overline{a}_i^{ectr}$ are real labels in the training data. $\theta_1 \triangleq \{\theta_{bert}, \theta_{td}\}$, $\theta_2 \triangleq \{\theta_{bert}, \theta_{te}\}$, $\theta_3 \triangleq \{\theta_{bert}, \theta_{re}\}$, $\theta_4 \triangleq \{\theta_{bert}, \theta_{ae}\}$, and θ_{bert}, θ_{td}, θ_{te}, θ_{re}, θ_{ae} denote the parameters from BERT, event detection, trigger identification, role pre-judgment, and argument extraction, respectively. We choose Adam [28] over shuffled mini-batches to minimize $\mathrm{Loss}_{\mathrm{all}}$.

4. Experiments and Analysis

4.1. Datasets

We use the FewFC dataset [11] to conduct comparative experiments. The reason for choosing FewFC is that other datasets do not completely cover the three overlapping situations mentioned in Section 1 like FewFC. For example, only 10% of events in the mainstream ACE2005 dataset have overlapping arguments, and there are no samples with overlapping triggers [12]. FewFC is a benchmark dataset in the Chinese financial field extracted for overlapping events, in which a total of 10 event types and 18 roles are annotated, and about 22% of the sentences contain overlapping events. Regardless of whether the event types are the same, the test set in FewFC contains 168 samples with overlapping triggers and 209 samples with overlapping arguments. Note that there is an intersection between overlapping samples. The dataset is split into training, validation, and testing by 80%, 10%, and 10%. See Table 1 for more detail.

Table 1. Statistics of FewFC. Each column represents the size of each subset in terms of overlapped triggers, arguments, samples, and events.

	Trigger Overlap	Argument Overlap	Samples	Event
Training	1314	1541	7185	10,277
Validation	168	203	899	1281
Testing	168	209	898	1332
All	1650	1953	8982	12,890

4.2. Implementation Details

We choose PyTorch for code implementation, and NVIDIA A100-PCIE-40GB for training. ROPEE uses the *bert-base-Chinese* model with a starting learning rate of 2×10^{-5}, a decoder learning rate of 1×10^{-4}, and a decoder dropout rate of 0.3. The batch size is eight. The hidden layer size (d) is 768, the size of the position embeddings (d_p) is 64, and the epochs for training are 20. All hyperparameters are turned on for the validation set. The event type embeddings and role embeddings are trained from scratch with random initialization.

4.3. Evaluation Metric

The evaluation metric includes four parts [8–10]: (1) Trigger Identification (TI): The trigger is considered correctly recognized if the predicted trigger span aligns with the ground truth label. (2) Trigger Classification* (TC): The trigger is deemed correctly classified when it is both accurately identified and assigned to the right event type. (3) Argument Identification (AI): The argument is considered correctly identified if the event type is accurately recognized and the predicted argument span aligns with the gold span. (4) Argument Classification (AC): The argument is considered correctly classified when it is both accurately identified and the predicted role matches the gold role. Each of these parts is evaluated by three metrics: micro precision (P), micro recall (R), and micro F1-measure (F1). The specific formula is as follows:

$$\begin{aligned} \text{Precision}_{micro} &= \frac{\sum_{i=1}^{n} \text{TP}_i}{\sum_{i=1}^{n} \text{TP}_i + \sum_{i=1}^{n} \text{FP}_i} \\ \text{Recall}_{micro} &= \frac{\sum_{i=1}^{n} \text{TP}_i}{\sum_{i=1}^{n} \text{TP}_i + \sum_{i=1}^{n} \text{FN}_i} \\ \text{F1}_{micro} &= 2 \cdot \frac{\text{Precision}_{micro} \cdot \text{Recall}_{micro}}{\text{Precision}_{micro} + \text{Recall}_{micro}} \end{aligned} \tag{17}$$

where $\text{Precision}_{\text{micro}}$ and $\text{Recall}_{\text{micro}}$ represent the average precision and recall across all categories.

4.4. Baselines

The following baseline models are chosen to compare with ROPEE over the FewFC dataset:

(1) BERT-softmax [24]: It uses BERT to obtain the feature representation of words for the classification of both trigger and event arguments.

(2) BERT-CRF [29]: It uses a CRF module based on BERT to catch the transfer rules between adjacent tags.

(3) BERT-CRF-joint: It extends the classic BIO labeling scheme by merging tags of event types and roles for sequence annotation, such as BIO-type roles [30].

These methods convert the EE task into a sequence labeling task by attaching a label to each token. The flattened sequence labeling approach cannot address the overlapping problem due to label conflicts.

(4) PLMEE [12]: It extracts triggers and arguments in a pipeline manner, and alleviates the argument overlapping issue by extracting role-aware arguments.

(5) MQAEE: It is extended based on Li et al. [15]. It first predicts overlapping triggers through question and answer, and then predicts overlapping arguments based on typed triggers.

(6) CasEE [9]: It performs all four subtasks sequentially with a cascade decoder based on the specific previous predictions.

The above three methods are multi-stage methods for overlapping event extraction.

4.5. Main Results

Table 2 shows experimental results comparing ROPEE with baseline models over FewFC. It can be revealed from the table below.

Table 2. Model comparative results of EE on four subtasks over FewFC. Highest scores are in bold.

	TI (%)			TC (%)			AI (%)			AC (%)		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
BERT-softmax	89.8	79.0	84.0	80.2	61.8	69.8	74.6	62.8	68.2	72.5	60.2	65.8
BERT-CRF	**90.8**	80.8	85.5	**81.7**	63.6	71.5	75.1	64.3	69.3	72.9	61.8	66.9
BERT-CRF-joint	89.5	79.8	84.4	80.7	63.0	70.8	**76.1**	63.5	69.2	**74.2**	61.2	67.1
PLMEE	83.7	85.8	84.7	75.6	74.5	75.1	74.3	67.3	70.6	72.5	65.5	68.8
MQAEE	89.1	85.5	87.4	79.7	76.1	77.8	70.3	68.3	69.3	68.2	66.5	67.3
CasEE	89.4	87.7	**88.6**	77.9	78.5	78.2	72.8	73.1	72.9	71.3	71.5	71.4
ROPEE	88.8	**88.2**	88.5	74.7	**82.8**	**78.6**	69.5	**78.6**	**73.8**	67.6	**76.9**	**72.0**

(1) In contrast to the flattened sequence labeling methods, ROPEE achieves superior recall and F1 scores. Specifically, ROPEE outperforms BERT-CRF-joint by 15.7% and 4.9% on recall and the F1 score of AC. ROPEE also achieves significantly better than the sequence labeling method on recall because the sequence labeling method can only solve the flat event extraction problem, which can cause label conflicts. This shows that ROPEE can effectively solve the problem of overlapping event extraction.

(2) Compared with the multi-stage methods for overlapping event extraction, the F1 scores of ROPEE in AI and AC are greater than that of CasEE by 0.9% and 0.6%, respectively. We believe that the role pre-judgment decoder in the model provides good help for argument extraction. In particular, ROPEE outperforms CasEE on the recall score of all four subtasks, especially AI and AC by 5.5% and 5.4%, respectively. This shows that the ROPEE model can better recall arguments that match the role type in the training of role pre-judgment. Overall, ROPEE outperforms all multi-stage methods for overlapping event extraction.

(3) We also conducted comparative experiments on the large language model ChatGLM2-6B [31,32]. Some parameters of ChatGLM were fine-tuned using the P-Tuning-v2 method, but the final result was not ideal. ChatGLM is more accurate in extracting core arguments and triggers, but the span position it extracts is seriously inconsistent with the original text, indicating that ChatGLM fails to understand the boundary meaning represented by span.

4.6. Results of Overlapped EE

Table 3 shows the comparative performance of ROPEE and CasEE in two overlapping situations. For sentences with overlapping triggers, ROPEE achieves improvements of 0.9% and 1.2% over CasEE on the F1 score of AI and AC, respectively. For overlapping arguments, ROPEE outperforms CasEE in all four subtasks. Experimental results illustrate the superiority of ROPEE in solving overlapping problems, and we argue that the role pre-judgment decoder predicts potential roles and promotes the performance of extraction in overlapping arguments.

Table 3. Comparative results of ROPEE and CasEE in two overlapping situations. Highest scores are in bold.

		TI (%)	TC (%)	AI (%)	AC (%)
Trigger Overlap	CasEE	**92.5**	**82.8**	75.5	74.2
	ROPEE	92.0	82.4	**76.4**	**75.4**
Argument Overlap	CasEE	88.6	78.2	74.2	72.8
	ROPEE	**89.2**	**78.7**	**74.9**	**73.4**

4.7. Ablation Studies

To verify the rationality of the role pre-judgment decoder, we conducted ablation experiments. In Table 4, *ROP-text_emb* only uses the text representation containing event-type embeddings as the predicted role type. *type_emb* denotes models that pre-judge roles using only event-type embeddings. As shown from this table, the *type_emb* used in the CasEE model performs worse than ROPEE on the F1 of TC, AI, and AC. It can be seen that event-type embeddings alone are not as efficient as the text embeddings containing event-type embeddings when used as the input for the role pre-judgment decoder, e.g., *ROP-text_emb*. ROPEE performs role pre-judgment by fusing the information of text representation and trigger embeddings, which outperforms *ROP-text_emb* in the F1 score of all four subtasks. It can be seen that trigger embeddings serve as supplementary information, allowing the model to grasp the knowledge hidden in the text.

Table 4. Comparative results on F1 using different role classifier strategies. Highest scores are in bold.

	TI (%)	TC (%)	AI (%)	AC (%)
type_emb	**88.6**	78.2	72.9	71.4
ROP-text_emb	88.2	78.4	73.4	71.7
ROPEE	88.5	**78.6**	**73.8**	**72.0**

In order to check the effectiveness of the role pre-judgment decoder, we adopt two training strategies while retaining the role pre-judgment decoder. The experimental results are given in Table 5. We use the correct role type for training and validation, such as *Role-x* (*x* can be 1, 2, 3, 4, or 5, representing the role pre-judgment decoder in the loss function after the correct role is input to the model weight value). With hyperparameters unchanged, the *Role-1* model converged quickly. It stopped iterating at 11 epochs, and its effects on four subtasks were not as good as ROPEE. To enable the model to learn more fully and pay attention to the learning of role types, we gradually increase the weight of the loss function in the role prediction part with the other hyperparameters remaining unchanged. An obvious trend of first increasing and then decreasing can be seen from *Role-x* in Table 5, and the best performance is achieved in *Role-3*, which is only 0.3% lower than ROPEE in both of AI and AC. It can be seen that the training strategy of using the correct role type requires adjusting parameters on the original basis to enable the model to focus on role-type learning. However, the performance of this training strategy is not as good as ROPEE. We argue that the correct role type information can greatly reduce the difficulty of the task. The rapid convergence of the model during training leads to insufficient learning and over-fitting.

Table 6 shows the comparative results between a model with trigger extraction and one without trigger extraction. In the stage of data annotation, if no triggers are annotated, the event extraction task is completed using only event types, roles, and arguments. Since AI and AC tasks need only event-type information, we use these two to measure the performance of ROPEE. According to Table 6, 20 epochs are also used for training. When the correct triggers are not provided, the F1 score is decreased by 1.8% and 1.5% respectively, in AI and AC. Due to the increased difficulty in model training, we increased the number of training epochs to 50, and the result was increased by 0.2% in the F1 of AI, in contrast with *20 epoch w/o trigger*. Thus, the lack of trigger annotation information will result in the inability to obtain the position embeddings of triggers during argument extraction, which also affects the model performance. Despite this, the model without trigger information achieves improvements than that of PLMEE and MQAEE, both of which are multi-stage methods with triggers involved.

Table 5. Comparative results on F1 using different training strategies. Highest scores are in bold.

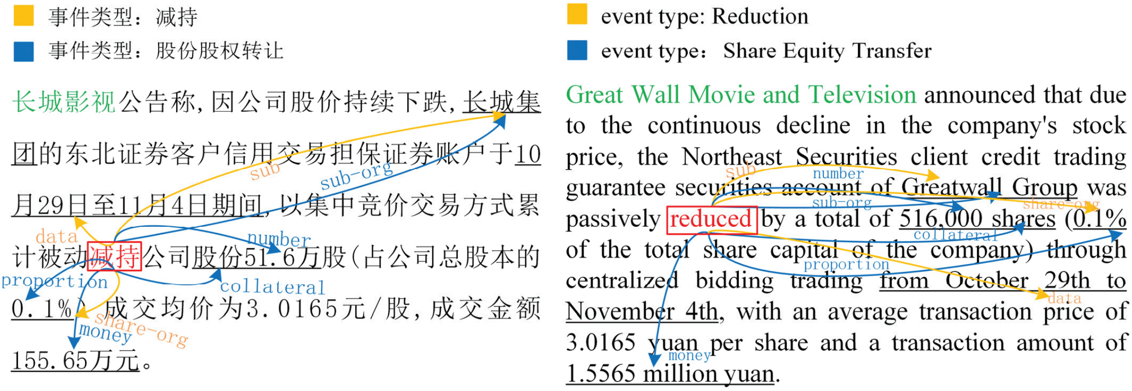
	TI (%)	TC (%)	AI (%)	AC (%)
Role-1	88.1	77.6	71.7	69.4
Role-2	**88.5**	77.7	72.7	70.6
Role-3	88.4	78.1	73.5	71.7
Role-4	87.8	76.6	72.0	70.3
Role-5	88.2	77.2	72.4	70.7
ROPEE	**88.5**	**78.6**	**73.8**	**72.0**

Table 6. Comparative results on F1 with and without trigger annotation.

		P (%)	R (%)	F1 (%)
ROPEE	AI	69.5	78.6	73.8
	AC	67.6	76.9	72.0
20 epoch w/o trigger	AI	71.4	72.6	72.0
	AC	70.0	71.0	70.5
50 epoch w/o trigger	AI	70.9	73.5	72.2
	AC	69.3	71.7	70.5

4.8. Case Analysis

Figure 3 shows a test case analysis on CasEE and ROPEE, two events both triggered by "reduce". For the "减持" (Reduction) event, both CasEE and ROPEE predict "Great Wall Film and Television" as the role of "target-company", and there is no such argument in the standard answer. This shows that these two models lack the ability to judge the semantic boundaries of events. For the "股份股权转让" (share equity transfer) event, CasEE failed to extract the remote argument "Great Wall Group" as the role *sub*, while ROPEE improved the recall of AC by adding a role pre-judgment decoder, and, thus, was able to correctly detect the role of *sub*.



🟨 事件类型：减持
🟦 事件类型：股份股权转让

长城影视公告称，因公司股价持续下跌，长城集团的东北证券客户信用交易担保证券账户于10月29日至11月4日期间，以集中竞价交易方式累计被动减持公司股份51.6万股(占公司总股本的0.1%)成交均价为3.0165元/股，成交金额155.65万元。

🟨 event type: Reduction
🟦 event type：Share Equity Transfer

Great Wall Movie and Television announced that due to the continuous decline in the company's stock price, the Northeast Securities client credit trading guarantee securities account of Greatwall Group was passively reduced by a total of 516,000 shares (0.1% of the total share capital of the company) through centralized bidding trading from October 29th to November 4th, with an average transaction price of 3.0165 yuan per share and a transaction amount of 1.5565 million yuan.

Figure 3. Golden data of test case analysis on CasEE and ROPEE.

5. Conclusions

We design a joint learning framework ROPEE for overlapping event extraction, in which the event detection decoder can identify potential event types and help extract overlapping triggers. Based on text embeddings and trigger embeddings, a role pre-judgment module is proposed to predict roles based on the corresponding relationship between event types and roles, thereby enhancing the extraction of overlapping arguments. ROPEE has a certain effectiveness in addressing the issue of trigger overlap in EE, as well as argument overlap. Unlike other pipeline models, we integrate the tasks of four modules in the loss optimization layer, avoiding the traditional problem of error propagation existing in other pipeline methods. On the FewFC dataset, which was compared with other flattened sequence labeling methods (such as BERT-softmax, BERT-CRF, and BERT-CRF-joint), ROPEE achieves excellent recall and F1 scores on all subtasks; compared with multi-stage methods for overlapping event extraction (such as PLMEE, MQAEE, and CasEE), and ROPEE is superior to the above methods in terms of F1 scores of TC, AI, and AC. The above results show the superiority of ROPEE in overlapping event extraction. The ablation experimental results show that our model can also be used in different training strategies and other task scenarios or datasets without trigger labeling. In the future, we plan to build one more overlapping datasets based on specific application scenarios to verify our model's performance. In addition, since trigger tagging requires additional manpower and material resources, the absence of trigger tagging will reduce the accuracy of the model. The scenario without trigger annotation is general (models should automatically find the core arguments in the sentence), and we plan to improve the model performance in this scenario using position embeddings relative to the core arguments. Finally, we implement joint learning by designing a joint loss of four modules in this article, and we plan to develop different joint strategies to strengthen the interaction between each subtask.

Author Contributions: Conceptualization, Q.C. and K.Y.; methodology, K.Y. and Q.C.; software, K.Y.; investigation, S.W.; resources, X.G.; writing—original draft preparation, K.Y.; writing—review and editing, J.Z.; supervision, X.G.; project administration, J.L.; funding acquisition, S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by National Key Research and Development Program of China (grant NO. 2022QY0300-01), National Natural Science Foundation of China (grant NO. 62076158), Natural Science Foundation of Shanxi Province of China (grant NOs. 202203021221021, 20210302123468 and 202203021221001), and CCF-Zhipu AI Large Model Fund (grant NO. 202310).

Data Availability Statement: The dataset used for this work is the FewFC dataset published by Zhou et al. [11]. It can be found at https://github.com/TimeBurningFish/FewFC (accessed on 17 November 2023). The codes are at https://github.com/yang-4074/ROPEE (accessed on 17 November 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Miwa, M.; Bansal, M. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv* **2016**, arXiv:1601.00770.
2. Katiyar, A.; Cardie, C. Investigating lstms for joint extraction of opinion entities and relations. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; pp. 919–929.
3. Fei, H.; Zhang, M.; Ji, D. Cross-lingual semantic role labeling with high-quality translated training corpus. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7014–7026.
4. Li, J.; Xu, K.; Li, F.; Fei, H.; Ren, Y.; Ji, D. MRN: A locally and globally mention-based reasoning network for document-level relation extraction. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, 1–6 August 2021; pp. 1359–1370.
5. Liu, Z.; Li, Y.; Zhang, Y.; Weng, Y.; Yang, K.; Wang, C. Effective Event Extraction Method via Enhanced Graph Convolutional Network Indication with Hierarchical Argument Selection Strategy. *Electronics* **2023**, *12*, 2981. [CrossRef]
6. Bosselut, A.; Le Bras, R.; Choi, Y. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In Proceedings of the 35th AAAI conference on Artificial Intelligence, Online, 2–9 February 2021; pp. 4923–4931.

7. Xiang, G.; Shi, C.; Zhang, Y. An APT Event Extraction Method Based on BERT-BiGRU-CRF for APT Attack Detection. *Electronics* **2023**, *12*, 3349. [CrossRef]
8. Chen, Y.; Xu, L.; Liu, K.; Zeng, D.; Zhao, J. Event extraction via dynamic multi-pooling convolutional neural networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; pp. 167–176.
9. Sheng, J.; Guo, S.; Yu, B.; Li, Q.; Hei, Y.; Wang, L.; Liu, T.; Xu, H. CasEE: A joint learning framework with cascade decoding for overlapping event extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*; Association for Computational Linguistics: Cedarville, OH, USA, 2021; pp. 164–174.
10. Cao, H.; Li, J.; Su, F.; Li, F.; Fei, H.; Wu, S.; Li, B.; Zhao, L.; Ji, D. OneEE: A one-stage framework for fast overlapping and nested event extraction. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 1953–1964.
11. Zhou, Y.; Chen, Y.; Zhao, J.; Wu, Y.; Xu, J.; Li, J. What the role is vs. what plays the role: Semi-supervised event argument extraction via dual question answering. In Proceedings of the 35th AAAI conference on Artificial Intelligence, Online, 2–9 February 2021; pp. 14638–14646.
12. Yang, S.; Feng, D.; Qiao, L.; Kan, Z.; Li, D. Exploring pre-trained language models for event extraction and generation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5284–5294.
13. Xu, N.; Xie, H.; Zhao, D. A novel joint framework for multiple Chinese events extraction. In Proceedings of the China National Conference on Chinese Computational Linguistics, Hainan, China, 30 October–1 November 2020; pp. 174–183.
14. Huang, K.H.; Yang, M.; Peng, N. Biomedical event extraction with hierarchical knowledge graphs. *arXiv* **2020**, arXiv:2009.09335.
15. Zhang, X.; Zhu, Y.H.; OuYang, K.; Kong, L.W. Chinese Event Extraction Based on Role Separation. *J. Shanxi Univ.* **2022**, *45*, 936–946.
16. Yang, H.J.; Jin, X.Y. A general model for entity relationship and event extraction. *Comput. Eng.* **2023**, *49*, 143–149.
17. Zhu, M.; Mao, Y.C.; Cheng, Y.; Chen, C.J.; Wang, L.B. Event Extraction Method Based on Dual Attention Mechanism. *Ruan Jian Xue Bao/J. Softw.* **2023**, *34*, 3226–3240.
18. Li, Q.; Li, J.; Sheng, J.; Cui, S.; Wu, J.; Hei, Y.; Peng, H.; Guo, S.; Wang, L.; Beheshti, A.; et al. A survey on deep learning event extraction: Approaches and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *14*, 1–21. [CrossRef]
19. Li, F.; Peng, W.; Chen, Y.; Wang, Q.; Pan, L.; Lyu, Y.; Zhu, Y. Event extraction as multi-turn question answering. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 829–838.
20. Paolini, G.; Athiwaratkun, B.; Krone, J.; Ma, J.; Achille, A.; Anubhai, R.; Santos, C.N.d.; Xiang, B.; Soatto, S. Structured prediction as translation between augmented natural languages. In Proceedings of the Ninth International Conference on Learning Representations, Online, 3–7 May 2021.
21. Hsu, I.; Huang, K.-H.; Boschee, E.; Miller, S.; Natarajan, P.; Chang, K.-W.; Peng, N. DEGREE: A data-efficient generation-based event extraction model. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, DC, USA, 10–15 July 2022; pp. 1890–1908.
22. Van Nguyen, M.; Min, B.; Dernoncourt, F.; Nguyen, T. Joint extraction of entities, relations, and events via modeling inter-instance and inter-label dependencies. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, DC, USA, 10–15 July 2022; pp. 4363–4374.
23. Liu, J.; Liang, C.; Xu, J.; Liu, H.; Zhao, Z. Document-level event argument extraction with a chain reasoning paradigm. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Toronto, ON, Canada, 9–14 July 2023; pp. 9570–9583.
24. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
25. Yu, B.; Zhang, Z.; Sheng, J.; Liu, T.; Wang, Y.; Wang, Y.; Wang, B. Semi-open information extraction. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 1661–1672.
26. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
27. Su, J.; Lu, Y.; Pan, S.; Murtadha, A.; Wen, B.; Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *arXiv* **2021**, arXiv:2104.09864.
28. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
29. Du, X.; Cardie, C. Document-level event role filler extraction using multi-granularity contextualized encoding. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 8010–8020.
30. Zheng, S.; Wang, F.; Bao, H.; Hao, Y.; Zhou, P.; Xu, B. Joint extraction of entities and relations based on a novel tagging scheme. *arXiv* **2017**, arXiv:1706.05075.

31. Zeng, A.; Liu, X.; Du, Z.; Wang, Z.; Lai, H.; Ding, M.; Yang, Z.; Xu, Y.; Zheng, W.; Xia, X. Glm-130b: An open bilingual pre-trained model. In Proceedings of the Eleventh International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
32. Du, Z.; Qian, Y.; Liu, X.; Ding, M.; Qiu, J.; Yang, Z.; Tang, J. Glm: General language model pretraining with autoregressive blank infilling. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 320–335.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

electronics

MDPI

Article

Resolving Agent Conflicts Using Enhanced Uncertainty Modeling Tools for Intelligent Decision Making

Yanhui Zhai [1,2,*], Zihan Jia [1,†] and Deyu Li [1,2,†]

[1] School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China; 202122407018@email.sxu.edu.cn (Z.J.); lidy@sxu.edu.cn (D.L.)
[2] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan 030006, China
* Correspondence: zhaiyh@sxu.edu.cn
† These authors contributed equally to this work.

Abstract: Conflict analysis in intelligent decision making has received increasing attention in recent years. However, few researchers have analyzed conflicts by considering trustworthiness from the perspective of common agreement and common opposition. Since *L*-fuzzy three-way concept lattice is able to describe both the attributes that objects commonly possess and the attributes that objects commonly do not possess, this paper introduces an *L*-fuzzy three-way concept lattice to capture the issues on which agents commonly agree and the issues which they commonly oppose, and proposes a hybrid conflict analysis model. In order to resolve conflicts identified by the proposed model, we formulate the problem as a knapsack problem and propose a method for selecting the optimal attitude change strategy. This strategy takes into account the associated costs and aims to provide the decision maker with the most favorable decision in terms of resolving conflicts and reaching consensus. To validate the effectiveness and feasibility of the proposed model, a case study is conducted, providing evidence of the model's efficacy and viability in resolving conflicts.

Keywords: intelligent decision making; conflict analysis; conflict resolving; three-way decisions; uncertainty modeling

check for updates

Citation: Zhai, Y.; Jia, Z.; Li, D. Resolving Agent Conflicts Using Enhanced Uncertainty Modeling Tools for Intelligent Decision Making. *Electronics* **2023**, *12*, 4547. https://doi.org/10.3390/electronics12214547

Academic Editor: Enrique Romero-Cadaval

Received: 12 October 2023
Revised: 26 October 2023
Accepted: 3 November 2023
Published: 5 November 2023

1. Introduction

Conflict analysis [1–3] aims to analyze complex conflict situations with appropriate models by studying the conflict relationships between individuals or groups on issues, identifying the internal causes of conflict and providing some guidance for conflict resolution in intelligent decision making such as labor negotiation [4], diplomatic relations [5], and urban planning [6].

Many scholars have proposed various models for conflict analysis from different perspectives. Pawlak [5] first considered the uncertainty of agents' attitudes toward issues and divided the agents into three groups (i.e., coalition, neutrality, and conflict). Yao [7] extended the Pawlak model [5] by examining three levels of conflict and proposed three-way conflict analysis. Lang et al. [8] further improved the Yao model [7] by employing an alliance measure and a conflict measure. In addition, considering uncertainty and data complexity in actual conflict situations, Lang et al. [9] used Pythagorean fuzzy sets to describe conflict situations and proposed a Bayesian minimal risk theory based conflict analysis method. Li et al. [10] proposed a conflict analysis model to cope with trapezoidal fuzzy numbers in agents' attitudes toward issues. Yang et al. [11] investigated a three-way conflict analysis to deal with diverse rating types in situation tables. Suo et al. [12] studied a three-way conflict analysis model to deal with incomplete three-valued situation tables. Furthermore, since psychological factors and risk attitudes of agents may affect the results of conflict analysis, Wang et al. [13] proposed a compound risk preference model for three-way decision based on different types of risk preferences.

Zhi et al. [14] introduced a three-way concept for conflict analysis [15]. Three-way concept analysis was proposed in [15] by combining three-way decision theory [16–20] and formal concept analysis [21,22], with the ability of describing the properties that objects possess in common and those that they do not possess in common [23,24]. Zhi et al. [14] analyzed the conflict relationships between agents and identified the binary relationships between sets of agents and sets of issues by using three-way concepts, and then analyzed the causes of conflict. The above studies were conducted in the case that agents are completely trustworthy. In some situations (for example, in network); however, agents may be untrustworthy. To this end, Zhi et al. [25] proposed a multilevel conflict analysis that analyzed and resolved uncertainty in agents' trustworthiness and uncertainty in agents' attitudes toward issues. However, in [25], when describing the consistency of agents using fuzzy concepts, only the common agreement consistency is considered and the common opposition consistency is ignored. This may lead to a misjudgment of conflict in some cases. For example, when agents oppose an issue, according to [25] the agents are inconsistent on this issue, but in fact, agents are consistent.

On the other hand, when agents have disputes, it is necessary to find appropriate conflict resolution strategies to promote cooperation among them. To this end, Sun et al. [26] constructed a probabilistic rough set model and provided an effective method to find the feasible consensus strategy to facilitate the resolution of conflict situations. To select an effective feasible strategy, Xu et al. [4] formulated the criteria for selecting feasible strategies based on consistency measurement of cliques. Based on game-theoretic rough sets, Bashir et al. [1] designed a novel conflict resolution model by constructing a game among all agents, computing the payoff of different strategies, and classifying them according to the equilibrium rules. From the perspective of multi-criteria decision analysis, Du et al. [2] introduced three kinds of relations among agents into multi-criteria large-scale group decision making in linguistic context, obtained the coalition of decision models and the weights of criteria and finally proposed a conflict coordination and feedback mechanism to solve conflicts.

Most of the existing solution strategies resolve conflicts by selecting an optimal subset of issues most agents agree on. However, if conflicts have not yet reached a serious level, some compromises can be made through third-party mediation to promote cooperation between the agents. For example, Iran and Saudi Arabia have resumed diplomatic relations after China-mediated talks. Therefore, this paper considers changing the attitudes of agents to make them reach a consensus. Since such changes will bring a certain cost, it is necessary to measure the costs to determine an optimal change. In addition, the cost may also change as trust degree is introduced and changed. Consequently, conflict resolution strategies that only consider costs, without considering trust, are often ineffective.

In order to solve the above problems, this paper introduces an L-fuzzy three-way concept lattice (L-3WCL) [27], which is mainly used in knowledge representation [28] and fuzzy three-way concept lattices reduction [29], to conflict analysis and resolves the conflict using the dynamic programming method of the knapsack problem [30–32]. As a result, a hybrid conflict analysis model is developed. The model first employs L-fuzzy three-way concept to capture the issues on which agents commonly agree and the issues which they commonly oppose, and then measures the relative inconsistency of a set of agents. By relative inconsistency, we identify the state of a set of agents and categorize the issues into different types, which may help us find the causes of conflict. To facilitate cooperation between agents, we act as a third-party mediator, seeking to compromise between the agents at minimal cost to reach a consensus. We model this problem as a knapsack problem, which is a combinatorial optimization problem that can be solved using dynamic programming method. Finally, we propose an optimal attitude change strategy based on dynamic programming and solve conflicts with minimum cost. Furthermore, we verify the effectiveness of this strategy in intelligent decision-making instances such as business decision making.

Section 2 will briefly review the models in [5,25]. Section 3 analyzes the shortcomings of the fuzzy-concept-lattice-based conflict analysis model and proposes an L-3WCL-based conflict analysis model. Section 4 develops an optimal attitude change strategy and illustrates the effectiveness of the strategy with a case study. Finally, Section 5 concludes the paper with an outlook.

2. Related Works

This section briefly reviews some basic conflict analysis models, including the Pawlak's conflict analysis model and the fuzzy-concept-lattice-based conflict analysis model.

2.1. Pawlak's Conflict Analysis Model

Pawlak's conflict analysis model is built on conflict analysis information system (CAIS).

Definition 1 ([5]). *A CAIS is a triple $K = (U, V, R)$, where $U = \{x_1, x_2, \dots, x_n\}$ is a non-empty finite set of agents, $V = \{a_1, a_2 \dots, a_m\}$ is a non-empty finite set of issues, and $R : U \times V \to \{-1, 0, 1\}$ is defined by*

$$R(x,a) = \begin{cases} 1, & x \text{ is positive towards } a \\ 0, & x \text{ is neutral towards } a \\ -1, & x \text{ is negative towards } a \end{cases} \tag{1}$$

where $x \in U$ and $a \in V$.

In order to determine the relationship between two agents, Pawlak the defined auxiliary function [5].

Definition 2 ([5]). *Let $K = (U, V, R)$ be a CAIS and $a \in V$. Define the auxiliary function ϕ_a with regard to a for x_i, $x_j \in U$ as*

$$\phi_a(x_i, x_j) = \begin{cases} 1, & \text{if } R(x_i, a) * R(x_j, a) = 1 \text{ or } x_i = x_j \\ 0, & \text{if } R(x_i, a) * R(x_j, a) = 0 \text{ and } x_i \neq x_j \\ -1, & \text{if } R(x_i, a) * R(x_j, a) = -1. \end{cases} \tag{2}$$

By Definition 2, $\phi_a(x_i, x_j) = 1$ means that both agents x_i and x_j have the same attitude towards issue a; $\phi_a(x_i, x_j) = -1$ means that agents x_i and x_j have different attitudes towards a; $\phi_a(x_i, x_j) = 0$ means that at least one agent has a neutral attitude towards a.

Based on the auxiliary function, the distance between agents can be measured.

Definition 3 ([5]). *Let $K = (U, V, R)$ be a CAIS and $B \subseteq V$. Define the distance function with regard to B for x_i, $x_j \in U$ as*

$$\rho_B(x_i, x_j) = \frac{1}{|B|} \sum_{a \in B} \frac{1 - \phi_a(x_i, x_j)}{2}. \tag{3}$$

According to the distance function, Pawlak divided the relationships between agents into three groups [5].

Definition 4 ([5]). *Let $K = (U, V, R)$ be a CAIS and $B \subseteq V$. For x_i, $x_j \in U$,*

1. *If $\rho_B(x_i, x_j) > 0.5$, then x_i, x_j are called in a conflict state;*
2. *If $\rho_B(x_i, x_j) = 0.5$, then x_i, x_j are called in a neutral state;*
3. *If $\rho_B(x_i, x_j) < 0.5$, then x_i, x_j are called in an alliance state.*

2.2. Fuzzy-Concept-Lattice-Based Conflict Analysis Model

Zhi et al. [25] employed fuzzy concepts to characterize the uncertainty of agents and proposed a fuzzy-concept-lattice-based conflict analysis model.

Definition 5 ([33])**.** *A complete residuated lattice (CRL) is a seven-tuple* $(L, \wedge, \vee, \otimes, \rightarrow, 0, 1)$ *with*

- $(L, \wedge, \vee, 0, 1)$ *is a complete lattice;*
- $(L, \otimes, 1)$ *is a commutative monoid;*
- $\otimes$ *and* $\rightarrow$ *are adjoint, i.e., for* x*,* y*,* $z \in L$*,*

$$x \otimes y \leq z \Longleftrightarrow x \leq y \rightarrow z. \tag{4}$$

Definition 6 ([34])**.** *Let* $(L, \wedge, \vee, \otimes, \rightarrow, 0, 1)$ *be a CRL. For* $a \in L$*, define* $\neg : L \rightarrow L$ *as* $\neg a = a \rightarrow 0$ *and call* $\neg$ *a negation operation of* L*.*

CRL has the following properties.

Proposition 1 ([34,35])**.** *If* L *is a CRL and* $\neg$ *is a negation operation of* L*, then the following conclusions hold*

1. $x \leq y \Longleftrightarrow ((x \rightarrow y) = 1)$;
2. $(x \rightarrow x) = (x \rightarrow 1) = (0 \rightarrow x) = 1$, $(1 \rightarrow x) = x$;
3. $\neg 1 = 0$, $\neg 0 = 1$.

CRL generalizes various types of truth structures. For example, the Łukasiewicz implication operator is:

$$x \rightarrow y = ((1 - x + y) \wedge 1) \tag{5}$$

where $x, y \in [0, 1]$. For a universe U and a CRL L, $S : U \rightarrow L$ is called an L-fuzzy set on U. The set of all the fuzzy sets on U is denoted by L^U.

Definition 7 ([36,37])**.** *A fuzzy formal context is a triple* $K = (U, V, R)$*, where* U *is the set of objects,* V *is the set of attributes, and* $R \in L^{U \times V}$ *is a fuzzy relation between* U *and* V*.*

Based on fuzzy formal context, fuzzy concepts can then be defined.

Definition 8 ([36,37])**.** *Let* $K = (U, V, R)$ *be a fuzzy formal context. For* $A \in L^U$ *and* $B \in L^V$*,* $\uparrow: L^U \rightarrow L^V$ *and* $\downarrow: L^V \rightarrow L^U$ *can be defined as*

$$A^{\uparrow}(y) = \bigwedge_{x \in U} (A(x) \rightarrow R(x, y)), y \in V \tag{6}$$

$$B^{\downarrow}(x) = \bigwedge_{y \in V} (B(y) \rightarrow R(x, y)), x \in U \tag{7}$$

If $A^{\uparrow} = B$ *and* $B \in L^V$*, then* (A, B) *is called a fuzzy concept in* K*, where* A *and* B *are called the extent and intent of* (A, B)*, respectively. For all the fuzzy concepts in* K*, the order defined by* $(A_1, B_1) \leq (A_2, B_2) \Longleftrightarrow A_1 \subseteq A_2 \Longleftrightarrow B_2 \subseteq B_1$ *forms a complete lattice* $F(K)$*, called the fuzzy concept lattice of* K*.*

Definition 9 ([25])**.** *Let* $K = (U, V, R)$ *be a CAIS. The inconsistency measure for* $X \in L^U$ *is defined as*

$$m(X) = \frac{1}{|V|} \sum_{y \in V} (1 - X^{\uparrow}(y)) \tag{8}$$

Given two thresholds t_1 *and* t_2 *such that* $0 \leq t_1 \leq t_2 \leq 1$*, for* $X \in L^U$

1. *If* $m(X) > t_2$*, then* X *is called in a conflict state;*

2. *If $t_1 \leq m(X) \leq t_2$, then X is called in a neutral state;*
3. *If $m(X) < t_1$, then X is called in an alliance state.*

If a set X of agents is in a conflict state, then we call X a conflict set of agents; if X is in a neutral state, then we call X a neutral set of agents; and if X is in an alliance state, then we call X an alliance set of agents.

In this paper, we refer to the conflict analysis model proposed in [25] as the Zhi model.

3. *L*-3WCL Based Conflict Analysis Model

The Zhi model can effectively analyze conflict situations with the uncertainty of agents in trustworthiness, but it may lead to misjudgment of conflicts. Example 1 illustrates the cause of misjudgment.

Example 1. *Let $K = (U, V, R)$ be a CAIS, where $U = \{x_1, x_2\}$, $V = \{y\}$ and $R(x_1, y) = R(x_2, y) = 0$.*

For K, since the attitudes of agents x_1 and x_2 towards issue y are both negative, they reach a consensus on y; therefore, the inconsistency should be 0. However, according to the Zhi model, when using the Łukasiewicz implication, for $X = \{1/x_1, 1/x_2\} \in L^U$, according to Definitions 8 and 9, $X^{\uparrow}(y) = 0$ and $m(X) = 1$. Thus, the inconsistency of agents x_1 and x_2 on y is 1, indicating that the attitudes of the two agents are inconsistent. Similar problems also exist in the Zhi model when using other commonly used implication operators.

From Example 1, we can see that the Zhi model misjudges the conflict. This is due to the fact that the Zhi model only considers the common agreement of agents as consistency and the common opposition of agents as inconsistency. Since *L*-3WCL [27] is able to characterize both the attributes that objects commonly possess and the attributes that they commonly do not possess, we will utilize *L*-3WCL to describe the issues that agents commonly agree on and commonly oppose, and propose a hybrid conflict analysis method based on *L*-3WCL.

Definition 10 ([27])**.** *For a fuzzy formal context $K = (U, V, R)$, define $\uparrow_T$: $L^U \to L^V \times L^V$ and $\downarrow_T$: $L^V \times L^V \to L^U$ as:*

$$X^{\uparrow_T}(y) = (X^+(y), X^-(y)) \tag{9}$$

$$(Y^+, Y^-)^{\downarrow_T}(x) = \bigwedge_{y \in V} (Y^+(y) \to R(x, y)) \wedge \bigwedge_{y \in V} (Y^-(y) \to \neg R(x, y)), \tag{10}$$

where $X \in L^U$, $y \in V$, $(Y^+, Y^-) \in L^V \times L^V$, $x \in U$, and

$$X^+(y) = \bigwedge_{x \in U} (X(x) \to R(x, y)) \tag{11}$$

$$X^-(y) = \bigwedge_{x \in U} (X(x) \to \neg R(x, y)). \tag{12}$$

When $X^{\uparrow_T} = (Y^+, Y^-)$ and $(Y^+, Y^-)^{\downarrow_T} = X$, $(X, (Y^+, Y^-))$ is called an object-derived L-fuzzy three-way concept and X and (Y^+, Y^-) are called the extent and intent of $(X, (Y^+, Y^-))$, respectively. All the fuzzy concepts of K form a complete lattice $ATWL(K)$, called the L-3WCL of K, defined as

$$(X_1, (Y_1^+, Y_1^-)) \leq (X_2, (Y_2^+, Y_2^-)) \Longleftrightarrow X_1 \subseteq X_2 \Longleftrightarrow (Y_2^+, Y_2^-) \subseteq (Y_1^+, Y_1^-). \tag{13}$$

A CAIS can be regarded as a fuzzy formal context, where $R(x, y)$ represents the attitude of the agent x towards the issue y, and $X(x)$ represents the trust degree of x. By Definition 10, $X^+(y)$ represents the consistency degree of X in agreeing with the issue y; $X^-(y)$ represents the consistency degree of X in opposing y. Since the larger the value of

$R(x,y)$ the higher the degree of agreement, when $X(x)$ remains constant, $X^+(y)$ increases with the increase of $R(x,y)$; in other words, $X^+(y)$ is the agreement consistency. Similarly, since the larger the value of $\neg R(x,y)$ the higher the degree of opposition, when $X(x)$ remains constant, $X^-(y)$ increases with the increase of $\neg R(x,y)$, and therefore $X^+(y)$ is the opposition consistency.

Similarly, in Definition 10, $Y^+(y)$ denotes the agreement consistency with the issue y and $Y^-(y)$ denotes the opposition consistency with y. If $Y^+(y) = X^+(y)$ and $Y^-(y) = X^-(y)$, i.e., $Y^+(y)$ is the agreement consistency of X with y and $Y^-(y)$ is the opposition consistency of X with y, $(Y^+, Y^-)^{\downarrow_T}(x)$ returns a new set $(Y^+, Y^-)^{\downarrow_T} \in L^U$ such that $X(x) \leq (Y^+, Y^-)^{\downarrow_T}(x) = X^{\uparrow_T \downarrow_T}(x)$ for any $x \in U$ (see [27]), implying that one can increase the trust degree of x while keeping the same agreement consistency and opposition consistency because $X^{\uparrow_T \downarrow_T \uparrow_T}(x) = X^{\uparrow_T}(x)$ (see [27]).

In our case, we require L to be {0, 0.5, 1}. In particular, we have the following conclusions.

1. Both $X^+(y)$ and $X^-(y)$ are also in {0, 0.5, 1}. If $X^+(y) = 1$, then the set X of agents completely reach a consensus in agreeing with the issue y; if $X^+(y) = 0.5$, then X partially reach a consensus in agreeing with y; and if $X^+(y) = 0$, then X does not reach a consensus in agreeing with y. Similar analysis applies to $X^-(y)$.
2. The value of $(Y^+, Y^-)^{\downarrow_T}(x)$ also falls in $\{0, 0.5, 1\}$. In this case, if $(Y^+, Y^-)^{\downarrow_T}(x) = 1$, then the agent x is fully trusted. If $(Y^+, Y^-)^{\downarrow_T}(x) = 0.5$, then x is partially trusted; if $(Y^+, Y^-)^{\downarrow_T}(x) = 0$, then x is not trusted.

Example 2. *Since the Gödel implication operator is not suitable for building fuzzy logic systems [38], we will choose the Łukasiewicz implication operator for illustration. In this case, for $X \in L^U$ and $y \in V$, we have*

$$X^+(y) = \bigwedge_{x \in U} (X(x) \rightarrow R(x,y)) = \bigwedge_{x \in U} ((1 - X(x) + R(x,y)) \wedge 1) \tag{14}$$

$$X^-(y) = \bigwedge_{x \in U} (X(x) \rightarrow \neg R(x,y)) = \bigwedge_{x \in U} ((1 - X(x) + \neg R(x,y)) \wedge 1). \tag{15}$$

When all the agents $x \in U$ are trustworthy, i.e., $X(x) = 1$, we have $X^+(y) = \wedge_{x \in U} R(x,y)$ and $X^-(y) = \wedge_{x \in U}(\neg R(x,y))$. This is reasonable because when all the agents are trustworthy, the consistency of X will completely depend on their attitudes.

If there is an agent $x \in U$ with $X(x) = 0$, then we have $X(x) \rightarrow R(x,y) = X(x) \rightarrow \neg R(x,y) = 1$, i.e., the values of $X^+(y)$ and $X^-(y)$ are not affected by $R(x,y)$. This is also reasonable because if the agent x is not trustworthy, their attitude can be ignored.

Based on the agreement consistency and opposition consistency, the comprehensive consistency $X^*(y)$ can be defined.

Definition 11. *Let $K = (U, V, R)$ be a CAIS and $X \in L^U$. The comprehensive consistency $X^*(y)$ of X on the issue $y \in V$ is defined as*

$$X^*(y) = X^+(y) \vee X^-(y). \tag{16}$$

According to Definition 11, we can see that

1. If $X^+(y) = 1$ and $X^-(y) = 1$, then the set X of agents reaches a complete consensus both on agreeing with y and on opposing y. In fact, when using the Łukasiewicz implication operator, for $X \in L^U$ and $y \in V$, if $X(x) = 0$ for all the agents $x \in U$, then we have $X^+(y) = 1$ and $X^-(y) = 1$, i.e., when the attitudes of all the agents are ignorable, one can conclude $X^+(y) = 1$ and $X^-(y) = 1$. Similarly, if for all the agents $x \in U$, $X(x) = 0.5$ implies $R(x,y) = 0.5$, then we have $X^+(y) = 1$ and $X^-(y) = 1$, i.e., when the trustworthiness and the attitudes of agents are uncertain, one can also

conclude $X^{+}(y) = 1$ and $X^{-}(y) = 1$. In both the cases, it can be considered either that X is unanimously agreeing on y or that X is unanimously opposing on y, and y is an alliance issue for X.

2. If $X^{+}(y) = 1$ and $X^{-}(y) \leq 0.5$, then X reaches a complete consensus on agreeing with y and does not reach a complete consensus on opposing y. In this case, the agents within X have achieved a unanimous agreement on y. Similarly, if $X^{+}(y) \leq 0.5$ and $X^{-}(y) = 1$, then X reaches a complete consensus on opposing y but does not reach a complete consensus on agreeing with y. It can be considered that the agents in X have achieved a unanimous opposition on y. In both the cases, X achieves a consensus, i.e., $X^{*}(y) = 1$, and y is an alliance issue for X.
3. If $X^{+}(y) = 0$ and $X^{-}(y) = 0$, then there must exist at least one pair of agents with opposite attitudes. Therefore, its comprehensive consistency is 0. In fact, when using the Łukasiewicz implication operator, if $X^{+}(y) = 0$, then there must exist $x_1 \in U$ such that $X(x_1) \to R(x_1, y) = 0$, which implies $R(x_1, y) = 0$. Similarly, if $X^{-}(y) = 0$, then there must exist $x_2 \in U$ such that $x_2 \in U$, which implies $\neg R(x_2, y) = 0$, yielding $R(x_2, y) = 1$. Hence, the attitudes of agents x_1 and x_2 towards y are opposite, indicating that X does not reach a consensus, i.e., $X^{*}(y) = 0$. At this point, y is a conflict issue of X.
4. If $X^{+}(y) = 0.5$ and $X^{-}(y) = 0$, then X reaches a partial consensus on agreeing with y but does not reach a consensus on opposing y. In this case, X only partially agrees on y. Similarly, if $X^{-}(y) = 0.5$ and $X^{+}(y) = 0.5$, then the agents reach a partial consensus on opposing y but does not reach a consensus on agreeing with y. Thus, X only partially agrees on y. If $X^{+}(y) = 0.5$ and $X^{-}(y) = 0.5$, X partially reaches a consensus on agreeing with and opposing y. This indicates that X reaches a partial consensus on y. In all the three cases, X only reaches a partial consensus on y, i.e., $X^{*}(y) = 0.5$, and y is a neutral issue of X.

In Example 1, because the Zhi model does not consider the opposition consistency, a discrepancy with the actual situation occurs when analyzing the consistency of agents. Definition 11 considers both the agreement and opposition consistency, leading to a more reasonable result than the Zhi model. For example, for the issue y in Example 1, two agents x_1 and x_2 have the agreement consistency $X^{+}(y) = 0$, the opposition consistency $X^{-}(y) = 1$, and the comprehensive consistency $X^{*}(y) = 1$, which is in line with the actual situation.

The relative inconsistency of a set of agents can be determined by the comprehensive consistency over all the issues.

Definition 12. *Let $K = (U, V, R)$ be a CAIS and $X \in L^{U}$. Define the relative inconsistency of X as*

$$m'(X) = \frac{1}{|V|} \sum_{y \in V} (1 - X^{*}(y)). \tag{17}$$

Let t_1 and t_2 be two thresholds such that $0 \leq t_1 \leq t_2 \leq 1$. For $X \in L^{U}$, if $m'(X) < t_1$, then X is called an alliance set of agents; if $t_1 \leq m'(X) \leq t_2$, then X is called a neutral set of agents; if $m'(X) > t_2$, then X is called a conflict set of agents.

The difference between the relative inconsistency (Equation (17)) and the inconsistency in the Zhi model (Equation (8)) is illustrated by Example 3.

Example 3. *Table 1 shows a CAIS $K = (U, V, R)$, where $U = \{x_1, x_2\}$ and $V = \{y_1, y_2, y_3, y_4, y_5, y_6\}$.*

Table 1. A CAIS for Example 3.

	y_1	y_2	y_3	y_4	y_5	y_6
x_1	0	0	0.5	1	1	1
x_2	0	0.5	0.5	1	0.5	0

If $X = \{1/x_1, 1/x_2\} \in L^U$, then according to the Zhi model, we have $X^{\uparrow}(y_1) = 0$, $X^{\uparrow}(y_2) = 0$, $X^{\uparrow}(y_3) = 0.5$, $X^{\uparrow}(y_4) = 1$, $X^{\uparrow}(y_5) = 0.5$, and $X^{\uparrow}(y_6) = 0$. By Equation (8) we can obtain $m(X) = 0.67$. According to the proposed model, we have $X^*(y_1) = 1$, $X^*(y_2) = 0.5$, $X^*(y_3) = 0.5$, $X^*(y_4) = 1$, $X^*(y_5) = 0.5$, and $X^*(y_6) = 0$, yielding $m'(X) = 0.42 < 0.67 = m(X)$. From the calculation process, it can be seen that the two models are consistent on the set $\{y_3, y_4, y_5, y_6\}$, but on the set $\{y_1, y_2\}$, the Zhi model considers that the set is completely inconsistent on y_1 and y_2 (i.e., $X^{\uparrow}(y_1) = 0$ and $X^{\uparrow}(y_2) = 0$), while the proposed model considers that the set is completely consistent on y_1 (i.e., $X^*(y_1) = 1$) and partially consistent on y_2 (i.e., $X^*(y_2) = 0.5$). Intuitively, the attitude of X on y_1 is consistent and therefore the calculation result of the proposed model is reasonable; for y_2, if the consistency of the attitudes of X towards this issue is considered to be 0, the same result can be obtained in the case of $R(x_1, y_2) = 0$ and $R(x_2, y_2) = 1$. In other words, the result of the Zhi model on y_2 cannot distinguish the two cases: 1. $R(x_1, y_2) = 0$ and $R(x_2, y_2) = 0.5$; 2. $R(x_1, y_2) = 0$ and $R(x_2, y_2) = 1$.

When considering the subset $V_1 = \{y_1, y_2, y_3, y_4, y_5\}$, by Equation (8), we can obtain $m(X) = 0.6$ and by Equation (17), we can obtain $m'(X) = 0.3$. Thus, when the issue y_6 is not taken into consideration, $m(X)$ will decrease from 0.67 to 0.6, while $m'(X)$ will decrease from 0.42 to 0.3. Obviously, the decrease of $m'(X)$ is greater than that of $m(X)$. This is because the Zhi model considers that X has conflicts not only on issue y_6, but also on issues y_1 and y_2, and thus removing issue y_6 does not eliminate the conflicts in X. The proposed model considers that X has the highest level of conflict on y_6 and thus removing y_6 results in a greater decrease in the inconsistency of X.

From Example 3, we can derive $m(X) \geq m'(X)$, a general relationship between Equations (8) and (17).

Theorem 1. *Let $K = (U, V, R)$ be a CAIS and $X \in L^U$. Then, we have $m'(X) \leq m(X)$.*

Proof of Theorem 1. For any $y \in V$, we have $X^*(y) = X^+(y) \vee X^-(y) \geq X^+(y) = X^{\uparrow}(y)$ and thus the conclusion holds by Definitions 11 and 12. □

Theorem 1 is obvious, because the proposed model captures both the agreement consistency and the opposition consistency of agents towards issues, while the Zhi model only captures the agreement consistency of agents towards issues. Thus, compared to the Zhi model, the proposed model reduces the inconsistencies of the sets of agents.

Theorem 2. *Let $K = (U, V, R)$ be a CAIS, and X, $X' \in L^U$. If $X \subseteq X'$, then we have $m'(X) \leq m'(X')$.*

Proof of Theorem 2. If $X \subseteq X'$, for any $u \in U$, then we have $X(u) \leq X'(u)$, and by the definition of $X^+(y)$ and $X^-(y)$, when $R(u, y)$ is constant, if $X(u)$ increases, then both $X^+(y)$ and $X^-(y)$ will decrease. Thus, we obtain $X^+(y) \geq X'^+(y)$ and $X^-(y) \geq X'^-(y)$. By Definitions 11 and 12, the conclusion holds. □

Theorem 2 shows that the more agents there are, the higher the probability of inconsistency. In fact, suppose that X, $X_1 \in L^U$ satisfy $X_1(u_1) \geq X(u_1)$, $X_1(u_2) = X(u_2)$, $X_1(u_3) = X(u_3)$, …, $X_1(u_{|U|}) = X(u_{|U|})$. One can easily obtain $X_1{}^*(y) \leq X^*(y)$ for any y and thus $m'(X_1) \geq m'(X)$. In other words, when the trust degree of agent u_1 increases from $X(u_1)$ to $X_1(u_1)$, since $X_1(u_1) \to R(u_1, y) \leq X(u_1) \to R(u_1, y)$, the importance of this agent's opinion in X increases, leading to a decrease in the agreement consistency

$X^+(y)$ or the opposition consistency $X^-(y)$, which in turn leads to an increase in the inconsistency of X. Similarly, suppose that $X_2 \in L^U$ satisfies $X_2(u_1) = X_1(u_1), X_2(u_2) \geq X_1(u_2)$, $X_2(u_3) = X_1(u_3), \ldots, X_2(u_{|U|}) = X_1(u_{|U|})$. One can easily obtain ${X_2}^*(y) \leq {X_1}^*(y)$ for any y and thus $m'(X_2) \geq m'(X_1)$, i.e., the increase of the trust degree in agent u_2 leads to the increase of the inconsistency in X. Continue the process and suppose that $X_{|U|} = X' \in L^U$ satisfies $X_{|U|}(u_1) = X_{|U|-1}(u_1)$, $X_{|U|}(u_2) = X_{|U|-1}(u_2)$, $X_{|U|}(u_3) = X_{|U|-1}(u_3)$, $\ldots, X_{|U|}(u_{|U|}) \geq X_{|U|-1}(u_{|U|})$. Then, one can easily obtain ${X_{|U|}}^*(y) \leq {X_{|U|-1}}^*(y)$ and thus $m'(X_{|U|}) \geq m'(X_{|U|-1})$. In summary, we have

$$m'(X') \geq m'(X_{|U|-1}) \geq m'(X_{|U|-2}) \geq \ldots \geq m'(X_1) \geq m'(X). \tag{18}$$

By Definition 12, one can determine the state of a set of agents. If the set is allied, one may be more concerned about whether the set is unanimously agreed on or opposed some issues; if it is neutral, then the set is partially agreed and one may be more concerned about the issues that the agents are partially agreed on and the issues that they oppose; if it is conflicted, then the set does not reach a consensus on some issues and one may have to analyze the reasons of the conflicts. For this purpose, one should further analyze issues according to their states.

Definition 13. *Let $K = (U, V, R)$ be a CAIS. For $X \in L^U$ and $y \in V$,*

1. *If $X^+(y) = 1$ and $X^-(y) \neq 1$, then y is called a unanimous agreement issue of X. The unanimous agreement set of issues of X is defined as*

$$\alpha^+(X) = \{y | X^+(y) = 1 \wedge X^-(y) \neq 1, y \in V\}; \tag{19}$$

2. *If $X^+(y) \neq 1$ and $X^-(y) = 1$, then y is called a unanimous opposition issue of X. The unanimous opposition set of issues of X is defined as*

$$\alpha^-(X) = \{y | X^+(y) \neq 1 \wedge X^-(y) = 1, y \in V\}; \tag{20}$$

3. *If $X^+(y) = 1$ and $X^-(y) = 1$, then y is called a unanimous issue of X. The unanimous set of issues of X is defined as*

$$\alpha(X) = \{y | X^+(y) = 1 \wedge X^-(y) = 1, y \in V\}; \tag{21}$$

4. *If $X^+(y) = 0.5$ and $X^-(y) = 0$, then y is called an agreement-neutral issue of X. The agreement-neutral set of issues of X is defined as*

$$\gamma^+(X) = \{y | X^+(y) = 0.5 \wedge X^-(y) = 0, y \in V\}; \tag{22}$$

5. *If $X^+(y) = 0$ and $X^-(y) = 0.5$, then y is called an opposition-neutral issue of X. The opposition-neutral set of issues of X is defined as*

$$\gamma^-(X) = \{y | X^+(y) = 0 \wedge X^-(y) = 0.5, y \in V\}; \tag{23}$$

6. *If $X^+(y) = 0.5$ and $X^-(y) = 0.5$, then y is called a completely neutral issue of X. The completely neutral set of issues of X is defined as*

$$\gamma(X) = \{y | X^+(y) = 0.5 \wedge X^-(y) = 0.5, y \in V\}; \tag{24}$$

7. *If $X^+(y) = 0$ and $X^-(y) = 0$, then y is called a conflict issue of X. The conflict set of issues of X is defined as*

$$\beta(X) = \{y | X^+(y) = 0 \wedge X^-(y) = 0, y \in V\}. \tag{25}$$

By Definition 13, the attitude of a set of agents on an issue can be derived. For $X \in L^U$ and $y \in V$, if $y \in \alpha^+(X)$, then X agrees unanimously on y; if $y \in \alpha^-(X)$, then X opposes unanimously y; if $y \in \alpha(X)$, as mentioned before, then either X agrees unanimously on y or X opposes unanimously y. Since X is unanimous on y when $y \in \alpha^+(X) \cup \alpha^-(X) \cup \alpha(X)$, these issues are collectively referred to as alliance issues. If $y \in \gamma^+(X)$, then X is neutral on agreeing on y; if $y \in \gamma^-(X)$, then X is neutral on opposing y; if $y \in \gamma(X)$, then X is neutral on either agreeing on or opposing y. Since X is neutral on y when $y \in \gamma^+(X) \cup \gamma^-(X) \cup \gamma(X)$, these issues are collectively referred to as neutral issues. If $y \in \beta(X)$, then X is in conflict on y, and these issues will be called conflict issues. Neutral and conflict issues are collectively referred to as non-coalition issues.

Theorem 3. *Let $K = (U, V, R)$ be a CAIS, and X, $X' \in L^U$. If $X \subseteq X'$, then*

1. $\alpha(X') \subseteq \alpha(X)$
2. $\beta(X) \subseteq \beta(X')$
3. $\alpha^+(X') \subseteq \alpha^+(X) \cup \alpha(X)$
4. $\alpha^-(X') \subseteq \alpha^-(X) \cup \alpha(X)$
5. $\gamma^+(X) \subseteq \gamma^+(X') \cup \beta(X')$
6. $\gamma^-(X) \subseteq \gamma^-(X') \cup \beta(X')$
7. $\gamma(X) \subseteq \gamma(X') \cup \gamma^+(X') \cup \gamma^-(X') \cup \beta(X')$
8. $\gamma(X') \subseteq \gamma(X) \cup \alpha^+(X) \cup \alpha^-(X) \cup \alpha(X)$.

Proof of Theorem 3. (1) If $y \in \alpha(X')$, then we have $X'^+(y) = 1$ and $X'^-(y) = 1$. Since $X \subseteq X'$, we obtain $X^+(y) \geq X'^+(y) = 1$ and $X^-(y) \geq X'^-(y) = 1$, i.e., $X^+(y) = 1$ and $X^-(y) = 1$, and hence $y \in \alpha(X)$.

(2) If $y \in \beta(X)$, then we have $X^+(y) = 0$ and $X^-(y) = 0$. Since $X \subseteq X'$, we obtain $X'^+(y) \leq X^+(y) = 0$ and $X'^-(y) \leq X^-(y) = 0$, i.e., $X'^+(y) = 0$ and $X'^-(y) = 0$, and hence $y \in \beta(X')$.

(3) If $y \in \alpha^+(X')$, then we have $X'^+(y) = 1$ and $X'^-(y) \neq 1$. Since $X \subseteq X'$, we obtain $X^+(y) \geq X'^+(y) = 1$ and $X^-(y) \geq X'^-(y) \neq 1$, i.e., $X^+(y) = 1$ and $X^-(y) \geq 0$. If $X^+(y) = 1$ and $X^-(y) \leq 0.5$, then we have $y \in \alpha^+(X)$; if $X^+(y) = 1$ and $X^-(y) = 1$, then we have $y \in \alpha(X)$.

The proofs of (4), (5), (6), (7), and (8) are similar. □

Theorem 3 (1) states that the more agents there are, the fewer issues they reach a consensus on. Theorem 3 (2) states that the more agents there are, the more conflict issues there are. Theorem 3 (3) and Theorem 3 (4) state that the unanimous agreement issues or unanimous opposition issues may become the unanimous issues when there are fewer agents. Theorem 3 (5) and Theorem 3 (6) state that the agreement-neutral issues or the opposition-neutral issues may become conflict issues when there are more agents. Theorem 3 (7) states that the completely neutral issues may become the agreement-neutral issues or the opposition-neutral issues or the conflict issues when there are more agents. Theorem 3 (8) states that the completely neutral issues may become the unanimous agreement issues or the unanimous opposition issues or the unanimous issues when there are fewer agents.

Example 4. *Table 2 shows a CAIS $K = (U, V, R)$, where $U = \{x_1, x_2\}$ and $V = \{y_1, y_2, y_3\}$.*

Table 2. A CAIS for Example 4.

	y_1	y_2	y_3
x_1	0.5	1	0.5
x_2	1	0	0.5

For $X = \{0.5/x_1\}$ and $X' = \{0.5/x_1, 0.5/x_2\}$, according to Definition 10, we can obtain $X^+(y_1) = 1$, $X^-(y_1) = 1$, $X^+(y_2) = 1$, $X^-(y_2) = 0.5$, $X^+(y_3) = 1$, and $X^-(y_3) = 1$;

therefore, $\alpha(X) = \{y_1, y_3\}$, $\alpha^+(X) = \{y_2\}$. Similarly, we have $X'^+(y_1) = 1$, $X'^-(y_1) = 0.5$, $X'^+(y_2) = 0.5$, $X'^-(y_2) = 0.5$, $X^+(y_3) = 1$, and $X^-(y_3) = 1$; therefore, $\alpha^+(X') = \{y_1\}$, $\alpha(X') = \{y_3\}$ and $\gamma(X') = \{y_2\}$. When the number of agents increases, i.e., X becomes X', according to Theorem 3 (1), the unanimous issues become fewer, from $\{y_1, y_3\}$ to $\{y_3\}$. When the number of agents decreases, i.e., X' becomes X, according to Theorem 3 (3), the unanimous agreement issue y_1 become the unanimous issue; according to Theorem 3 (8), the completely neutral issues y_2 become the unanimous agreement issues. In addition, it can be found that although $X \subseteq X'$, $\alpha^+(X)$ has no inclusion relation with $\alpha^+(X')$.

According to the above discussion, for a given set X of agents, the state of X and its attitude towards each issue can be determined. When the trust degrees of agents changes frequently; however, the workload of performing conflict analysis may be huge. It is easy to conclude that for a set of agents containing n agents, the number of sets of agents under $|L|$ is $|L|^n$; even when $L = \{0, 0.5, 1\}$, the number of sets of agents will be 3^n. In fact, the basic nature of L-3WCL states that sets of agents with different trust degrees may produce the same conflict analysis results. Specially, for the sets X_1 and X_2 of agents with different trust degrees, if $X_1^+ = X_2^+$ and $X_1^- = X_2^-$, then their conflict analysis results are exactly the same. In other words, the number of conflict analysis results is equal to the number of L-fuzzy three-way concepts, much smaller than $|L|^n$. Therefore, in order to reduce the workload, we can establish L-3WCL to calculate all possible conflict analysis results, and when trust degrees change, we only need to find the result corresponding to X. In addition, by analyzing L-3WCL, we can also observe the change of conflict analysis results when trust degree changes.

In $ATWL(K)$, the extent of a concept represent a set of agents with different trust degrees, and the intent represent the agreement degree and opposition degree. For any set X of agents, the corresponding fuzzy three-way concept $(X^{\uparrow_T\downarrow_T}, X^{\uparrow_T\downarrow_T\uparrow_T})$ can be found in $ATWL(K)$, and the relative inconsistency $m'(X^{\uparrow_T\downarrow_T})$ and sets of issues $\alpha^+(X^{\uparrow_T\downarrow_T})$, $\alpha^-(X^{\uparrow_T\downarrow_T})$, $\alpha(X^{\uparrow_T\downarrow_T})$, $\gamma^+(X^{\uparrow_T\downarrow_T})$, $\gamma^-(X^{\uparrow_T\downarrow_T})$, $\gamma(X^{\uparrow_T\downarrow_T})$, and $\beta(X^{\uparrow_T\downarrow_T})$ can be derived by Definitions 12 and 13.

For three concepts $(X_1, X_1^{\uparrow_T})$, $(X_2, X_2^{\uparrow_T})$ and $(X_3, X_3^{\uparrow_T})$ in $ATWL(K)$, if $(X_3, X_3^{\uparrow_T}) \subseteq (X_1, X_1^{\uparrow_T}) \subseteq (X_2, X_2^{\uparrow_T})$, then $X_3 \subseteq X_1 \subseteq X_2$. According to Theorems 1 and 2, if the trust degrees of X_1 raises to X_2, the conflict issues in X_1 must be the conflict issues in X_2, the opposition-neutral issues must be the opposition-neutral issues or the conflict issues in X_2, the agreement-neutral issues must be the agreement-neutral issues or the conflict issues in X_2, the complete neutral issues must be the complete neutral issues, the opposition-neutral issues, and the agreement-neutral issues or the conflict issues in X_2. If the trust degrees of X_1 decreases to X_3, then the unanimous issues in X_1 must be the unanimous issues in X_3, the unanimous agreement issues must be the unanimous agreement issues or unanimous issues in X_3, the unanimous opposition issues must be the unanimous opposition issues or the unanimous issues in X_3, the completely neutral issues must be the completely neutral issues, the unanimous agreement issues, the unanimous opposition issues, or the unanimous issues in X_3.

In the following, we illustrate the L-3WCL based conflict analysis model in Example 5.

Example 5. *Consider the situation in which some agents are discussing some strategies for the development of an enterprise. Since the circumstances and the interests of agents may be different, there may be some conflicts among agents. Assume that three agents and four development strategies constitute a CAIS $K = (U, V, R)$, where $U = \{x_1, x_2, x_3\}$ and $V = \{y_1, y_2, y_3, y_4\}$ with y_1 representing talent introduction, y_2 representing project management, y_3 representing marketing and sales, and y_4 representing technology development. The attitude of each agent towards each issue is shown in Table 3. For example, agent x_1 has a positive attitude towards y_1, agent x_2 has a negative attitude towards y_1, and agent x_3 has a neutral attitude towards y_1.*

Table 3. A CAIS for Example 5.

	y_1	y_2	y_3	y_4
x_1	1	0	0.5	0
x_2	0	0.5	0.5	1
x_3	0.5	1	0	0

Since agents have different discourse powers, we adopt three levels of analysis, i.e., agent has very large discourse power, agent has relatively small discourse power, and agent has no discourse power. For example, when the discourse power of agent x_1 is relatively small, the discourse power of agent x_2 is very large, and agent x_3 has no discourse power, we represent it as $\{0.5/x_1, 1/x_2, 0/x_3\}$. When Łukasiewicz implication operator is used, $ATWL(K)$ of K is shown in Figure 1.

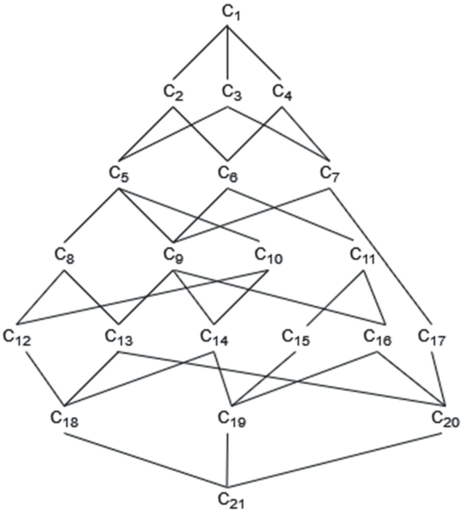


Figure 1. $ATWL(K)$.

The concepts in Figure 1 and the corresponding conflict analysis results are shown in Table 4.

In Table 4, for the concepts with at least two agents in the extents, the conflict analysis results obtained are the consistency of different agents; for the concepts with only one agent in the extents, the conflict analysis results obtained are the consistency of the agent for different issues; for the concept having no agents in the extent, the conflict analysis results are not meaningful.

The conflict analysis result of a set of agents can be used to infer the conflict analysis results of other sets of agents. For example, the conflict issue of $X_4 = \{0.5/x_1, 1/x_2, 1/x_3\}$ is y_4; according to Theorem 3, we can infer that issue y_4 must also be a conflict issue of $X_1 = \{1/x_1, 1/x_2, 1/x_3\}$; the opposition-neutral issues y_1 and y_3 of X_4 must be the opposition-neutral issues or the conflict issues of X_1; and the agreement-neutral issue y_2 of X_4 must be an agreement-neutral issue or a conflict issue of X_1.

The extents of the concepts in Table 4 contain only some sets in L^U and other sets of agents can be described by the corresponding concepts in $ATWL(K)$, as shown in Table 5.

Combining Tables 4 and 5, all the conflict analysis results of all sets of agents can be calculated. For example, in Table 5, we can see that for the set of agents $X = \{0.5/x_1, 1/x_3\}$, i.e., x_1 has a relatively large discourse power, x_3 has a very large discourse power, and x_2 has no discourse power, the corresponding concept can be obtained as $(\{0.5/x_1, 0.5/x_2, 1/x_3\}, (\{0.5/y_1, 0.5/y_2\}, \{0.5/y_1, 1/y_3, 0.5/y_4\}))$, and $\alpha^+(X) = \varnothing$, $\alpha^-(X) = \{y_3\}$, $\alpha(X) = \varnothing$, $\gamma^+(X) = \{y_2\}$, $\gamma^-(X) = \{y_4\}$, $\gamma(X) = \{y_1\}$, and $\beta(X) = \varnothing$. This indicates that the agents in X reach a complete consensus on opposing the marketing and sales issue y_3, a partial consensus on agreeing on the project management issue y_2 and opposing the technology development issue y_4, and a partial *consensus on the talent introduction issue y_1. Since $m'(X) = 0.375$, the conflict between agents is small.*

Table 4. Conflict analysis results.

$(X, (X^+, X^-))$	$\alpha^+(X)$	$\alpha^-(X)$	$\alpha(X)$	$\gamma^+(X)$	$\gamma^-(X)$	$\gamma(X)$	$\beta(X)$	$m'(X)$
$C_1 : (\{\frac{1}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}\}, (\varnothing, \{\frac{0.5}{y_3}\}))$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\{y_3\}$	$\varnothing$	$\{y_1, y_2, y_4\}$	0.875
$C_2 : (\{\frac{1}{x_1}, \frac{1}{x_2}, \frac{0.5}{x_3}\}, (\{\frac{0.5}{y_3}\}, \{\frac{0.5}{y_2}, \frac{0.5}{y_3}\}))$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\{y_2\}$	$\{y_3\}$	$\{y_1, y_4\}$	0.75
$C_3 : (\{\frac{1}{x_1}, \frac{0.5}{x_2}, \frac{1}{x_3}\}, (\{\frac{0.5}{y_1}\}, \{\frac{0.5}{y_3}, \frac{0.5}{y_4}\}))$	$\varnothing$	$\varnothing$	$\varnothing$	$\{y_1\}$	$\{y_3, y_4\}$	$\varnothing$	$\{y_2\}$	0.625
$C_4 : (\{\frac{0.5}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}\}, (\{\frac{0.5}{y_2}\}, \{\frac{0.5}{y_1}, \frac{0.5}{y_3}\}))$	$\varnothing$	$\varnothing$	$\varnothing$	$\{y_2\}$	$\{y_1, y_3\}$	$\varnothing$	$\{y_4\}$	0.625
$C_5 : (\{\frac{1}{x_1}, \frac{0.5}{x_2}, \frac{0.5}{x_3}\}, (\{\frac{0.5}{y_1}, \frac{0.5}{y_3}\}, \{\frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\}))$	$\varnothing$	$\varnothing$	$\varnothing$	$\{y_1\}$	$\{y_2, y_4\}$	$\{y_3\}$	$\varnothing$	0.5
$C_6 : (\{\frac{0.5}{x_1}, \frac{1}{x_2}, \frac{0.5}{x_3}\}, (\{\frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\}, \{\frac{0.5}{y_1}, \frac{0.5}{y_2}, \frac{0.5}{y_3}\}))$	$\varnothing$	$\varnothing$	$\varnothing$	$\{y_4\}$	$\{y_1\}$	$\{y_2, y_3\}$	$\varnothing$	0.5
$C_7 : (\{\frac{0.5}{x_1}, \frac{0.5}{x_2}, \frac{1}{x_3}\}, (\{\frac{0.5}{y_1}, \frac{0.5}{y_2}\}, \{\frac{0.5}{y_1}, \frac{1}{y_3}, \frac{0.5}{y_4}\}))$	$\varnothing$	$\{y_3\}$	$\varnothing$	$\{y_2\}$	$\{y_4\}$	$\{y_1\}$	$\varnothing$	0.375
$C_8 : (\{\frac{1}{x_1}, \frac{0.5}{x_3}\}, (\{\frac{1}{y_1}, \frac{0.5}{y_3}\}, \{\frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{1}{y_4}\}))$	$\{y_1\}$	$\{y_4\}$	$\varnothing$	$\varnothing$	$\{y_2\}$	$\{y_3\}$	$\varnothing$	0.25
$C_9 : (\{\frac{0.5}{x_1}, \frac{0.5}{x_2}, \frac{0.5}{x_3}\}, (\{\frac{0.5}{y_1}, \frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\}, \{\frac{0.5}{y_1}, \frac{0.5}{y_2}, \frac{1}{y_3}, \frac{0.5}{y_4}\}))$	$\varnothing$	$\{y_3\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\{y_1, y_2, y_4\}$	$\varnothing$	0.375
$C_{10} : (\{\frac{1}{x_1}, \frac{0.5}{x_2}\}, (\{\frac{0.5}{y_1}, \frac{0.5}{y_3}\}, \{\frac{1}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\}))$	$\varnothing$	$\{y_2\}$	$\varnothing$	$\{y_1\}$	$\{y_4\}$	$\{y_3\}$	$\varnothing$	0.375
$C_{11} : (\{\frac{1}{x_2}, \frac{0.5}{x_3}\}, (\{\frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\}, \{\frac{1}{y_1}, \frac{0.5}{y_2}, \frac{0.5}{y_3}\}))$	$\varnothing$	$\{y_1\}$	$\varnothing$	$\{y_4\}$	$\varnothing$	$\{y_2, y_3\}$	$\varnothing$	0.375
$C_{12} : (\{\frac{1}{x_1}\}, (\{\frac{1}{y_1}, \frac{0.5}{y_3}\}, \{\frac{1}{y_2}, \frac{0.5}{y_3}, \frac{1}{y_4}\}))$	$\{y_1\}$	$\{y_2, y_4\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\{y_3\}$	$\varnothing$	0.125
$C_{13} : (\{\frac{0.5}{x_1}, \frac{0.5}{x_3}\}, (\{\frac{1}{y_1}, \frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\}, \{\frac{0.5}{y_1}, \frac{0.5}{y_2}, \frac{1}{y_3}, \frac{1}{y_4}\}))$	$\{y_1\}$	$\{y_3, y_4\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\{y_2\}$	$\varnothing$	0.125
$C_{14} : (\{\frac{0.5}{x_1}, \frac{0.5}{x_2}\}, (\{\frac{0.5}{y_1}, \frac{0.5}{y_2}, \frac{1}{y_3}, \frac{0.5}{y_4}\}, \{\frac{0.5}{y_1}, \frac{1}{y_2}, \frac{1}{y_3}, \frac{0.5}{y_4}\}))$	$\varnothing$	$\{y_2\}$	$\{y_3\}$	$\varnothing$	$\varnothing$	$\{y_1, y_4\}$	$\varnothing$	0.25
$C_{15} : (\{\frac{1}{x_2}\}, (\{\frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{1}{y_4}\}, \{\frac{1}{y_1}, \frac{0.5}{y_2}, \frac{0.5}{y_3}\}))$	$\{y_4\}$	$\{y_1\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\{y_2, y_3\}$	$\varnothing$	0.25
$C_{16} : (\{\frac{0.5}{x_2}, \frac{0.5}{x_3}\}, (\{\frac{0.5}{y_1}, \frac{1}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\}, \{\frac{1}{y_1}, \frac{0.5}{y_2}, \frac{1}{y_3}, \frac{0.5}{y_4}\}))$	$\{y_2\}$	$\{y_1, y_3\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\{y_4\}$	$\varnothing$	0.125
$C_{17} : (\{\frac{1}{x_3}\}, (\{\frac{0.5}{y_1}, \frac{1}{y_2}\}, \{\frac{0.5}{y_1}, \frac{1}{y_3}, \frac{1}{y_4}\}))$	$\{y_2\}$	$\{y_3, y_4\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\{y_1\}$	$\varnothing$	0.125
$C_{18} : (\{\frac{0.5}{x_1}\}, (\{\frac{1}{y_1}, \frac{0.5}{y_2}, \frac{1}{y_3}, \frac{0.5}{y_4}\}, \{\frac{0.5}{y_1}, \frac{1}{y_2}, \frac{1}{y_3}, \frac{1}{y_4}\}))$	$\{y_1\}$	$\{y_2, y_4\}$	$\{y_3\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	0
$C_{19} : (\{\frac{0.5}{x_2}\}, (\{\frac{0.5}{y_1}, \frac{1}{y_2}, \frac{1}{y_3}, \frac{1}{y_4}\}, \{\frac{1}{y_1}, \frac{1}{y_2}, \frac{1}{y_3}, \frac{0.5}{y_4}\}))$	$\{y_4\}$	$\{y_1\}$	$\{y_2, y_3\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	0
$C_{20} : (\{\frac{0.5}{x_3}\}, (\{\frac{1}{y_1}, \frac{1}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\}, \{\frac{1}{y_1}, \frac{0.5}{y_2}, \frac{1}{y_3}, \frac{1}{y_4}\}))$	$\{y_2\}$	$\{y_3, y_4\}$	$\{y_1\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	0
$C_{21} : (\varnothing, (\{\frac{1}{y_1}, \frac{1}{y_2}, \frac{1}{y_3}, \frac{1}{y_4}\}, \{\frac{1}{y_1}, \frac{1}{y_2}, \frac{1}{y_3}, \frac{1}{y_4}\}))$	$\varnothing$	$\varnothing$	$\{y_1, y_2, y_3, y_4\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	0

Table 5. Sets of agents and the corresponding concepts.

The Set X of Agents	The Corresponding Concepts $(X^{\uparrow_T\downarrow_T}, X^{\uparrow_T\downarrow_T\uparrow_T})$
$\{\frac{1}{x_1}, \frac{1}{x_2}\}$	$C_2 : (\{\frac{1}{x_1}, \frac{1}{x_2}, \frac{0.5}{x_3}\}, (\{\frac{0.5}{y_3}\}, \{\frac{0.5}{y_2}, \frac{0.5}{y_3}\}))$
$\{\frac{1}{x_1}, \frac{1}{x_3}\}$	$C_3 : (\{\frac{1}{x_1}, \frac{0.5}{x_2}, \frac{1}{x_3}\}, (\{\frac{0.5}{y_1}\}, \{\frac{0.5}{y_3}, \frac{0.5}{y_4}\}))$
$\{\frac{1}{x_2}, \frac{1}{x_3}\}$	$C_4 : (\{\frac{0.5}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}\}, (\{\frac{0.5}{y_2}\}, \{\frac{0.5}{y_1}, \frac{0.5}{y_3}\}))$
$\{\frac{0.5}{x_1}, \frac{1}{x_2}\}$	$C_6 :$ $(\{\frac{0.5}{x_1}, \frac{1}{x_2}, \frac{0.5}{x_3}\}, (\{\frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\}, \{\frac{0.5}{y_1}, \frac{0.5}{y_2}, \frac{0.5}{y_3}\}))$
$\{\frac{0.5}{x_2}, \frac{1}{x_3}\}$	$C_7 : (\{\frac{0.5}{x_1}, \frac{0.5}{x_2}, \frac{1}{x_3}\}, (\{\frac{0.5}{y_1}, \frac{0.5}{y_2}\}, \{\frac{0.5}{y_1}, \frac{1}{y_3}, \frac{0.5}{y_4}\}))$
$\{\frac{0.5}{x_1}, \frac{1}{x_3}\}$	$C_7 : (\{\frac{0.5}{x_1}, \frac{0.5}{x_2}, \frac{1}{x_3}\}, (\{\frac{0.5}{y_1}, \frac{0.5}{y_2}\}, \{\frac{0.5}{y_1}, \frac{1}{y_3}, \frac{0.5}{y_4}\}))$

When applying the Zhi model to Table 3, the conflict analysis results can be obtained, as shown in Table 6. The comparison with the proposed model is also shown in Table 6.

According to Table 6, among the 17 results obtained by the Zhi model, 16 results are different from those obtained by the proposed model, indicating that there is a 94% probability that the Zhi model is problematic. The proposed model reduces the inconsistency in 16 results compared with the Zhi model, as shown in Figure 2. According to Figure 2 and Table 6, it is clear that the inconsistency decreases as the number of agents decreases, which verifies the correctness of Theorem 2. It can be observed that the difference between the two models occurs when the opposition consistency $X^-(y)$ of X for an issue y is greater than the agreement consistency $X^+(y)$. This is because the consistency in the Zhi model is the agreement consistency, and the consistency in the proposed model is the maximum of the opposition consistency and the agreement consistency. By examining the disparity distribution of two inconsistencies which is shown in Figure 3, we can *observe that the disparities between the two inconsistencies are primarily situated in the interval* $[0.125, 0.375)$. *These variations in value are relatively minor. On the interval* $[0.375, 0.5)$, *there are two results with a large disparity between the two inconsistencies, which is mainly caused by the Zhi method misclassifying an alliance issue as a conflict issue. In particular, the disparity between the two inconsistencies in the 14th result is 0.5, the largest disparity among all the results. This is because the Zhi model treats the two issues* y_3 *and* y_4*, which are unanimously opposed by X, as conflict issues, and the proposed model considers the two issues as the unanimous opposition issues. This means that if the Zhi model determines that X is a non-coalition, the proposed model must consider that X is an alliance. Thus, the disparity between the two inconsistencies directly affects the disparity in the states of X. Similarly, in the 15th-16th results, the inconsistency of the Zhi model indicates that the sets of agents are divergent and neutral on several issues, and no consensus is reached. The inconsistencies of the proposed model are all 0 and the sets of agents reach a consensus on all the issues.*

Table 6. Comparison results.

NO.	Results of Zhi Model	$m(X)$	Results of the Proposed Model	$m(X)'$
1	$(\{\frac{1}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}\}, \varnothing)$	1	$(\{\frac{1}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}\}, (\varnothing, \{\frac{0.5}{y_3}\}))$	0.875
2	$(\{\frac{1}{x_1}, \frac{1}{x_2}, \frac{0.5}{x_3}\}, \{\frac{0.5}{y_3}\})$	0.875	$(\{\frac{1}{x_1}, \frac{1}{x_2}, \frac{0.5}{x_3}\}, (\{\frac{0.5}{y_3}\}, \{\frac{0.5}{y_2}, \frac{0.5}{y_3}\}))$	0.75
3	$(\{\frac{1}{x_1}, \frac{0.5}{x_2}, \frac{1}{x_3}\}, \{\frac{0.5}{y_1}\})$	0.875	$(\{\frac{1}{x_1}, \frac{0.5}{x_2}, \frac{1}{x_3}\}, (\{\frac{0.5}{y_1}\}, \{\frac{0.5}{y_3}, \frac{0.5}{y_4}\}))$	0.625
4	$(\{\frac{0.5}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}\}, \{\frac{0.5}{y_2}\})$	0.875	$(\{\frac{0.5}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}\}, (\{\frac{0.5}{y_2}\}, \{\frac{0.5}{y_1}, \frac{0.5}{y_3}\}))$	0.625
5	$(\{\frac{1}{x_1}, \frac{0.5}{x_2}, \frac{0.5}{x_3}\}, \{\frac{0.5}{y_1}, \frac{0.5}{y_3}\})$	0.75	$(\{\frac{1}{x_1}, \frac{0.5}{x_2}, \frac{0.5}{x_3}\}, (\{\frac{0.5}{y_1}, \frac{0.5}{y_3}\}, \{\frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\}))$	0.5
6	$(\{\frac{0.5}{x_1}, \frac{1}{x_2}, \frac{0.5}{x_3}\}, \{\frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\})$	0.625	$(\{\frac{0.5}{x_1}, \frac{1}{x_2}, \frac{0.5}{x_3}\}, (\{\frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\}, \{\frac{0.5}{y_1}, \frac{0.5}{y_2}, \frac{0.5}{y_3}\}))$	0.5
7	$(\{\frac{0.5}{x_1}, \frac{0.5}{x_2}, \frac{1}{x_3}\}, \{\frac{0.5}{y_1}, \frac{0.5}{y_2}\})$	0.75	$(\{\frac{0.5}{x_1}, \frac{0.5}{x_2}, \frac{1}{x_3}\}, (\{\frac{0.5}{y_1}, \frac{0.5}{y_2}\}, \{\frac{0.5}{y_1}, \frac{1}{y_3}, \frac{0.5}{y_4}\}))$	0.375
8	$(\{\frac{1}{x_1}, \frac{0.5}{x_3}\}, \{\frac{1}{y_1}, \frac{0.5}{y_3}\})$	0.625	$(\{\frac{1}{x_1}, \frac{0.5}{x_3}\}, (\{\frac{1}{y_1}, \frac{0.5}{y_3}\}, \{\frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{1}{y_4}\}))$	0.25
9	$(\{\frac{0.5}{x_1}, \frac{0.5}{x_2}, \frac{0.5}{x_3}\}, \{\frac{0.5}{y_1}, \frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\})$	0.5	$(\{\frac{0.5}{x_1}, \frac{0.5}{x_2}, \frac{0.5}{x_3}\}, (\{\frac{0.5}{y_1}, \frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\}, \{\frac{0.5}{y_1}, \frac{0.5}{y_2}, \frac{1}{y_3}, \frac{0.5}{y_4}\}))$	0.375
10	$(\{\frac{0.5}{x_1}, \frac{0.5}{x_3}\}, \{\frac{1}{y_1}, \frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\})$	0.375	$(\{\frac{0.5}{x_1}, \frac{0.5}{x_3}\}, (\{\frac{1}{y_1}, \frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\}, \{\frac{0.5}{y_1}, \frac{0.5}{y_2}, \frac{1}{y_3}, \frac{1}{y_4}\}))$	0.125
11	$(\{\frac{0.5}{x_1}, \frac{0.5}{x_2}\}, \{\frac{0.5}{y_1}, \frac{0.5}{y_2}, \frac{1}{y_3}, \frac{0.5}{y_4}\})$	0.375	$(\{\frac{0.5}{x_1}, \frac{0.5}{x_2}\}, (\{\frac{0.5}{y_1}, \frac{0.5}{y_2}, \frac{1}{y_3}, \frac{0.5}{y_4}\}, \{\frac{0.5}{y_1}, \frac{1}{y_2}, \frac{1}{y_3}, \frac{0.5}{y_4}\}))$	0.25
12	$(\{\frac{1}{x_2}\}, \{\frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{1}{y_4}\})$	0.5	$(\{\frac{1}{x_2}\}, (\{\frac{0.5}{y_2}, \frac{0.5}{y_3}, \frac{1}{y_4}\}, \{\frac{1}{y_1}, \frac{0.5}{y_2}, \frac{0.5}{y_3}\}))$	0.25
13	$(\{\frac{0.5}{x_2}, \frac{0.5}{x_3}\}, \{\frac{0.5}{y_1}, \frac{1}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\})$	0.375	$(\{\frac{0.5}{x_2}, \frac{0.5}{x_3}\}, (\{\frac{0.5}{y_1}, \frac{1}{y_2}, \frac{0.5}{y_3}, \frac{0.5}{y_4}\}, \{\frac{1}{y_1}, \frac{0.5}{y_2}, \frac{1}{y_3}, \frac{0.5}{y_4}\}))$	0.125
14	$(\{\frac{1}{x_3}\}, \{\frac{0.5}{y_1}, \frac{1}{y_2}\})$	0.625	$(\{\frac{1}{x_3}\}, (\{\frac{0.5}{y_1}, \frac{1}{y_2}\}, \{\frac{0.5}{y_1}, \frac{1}{y_3}, \frac{1}{y_4}\}))$	0.125
15	$(\{\frac{0.5}{x_1}\}, \{\frac{1}{y_1}, \frac{0.5}{y_2}, \frac{1}{y_3}, \frac{0.5}{y_4}\})$	0.25	$(\{\frac{0.5}{x_1}\}, (\{\frac{1}{y_1}, \frac{0.5}{y_2}, \frac{1}{y_3}, \frac{0.5}{y_4}\}, \{\frac{0.5}{y_1}, \frac{1}{y_2}, \frac{1}{y_3}, \frac{1}{y_4}\}))$	0
16	$(\{\frac{0.5}{x_2}\}, \{\frac{0.5}{y_1}, \frac{1}{y_2}, \frac{1}{y_3}, \frac{1}{y_4}\})$	0.125	$(\{\frac{0.5}{x_2}\}, (\{\frac{0.5}{y_1}, \frac{1}{y_2}, \frac{1}{y_3}, \frac{1}{y_4}\}, \{\frac{1}{y_1}, \frac{1}{y_2}, \frac{1}{y_3}, \frac{0.5}{y_4}\}))$	0
17	$(\varnothing, \{\frac{1}{y_1}, \frac{1}{y_2}, \frac{1}{y_3}, \frac{1}{y_4}\})$	0	$(\varnothing, (\{\frac{1}{y_1}, \frac{1}{y_2}, \frac{1}{y_3}, \frac{1}{y_4}\}, \{\frac{1}{y_1}, \frac{1}{y_2}, \frac{1}{y_3}, \frac{1}{y_4}\}))$	0

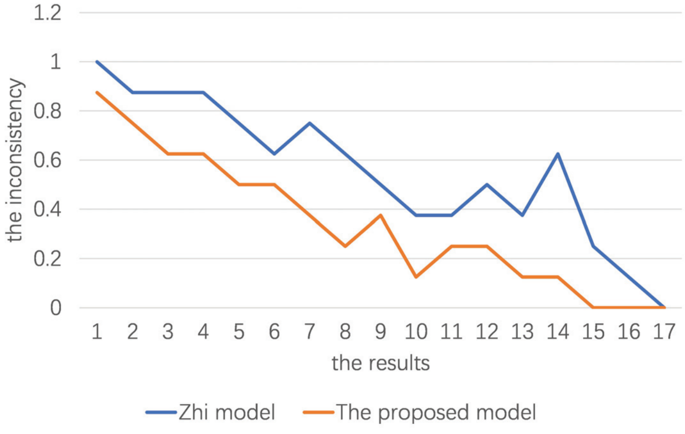


Figure 2. The differences in inconsistency between two models.

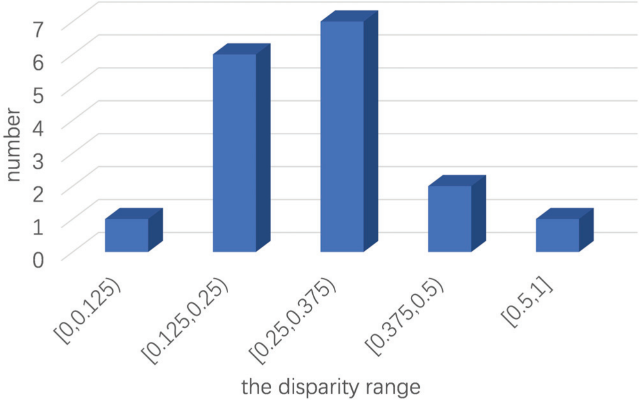


Figure 3. The disparity distribution of two inconsistencies.

4. Conflict Resolution Based on the Proposed Model

In Section 3, we examined the alliance issues, the neutrality issues, and the conflict issues of a given set of agents. For a set of agents in alliance, minimal divergence exists in the set, enabling the agents to easily reach a consensus. A non-alliance set of agents may experience substantial divergence in issues, presenting challenges for consensus-building among the agents. To address these conflicts, it becomes crucial to develop strategies that eliminate the divergence between agents and mitigate the relative inconsistency, thereby transforming the non-alliance set of agents into an alliance set.

For $X \in L^U$, it follows from Definition 12 that the relative inconsistency $m'(X)$ is determined by the comprehensive consistency $X^*(y)$ of all the issues y, and thus $m'(X)$ can be decreased only by increasing $X^*(y)$. According to Definition 13, if $y \in \alpha^+(X) \cup \alpha^-(X) \cup \alpha(X)$, then we have $X^*(y) = 1$, so the comprehensive consistency $X^*(y)$ of an alliance issue cannot be increased; if $y \in \gamma^+(X) \cup \gamma^-(X) \cup \gamma(X)$, then we have $X^*(y) = 0.5$, and the comprehensive consistency $X^*(y)$ can be increased from 0.5 to 1, i.e., a neutral issue can be transformed into an alliance issue, thus decreasing the relative inconsistency $m'(X)$; if $y \in \beta(X)$, then we have $X^*(y) = 0$, and the comprehensive consistency $X^*(y)$ can be increased from 0 to 0.5 or 1, which transforms a conflict issue into a neutral issue or an alliance issue, thus decreasing the relative inconsistency $m'(X)$. Therefore, the relative inconsistency can be effectively decreased only by increasing the comprehensive consistency of non-alliance issues. In the following, we consider how to increase the comprehensive consistency of the issues in $\gamma^+(X)$, $\gamma^-(X)$, $\gamma(X)$, and $\beta(X)$.

According to Definition 11, the comprehensive consistency $X^*(y)$ is determined by the agreement consistency $X^+(y)$ and the opposition consistency $X^-(y)$, and from Definition 10, it follows that $X^+(y)$ and $X^-(y)$ are jointly determined by the trust degree $X(x)$ and the attitude $R(x, y)$. To increase $X^*(y)$, it is necessary to increase $X^+(y)$ or $X^-(y)$. Depending on the nature of implication operator, $X^+(y)$ and $X^-(y)$ can be increased by decreasing $X(x)$; however, in most conflict situations, $X(x)$ cannot be easily changed, e.g., when an enterprise makes a development plan, the discourse powers of agents in different positions are generally fixed. Therefore, in order to increase $X^+(y)$ or $X^-(y)$, it is necessary to change agents' attitudes towards issues.

Let $K = (U, V, R)$ be a CAIS and $X \in L^U$. In order to increase $X^*(y)$, $X^+(y)$ can be increased by changing the attitudes of agents. If $X^+(y) = 0$, then in order to increase $X^+(y)$ to 0.5, according to the definition of $X^+(y)$, the set of agents whose attitudes should be changed is

$$X_0^+(y) = \{x | X(x) \to R(x, y) = 0\} \tag{26}$$

For $x \in X_0^+(y)$, we have $X(x) \neq 0$; otherwise, we have $X(x) \to R(x,y) = 0 \to R(x,y) = 1$. Therefore, we denote

$$\dot{X}_1^+(y) = \{x | X(x) \to R(x,y) = 0, X(x) = 1\} \tag{27}$$

$$\dot{X}_{0.5}^+(y) = \{x | X(x) \to R(x,y) = 0, X(x) = 0.5\} \tag{28}$$

and then $X_0^+(y) = \dot{X}_1^+(y) \cup \dot{X}_{0.5}^+(y)$.

If $X^+(y) = 0$, then in order to increase $X^+(y)$ to 1, the set of agents whose attitudes should be changed is

$$X_{0,0.5}^+(y) = X_0^+(y) \cup X_{0.5}^+(y) = \dot{X}_1^+(y) \cup \dot{X}_{0.5}^+(y) \cup X_{0.5}^+(y) \tag{29}$$

where $X_{0.5}^+(y) = \{x | X(x) \to R(x,y) = 0.5\}$.

If $X^+(y) = 0.5$, then in order to increase $X^+(y)$ to 1, the set of agents whose attitudes should be changed is $X_{0.5}^+(y)$.

Similarly, in order to increase $X^*(y)$, it is also feasible to increase $X^-(y)$ by changing the attitudes of the agents. If $X^-(y) = 0$, then in order to increase $X^-(y)$ to 0.5, according to the definition of $X^-(y)$, the set of agents whose attitude should be changed is

$$X_0^-(y) = \{x | X(x) \to \neg R(x,y) = 0\} \tag{30}$$

For $x \in X_0^-(y)$, we have $R(x,y) \neq 0$; otherwise, we have $X(x) \to \neg 0 = X(x) \to 1 = 1$. Therefore, we denote

$$\dot{X}_{0.5}^-(y) = \{x | X(x) \to \neg R(x,y) = 0, R(x,y) = 0.5\} \tag{31}$$

$$\dot{X}_1^-(y) = \{x | X(x) \to \neg R(x,y) = 0, R(x,y) = 1\} \tag{32}$$

and then $X_0^-(y) = \dot{X}_{0.5}^-(y) \cup \dot{X}_1^-(y)$.

If $X^-(y) = 0$, then in order to increase $X^-(y)$ to 1, the set of agents whose attitudes should be changed is

$$X_{0,0.5}^-(y) = X_0^-(y) \cup X_{0.5}^-(y) = \dot{X}_{0.5}^-(y) \cup \dot{X}_1^-(y) \cup X_{0.5}^-(y) \tag{33}$$

where $X_{0.5}^-(y) = \{x | X(x) \to \neg R(x,y) = 0.5\}$.

If $X^-(y) = 0.5$, then in order to increase $X^-(y)$ to 1, the set of agents whose attitudes should be changed is $X_{0.5}^-(y)$.

The following conclusions hold.

Theorem 4. *Let $K = (U, V, R)$ be a CAIS, $x \in U$ and $X \in L^U$.*

1. *Suppose $y \in \gamma^+(X)$.*
   - (a) *If $|X_{0.5}^+(y)| \leq |\dot{X}_{0.5}^-(y)| + 2|\dot{X}_1^-(y)| + |X_{0.5}^-(y)|$, then $X^*(y)$ can be increased from 0.5 to 1 at a cost of at least $0.5 \cdot |X_{0.5}^+(y)|$. In this case, the attitude of x needs to be increased to 1 for $X(x) = 1$ and $R(x,y) = 0.5$ and to 0.5 for $X(x) = 0.5$ and $R(x,y) = 0$.*
   - (b) *If $|X_{0.5}^+(y)| > |\dot{X}_{0.5}^-(y)| + 2|\dot{X}_1^-(y)| + |X_{0.5}^-(y)|$, then $X^*(y)$ can be increased from 0.5 to 1 at a cost of at least $0.5 \cdot (|\dot{X}_{0.5}^-(y)| + 2|\dot{X}_1^-(y)| + |X_{0.5}^-(y)|)$. In this case, if $\neg 0.5 = 0$, then the attitude of x needs to be reduced to 0 for $X(x) \geq 0.5$ and $R(x,y) \geq 0.5$; if $\neg 0.5 = 0.5$, then the attitude of x needs to be reduced to 0 for $X(x) = 1$ and $R(x,y) \geq 0.5$ and to 0.5 for $X(x) = 0.5$ and $R(x,y) = 1$.*
2. *Suppose $y \in \gamma^-(X)$.*
   - (a) *If $|X_{0.5}^-(y)| \leq 2|\dot{X}_1^+(y)| + |\dot{X}_{0.5}^+(y)| + |X_{0.5}^+(y)|$, then $X^*(y)$ can be increased from 0.5 to 1 at a cost of at least $0.5 \cdot |X_{0.5}^-(y)|$. In this case, the attitude of x needs to*

be reduced to 0 for $X(x) = 1$ and $R(x,y) = 0.5$ and to 0.5 for $X(x) = 0.5$ and $R(x,y) = 1$.

(b) *If $|X^-_{0.5}(y)| > 2|\dot{X}^+_1(y)| + |\dot{X}^+_{0.5}(y)| + |X^+_{0.5}(y)|$, then $X^*(y)$ can be increased from 0.5 to 1 at a cost of at least $0.5 \cdot (2|\dot{X}^+_1(y)| + |\dot{X}^+_{0.5}(y)| + |X^+_{0.5}(y)|)$. In this case, the attitude of x needs to be increased to 1 for $X(x) = 1$ and $R(x,y) \leq 0.5$ and to 0.5 for $X(x) = 0.5$ and $R(x,y) = 0$.*

3. *Suppose $y \in \gamma(X)$.*

   (a) *If $|X^+_{0.5}(y)| \leq |X^-_{0.5}(y)|$, then $X^*(y)$ can be increased from 0.5 to 1 at a cost of at least $0.5 \cdot |X^+_{0.5}(y)|$. In this case, the attitude of x needs to be increased to 1 for $X(x) = 1$ and $R(x,y) = 0.5$ and to 0.5 for $X(x) = 0.5$ and $R(x,y) = 0$.*

   (b) *If $|X^+_{0.5}(y)| > |X^-_{0.5}(y)|$, then $X^*(y)$ can be increased from 0.5 to 1 at a cost of at least $0.5 \cdot |X^-_{0.5}(y)|$. In this case, the attitude of x needs to be reduced to 0 for $X(x) = 1$ and $R(x,y) = 0.5$ and to 0.5 for $X(x) = 0.5$ and $R(x,y) = 1$.*

4. *Suppose $y \in \beta(X)$.*

   (a) *If $|\dot{X}^+_1(y)| \leq |X^-_0(y)|$, then $X^*(y)$ can be increased from 0 to 0.5 at a cost of at least $0.5 \cdot |\dot{X}^+_1(y)|$. In this case, the attitude of x needs to be increased to 0.5 for $X(x) = 1$ and $R(x,y) = 0$.*

   (b) *If $|\dot{X}^+_1(y)| > |X^-_0(y)|$, then $X^*(y)$ can be increased from 0 to 0.5 at a cost of at least $0.5 \cdot |X^-_0(y)|$. In this case, the attitude of x needs to be reduced to 0.5 for $X(x) = 1$ and $R(x,y) = 1$.*

   (c) *If $2|\dot{X}^+_1(y)| + |\dot{X}^+_{0.5}(y)| + |X^+_{0.5}(y)| \leq |\dot{X}^-_{0.5}(y)| + 2|\dot{X}^-_1(y)| + |X^-_{0.5}(y)|$, then $X^*(y)$ can be increased from 0 to 1 at a cost of at least $0.5 \cdot (2|\dot{X}^+_1(y)| + |\dot{X}^+_{0.5}(y)| + |X^+_{0.5}(y)|)$. In this case, the attitude of x needs to be increased to 1 for $X(x) = 1$ and $R(x,y) \leq 0.5$ and to 0.5 for $X(x) = 0.5$ and $R(x,y) = 0$.*

   (d) *If $2|\dot{X}^+_1(y)| + |\dot{X}^+_{0.5}(y)| + |X^+_{0.5}(y)| > |\dot{X}^-_{0.5}(y)| + 2|\dot{X}^-_1(y)| + |X^-_{0.5}(y)|$, then $X^*(y)$ can be increased from 0 to 1 at a cost of at least $0.5 \cdot (|\dot{X}^-_{0.5}(y)| + 2|\dot{X}^-_1(y)| + |X^-_{0.5}(y)|)$. In this case, if $\neg 0.5 = 0$, then the attitude of x needs to be reduced to 0 for $X(x) \geq 0.5$ and $R(x,y) \geq 0.5$; if $\neg 0.5 = 0.5$, then the attitude of x needs to be reduced to 0 for $X(x) = 1$ and $R(x,y) \geq 0.5$ and to 0.5 for $X(x) = 0.5$ and $R(x,y) = 1$.*

Proof of Theorem 4. 1. For $y \in \gamma^+(X)$, in order to increase $X^*(y)$ from 0.5 to 1, we can increase $X^+(y)$ from 0.5 to 1, or increase $X^-(y)$ from 0 to 1.

In order to increase $X^+(y)$ from 0.5 to 1, according to the definition of $X^+(y)$, the values of $X(x) \to R(x,y)$ of all the agents in $X^+_{0.5}(y)$ must be increased from 0.5 to 1. If $x \in X^+_{0.5}(y)$, then consider the following three cases.

(1) If $X(x) = 1$, then we have $1 \to R(x,y) = 0.5$, and by Proposition 1 (2), we have $R(x,y) = 0.5$. In order to increase the value of $X(x) \to R(x,y)$ from 0.5 to 1, we need to change $R(x,y)$ to $R^+_1(x,y) = \wedge\{a \in \{0, 0.5, 1\} | X(x) \to a = 1\}$. Since $1 \to a = 1$, we have $a = 1$ and $R^+_1(x,y) = 1$. Thus, the cost is $R^+_1(x,y) - R(x,y) = 0.5$.

(2) If $X(x) = 0.5$, then we have $0.5 \to R(x,y) = 0.5$ and thus $R(x,y) = 0$; otherwise, by Proposition 1 (1), if $0.5 \leq R(x,y)$, then we have $0.5 \to R(x,y) = 1$. In order to increase the value of $X(x) \to R(x,y)$ from 0.5 to 1, we need to change $R(x,y)$ to $R^+_1(x,y) = \wedge\{a \in \{0, 0.5, 1\} | X(x) \to a = 1\}$. Since $0.5 \to a = 1$, we have $a = 0.5$ or $a = 1$ and thus $R^+_1(x,y) = 0.5$. Therefore, the cost is $R^+_1(x,y) - R(x,y) = 0.5$.

(3) If $X(x) = 0$, then by Proposition 1 (1), $R(x,y)$ does not exist, such that $0 \to R(x,y) = 0.5$.

Therefore, the total cost is $\sum_{x \in X^+_{0.5}(y)} (R^+_1(x,y) - R(x,y)) = 0.5 \cdot |X^+_{0.5}(y)|$.

In order to increase $X^-(y)$ from 0 to 1, according to the definition of $X^-(y)$, the values of $X(x) \to \neg R(x,y)$ of all the agents in $X^-_{0,0.5}(y)$ must be increased to 1. Since $X^-_{0,0.5}(y) = \dot{X}^-_{0.5}(y) \cup \dot{X}^-_1(y) \cup X^-_{0.5}(y)$ and $\dot{X}^-_{0.5}(y)$, $\dot{X}^-_1(y)$ and $X^-_{0.5}(y)$ are disjoint, we need to consider the three sets, respectively.

If $x \in \dot{X}^-_{0.5}(y)$, then consider the following three cases.

(1) If $X(x) = 1$, then, since $R(x,y) = 0.5$, we have $1 \to \neg R(x,y) = \neg R(x,y) = \neg 0.5 = 0$. In order to increase $X(x) \to R(x,y)$ of all the agents in $\dot{X}^-_{0.5}(y)$ from 0 to 1, we need to change $R(x,y)$ to $R^-_1(x,y) = \vee\{a \in \{0, 0.5, 1\} | X(x) \to \neg a = 1\}$. Since $1 \to \neg a = \neg a = 1$, we have $a = 0$ and thus $R^-_1(x,y) = 0$. The cost is $R(x,y) - R^-_1(x,y) = 0.5$.

(2) If $X(x) = 0.5$, then because $R(x,y) = 0.5$, we have $0.5 \to \neg R(x,y) = 0$. Since $0.5 \to 0.5 = 0.5 \to 1 = 1$, we have $\neg 0.5 = 0$. At this time, we need to change $R(x,y)$ to $R^-_1(x,y) = \vee\{a \in \{0, 0.5, 1\} | X(x) \to \neg a = 1\}$, so we have $\neg a = 0.5$ or $\neg a = 1$. If $\neg a = 0.5$, then according to Proposition 1 (3), we have $a = 0.5$ and thus $\neg 0.5 = 0.5$, which is contradictory to $\neg 0.5 = 0$. Thus, we have $R^-_1(x,y) = 0$, and the cost is $R(x,y) - R^-_1(x,y) = 0.5$.

(3) If $X(x) = 0$, then $R(x,y)$ does not exist, such that $0 \to \neg R(x,y) = 0.5$.

Therefore, the total cost is $\sum_{x \in \dot{X}^-_{0.5}(y)} (R(x,y) - R^-_1(x,y)) = 0.5 \cdot |\dot{X}^-_{0.5}(y)|$.

If $x \in \dot{X}^-_1(y)$, then consider the following three cases.

(1) If $X(x) = 1$, then in order to increase $X(x) \to \neg R(x,y)$ from 0 to 1, we need to change $R(x,y)$ to $R^-_1(x,y) = \vee\{a \in \{0, 0.5, 1\} | X(x) \to \neg a = 1\}$. Since $1 \to \neg a = \neg a = 1$, we have $a = 0$ and thus $R^-_1(x,y) = 0$. The cost is $R(x,y) - R^-_1(x,y) = 1$.

(2) If $X(x) = 0.5$, then since $R(x,y) = 1$, we have $0.5 \to \neg R(x,y) = 0.5 \to \neg 1 = 0.5 \to 0 = \neg 0.5 = 0$. Since $0.5 \to \neg a = 1$, we have $\neg a = 0.5$ or $\neg a = 1$. If $\neg a = 0.5$, according to Proposition 1 (3), we can conclude $a = 0.5$ and thus $\neg 0.5 = 0.5$, which is contradictory to $\neg 0.5 = 0$. Therefore, we have $\neg a = 1$, i.e., $a = 0$. In this case, $R^-_1(x,y) = 0$ and the cost is $R(x,y) - R^-_1(x,y) = 1$.

(3) If $X(x) = 0$, then $R(x,y)$ does not exist, such that $0 \to \neg R(x,y) = 0.5$.

Therefore, the total cost is $\sum_{x \in \dot{X}^-_1(y)} (R(x,y) - R^-_1(x,y)) = \sum_{x \in \dot{X}^-_1(y)} 1 = |\dot{X}^-_1(y)|$.

If $x \in X^-_{0.5}(y)$, then consider the following three cases.

(1) If $X(x) = 1$, we have $1 \to \neg R(x,y) = \neg R(x,y) = 0.5$, and only when $R(x,y) = 0.5$, $\neg R(x,y) = 0.5$ holds. In order to increase the value of $X(x) \to \neg R(x,y)$ from 0.5 to 1, we need to change $R(x,y)$ to $R^-_1(x,y) = \vee\{a \in \{0, 0.5, 1\} | X(x) \to \neg a = 1\}$. Since $1 \to \neg a = 1$, we have $\neg a = 1$ and thus $a = 0$. Thus, we have $R^-_1(x,y) = 0$, and the cost is $R(x,y) - R^-_1(x,y) = 0.5$.

(2) If $X(x) = 0.5$, then we have $0.5 \to \neg R(x,y) = 0.5$ and thus $\neg R(x,y) = 0$, yielding $\neg 0.5 = 0.5 \to 0 = 0.5$. Since $\neg R(x,y) = 0$, we have $R(x,y) = 1$ or $R(x,y) = 0.5$. If $R(x,y) = 0.5$, then we have $\neg 0.5 = 0$, which is contradictory to $\neg 0.5 = 0.5$, and thus $R(x,y) = 1$. In order to increase the value of $X(x) \to \neg R(x,y)$ from 0.5 to 1, we need to change $R(x,y)$ to $R^-_1(x,y) = \vee\{a \in \{0, 0.5, 1\} | X(x) \to \neg a = 1\}$. Since $0.5 \to \neg a = 1$, we have $\neg a = 0.5$ or $\neg a = 1$, and thus $a = 0.5$ or $a = 0$. At this time, we have $R^-_1(x,y) = 0.5$, and the cost is $R(x,y) - R^-_1(x,y) = 0.5$.

(3) If $X(x) = 0$, then $R(x,y)$ does not exist, such that $0 \to \neg R(x,y) = 0.5$.

Therefore, the total cost is $\sum_{x \in X^-_{0.5}(y)} (R(x,y) - R^-_1(x,y)) = 0.5 \cdot |X^-_{0.5}(y)|$.

In summary, in order to increase $X^*(y)$ from 0.5 to 1, we can increase $X^+(y)$ to 1 at the cost of $0.5 \cdot |X^+_{0.5}(y)|$, or increase $X^-(y)$ to 1 at the cost of $0.5 \cdot (|\dot{X}^-_{0.5}(y)| + 2|\dot{X}^-_1(y)| + |X^-_{0.5}(y)|)$. If $|X^+_{0.5}(y)| \leq |\dot{X}^-_{0.5}(y)| + 2|\dot{X}^-_1(y)| + |X^-_{0.5}(y)|$, then $X^+(y)$ can be increased to 1 at a cost of at least $0.5 \cdot |X^+_{0.5}(y)|$. Otherwise, $X^-(y)$ can be increased to 1 at a cost of at least $0.5 \cdot (|\dot{X}^-_{0.5}(y)| + 2|\dot{X}^-_1(y)| + |X^-_{0.5}(y)|)$.

Thus, if $X^+(y)$ is increased to 1, then we need to increase the attitude of x with $X(x) = 1$ and $R(x,y) = 0.5$ to 1, as well as increasing the attitude of x with $X(x) = 0.5$ and $R(x,y) = 0$ to 0.5. If $X^-(y)$ is increased to 1, and if $\neg 0.5 = 0$, then we need to reduce the attitude of x with $X(x) \geq 0.5$ and $R(x,y) \geq 0.5$ to 0. If $\neg 0.5 = 0.5$, then the attitude of x with $X(x) = 1$ and $R(x,y) \geq 0.5$ should be reduced to 0, and the attitude of x with $X(x) = 0.5$ and $R(x,y) = 1$ should be reduced to 0.5.

Other results can be proved similarly. □

By Definition 13, if $m'(X) > t_1$, then X is a neutral or conflict set of agents. In order to transform X into an alliance set, it is necessary to reduce $m'(X)$ to t_1, with the difference of $m'(X) - t_1$.

In order to reduce $m'(X)$, it is necessary to increase $X^*(y)$. If $X^*(y) = 0$, then $X^*(y)$ can be increased either to 0.5 or to 1; if $X^*(y) = 0.5$, then $X^*(y)$ can be increased to 1.

Now, denote the number of issues such that $X^*(y) = 0$ as n_0 and the number of issues such that $X^*(y) = 0.5$ as $n_{0.5}$. According to Definition 12, we have

$$m'(X) = \frac{\sum_{y \in V}(1 - X^*(y))}{|V|} = \frac{n_0 + 0.5n_{0.5}}{|V|} \tag{34}$$

Denote by $n_{0\to0.5}$ the number of issues whose $X^*(y)$ need to be increased from 0 to 0.5; denote by $n_{0\to1}$ the number of issues whose $X^*(y)$ need to be increased from 0 to 1; denote by $n_{0.5\to1}$ the number of issues whose $X^*(y)$ need to be increased from 0.5 to 1. Thus, the relative inconsistency will become

$$m'_{\uparrow}(X) = \frac{0.5n_{0\to0.5} + n_0 - n_{0\to0.5} - n_{0\to1} + 0.5(n_{0.5} - n_{0.5\to1})}{|V|} \tag{35}$$

and the difference is

$$m'(X) - m'_{\uparrow}(X) = \Delta m'(X) = \frac{0.5n_{0\to0.5} + 0.5n_{0.5\to1} + n_{0\to1}}{|V|} \tag{36}$$

We require $\Delta m'(X) \geq m'(X) - t_1$, i.e., $n_{0\to0.5}$, $n_{0\to1}$ and $n_{0.5\to1}$ satisfy Equation (37)

$$0.5n_{0\to0.5} + 0.5n_{0.5\to1} + n_{0\to1} \geq (m'(X) - t_1)|V| \tag{37}$$

Obviously, there may be different values of $n_{0\to0.5}$, $n_{0.5\to1}$ and $n_{0\to1}$ that satisfy Equation (37) and can transform X into an alliance set. Since the cost of increasing $X^*(y)$ varies for different issues y, in real life, people tend to seek the least costly way to make decisions. Therefore, it is necessary to consider which issues to change so that the cost of transforming X into an alliance set is minimal.

For an issue y, there may be more than one way to increase $X^*(y)$. For $y \in \gamma^+(X) \cup \gamma^-(X) \cup \gamma(X)$, $X^*(y)$ can only be increased from 0.5 to 1 with a cost of 0.5; for $y \in \beta(X)$, $X^*(y)$ can be increased from 0 to 0.5 or from 0 to 1 with a cost of 0.5 or 1. Therefore, there are at most two methods to increase $X^*(y)$ for issue y, but at most one method can be chosen.

The problem can be formulated as follows. For a set of s issues, $r_i \in \{1, 2\}$ denotes the number of increasing methods for the i-th issue; the variation of the k-th increasing method for the i-th issue is w_i^k, and the corresponding cost is c_i^k. The objective is to choose an optimal solution $M = \{p_1^1, ..., p_1^{r_1}, p_2^1, \ldots, p_2^{r_2}, \ldots, p_s^1, \ldots, p_s^{r_s}\}$, such that the sum of the costs $\sum_{i=1}^{s}\sum_{k=1}^{r_i} p_i^k c_i^k$ is minimized while the sum of variations $\sum_{i=1}^{s}\sum_{k=1}^{r_i} p_i^k w_i^k$ is not less than $W = (m^{'}(X) - t_1)|V|$, where $p_i^k \in \{0, 1\}$ indicates whether the optimal solution chooses the k-th increasing method for the i-th issue, i.e.,

$$\min \sum_{i=1}^{s}\sum_{k=1}^{r_i} p_i^k c_i^k \tag{38}$$

$$s.t. \sum_{i=1}^{s}\sum_{k=1}^{r_i} p_i^k w_i^k \geq W \tag{39}$$

It can be observed that the problem is actually a grouped knapsack problem, which is a special kind of knapsack problem. There are several solutions to solve the knapsack problem, such as branch-and-bound method [39], dynamic programming method [40], and approximation algorithm [41]. Since the dynamic programming method can solve the optimal solution quickly for smaller knapsack problems, we adopt the dynamic programming method to select the least costly set of issues. The method of solving the optimal attitude change strategy based on the dynamic programming method is shown in Algorithm 1.

Algorithm 1 The optimal attitude change strategy based on dynamic programming method.

```
Require: A CAIS $K = (U, V, R)$, threshold $t_1$, a set $X \in L^U$ of agents
Ensure: An optimal attitude change strategy
 1: Compute $\gamma^+(X)$, $\gamma^-(X)$, $\gamma(X)$, and $\beta(X)$ according to Definition 13
 2: for each $y$ in $\gamma^+(X)$ do
 3:     Compute $|X^+_{0.5}(y)|$ and $|\dot{X}^-_{0.5}(y)| + 2|\dot{X}^-_1(y)| + |X^-_{0.5}(y)|$
 4:     $r[y] = 1$    ▷ $r[y]$ denotes the number of increasing methods of $y$
 5:     $w[y][1] = 0.5$    ▷ $w[y][1]$ denotes the variation of the first increasing method of $y$
 6:     $c[y][1] = \min(0.5 \cdot |X^+_{0.5}(y)|, 0.5 \cdot (|\dot{X}^-_{0.5}(y)| + 2|\dot{X}^-_1(y)| + |X^-_{0.5}(y)|))$    ▷ $c[y][1]$ denotes the cost of the first increasing method of $y$
 7: end for
 8: for each $y$ in $\gamma^-(X)$ do
 9:     Compute $|X^-_{0.5}(y)|$ and $2|\dot{X}^+_1(y)| + |\dot{X}^+_{0.5}(y)| + |X^+_{0.5}(y)|$
10:     $r[y] = 1$
11:     $w[y][1] = 0.5$
12:     $c[y][1] = \min(0.5 \cdot |X^-_{0.5}(y)|, 0.5 \cdot (2|\dot{X}^+_1(y)| + |\dot{X}^+_{0.5}(y)| + |X^+_{0.5}(y)|))$
13: end for
14: for each $y$ in $\gamma(X)$ do
15:     Compute $|X^+_{0.5}(y)|$ and $|X^-_{0.5}(y)|$
16:     $r[y] = 1$
17:     $w[y][1] = 0.5$
18:     $c[y][1] = \min(0.5 \cdot |X^+_{0.5}(y)|, 0.5 \cdot |X^-_{0.5}(y)|)$
19: end for
20: for each $y$ in $\beta(X)$ do
21:     Compute $|\dot{X}^+_1(y)|$, $|X^-_0(y)|$, $2|\dot{X}^+_1(y)| + |\dot{X}^+_{0.5}(y)| + |X^+_{0.5}(y)|$ and $|\dot{X}^-_{0.5}(y)| + 2|\dot{X}^-_1(y)| + |X^-_{0.5}(y)|$
22:     $r[y] = 2$    ▷ the number of increasing methods is 2
23:     $w[y][1] = 0.5$
24:     $c[y][1] = \min(0.5 \cdot |\dot{X}^+_1(y)|, 0.5 \cdot |X^-_0(y)|)$
25:     $w[y][2] = 1$    ▷ $w[y][2]$ denotes the variation of the second increasing method of $y$
26:     $c[y][2] = \min(0.5 \cdot (2|\dot{X}^+_1(y)| + |\dot{X}^+_{0.5}(y)| + |X^+_{0.5}(y)|), 0.5 \cdot (|\dot{X}^-_{0.5}(y)| + 2|\dot{X}^-_1(y)| + |X^-_{0.5}(y)|))$    ▷ $c[y][2]$ denotes the cost of the second increasing method of $y$
27: end for
28: Compute $m'(X)$ according to Definition 12
29: $\Delta = (m'(X) - t_1)|V|$    ▷ $\Delta$ is the amount of variations needed to transform $X$ into an alliance set
30: Utilize the dynamic programming method in [30] or [31,32] to solve the knapsack problem and obtain the optimal solution $M = \{p_1^1, \dots, p_1^{r_1}, p_2^1, \dots, p_2^{r_2}, \dots, p_s^1, \dots, p_s^{r_s}\}$
31: According to $M$, determine the agents that need to change their attitudes, and add the agents and the issues to the corresponding sets
32: return $R_{0\to0.5}$, $R_{0.5\to1}$, $R_{0\to1}$, $R_{0.5\to0}$, $R_{1\to0.5}$, $R_{1\to0}$
```

Steps 2–27 of Algorithm 1 first determine the number of increasing methods $r[y]$, the variation $w[y][k]$, and the cost $c[y][k]$ of the k-th increasing method for issue y according to Theorem 4. Then, Algorithm 1 solves the optimal attitude change strategy by dynamic programming method in Step 30 and obtains an optimal solution $M = \{p_1^1, \dots, p_1^{r_1}, p_2^1, \dots, p_2^{r_2}, \dots, p_s^1, \dots, p_s^{r_s}\}$. Finally, in Step 31, according to the optimal solution M and Theorem 4, Algorithm 1 identifies the agents that are required to change their attitudes towards some specific issues, and adds the pairs of agent and issue to the corresponding sets: $R_{0\to0.5}$, $R_{0.5\to1}$, $R_{0\to1}$, $R_{0.5\to0}$, $R_{1\to0.5}$, and $R_{1\to0}$.

Example 6 verifies the validity of Algorithm 1.

Example 6. *For the CAIS in Example 5, given the threshold $t_1 = 0.35$, we have $X = \{1/x_1, 0.5/x_2, 1/x_3\}$. According to Table 3, we have $m'(X) = 0.625$, $\gamma^+(X) = \{y_1\}$, $\gamma^-(X) = \{y_3, y_4\}$, and $\beta(X) = \{y_2\}$. Since $m'(X) = 0.625 > 0.35$, X should be transformed into an alliance set.*

According to Algorithm 1, $X^*(y_1)$ of issue y_1 can be increased from 0.5 to 1 with the variation $w[y_1][1] = 0.5$ and the cost $c[y_1][1] = 0.5$; $X^*(y_3)$ of issue y_3 can be increased from 0.5 to 1 with the variation $w[y_3][1] = 0.5$ and the cost $c[y_3][1] = 0.5$; $X^*(y_4)$ of issue y_4 can be increased from 0.5 to 1 with the variation $w[y_4][1] = 0.5$ and the cost $c[y_4][1] = 0.5$; $X^*(y_2)$ of issue y_2 can be increased from 0 to 0.5 with the variation $w[y_2][1] = 0.5$ and the cost $c[y_2][1] = 0.5$, or increased from 0 to 1 with the variation $w[y_2][2] = 1$ and the cost $c[y_2][2] = 1$. According to the dynamic programming method, only $p^2_{y_2} = 1$ and $p^1_{y_4} = 1$ can be obtained in M.

Since $y_2 \in \beta(X)$, we can compute $2|\dot{X}^+_1(y_1)| + |\dot{X}^+_{0.5}(y_1)| + |X^+_{0.5}(y_1)| = 2 = |\dot{X}^-_{0.5}(y_1)| + 2|\dot{X}^-_1(y_1)| + |X^-_{0.5}(y_1)|$. Furthermore, since $R(x_1, y_2) = 0$ and $X(x_2) = 1$, we have $R_{0\to1} = \{(x_1, y_2)\}$. Since $y_4 \in \gamma^-(X)$, we can compute $|\dot{X}^-_{0.5}(y_4)| = 1 < 4 = 2|\dot{X}^+_1(y_4)| + |\dot{X}^+_{0.5}(y_4)| + |X^+_{0.5}(y_4)|$. Since $R(x_2, y_4) = 1$ and $X(x) = 0.5$, we have $R_{1\to0.5} = \{(x_2, y_4)\}$.

Therefore, the optimal attitude change strategy is $R_{0\to1} = \{(x_1, y_2)\}$, $R_{1\to0.5} = \{(x_2, y_4)\}$, i.e., changing the attitude of x_1 towards y_2 from opposing to agreeing and changing the attitude of x_2 towards y_4 from agreeing to neutral. After changing attitudes, we have $m'(X) = 0.25 < 0.35$, $\alpha^+(X) = \{y_2\}$, $\alpha^-(X) = \{y_4\}$, $\gamma^+(X) = \{y_1\}$ and $\gamma^-(X) = \{y_3\}$, i.e., X becomes an alliance set with the minimal cost 2. The changes in conflict analysis results caused by attitude changes is shown as Figure 4.

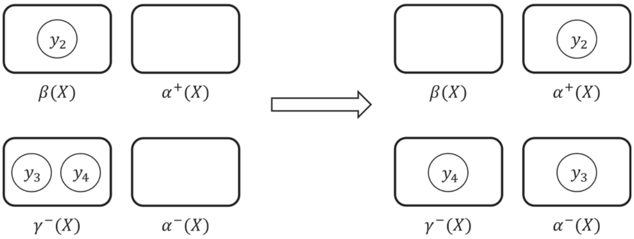


Figure 4. Changes in conflict analysis results caused by attitude changes.

5. Conclusions

In this paper, a hybrid conflict analysis model is proposed by combining conflict analysis and *L*-3WCL. The model employs *L*-fuzzy three-way concepts to capture the issues on which agents commonly agree and the issues which they commonly oppose, and identifies the state of a set of agents. In addition, this paper seeks minimal cost to reach a consensus of agents by developing an optimal attitude change strategy.

In fact, the existing studies [9–11] used different fuzzy numbers to describe agents' uncertainty about issues in conflict analysis situations, and these studies were conducted in the case that agents are completely trustworthy. The study [25] further considered to what extent the agent can be trusted. However, [25] used fuzzy concepts to describe agents' consistency by considering only the common agreement consistency and ignoring the common opposition consistency. In this study, we use *L*-3WCL to describe both agreement consistency and disagreeing consistency, which can help us analyze conflict situations more comprehensively.

On the other hand, the existing conflict analysis solution methods [1,4,6,26] used as a solution strategy the selection of the optimal subsets of issues most agents agree on. Our method introduces a third-party assumption to mediate between agents in search of an optimal attitude change strategy, and is better suited to deal with situations where the conflict has not yet reached a serious level. Thus, our study provides a new perspective on conflict resolution.

However, there are some problems that need to be explored and improved. For example, in the proposed model, both the agreement consistency and the opposition consistency may reach the value 1, i.e., the set of agents may agree on and oppose an issue unanimously. Although we have explained this situation as a special case, this may be unreasonable. In fact, by Proposition 1 (1), this is a result of introducing CRL to characterize the problem of conflict analysis. In addition, the proposed model employs CRL to characterize both attitudes and trust degrees to $\{0, 0.5, 1\}$. Since attitude values and trust values have different meanings and operators, for example, it seems unreasonable that the negative attitude 0 is less than the positive attitude 1, but it seems reasonable that the trust degree 0 is less than the trust degree 1. Furthermore, the different meanings of their values means that different operators are needed. This may lead to unreasonable results and therefore needs to be further explored.

Author Contributions: Conceptualization, Y.Z. and Z.J.; methodology, Y.Z., Z.J. and D.L.; software, Y.Z.; validation, Y.Z. and Z.J.; formal analysis, Y.Z., Z.J. and D.L.; investigation, Z.J.; resources, Y.Z. and D.L.; data curation, Y.Z. and Z.J.; writing—original draft preparation, Y.Z. and Z.J.; writing—review and editing, Y.Z., Z.J. and D.L.; visualization, Y.Z. and Z.J.; supervision, Y.Z. and D.L.; project administration, Y.Z.; funding acquisition, Y.Z. and D.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant numbers 61972238 and 62072294).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zia, B.; Saima, M.; Malik, M.G.A. Conflict resolution using game theory and rough sets. *Int. J. Intell. Syst.* **2020**, *36*, 237–259.
2. Du, J.; Liu, S.; Liu, Y.; Tao, L. Multi-criteria large-scale group decision-making in linguistic contexts: A perspective of conflict analysis and resolution. *Group Decis. Negot.* **2022**, *32*, 177–207. [CrossRef]
3. Pawlak, Z. On conflicts. *Int. J. Man–Mach. Stud.* **1984**, *21*, 127–134. [CrossRef]
4. Xu, F.; Cai, M.; Song, H.; Dai, J. The selection of feasible strategies based on consistency measurement of cliques. *Inf. Sci.* **2022**, *583*, 33–55. [CrossRef]
5. Pawlak, Z. An inquiry into anatomy of conflicts. *Inf. Sci.* **1998**, *109*, 65–78. [CrossRef]
6. Du, J.; Liu, S.; Liu, Y.; Yi, J. A novel approach to three-way conflict analysis and resolution with pythagorean fuzzy information. *Inf. Sci.* **2022**, *584*, 65–88. [CrossRef]
7. Yao, Y. Three-way conflict analysis: Reformulations and extensions of the pawlak model. *Knowl. Based Syst.* **2019**, *180*, 26–37. [CrossRef]
8. Lang, G.; Yao, Y. New measures of alliance and conflict for three-way conflict analysis. *Int. J. Approx. Reason.* **2021**, *132*, 49–69. [CrossRef]
9. Lang, G.; Miao, D.; Fujita, H. Three-way group conflict analysis based on pythagorean fuzzy set theory. *IEEE Trans. Fuzzy Syst.* **2019**, *28*, 447–461. [CrossRef]
10. Li, X.; Yang, Y.; Yi, H.; Yu, Q. Conflict analysis based on three-way decision for trapezoidal fuzzy information systems. *Int. J. Mach. Learn. Cybern.* **2022**, *13*, 929–945. [CrossRef]
11. Yang, H.; Wang, Y.; Guo, Z. Three-way conflict analysis based on hybrid situation tables. *Inf. Sci.* **2023**, *628*, 522–541. [CrossRef]
12. Suo, L.; Yang, H. Three-way conflict analysis based on incomplete situation tables: A tentative study. *Int. J. Approx. Reason.* **2022**, *145*, 51–74. [CrossRef]
13. Wang, T.; Huang, B.; Li, H.; Liu, D.; Yu, H. Three-way decision for probabilistic linguistic conflict analysis via compounded risk preference. *Inf. Sci.* **2023**, *631*, 65–90. [CrossRef]
14. Zhi, H.; Qi, J.; Qian, T.; Ren, R. Conflict analysis under one-vote veto based on approximate three-way concept lattice. *Inf. Sci.* **2019**, *516*, 316–330. [CrossRef]
15. Qi, J.; Qian, T.; Wei, L. The connections between three-way and classical concept lattices. *Knowl. Based Syst.* **2016**, *91*, 143–151. [CrossRef]
16. Yao, Y. An outline of a theory of three-way decisions. In *Rough Sets and Current Trends in Computing: Proceedings of the 8th International Conference, RSCTC 2012, Chengdu, China, 17–20 August 2012*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7413, pp. 1–17.
17. Yao, Y. Three-way decision and granular computing. *Int. J. Approx. Reason.* **2018**, *103*, 107–123. [CrossRef]
18. Yang, Z.; Ren, J.; Zhang, Z.; Sun, Y.; Zhang, C.; Wang, M.; Wang, L. A New Three-Way Incremental Naive Bayes Classifier. *Electronics* **2023**, *12*, 1730. [CrossRef]
19. Wu, J.; Xu, J.; Lin, D.; Tu, M. Optical Flow Filtering-Based Micro-Expression Recognition Method. *Electronics* **2020**, *9*, 2056. [CrossRef]

20. Zhang, C.; Zhang, M.; Yang, G.; Xue, T.; Zhang, Z.; Liu, L.; Wang, L.; Hou, W.; Chen, Z. Three-Way Selection Random Forest Optimization Model for Anomaly Traffic Detection. *Electronics* **2023**, *12*, 1788. [CrossRef]
21. Wille, R. Restructuring lattice theory: An approach based on hierarchies of concepts. In *Ordered Sets (Banff, Alta., 1981)*; D. Reidel Publishing Company: Dordrecht, The Netherlands, 1982; pp. 445–470.
22. Wille, R.; Ganter, B. Fuzzy concept lattice. In *Formal Concept Analysis: Mathematical Foundations*; Springer-Verlag: Berlin, Germany, 1999; pp. 183–217.
23. Wei, L.; Liu, L.; Qi, J.; Qian, T. Rules acquisition of formal decision contexts based on three-way concept lattices. *Inf. Sci.* **2020**, *516*, 529–544. [CrossRef]
24. Zhai, Y.; Qi, J.; Li, D.; Zhang, C.; Xu, W. Three-way decision and granular computing. *Int. J. Approx. Reason.* **2022**, *146*, 157–173. [CrossRef]
25. Zhi, H.; Li, J.; Li, Y. Multilevel conflict analysis based on fuzzy formal contexts. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 5128–5142. [CrossRef]
26. Sun, B.; Chen, X.; Zhang, L.; Ma, W. Three-way decision making approach to conflict analysis and resolution using probabilistic rough set over two universes. *Inf. Sci.* **2020**, *507*, 809–822. [CrossRef]
27. He, X.; Wei, L.; She, Y. L-fuzzy concept analysis for three-way decisions: basic definitions and fuzzy inference mechanisms. *Int. J. Mach. Learn. Cybern.* **2018**, *9*, 1857–1867. [CrossRef]
28. Wang, Z.; Qi, J.; Shi, C.; Ren, R.; Wei, L. Multiview granular data analytics based on three-way concept analysis. *Appl. Intell.* **2023**, *53*, 14645–14667. [CrossRef]
29. Zhang, C; Li, J; Lin, Y. Matrix-based reduction approach for one-sided fuzzy three-way concept lattices. *J. Intell. Fuzzy Syst.* **2021**, *40*, 11393–11410. [CrossRef]
30. Ezugwu, A.E.; Pillay, V.; Hirasen, D.; Sivanarain, K.; Govender, M. A comparative study of meta-heuristic optimization algorithms for 0 -1 knapsack problem: Some initial results. *IEEE Access* **2019**, *7*, 43979–44001. [CrossRef]
31. Sin, S.T.T. The parallel processing approach to the dynamic programming algorithm of knapsack problem. In Proceedings of the IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), St. Petersburg, Russia, 26–29 January 2021; pp. 2252–2256.
32. Clautiaux, F.; Detienne, B.; Guillot, G. An iterative dynamic programming approach for the temporal knapsack problem. *Eur. J. Oper. Res.* **2021**, *293*, 442–456. [CrossRef]
33. Ward, M.; Dilworth, R.P. Residuated Lattices. In *The Dilworth Theorems*; Kenneth, P.B., Freese, R., Joseph, P.S.K., Eds.; Birkhäuser: Boston, MA, USA, 1990; pp. 317–336
34. Pei, D. The characterization of residuated lattices and regular residuated lattices. *Acta Math. Sin.* **2002**, *45*, 271–278.
35. Bělohlávek, R. Some properties of residuated lattices. *Czechoslov. Math. J.* **2003**, *53*, 161–171. [CrossRef]
36. Bělohlávek, R. Algorithms for fuzzy concept lattices. *Acta Math. Sin.* **2002**, *45*, 271–278.
37. Bělohlávek, R.; De Baets, B.; Outrata, J.; Vychodil, V. Computing the lattice of all fixpoints of a fuzzy closure operator. *IEEE Trans. Fuzzy Syst.* **2010**, *18*, 546–557. [CrossRef]
38. Wang, W.; Wang, G.J. Gödel implication operator is not suitable for establishing fuzzy propositional logic. *Fuzzy Sets Syst.* **2005**, *19*, 14–18.
39. Stefano, C.; Fabio, F.; Pablo, S. A new combinatorial branch-and-bound algorithm for the knapsack problem with conflicts. *Eur. J. Oper. Res.* **2021**, *289*, 435–455.
40. Djeumou, F.F. A lifted-space dynamic programming algorithm for the quadratic knapsack problem. *Discret. Appl. Math.* **2023**, *335*, 52–68. [CrossRef]
41. Boeckmann, J.; Thielen, C.; Pferschy, U. Approximating single- and multi-objective nonlinear sum and product knapsack problems. *Discret. Optim.* **2023**, *48*, 100771. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

Electronics Editorial Office
E-mail: electronics@mdpi.com
www.mdpi.com/journal/electronics

Disclaimer/Publisher's Note: The title and front matter of this reprint are at the discretion of the Guest Editors. The publisher is not responsible for their content or any associated concerns. The statements, opinions and data contained in all individual articles are solely those of the individual Editors and contributors and not of MDPI. MDPI disclaims responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

MDPI

Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-3138-8