electronics

Special Issue Reprint

Future Technologies for Data Management, Processing and Application

Edited by
Duc Tai Le

mdpi.com/journal/electronics

MDPI

Future Technologies for Data Management, Processing and Application

Future Technologies for Data Management, Processing and Application

Guest Editor

Duc Tai Le

MDPI

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

Guest Editor
Duc Tai Le
Department of AI Systems Engineering
Sungkyunkwan University
Suwon
Republic of Korea

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/JLNB43G2JT.

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 978-3-7258-8031-7 (Hbk)
ISBN 978-3-7258-8032-4 (PDF)
https://doi.org/10.3390/books978-3-7258-8032-4

© 2026 by the authors. Articles in this reprint are Open Access and distributed under the Creative Commons Attribution (CC BY) license. The reprint as a whole is distributed by MDPI under the terms and conditions of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) license (https://creativecommons.org/licenses/by-nc-nd/4.0/).

Contents

About the Editor . . . . . **vii**

Duc-Tai Le
Future Technologies for Data Management, Processing, and Application
Reprinted from: *Electronics* **2026**, *15*, 2117, https://doi.org/10.3390/electronics15102117 . . . . . **1**

Bo Fan, Linfu Sun, Dong Tan and Meng Pan
Optimal Selection Technology of Business Data Resources for Multi-Value Chain Data Space—Optimizing Future Data Management Methods
Reprinted from: *Electronics* **2024**, *13*, 4690, https://doi.org/10.3390/electronics13234690 . . . . . **6**

Ang Ma, Yanhua Yu, Chuan Shi, Shuai Zhen, Liang Pang and Tat-Seng Chua
PMHR: Path-Based Multi-Hop Reasoning Incorporating Rule-Enhanced Reinforcement Learning and KG Embeddings
Reprinted from: *Electronics* **2024**, *13*, 4847, https://doi.org/10.3390/electronics13234847 . . . . . **41**

Yuanyuan Wang
Leveraging SNS Data for E-Sports Recommendation: Analyzing Popularity and User Satisfaction Metrics
Reprinted from: *Electronics* **2025**, *14*, 94, https://doi.org/10.3390/electronics14010094 . . . . . **60**

Yiluo Liu, Yaokai Feng and Kouichi Sakurai
Hyperledger Fabric-Based Multi-Channel Structure for Data Exchange in Internet of Vehicles
Reprinted from: *Electronics* **2025**, *14*, 572, https://doi.org/10.3390/electronics14030572 . . . . . **96**

Bo Hu, Shigang Tang, Fangzheng Huang, Guangqiang Yin and Jingye Cai
Implicit Identity Authentication Method Based on User Posture Perception
Reprinted from: *Electronics* **2025**, *14*, 835, https://doi.org/10.3390/electronics14050835 . . . . . **121**

Homayoon Beigi and Xing Yi Liu
Efficient Ensemble of Deep Neural Networks for Multimodal Punctuation Restoration and the Spontaneous Informal Speech Dataset
Reprinted from: *Electronics* **2025**, *14*, 973, https://doi.org/10.3390/electronics14050973 . . . . . **134**

Nikolay Metodiev Sirakov and Long H. Ngo
Automatic Image Distillation with Wavelet Transform and Modified Principal Component Analysis
Reprinted from: *Electronics* **2025**, *14*, 1357, https://doi.org/10.3390/electronics14071357 . . . . . **153**

Teng Li, Jiajia Nie, Guozhi Qiu, Zhen Li, Cun Ji and Xueqing Li
Time Series Forecasting via an Elastic Optimal Adaptive GM(1,1) Model
Reprinted from: *Electronics* **2025**, *14*, 2071, https://doi.org/10.3390/electronics14102071 . . . . . **166**

Lucija Petricioli, Luka Humski and Mihaela Vranić
Preserving Clusters in Synthetic Data Sets Based on Correlations and Distributions
Reprinted from: *Electronics* **2025**, *14*, 2230, https://doi.org/10.3390/electronics14112230 . . . . . **183**

Yu Zhang, Jinyue Shi, Shicheng Huang, Liang Geng, Zexiong Wang, Hao Sun, et al.
Data Driven Analytics for Distribution Network Power Supply Reliability Assessment Method Considering Frequency Regulating Scenario
Reprinted from: *Electronics* **2025**, *14*, 4009, https://doi.org/10.3390/electronics14204009 . . . . . **215**

Hayong Jeong, Binwon Song, Minwoo Jo and Heeseung Jo
LACX: Locality-Aware Shared Data Migration in NUMA +CXL Tiered Memory
Reprinted from: *Electronics* **2025**, *14*, 4235, https://doi.org/10.3390/electronics14214235 . . . . . **231**

About the Editor

Duc Tai Le

Duc Tai Le received an M.S. degree in Computer Science from the University of Science, Vietnam National University Ho Chi Minh City, Vietnam, in 2010, and a Ph.D. degree in Computer Engineering from Sungkyunkwan University, South Korea, in 2016. He was a postdoctoral researcher at the Convergence Research Institute from 2016 to 2019. He has since held research professor appointments at Sungkyunkwan University, serving at the College of Computing and Informatics from 2019 to 2024, the Department of Electrical and Computer Engineering from 2024 to 2026, and, since March 2026, the Department of AI Systems Engineering. His research interests span wireless ad hoc and sensor networks, software-defined networking, intelligent networking, and medical image processing. He serves as a Special Issue Guest Editor for the MDPI journal *Electronics* and an Area Editor for *KSII Transactions on Internet and Information Systems*.

electronics

MDPI

Editorial

Future Technologies for Data Management, Processing, and Application

Duc-Tai Le

Department of AI Systems Engineering, Sungkyunkwan University, Suwon 16419, Republic of Korea; ldtai@skku.edu

1. Introduction

Data management, processing, and application are increasingly interconnected in modern electronic systems. In industrial platforms, connected vehicles, smart terminals, energy systems, social media environments, and high-performance computing infrastructures, the value of data depends not only on its availability but also on whether it can be appropriately selected, efficiently processed, securely protected, and reliably applied. As data becomes more heterogeneous and application environments more dynamic, research in this area must move beyond algorithmic performance alone and consider the full pathway from data-resource selection to trustworthy processing and practical deployment.

This Special Issue, "Future Technologies for Data Management, Processing, and Application," brings together eleven contributions that address this pathway from complementary perspectives. The papers examine data resource evaluation, knowledge graph reasoning, recommendation, secure data exchange, implicit authentication, multimodal speech processing, image distillation, adaptive forecasting, synthetic data generation, microgrid reliability assessment, and memory system optimization. Collectively, they show that data-oriented electronic systems require coordinated advances in data quality, domain-aware processing, security, reliability, and resource-efficient implementation.

This perspective is consistent with earlier research emphasizing that data quality should be understood from the viewpoint of data consumers rather than only as a technical property [1]. It also reflects the broader transition from conventional database management toward dataspaces capable of organizing heterogeneous and distributed information resources [2], as well as the growing role of big data analytics in decision-making and system optimization [3]. Related studies in electronics have similarly highlighted the importance of reliable information interaction, timely data delivery, and multisource information fusion in connected and sensor-based environments [4–6]. Building on these directions, the present Special Issue illustrates how data technologies can be developed across both algorithmic and system-level contexts.

2. Overview of the Contributions

Several contributions focus on data as a managed and valuable resource. Contribution 1 addresses business data-resource selection in multi-value-chain data spaces. Instead of treating data management as passive storage, the study combines data quality evaluation, data utility assessment, and collaborative filtering to recommend resources that better match business analysis needs. Contribution 9 extends the concern with data usefulness to synthetic data generation. It shows that preserving distributions and correlations alone may be insufficient when real data contain meaningful clusters, and it proposes a cluster-wise generation method to better retain structural properties for downstream analysis.

Contribution 6 also contributes to data-resource quality by introducing SponSpeech, a spontaneous, informal speech dataset designed to support more realistic punctuation-restoration evaluation. These contributions underline a common point: data are useful only when their quality, structure, and context are aligned with the intended analytical task.

A second group of contributions develops methods for intelligent and efficient data processing. Contribution 2 proposes PMHR, a path-based multi-hop reasoning framework for knowledge graphs that combines reinforcement learning, knowledge graph embeddings, and rule-enhanced rewards. By addressing sparse rewards, spurious paths, large action spaces, and runtime cost, the work contributes to interpretable and efficient knowledge reasoning. Contribution 3 applies social-network data to e-sports recommendation, using both popularity and satisfaction metrics derived from Twitter/X metadata. This dual-metric approach enables the identification of both mainstream and niche e-sports and demonstrates how social media data can support more nuanced recommendations.

Efficiency is a recurring concern in this group. In Contribution 6, EfficientPunct combines text-based predictions with multimodal acoustic and textual embeddings for punctuation restoration while reducing inference complexity. Contribution 7 approaches efficiency through image distillation, using discrete wavelet transform and modified principal component analysis to obtain compact image representations for classification. Contribution 8 addresses forecasting under limited and fluctuating data conditions through an elastic optimal adaptive GM(1,1) model that adjusts sequence length and incorporates stationarity testing. Although these studies differ in domain, they share a practical concern: data-processing methods must balance accuracy with computational feasibility.

A third group of contributions addresses trustworthy and application-oriented systems. Contribution 4 proposes a Hyperledger Fabric-based multi-channel architecture for Internet of Vehicles data exchange. By organizing data into channel-specific ledgers and enabling controlled cross-channel interaction, the study supports confidentiality, integrity, and interoperability among stakeholders such as manufacturers, insurers, and traffic-management organizations. Contribution 5 examines smart-terminal security through implicit identity authentication based on user posture perception and the fusion of keystroke and motion-sensor data. Together, these papers show that trust in data systems may be achieved through both secure infrastructure and continuous user verification.

The final two contributions emphasize reliability and system-level performance. Contribution 10 proposes a data-driven reliability assessment method for islanded microgrids under frequency-regulation scenarios. By integrating a system frequency response model with sequential Monte Carlo simulation, the study shows that static reliability assessment may underestimate operational risk when frequency dynamics are ignored. Contribution 11 addresses memory management in NUMA and CXL-based tiered-memory systems. Its LACX mechanism identifies shared pages and migrates them to CXL memory, improving bandwidth utilization while preserving DRAM locality for private data. These contributions demonstrate that data applications depend not only on models and datasets, but also on the reliability and efficiency of the infrastructures that support them.

3. Discussion

The contributions in this Special Issue point to a clear development in data-oriented electronics research: the main challenge is no longer limited to improving computational models in isolation. Instead, data must be made usable, trustworthy, and efficient throughout the complete lifecycle of management, processing, and application.

One important theme is data-centric design. Contributions 1, 6, and 9 show that the usefulness of later processing stages depends strongly on the quality, structure, and relevance of the underlying data. Industrial data must be evaluated according to both

https://doi.org/10.3390/electronics15102117

quality and utility; speech datasets should reflect realistic communication conditions; and synthetic data should preserve structural properties that matter for downstream tasks. These studies suggest that better data preparation, evaluation, and representation are as important as model improvement.

A second theme is domain-aware intelligence. The contributions adapt methods to the structures of specific application domains: paths and rules in knowledge graphs, engagement signals in e-sports communities, acoustic and textual cues in speech, wavelet-domain features in image classification, stationarity in time series, and frequency dynamics in microgrids. This domain awareness is important because data rarely becomes actionable through general-purpose computation alone. They must be interpreted within the constraints, risks, and objectives of particular environments.

A third theme is trust under practical constraints. Secure Internet of Vehicles data exchange, continuous authentication, interpretable knowledge graph reasoning, privacy-aware synthetic data generation, and frequency-sensitive reliability assessment all show that trust is multidimensional. It includes cybersecurity, privacy, explainability, operational reliability, and resilience to changing conditions. The contributions support a broader view of trustworthy data systems, in which security mechanisms, analytical models, and deployment environments must be considered together.

Finally, this Special Issue highlights resource-aware deployment. Efficient punctuation restoration, image distillation, adaptive forecasting, and CXL-based memory management all respond to the practical costs of data processing. As data-intensive applications expand, performance will increasingly depend on the interaction between algorithms, datasets, operating systems, hardware resources, and application requirements. This reinforces the need to study data management, processing, and application as connected components of a single technological ecosystem.

4. Future Research Directions

The studies collected in this Special Issue suggest several directions for further work.

First, data management systems should become more adaptive and utility-driven. In large industrial and organizational data spaces, users may not know in advance which data resources are most relevant to a specific task. Future systems should therefore assess data according to completeness, consistency, relevance, timeliness, uncertainty, privacy risk, cost, and analytical value.

Second, intelligent processing methods should more explicitly integrate learning with domain knowledge. The contributions on knowledge graph reasoning, time-series forecasting, speech processing, and microgrid reliability show that robust performance often depends on structural information, physical behavior, temporal stability, or task-specific metrics. Hybrid approaches may be particularly valuable when data are limited, noisy, incomplete, or changing over time.

Third, privacy-preserving and secure data use will remain a major challenge. Blockchain-based data exchange, synthetic data generation, continuous authentication, federated learning, differential privacy, access-control mechanisms, and edge computing represent complementary directions. A key issue will be how to preserve analytical utility while ensuring confidentiality, accountability, and operational feasibility.

Fourth, future applications will require closer connections between data processing and computing infrastructure. The work on CXL-based tiered memory illustrates that data movement and memory placement can directly influence application performance. Similar concerns are likely to arise in edge computing, distributed analytics, sensor networks, and high-throughput AI systems. For latency-sensitive, energy-constrained, or memory-intensive applications, algorithm design and system design should be considered jointly.

https://doi.org/10.3390/electronics15102117

Finally, a realistic evaluation will be essential. Several contributions rely on application-oriented data sources or environments, including industrial production datasets, social media data, spontaneous speech, user behavior data, energy-system simulations, and memory workloads. Continued progress will require benchmarks and evaluation protocols that better reflect the complexity of deployment conditions.

5. Conclusions

This Special Issue, "Future Technologies for Data Management, Processing, and Application," presents eleven contributions that examine how data can be selected, processed, secured, and applied in modern electronic systems. The papers cover industrial data-resource optimization, knowledge graph reasoning, social-media-based recommendation, blockchain-enabled Internet of Vehicles data exchange, implicit authentication, multimodal punctuation restoration, image distillation, adaptive forecasting, synthetic data generation, microgrid reliability assessment, and CXL-based memory management.

Taken together, the contributions show that future data-oriented research should not be judged by algorithmic accuracy alone. Data must first be evaluated and made usable; processing methods must reflect domain structure and computational constraints; and applications must operate securely and reliably in practical environments. This Special Issue presents data technologies not as isolated computational tools, but as integrated systems that connect data quality, intelligent processing, trustworthy operation, and deployment-aware optimization.

We hope that this collection will serve as a useful reference for researchers and practitioners working on data management, intelligent processing, and application-oriented electronic systems. We also hope that it will encourage further research on methods and infrastructures that connect data utility, trustworthy analytics, and practical implementation.

Funding: This research received no external funding.

Acknowledgments: The Guest Editors sincerely thank the authors who contributed their work to this Special Issue. We also acknowledge the anonymous reviewers for their careful evaluations and constructive comments, and the editorial team of Electronics for their professional support throughout the editorial process.

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

List of Contributions

1. Fan, B.; Sun, L.; Tan, D.; Pan, M. Optimal Selection Technology of Business Data Resources for Multi-Value Chain Data Space—Optimizing Future Data Management Methods. *Electronics* **2024**, *13*, 4690. https://doi.org/10.3390/electronics13234690.
2. Ma, A.; Yu, Y.; Shi, C.; Zhen, S.; Pang, L.; Chua, T.-S. PMHR: Path-Based Multi-Hop Reasoning Incorporating Rule-Enhanced Reinforcement Learning and KG Embeddings. *Electronics* **2024**, *13*, 4847. https://doi.org/10.3390/electronics13234847.
3. Wang, Y. Leveraging SNS Data for E-Sports Recommendation: Analyzing Popularity and User Satisfaction Metrics. *Electronics* **2025**, *14*, 94. https://doi.org/10.3390/electronics14010094.
4. Liu, Y.; Feng, Y.; Sakurai, K. Hyperledger Fabric-Based Multi-Channel Structure for Data Exchange in Internet of Vehicles. *Electronics* **2025**, *14*, 572. https://doi.org/10.3390/electronics14030572.
5. Hu, B.; Tang, S.; Huang, F.; Yin, G.; Cai, J. Implicit Identity Authentication Method Based on User Posture Perception. *Electronics* **2025**, *14*, 835. https://doi.org/10.3390/electronics14050835.
6. Beigi, H.; Liu, X.Y. Efficient Ensemble of Deep Neural Networks for Multimodal Punctuation Restoration and the Spontaneous Informal Speech Dataset. *Electronics* **2025**, *14*, 973. https://doi.org/10.3390/electronics14050973.
7. Sirakov, N.M.; Ngo, L.H. Automatic Image Distillation with Wavelet Transform and Modified Principal Component Analysis. *Electronics* **2025**, *14*, 1357. https://doi.org/10.3390/electronics14071357.

8. Li, T.; Nie, J.; Qiu, G.; Li, Z.; Ji, C.; Li, X. Time Series Forecasting via an Elastic Optimal Adaptive GM(1,1) Model. *Electronics* **2025**, *14*, 2071. https://doi.org/10.3390/electronics14102071.
9. Petricioli, L.; Humski, L.; Vranić, M. Preserving Clusters in Synthetic Data Sets Based on Correlations and Distributions. *Electronics* **2025**, *14*, 2230. https://doi.org/10.3390/electronics14112230.
10. Zhang, Y.; Shi, J.; Huang, S.; Geng, L.; Wang, Z.; Sun, H.; Yu, Q.; Yao, X.; Liu, D.; Zuo, W.; et al. Data Driven Analytics for Distribution Network Power Supply Reliability Assessment Method Considering Frequency Regulating Scenario. *Electronics* **2025**, *14*, 4009. https://doi.org/10.3390/electronics14204009.
11. Jeong, H.; Song, B.; Jo, M.; Jo, H. LACX: Locality-Aware Shared Data Migration in NUMA + CXL Tiered Memory. *Electronics* **2025**, *14*, 4235. https://doi.org/10.3390/electronics14214235.

References

1. Wang, R.Y.; Strong, D.M. Beyond Accuracy: What Data Quality Means to Data Consumers. *J. Manag. Inf. Syst.* **1996**, *12*, 5–33. [CrossRef]
2. Franklin, M.; Halevy, A.; Maier, D. From Databases to Dataspaces: A New Abstraction for Information Management. *SIGMOD Rec.* **2005**, *34*, 27–33. [CrossRef]
3. Gandomi, A.; Haider, M. Beyond the Hype: Big Data Concepts, Methods, and Analytics. *Int. J. Inf. Manag.* **2015**, *35*, 137–144. [CrossRef]
4. Kovtun, V.; Grochla, K.; Połys, K. Investigation of the Information Interaction of the Sensor Network End IoT Device and the Hub at the Transport Protocol Level. *Electronics* **2023**, *12*, 4662. [CrossRef]
5. Pu, C.; Yang, H.; Wang, P.; Dong, C. AoI-Bounded Scheduling for Industrial Wireless Sensor Networks. *Electronics* **2023**, *12*, 1499. [CrossRef]
6. El Boudani, B.; Dagiuklas, T.; Kanaris, L.; Iqbal, M.; Chrysoulas, C. Information Fusion for 5G IoT: An Improved 3D Localisation Approach Using K-DNN and Multi-Layered Hybrid Radiomap. *Electronics* **2023**, *12*, 4150. [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

Optimal Selection Technology of Business Data Resources for Multi-Value Chain Data Space—Optimizing Future Data Management Methods

Bo Fan [1,2,3,*], Linfu Sun [1,2], Dong Tan [3,4] and Meng Pan [3,4]

[1] School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 610032, China; sunlf@swjtu.edu.cn
[2] Manufacturing Industry Chains Collaboration and Information Support Technology Key Laboratory of Sichuan Province, Southwest Jiaotong University, Chengdu 610032, China
[3] National Engineering Laboratory of Industrial Big-Data Application Technology, CASIC, Chengdu 611756, China; tan.dong@htdata.com (D.T.); pan.meng@htdata.com (M.P.)
[4] Chengdu Aerospace Science & Industry Big Data Institute Ltd., CASIC, Chengdu 610037, China
* Correspondence: fanbo@my.swjtu.edu.cn

Abstract: In the field of industrial big data, the key issue in discovering data value lies not in overcoming the bottlenecks formed by analysis methods and data mining algorithms but in the difficulty of providing data element resources that meet business analysis needs. Due to the surge in data volume and the increasing reliance of enterprises on data-driven decision-making, future data management strategies are constantly evolving to meet higher quality and efficiency requirements. Data metadata resources that meet business analysis needs require high-quality data integration, standardization, and metadata management. The key is to ensure the consistency and availability of data to support accurate analysis and decision-making. By leveraging automation and machine learning, organizations can more effectively integrate and manage data metadata resources, thereby improving data quality and analytical capabilities. The multi-value chain data space is a digital ecological platform for organizing and managing industrial big data. Research on optimizing the supply of its business data resources is a significant topic. This paper studies the evaluation index system of data quality and data utility, constructs an evaluation matrix of business data resources, and addresses the issues of data sparsity and cold start in evaluation calculations through a data quality-utility-based evaluation model of business data resources. It investigates a business data resource algorithm based on collaborative filtering, forming a recommendation set of similar data quality-utility data resources to provide to data analysis users. Finally, using actual production datasets, the paper validates the business data resource evaluation model, compares the performance and effectiveness of three business data resource recommendation algorithms based on collaborative filtering, empirically demonstrates the recommendation accuracy and stability performance of the combined improved data quality-utility collaborative filtering algorithm (CFA-DQU), and provides technical research recommendations for optimization of business data resources.

Keywords: multi-value chain data space; data quality; data utility; evaluation model; recommendation algorithm

1. Introduction

Data are a crucial production factor that promotes value chain collaboration and enables full-factor production [1]. Therefore, in the era of digital transformation, the importance of data is becoming increasingly prominent. To better and faster uncover data value, analysts from Daimler–Chrysler, SPSS, and NCR jointly developed the Cross Industry Standard Process for Data Mining (CRISP-DM), providing a methodology for data mining and knowledge discovery [2]. In CRISP-DM, "business understanding and

data understanding" are placed at the forefront. In other words, in the field of big data for industry/manufacturing, the key issue in uncovering data value is not the bottleneck caused by analytical methods and data mining algorithms, but the difficulty in providing data elements that meet business analysis needs [3,4].

Data space, as a next-generation data management concept (focusing not on the storage structure and data structure themselves but on the value and application of the data), aligns with the management and service requirements for business data resources within the multi-value chain collaborative system, effectively addressing this shortcoming [5].

The main function of a data space is to process data into higher quality and more valuable data resources, transforming them into data assets, which data consumers can then access to meet their business analysis needs with the support of the data space. The multi-value chain data space is a cloud platform-based system that integrates multi-value chain business logic data, physical logic data, and data resource mapping relationships, providing an effective data management tool for retrieving business data resources. It serves as the foundational platform for multi-value chain business analysis in dynamic enterprise alliances within the industrial ecosystem, and as an enabling tool for shifting from business-driven to data-driven decision-making [6]. As the business collaboration network among enterprises in the multi-value data space continues to expand, the resulting business data resources are growing explosively [7].

However, in the field of multi-value chain business analysis, the uncertainty of cross-enterprise value chain business analysis imposes higher demands on data resources themselves. Determining what type of data resources are needed in the current context becomes the primary question facing data consumers. Against this backdrop, enhancing data quality and utility to strengthen the support for data analysis, based on the characteristics of business data resources and the requirements of business analysis, and evaluating and optimizing business data resources to provide enterprises' data consumers with business data resources that meet the needs of business analysis, constitutes the primary focus of this study.

Currently, the evaluation of business data resources in a multi-value chain data space mostly involves judging and evaluating data consistency and data missingness based on physical structure logic. However, this method only assesses data resources from the dimension of data quality management. The requirements for data quality vary in different fields and scenarios, resulting in poor comparability of evaluation results [8]. Existing data quality evaluation methods mostly focus on data completeness, accuracy, and consistency, with insufficient attention to data timeliness, relevance, and applicability [9]. Meanwhile, research on data resource recommendation requires mining a large amount of usage experience data to achieve [10]; data sparsity and cold start issues are also major obstacles to the recommendation of data resources [11].

Data analysis is a process that requires repeated experimentation and tuning of analytical algorithm models, involving substantial pre-processing work such as data extraction and acquisition. In the business analysis scenario of multi-value chain collaboration, how to quickly and efficiently provide business data resources in the data space is the core of this research. To address these needs, this paper will study the optimal model framework based on the evaluation and recommendation of business data resources from the perspectives of business logic and physical logic. We proposed an optimal selection technology for business data resources in a multi-value chain data space and, finally, verified the effectiveness and feasibility of the proposed methods and technologies through empirical research.

Fricker S.A. and Maksimov Y.V. [12] conducted research on a pricing model for data resources based on data quality. This article approaches the issue from the perspective of data transactions, analyzing the impact of quality levels on data analysis, but it does not address the impact of data utility on data analysis. Xiao Yunlong et al. [13] proposed a data selection method based on data utility evaluation to address the issue of unstable data utility. The primary goal of this method is to improve the recognition rate of data during selection. However, it overlooks the impact of data quality on data analysis. In

current research on data resource optimization, data quality and data utility are often treated as separate aspects, with frameworks and models developed independently. There is a lack of research that integrates data quality and utility into a unified perspective. This paper constructs an evaluation model based on "data quality utility" and conducts research on recommendation algorithms for business data resources, aiming to provide users of multi-value chain data spaces with data resources that meet their business needs.

The multi-value chain data space is a data-driven enabling platform. This paper addresses issues such as low data quality, low utility value, and imprecise matching of business data resource supply in the multi-value chain data space by proposing an optimal model framework for business data resources driven by data requirements. The framework is divided into two modules: a business data resource evaluation module and a recommendation module. The evaluation module employs a combination of quantitative and qualitative methods, along with dynamic evaluation updates, to address adaptability issues across multiple scenarios. It enables the evaluation of data quality, data utility, and the combined metric of data quality utility, laying a foundation for the supply of business data resources. Additionally, it employs collaborative filtering algorithms based on data quality, data utility, and the combined quality-utility metric to avoid the cold-start problem associated with user-preference-based collaborative filtering algorithms. The recommendation module compares the recommendation effects of three types of collaborative filtering algorithms: those based on data quality, data utility, and the combined quality-utility metric. Through comparisons in various scenarios, the applicability of the proposed model framework is demonstrated. The main contributions of this paper are summarized as follows:

(1) A business data resource evaluation model combining quantitative and qualitative indicators of data quality and utility was established, and the evaluation of business data resources in the multi-value chain data space was completed.
(2) Based on the evaluated business data resources, collaborative filtering research was conducted to address the issues of data sparsity and cold start in business data resource recommendation algorithms.
(3) A combined optimal framework for the evaluation and recommendation of business data resources was constructed. The study developed business data resource recommendation algorithms based on data quality, data utility, and an improved combination of data quality utility. Using real datasets, a comparative analysis of the application effects of the three algorithms in different scenarios was conducted. The technical validation was successfully completed.

The organization of this paper is as follows:

Section 1 is the introduction, which provides an overview of the multi-value chain data space as a data-driven enabling platform, highlights its crucial role in proactive data management and services, analyzes the challenges in matching business data resource supply with demand, identifies the main issues to be addressed in this paper, and emphasizes the significance of studying optimal techniques for business data resources in a multi-value chain data space.

Section 2 reviews the research progress in related fields, analyzing the contributions and limitations of business data resource optimization research in four scenarios: business resource matching and recommendation, data quality and utility evaluation, business scenarios, and big data analysis. It points out the lack of research on proactive business data resource optimization techniques that meet data-driven decision-making needs.

Section 3 provides an overview of the optimal model framework and its main modules (the business data resource evaluation module and the recommendation module), clarifying that this framework serves as a proactive data management and service solution for multi-value chain data spaces.

Sections 4 and 5 discuss the evaluation and recommendation modules for business data resources. Section 4 elaborates on the evaluation metrics, models, and dynamic

update algorithms, while Section 5 provides a detailed explanation of three collaborative filtering-based recommendation algorithms.

Section 6 validates the overall optimization model and recommendation algorithms. Through experimental analysis, the evaluation results of business data resources were calculated. Based on these results, the paper verifies and analyzes the outcomes of three collaborative filtering algorithms based on data quality, data utility, and combined quality-utility evaluation.

Section 7 presents the conclusions of this paper.

2. Related Research Work

2.1. Multi-Value Chain Business Resource Matching and Recommendation

Third-party cloud platforms provide a means for integrating and sharing business data resources among enterprise clusters in multi-service value chains. As the number of enterprises on third-party platforms continues to increase, a large amount of business data resources is also accumulated. These business data resources are of great significance to enterprises in achieving cost reduction and efficiency improvement. However, faced with a large volume of business data resources, enterprises in the multi-value chain data space find it difficult to quickly and accurately obtain valuable and high-quality business data resources that meet their needs.

In response to the overload of business data resources in multi-service value chains, Wang Shuhai [14] employed recommendation algorithms to recommend high-quality business resources to different enterprise users, aiding in cost reduction and efficiency improvement. Yu Yang [15] proposed a bilateral matching model for business resources in multi-service value chains, which integrates the personalized business needs of both supply and demand sides. This model filters the most satisfactory data resources for the demand side from the vast information resources, optimizing the collaborative effect between the supply and demand sides.

The aforementioned research focuses on business needs and achieves inter-enterprise business collaboration through the matching of business data resources. Such collaborative enterprise decisions require analysts to have a comprehensive understanding of business operations to address and decide on inter-enterprise collaboration issues, limiting the ability to transfer the approach across scenarios. Moreover, the collaborative business needs between enterprises are often uncertain: if the business requirements are unclear, data-driven business collaboration is unlikely to occur. In the context of data-driven business decision-making, business decisions based on data insights represent the true need for enterprises to enhance operational efficiency. Therefore, the optimal model framework for business data resources proposed in this paper provides high-quality and high-value data foundational conditions for achieving effective data insights.

2.2. Optimal Selection of Data Resources Based on Data Quality and Utility Evaluation

In the context of digital transformation, the evaluation of data quality and data value is an important basis for the optimal selection of business data resources. Particularly, research on recommending data resources from the perspective of utility value has garnered increasing attention. Sadiq proposed a data quality management framework that covers multiple dimensions, such as completeness, accuracy, consistency, and timeliness [8]. This framework provides a systematic method for assessing the fundamental quality of data resources. However, the evaluation of data utility needs to consider more business value and application scenarios. Zhang studied the evaluation of data quality in collaborative information systems and proposed evaluation indicators that comprehensively consider data relevance and applicability [9]. These studies indicate that high-quality data should not only have basic accuracy and consistency but also high business relevance and applicability. From the perspective of utility value, researchers have proposed various data resource recommendation methods. Grover proposed a data utility evaluation framework that assesses the potential value of data by analyzing its actual application effects in different

business scenarios [1]. This method emphasizes that the selection of data resources should be based not only on their quality but also on their contribution to actual business decisions. Chen further studied data resource recommendation methods based on machine-learning technologies [16]. They proposed dynamically adjusting data resource recommendation strategies by constructing data utility models and using historical data and business feedback. This research shows that combining machine-learning and data mining techniques can significantly improve the accuracy and effectiveness of data resource recommendations.

Despite significant progress in data quality and utility evaluation, there are still some problems and challenges in this field. First, the standards for data quality and utility are not yet unified. Different evaluation methods in research and application scenarios lead to poor comparability of results [8]. Second, data resource recommendation methods still face many challenges in practical applications, such as data privacy protection and imperfect data-sharing mechanisms [17]. Third, most studies propose evaluation metrics for data quality and data utility separately to assess data resources. This reflects a passive management approach to data resources, lacking a proactive service perspective to unlock data value.

Starting with the goal of maximizing the release of business data value, this paper compares and integrates data quality and data utility evaluation metrics. By combining quantitative and qualitative methods with dynamic evaluation updates, it addresses the adaptability issues in multi-scenario evaluations.

2.3. Research on Optimal Selection of Data Resources for Business Scenarios

Different business scenarios have different requirements for data resources. Therefore, the optimal selection of data resources must be analyzed in conjunction with specific application scenarios. In recent years, many studies have emphasized the importance of scenario needs in data selection. Kraus studied the data application needs in different business fields and pointed out that data types and sources differ significantly in the retail, manufacturing, and healthcare sectors [18]. For instance, in the retail industry, customer behavior data and sales data are key resources, while in manufacturing, production data and equipment sensor data are more important. Understanding the data needs in different scenarios helps to more accurately select and configure data resources.

In the field of smart manufacturing, Wang proposed a data resource optimization method based on scenario needs [19]. By analyzing the data flows and information requirements in smart manufacturing systems, they proposed a strategy for dynamically adjusting data resources to meet the needs of different production stages and tasks. This research provides practical guidance for data resource configuration in different scenarios. Similarly, in the financial sector, Liu studied the needs of financial institutions for real-time transaction data and market intelligence data and proposed a framework for the optimal selection of data resources to ensure that data resources can support fast and accurate financial analysis and decision-making [17].

Current research on data resource optimization for business scenarios primarily focuses on the business domain, emphasizing responsiveness to business needs while neglecting the evaluation of data quality and utility. Such studies often assume that the data quality and value are already sufficient. However, from the perspective of industrial practice, environments with low-quality, low-value, and fragmented data are the actual norm. Consequently, there remains a lack of comprehensive research on data resource optimization based on dynamic data evaluation.

2.4. Research on Optimal Selection of Data Resources for Big Data Analysis and Processing

The demand for data analysis directly influences the optimal selection of data resources. With the rapid development of big data technologies and machine-learning algorithms, the demand for high-quality data has become increasingly urgent. Recent studies show that the effectiveness of data analysis largely depends on data quality and suitability. For example, Janssen et al. (2017) [20] pointed out that high-quality data can

significantly improve the accuracy and efficiency of decision-making (Emerald). Additionally, Bevilacqua and Ciarapica (2018) [21], through their research on big data analytics tools, emphasized the critical role of data adaptability in quality management, noting that improving data quality can directly optimize business decision-making (Life Science Alliance). These studies indicate that when enterprises rely on complex data ecosystems, the quality and adaptability of data are key factors in obtaining effective analytical results. Sun studied data preprocessing techniques in big data analysis and pointed out the importance of data cleaning, data integration, and data transformation in improving data quality and analysis accuracy [22]. Their research emphasizes that when selecting data resources, the analyzability of the data must be considered; that is, whether the data have been sufficiently preprocessed and prepared to support efficient analysis processes.

Furthermore, for the application of machine-learning and deep-learning models, Huang proposed a data selection framework to optimize the datasets for model training and testing [23]. Their research shows that the diversity and representativeness of data are crucial for building robust machine-learning models. Therefore, in the process of optimal data resource selection, attention should be paid to choosing datasets that cover a wide range of business scenarios and data characteristics, in order to enhance the model's generalization ability and predictive accuracy.

The optimal selection of business data resources is a complex and multidimensional task. By combining data quality and value evaluation, the application needs of different scenarios, and the specific requirements of data analysis, a comprehensive and effective data resource optimization framework can be constructed. This not only enhances the effectiveness of data analysis but also provides strong support for business decision-making, thereby achieving data-driven business innovation and enhancing competitive advantages in industry collaboration. This raises specific requirements for data resource supply: the required data must be massive, multidimensional, high-quality, and highly adaptable. However, data resource optimization frameworks for big data analysis primarily focus on improving data quality and synthesizing data, with an emphasis on data model optimization, rather than directly unlocking the business value of the data.

From the review and analysis of various studies on business data resource optimization, it is evident that optimizing business data resources is a complex and multidimensional task. The aforementioned studies addressed the optimization issues of business data resources in terms of precise matching, data quality improvement, multi-scenario reuse, and model optimization from different perspectives. However, they lack a comprehensive solution for business data resource optimization under a data-driven decision-making paradigm. This paper integrates data quality and value evaluation, the application requirements of various scenarios, and specific data analysis needs to develop a comprehensive and effective data resource optimization framework. This framework not only enhances the effectiveness of data analysis but also provides strong support for business decision-making, serving as a technical reference for achieving data-driven business collaboration and innovation.

3. Optimal Model Framework for Business Data Resource Selection in Multi-Value Chain Data Spaces

The primary work of the optimal selection technology for business data resources in multi-value chain data spaces is conducted within the framework of the business data resource optimal model. By studying the evaluation system of data quality and data utility, a business data resource evaluation model based on data quality utility is constructed. Through the evaluation results of data quality, data utility, and data quality utility of business data resources, the current status of business data resources can be objectively understood. At the same time, an evaluation matrix of the business data resource recommendation algorithm is generated to obtain the recommendation set of business data resources.

The optimal model framework for business data resources in multi-value chain data spaces mainly consists of two modules: a business data resource evaluation model based on data quality utility and a business data resource recommendation algorithm based on collaborative filtering. The evaluation model extracts data from the multi-value chain data space and evaluates business data resources from the perspectives of data quality and data utility. The uncertainty of multi-value chain business analysis requirements results in difficulty in accumulating historical usage data, making it challenging to study recommendation algorithms by constructing "user-rating" or "user-item" matrices for the optimal selection of business data resources.

Therefore, it is necessary to start with the current state of data and provide business personnel with data resources that meet their needs as much as possible. Evaluating business data resources not only helps to understand the current status of these resources but also addresses the cold-start problem in the optimal recommendation. We can further study the business data resource recommendation algorithm based on collaborative filtering through the data quality evaluation matrix, data utility evaluation matrix, and data quality-utility evaluation matrix. The optimal model framework for business data resources is shown in Figure 1 below:

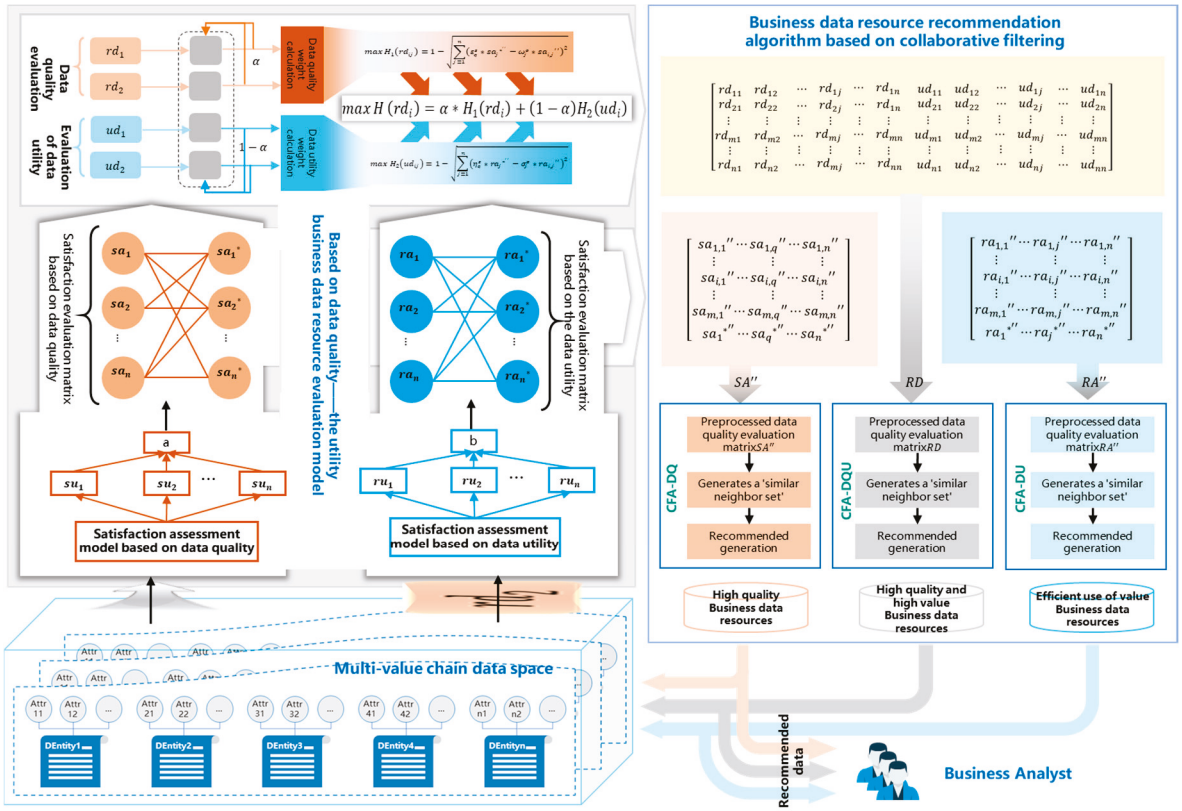


Figure 1. The optimal model framework for business data resources oriented to multi-value chain data space.

4. Business Data Resource Evaluation Model Based on Data Quality-Utility

This paper aims to construct a comprehensive evaluation system for business data resources based on "data quality-utility". It combines fuzzy theory, analytic hierarchy process (AHP), and entropy weight method to build the Business Data Resource Optimization Model (BDROM). Then, from the perspective of data quality-utility factors, BDROM is transformed into a multi-task optimization model based on data quality-utility factors. Finally, the "data quality-utility" evaluation matrix and its results are derived.

Definition 1. *The set of metadata quality factors* $SG = \{sg_1, sg_2, \cdots, sg_i, \cdots, sg_m\}$*, where data quality factors include accuracy, completeness, availability of data, etc.,* sg_i *represents the i-th data quality factor, N-dimensional data quality factor index set,* $SU = \{su_1, su_2, \cdots, su_q, \cdots, su_n\}$ *represents the q-th dimension index;* $Sa = \{sa_1, sa_2, \cdots, sa_q, \cdots, sa_n\}$*, Sa is the index evaluation set corresponding to SU,* sa_q *represents the q-th dimension index valuation;* $Sa^* = \{sa_1{}^*, sa_2{}^*, \cdots, sa_j{}^*, \cdots, sa_n{}^*\}$*, Sa* represents the n-dimensional evaluation ideal set corresponding to all indicators of business data resources,* $sa_q{}^*$ *is the q-th dimension indicator evaluation ideal value. The initial satisfaction matrix based on data quality factors is composed of the m-dimensional data quality factor set SG, the n-dimensional evaluation set Sa corresponding to SU, and the n-dimensional evaluation ideal set* $Sa^{*\prime}$ *corresponding to all indicators of data quality attributes:*

$$SA = \begin{bmatrix} sa_{1,1} \cdots sa_{1,q} \cdots sa_{1,n} \\ \vdots\ \vdots\ \vdots \\ sa_{i,1} \cdots sa_{i,q} \cdots sa_{i,n} \\ \vdots\ \vdots\ \vdots \\ sa_{m,1} \cdots sa_{m,q} \cdots sa_{m,n} \\ sa_1{}^* \cdots sa_q{}^* \cdots sa_n{}^* \end{bmatrix}$$

Definition 2. *The m-dimensional business data resource set* $RD = \{rd_1, rd_2, \cdots, rd_i, \cdots, rd_m\}$*, where* rd_i *represents the i-th business data resource, mainly including business data entities containing business domain, business process, and business entity mapping relationships; the n-dimensional business data resource index set* $RU = \{ru_1, ru_2, \cdots, ru_j, \cdots, ru_n\}$*, where* ru_j *represents the j-th dimension index;* $Ra = \{ra_1, ra_2, \cdots, ra_j, \cdots, ra_n\}$*, Ra is the index evaluation set corresponding to RU,* ra_j *represents the j-th dimension index evaluation value;* $Ra^* = \{ra_1{}^*, ra_2{}^*, \cdots, ra_j{}^*, \cdots, ra_n{}^*\}$*, Ra* represents the n-dimensional utility evaluation ideal set corresponding to all indicators of business data resources,* $ra_j{}^*$ *is the j-th dimension utility index evaluation ideal value. The initial satisfaction matrix based on data utility is composed of the m-dimensional business data resource set RD, the n-dimensional utility evaluation set Ra corresponding to RU, and the n-dimensional utility evaluation set Ra* corresponding to all indicators of business data resources:*

$$RA = \begin{bmatrix} ra_{1,1} \cdots ra_{1,j} \cdots ra_{1,n} \\ \vdots\ \vdots\ \vdots \\ ra_{i,1} \cdots ra_{i,j} \cdots ra_{i,n} \\ \vdots\ \vdots\ \vdots \\ ra_{m,1} \cdots ra_{m,j} \cdots ra_{m,n} \\ ra_1{}^* \cdots ra_j{}^* \cdots ra_n{}^* \end{bmatrix}$$

The business data resource selection model aims to identify the business data resource with the highest comprehensive satisfaction based on data quality and business value factors. The construction process of the business data resource selection model, as shown in Figure 2, consists of four main parts: (1) constructing a satisfaction evaluation index system based on data quality and utility factors; (2) forming a normalized satisfaction evaluation matrix based on data quality and utility; (3) solving a multi-objective optimization model based on data quality and utility factors; and (4) outputting the optimal business data resource.

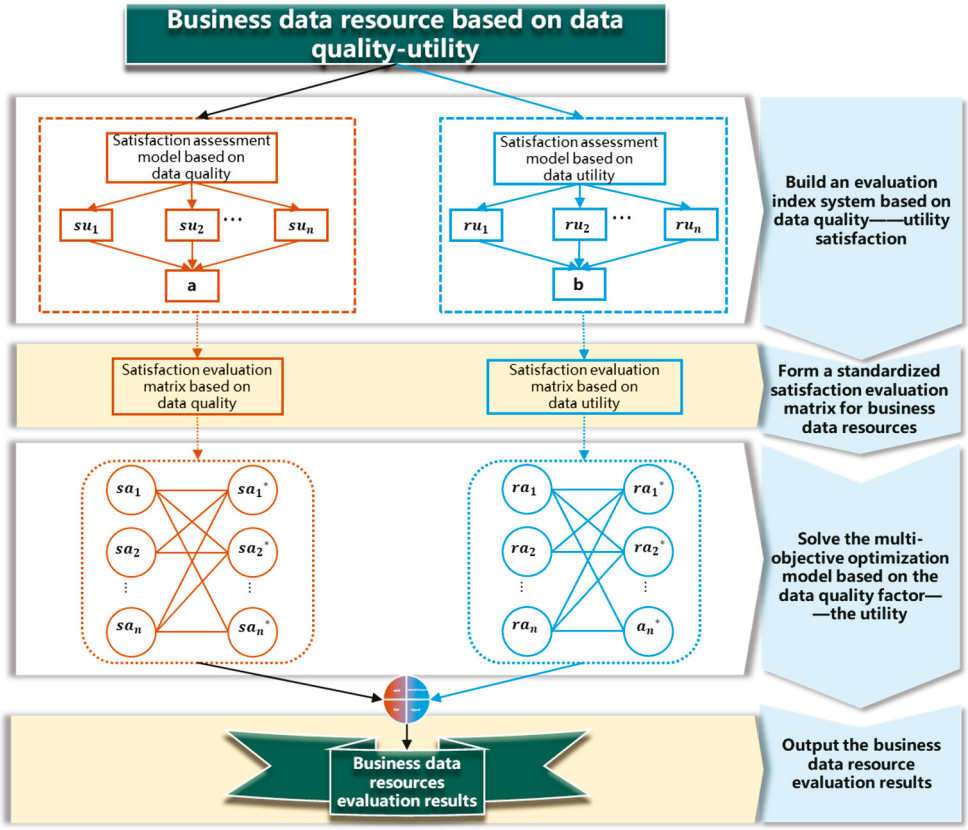


Figure 2. The framework for constructing business data resource evaluation models.

4.1. Satisfaction Evaluation Index System for Data Quality and Utility

In a multi-value chain data space, BDROM needs to meet both the requirements of the demand side for business data resources and the supply side's control over data quality factors. Therefore, this paper conducts an optimal analysis of business data resources from the perspectives of data quality and utility factors. The indicators are divided into quantitative and qualitative indicators based on their nature, and into positive/negative quantitative and qualitative indicators based on their type. For positive indicators, the larger the better, while for negative indicators, the smaller the better.

4.1.1. Satisfaction Evaluation of Business Data Resources

Based on data quality in the field of digital operations, the optimization of data quality is a crucial research topic. High-quality data are the foundation for ensuring the scientific nature of data analysis and decision-making. Researchers have systematically evaluated and optimized data quality from multiple dimensions, mainly including accuracy (Wang, R.Y., and Strong, D.M. proposed a data accuracy evaluation model focusing on the correctness and precision of data records [24]), completeness (Pipino, L.L., Lee, Y.W., and Wang, R.Y. studied data completeness issues and proposed a comprehensive method for detecting and filling data gaps [25]), consistency (Batini, C. and Scannapieco, M. proposed a data consistency checking framework capable of automatically detecting and fixing consistency issues in data [26]), timeliness (Ge, M. and Helfert, M. studied the impact of data timeliness on decision-making and proposed a real-time data quality monitoring system [27]), and accessibility (Oliveira, P., Rodrigues, F., and Henriques, P.R. proposed a data accessibility evaluation model that continuously optimizes data access paths and speeds through user feedback mechanisms [28]).

The main task of optimization based on data quality factors is to find the data that best meets business needs from the set of data quality factors. Combining the main research results in the current data quality evaluation, this paper determines the construction

of a satisfaction evaluation index system based on data quality from the dimensions of completeness, accuracy, usability, and consistency, as shown in Table 1.

Table 1. Satisfaction evaluation indicators for business data resources based on data quality.

Indicator	Indicator Value	Indicator Description	Indicator Nature	Calculation Rules	Rule Description
Data Integrity su_1	sa_1	Status of Missing Data Items	Quantitative Positive	$sa_1 = 1 - \frac{C_m}{C_w}$	C_m is the number of missing data items: C_w is the number of rows in the business data entity
Data Accuracy su_2	sa_2	Data Format Compliance, within the value range	Quantitative Positive	$sa_2 = 1 - \frac{C_{ex}}{C_w}$	C_{ex} is the number of data items exceeding the value range; data types, ranges, lengths, and precisions that exceed the set value range
	sa_3	Data Redundancy Rate	Quantitative Positive	$sa_3 = \frac{C_{reDE}}{C_{DE}}$	C_{reDE} represents the number of duplicate data entities, C_{DE} is the number of data entities from statistics
Data Usability su_3	sa_4	Whether Data is Reliably Provided to Users	Quantitative Positive	$sa_4 = 1 - \frac{C_{inv-w}}{C_{inv}}$	C_{inv-w} is the number of times data retrieval failed within a given period; C_{inv} is the total number of times the data was retrieved within the same period
Data Consistency su_4	sa_5	Whether Data Complies with Business Norms and Standards	Quantitative Positive	$sa_5 = 1 - \frac{C_{quw}}{C_w}$	C_{quw} represents the number of inconsistencies for the same data item across different data entities

The following explanations are provided for the quantitative index calculations in Table 1:

① su_1 represents data integrity, and its indicator value sa_1 is calculated as the ratio of missing values of data items to the number of rows in the data entity, i.e., $sa_1 = 1 - \frac{C_m}{C_w}$, where C_m is the number of missing data items and C_w is the number of rows in the business data entity. This metric measures the extent to which data elements are populated with values.

② su_2 represents data accuracy, and its indicator value is obtained through a composite calculation of sa_2 and sa_3 ($su_2 = sa_2 - 0.5 \cdot sa_3$). sa_2 is calculated as the ratio of the number of data items exceeding the preset value range to the number of rows in the data entity, i.e., $sa_2 = 1 - \frac{C_{ex}}{C_w}$, where C_{ex} represents the number of data items exceeding the value range. The data type, range, length, and precision exceed the preset value range, and this metric measures the extent to which the described data entity reflects the true value. sa_3 represents the duplication rate of the data entity, i.e., $sa_3 = \frac{C_{reDE}}{C_{DE}}$, where C_{reDE} is the number of duplicate data entities and C_{DE} is the total number of data entities counted.

③ su_3 represents data availability, and its indicator value sa_4 represents the success rate of data calls over a period of time. The higher the success rate, the higher the availability.

④ su_4 represents data consistency, and indicator value sa_4 is qualitatively determined by whether the data conform to business standards and norms, i.e., $sa_4 = 1 - \frac{C_{inv-w}}{C_{inv}}$, where C_{inv-w} is the number of failed data calls over a period of time and C_{inv} is the total number of data calls during that period. This metric primarily reflects the inconsistency rate of identical data items across different data entities at the same stage.

4.1.2. Satisfaction Evaluation of Business Data Resources Based on Data Utility

The utility evaluation of data resources is a crucial aspect of data management and data science, directly influencing the decision-making processes and competitiveness of

enterprises and organizations. Olson John A. proposed the concept of "Data-as-a-Service" (DaaS) in 2010, emphasizing the value of data sharing and reuse [29]. When conducting business-oriented data analysis, business personnel engage in extensive trial and error in data selection, thereby forming experience in quickly selecting data resources. Evaluating business data resources based on this experience directly reflects the utility value of the data.

This paper understands that the essence of optimizing business data resources from the perspective of data utility is for the demand side to find the optimal business data resource that meets its business needs from a set of business data resources. To determine the evaluation indicators for data utility, Chen [16] quantified the business value of data by calculating its utility in different application scenarios. Drawing on this approach, this study identifies factors influencing data utility as follows: the quality of business data resources provided to customers, cost-utility, timeliness, access volume, data coverage, customer satisfaction, and data security. Therefore, the satisfaction evaluation index system for business data resources based on data utility is shown in Table 2.

Table 2. Satisfaction evaluation indicators for business data resources based on data utility.

Indicator	Indicator Value	Indicator Description	Indicator Nature
Quality ru_1	ra_1	Quality of Business Data Resources Provided to Users	Quantitative Positive
Maintenance Cost ru_2	ra_2	Cost of Business Data Resources Provided to Users	Quantitative Negative
Timeliness ru_3	ra_3	Speed of Business Data Resources Provided to Users	Quantitative Positive
Access Volume ru_4	ra_4	Number of User Clicks on Business Data Resources	Quantitative Positive
User Satisfaction ru_5	ra_5	User Satisfaction with Business Data Resources Provided	Qualitative Positive
Data Security ru_6	ra_6	Security of Business Data Resources Provided to Users	Qualitative Positive
⋮	⋮	⋮	⋮

The following explanations are provided for the calculation of the indicators in Table 2:

① ru_1 represents data quality, and its index value ra_1 is determined by the accuracy, completeness, consistency, and usability of business data resources.

② ru_2 represents maintenance cost, and its index value ra_2 is primarily determined by the time cost of performing quality checks on business data resources, where $ru_2 = R_C$, with R_C representing the time expenditure for maintaining business data resources.

③ ru_3 represents timeliness, and its index value ra_3 is determined by the length of the business data update cycle and the number of updates within the cycle, where $ru_3 = \frac{F_d}{T_{Ud}}$, with T_{Ud} representing the length of the business data update cycle and F_d representing the number of updates within the cycle.

④ ru_4 represents access volume, and its index value ra_4 is determined by the number of user clicks on business data resources, expressed as $ru_4 = N_c$, with N_c representing the number of user clicks on business data resources.

⑤ ru_5 represents user satisfaction, and its indicator value ra_5 is qualitatively determined by users' satisfaction with data and services. The processing of this indicator is based on fuzzy theory, which quantifies fuzzy linguistic terms. The specific calculation process refers to Equation (2).

⑥ ru_6 represents data security, and its indicator value ra_6 is qualitatively determined by the confidentiality, integrity, and availability of business data resources. The calculation of these qualitative indicators follows the same process as ru_5.

4.2. Constructing a Normalized Evaluation Matrix Based on Data Quality and Utility

Due to the difficulty of integrating qualitative indicators such as data security and user satisfaction with other objective quantitative indicators for a comprehensive quantitative calculation of the utility of business data resources, this paper chooses to use fuzzy linguistic terms such as "very good", "good", "average", "poor", and "very poor" to subjectively describe user satisfaction and data security. The purpose of using fuzzy terms for subjective

evaluation is to utilize fuzzy theory to handle the fuzziness of these indicators and quantify these qualitative indicators. The relevant concepts of fuzzy theory are defined as follows:

Definition 3. *With a domain U, where the elements in the domain are x, any mapping μ_A from U to the closed interval $[0,1]$ is defined as follows:*

$$\mu_A : U \to [0,1]$$

Here, a fuzzy subset A on the domain U is referred to as a fuzzy set, and $\mu_A(x)$ is called the membership degree. When x varies within the domain U, $\mu_A(x)$ is called the membership function of the fuzzy set A (the membership function reflects the degree to which an element x in the domain belongs to the set A). Similarly, the domain of qualitative indicators $DOM^U = \left\{ Dom_1^U,\ Dom_2^U,\ \cdots,\ Dom_k^U,\ \cdots,\ Dom_n^U \right\}$ represents all qualitative indicator sets, where Dom_k^U is the k-th qualitative indicator in DOM^U and $Dom_k^U \in (RD \cup SG)$.

Definition 4. *A hedonic operator is an operation or function that modifies the membership degree of a fuzzy set. It is used to alter or adjust the membership function $\mu_A(x)$. Using H_γ as a hedonic operator to quantitatively describe the membership function of a fuzzy set, the transformed membership function is $H_\gamma(\mu_A(x))$, and the calculation formula is as follows:*

$$H_\gamma(\mu_A(x)) = \frac{\mu_\gamma(x)}{\mu_\tau(x)} \tag{1}$$

Here, γ represents different levels of hedonic operator fuzzy linguistic terms, $\mu_\gamma(x)$ represents the number of users evaluating x with the γ hedonic operator fuzzy linguistic terms, $\mu_\tau(x)$ represents the total number of users evaluating x, and $H_\gamma(\mu_A(x))$ represents the membership degree of the γ-level hedonic operator fuzzy linguistic terms. The membership degrees for different hedonic operator fuzzy linguistic term levels for each qualitative indicator are different, as shown in Table 3.

Table 3. Membership degrees of different hedonic operator fuzzy linguistic terms.

Hedonic Operator	γ	$H_\gamma(\mu_A(x))$
Very Poor	0.1	$H_{0.1}(\mu_A(x)) = \frac{\mu_{0.1}(x)}{\mu_\tau(x)}$
Poor	0.3	$H_{0.3}(\mu_A(x)) = \frac{\mu_{0.3}(x)}{\mu_\tau(x)}$
Average	0.5	$H_{0.5}(\mu_A(x)) = \frac{\mu_{0.5}(x)}{\mu_\tau(x)}$
Good	0.7	$H_{0.7}(\mu_A(x)) = \frac{\mu_{0.7}(x)}{\mu_\tau(x)}$
Very Good	0.9	$H_{0.9}(\mu_A(x)) = \frac{\mu_{0.9}(x)}{\mu_\tau(x)}$

The evaluation value of qualitative indicators is represented by the sum of the membership degrees of all fuzzy linguistic terms. The calculation formula is as follows:

$$Dom_k = \sum \gamma \cdot H_\gamma\left(\mu_A\left(Dom_k^U\right)\right) \tag{2}$$

where Dom_k represents the evaluation value of the k-th qualitative indicator.

According to the data utility evaluation indicator system, all qualitative indicators are positive qualitative indicators. Therefore, the qualitative indicators in the data utility factors are quantified using fuzzy theory on fuzzy linguistic terms.

Using the calculation formulas for $ra_1, \cdots, ra_4$, and Formulas (1) and (2), the initial satisfaction evaluation matrices SA based on data quality factors and RA based on data

utility factors are quantified, resulting in the quantified satisfaction evaluation matrices SA' and RA' as follows:

$$SA' = \begin{bmatrix} {sa_{1,1}}' \cdots {sa_{1,q}}' \cdots {sa_{1,n}}' \\ \vdots\vdots\vdots \\ {sa_{i,1}}' \cdots {sa_{i,q}}' \cdots {sa_{i,n}}' \\ \vdots\vdots\vdots \\ {sa_{m,1}}' \cdots {sa_{m,q}}' \cdots {sa_{m,n}}' \\ {sa_1}^{*\prime} \cdots {sa_q}^{*\prime} \cdots {sa_n}^{*\prime} \end{bmatrix}$$

$$RA' = \begin{bmatrix} {ra_{1,1}}' \cdots {ra_{1,j}}' \cdots {ra_{1,n}}' \\ \vdots\vdots\vdots \\ {ra_{i,1}}' \cdots {ra_{i,j}}' \cdots {ra_{i,n}}' \\ \vdots\vdots\vdots \\ {ra_{m,1}}' \cdots {ra_{m,j}}' \cdots {ra_{m,n}}' \\ {ra_1}^{*\prime} \cdots {ra_j}^{*\prime} \cdots {ra_n}^{*\prime} \end{bmatrix}$$

Among them, SA' is the satisfaction evaluation matrix based on data quality factors, ${sa_{i,q}}'$ represents the evaluated value of the q-th dimension indicator of the i-th business data resource after quantification, and ${sa_q}^{*''}$ is the ideal evaluated value of the q-th dimension indicator after quantification; RA' is the satisfaction evaluation matrix based on data utility, ${ra_{i,j}}'$ represents the evaluated value of the j-th dimension indicator of the i-th business data resource after quantification, and ${ra_j}^{*''}$ is the ideal evaluated value of the j-th dimension indicator after quantification.

Due to the different dimensions of indicators in the satisfaction evaluation system for data quality-utility factors, in order to achieve the highest satisfaction in comprehensive indicator evaluation, $BDROM$ eliminates the dimensional impact by non-dimensionalizing the indicator values. At the same time, to eliminate the impact of inconsistent indicator types, the reverse indicator values are converted to positive indicator values. This study adopts the non-dimensionalization of positive indicators (data compressed within the $[0, 1]$ range) and reverse indicators (data compressed within the $[0, 1]$ range and the data direction reversed). The non-dimensional values of positive and reverse indicators are obtained through Equations (3) and (4), and then the maximum value is used as the standard for evaluating the quality-utility factors of the data.

$$f'_{a_{u,v}} = \frac{fa_{u,v} - min\left\{fa_{u,v}\right\}}{max\left\{fa_{u,v}\right\} - min\left\{fa_{u,v}\right\}}, max\left\{fa_{u,v}\right\} - min\left\{fa_{u,v}\right\} \neq 0 \tag{3}$$

$$f'_{a_{u,v}} = \frac{max\left\{fa_{u,v}\right\} - fa_{u,v}}{max\left\{fa_{u,v}\right\} - min\left\{fa_{u,v}\right\}}, max\left\{fa_{u,v}\right\} - min\left\{fa_{u,v}\right\} \neq 0 \tag{4}$$

Among them, $fa_{u,v}$ represents the value of the v-th dimension indicator of the u-th object, $max\left\{fa_{u,v}\right\}$ represents the maximum value of the v-th dimension indicator of the u-th object, $min\left\{fa_{u,v}\right\}$ represents the minimum value of the v-th dimension indicator of the u-th object, and $f'_{a_{u,v}}$ represents the non-dimensional value of the v-th dimension indicator of the u-th object. These non-dimensionalized indicator values range within $[0, 1]$, with values closer to 1 being more favorable.

Therefore, the non-dimensionalized satisfaction evaluation matrices RA'' and SA'' are as follows:

$$SA'' = \begin{bmatrix} sa_{1,1}'' \cdots sa_{1,q}'' \cdots sa_{1,n}'' \\ \vdots\vdots\vdots \\ sa_{i,1}'' \cdots sa_{i,q}'' \cdots sa_{i,n}'' \\ \vdots\vdots\vdots \\ sa_{m,1}'' \cdots sa_{m,q}'' \cdots sa_{m,n}'' \\ {sa_1}^{*''} \cdots {sa_q}^{*''} \cdots {sa_n}^{*''} \end{bmatrix}$$

$$RA'' = \begin{bmatrix} ra_{1,1}'' \cdots ra_{1,j}'' \cdots ra_{1,n}'' \\ \vdots\vdots\vdots \\ ra_{i,1}'' \cdots ra_{i,j}'' \cdots ra_{i,n}'' \\ \vdots\vdots\vdots \\ ra_{m,1}'' \cdots ra_{m,j}'' \cdots ra_{m,n}'' \\ {ra_1}^{*''} \cdots {ra_j}^{*''} \cdots {ra_n}^{*''} \end{bmatrix}$$

Among them, SA'' is the non-dimensionalized satisfaction evaluation matrix based on data quality factors, $sa_{i,q}{}''$ represents the non-dimensionalized evaluation value of the q-th dimension indicator of the i-th business data resource, and ${sa_q}^{*''}$ is the ideal non-dimensionalized evaluation value of the q-th dimension indicator. RA'' is the non-dimensionalized satisfaction evaluation matrix based on data utility, $ra_{i,j}{}''$ represents the non-dimensionalized evaluation value of the j-th dimension indicator of the i-th business data resource, and ${ra_j}^{*''}$ is the ideal non-dimensionalized evaluation value of the j-th dimension indicator. This is used to solve the multi-task optimization model based on data quality utility.

The quantitative evaluation indicators of business data resources have clear numerical values (such as missing rate, duplication rate, access volume, etc.) with large data sample dispersion and a high degree of data quantification. The entropy weight method can automatically determine the weight of each indicator based on actual changes in the data, making it suitable for datasets with a high degree of quantification. Therefore, this paper adopts the entropy weight method [30] to calculate the objective weights of the evaluation indicators involved in data quality and data utility factors. The objective weights of these evaluation indicators are assigned as shown in Algorithm 1.

Algorithm 1: Entropy Weight-Based Weight Allocation Algorithm for Quantitative Indicators.

Input	Quantitative indicator elements of SA'' or RA''
Output	Objective weights ω_j of quantitative indicators and objective evaluation results BDR_{Score} for each indicator
Step 1	Perform data standardization on the elements of the quantitative indicator columns in SA'' and RA'', $P_{ij} = \frac{ra_{i,j}{}''}{\sum_{i=1}^{m} ra_{i,j}{}''}$
Step 2	Calculate the information entropy weight of each indicator using the entropy value formula, $g_j = 1 - \left(-\frac{1}{ln(m)} \sum_{i=1}^{m} P_{ij} ln\left(P_{ij}\right)\right)$
Step 3	Normalize the weights to determine the final weights. The normalization formula is $\omega_j = \frac{g_j}{\sum_{j=1}^{n} g_j}$
Step 4	Perform-weighted summation of scores to calculate the result of BDR_{QScore}

The dataset of qualitative indicators for business data resources consists of clearly defined descriptive levels, such as user satisfaction and data security. Users select their experience with the data based on categories like "excellent, good, average, poor, very poor", which have already been clearly structured into hierarchical levels. Therefore, this paper chooses to use the Analytic Hierarchy Process (AHP) [31] to calculate the subjective

weights of the evaluation indicators involved in data quality and utility factors. The construction process is shown in Algorithm 2.

Algorithm 2: Principal Component Analysis-Based Weight-Allocation Algorithm for Qualitative Indicators.	
Input	Qualitative indicator elements of SA'' or RA''.
Output	Indicator weights μ_j and objective evaluation results BDR_{UScore} for each indicator.
Step 1	Extract elements from the qualitative indicator columns in SA'' and RA'' and construct a judgment matrix. $R = \begin{bmatrix} ra_1 & ra_2 & \cdots & ra_j & \cdots & ra_n \end{bmatrix}$
Step 2	Calculate the eigenvector $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_j & \cdots & \lambda_n \end{bmatrix}$ of the judgment matrix R, where the eigenvector represents the weights of the indicators.
Step 3	Normalize the weights to determine the final weights, $\lambda_j = \frac{\lambda_j}{\sum_{j=1}^{n} \lambda_j}$
Step 4	Calculate the consistency index CI and the consistency ratio CR to check the consistency of the judgment matrix.Where $CI = \frac{\lambda_{max} - \boldsymbol{n}}{n-1}$($n$ is the order of the judgment matrix), $CR = \frac{CI}{RI}$ and RI is the random consistency index.
Step 5	If the consistency check is not passed, adjust the judgment matrix using pairwise comparisons and then repeat Steps 2 to 4 until the consistency check is passed.
Step 6	Perform weighted summation of scores to calculate the result of BDR_{UScore}

To adapt to the dynamic environment of business data resource evaluation, especially when the data's missing rate increases or user satisfaction decreases over a certain period, the evaluation of business data resources is updated accordingly. The process of updating the indicator weight is shown in Algorithm 3.

Algorithm 3: Indicator Weight Update Algorithm.	
Input	Data quality ru_1, user satisfaction ru_5.
Output	Objective weight of quantitative indicators ω_j and weight of qualitative indicators μ_j.
Step 1	Judging trigger conditions for the data update cycle.
Step 1.1	Set the data update cycle T = 24.
Step 1.2	Set the trigger condition for indicator weight recalculation : $ra_1 \leq \rho$; $ra_5 \leq \sigma$.
Step 2	Start Algorithm 1.
Step 3	Start Algorithm 2.
Step 4	Complete the recalculation of the indicator weights.

4.3. Construction of a Business Data Resource Evaluation Model Based on Data Quality Utility

On the basis of RA'' and SA'', solving the problem of maximizing satisfaction with data quality-utility factors is essentially a multi-objective optimization problem. Therefore, this paper transforms $BDROM$ into a multi-objective optimization model based on data quality-utility factors. From the perspective of data quality-utility factors, it constructs the corresponding objective functions for the multi-objective optimization model and solves the multi-objective optimization model using a linear weighting method. In summary, this study uses subjective and objective weights to reflect the importance of each indicator and constructs $BDROM$ based on the satisfaction evaluation matrices RA'' and SA''.

$$max\ H_1(rd_{ij}) = 1 - \sqrt{\sum_{j=1}^{n} \left(\varepsilon_q^s * {sa_j}^{*''} - \omega_j^o * sa_{i,j}'' \right)^2} \tag{5}$$

Subject. to. $\varepsilon_q^s \geq 0, \sum_{j=1}^{n} \varepsilon_q^s = 1, \omega_j^o \geq 0, \sum_{j=1}^{n} \omega_j^o = 1;$

$$max\ H_2(ud_{ij}) = 1 - \sqrt{\sum_{j=1}^{n} \left(\eta_q^s * {ra_j}^{*''} - \sigma_j^o * {ra_{i,j}}'' \right)^2} \tag{6}$$

Subject. to. $\eta_q^s \geq 0, \sum_{q=1}^{n} \eta_q^s = 1, \sigma_j^o \geq 0, \sum_{j=1}^{n} \sigma_j^o = 1$

Among them, $H_1(rd_i)$ represents the maximization of satisfaction with the data quality evaluation of business data resources, ε_q^s represents the subjective weight of the q-th dimension indicator of data quality obtained through the primary analytic hierarchy process, and ω_j^o represents the objective weight of the dimension indicator of business data resources calculated by the demand side using the entropy weight method. $H_2(rd_i)$ represents the maximization of satisfaction with the data utility evaluation, η_q^s represents the subjective weight of the q-th dimension indicator of data quality factors obtained by the demand side through the primary analytic hierarchy process, and σ_j^o represents the objective weight of the dimension indicator of data quality factors calculated by the demand side using the entropy weight method.

This paper transforms $BDROM$ into a multi-objective optimization model based on data quality utility. Equations (5) and (6) are the objective functions of the multi-objective optimization model. Then, the linear weighting method is used to handle the above objective functions, forming a new objective function, and the optimal result is obtained by solving the new objective function:

$$maxH(rd_i) = \alpha * H_1(rd_i) + (1 - \alpha)H_2(ud_i) \tag{7}$$

Subject. to. $\alpha \geq 0, (1 - \alpha) \geq 0, \alpha \in [0, 1]$

Among them, α represents the weight based on data quality factor matching, and $1 - \alpha$ represents the weight based on data utility matching. The weight α can be obtained by referring to the entropy weight method shown in Algorithm 1.

5. Business Data Resource Recommendation Algorithm Based on Collaborative Filtering

The value of data analysis in supporting business operations is crucial. IBM emphasizes that high-quality data can help enterprises make better business decisions, improve operational efficiency, and enhance customer satisfaction [32]. Furthermore, McKinsey highlights that as companies increasingly rely on complex data ecosystems, evaluating the quality, availability, and reliability of data is essential for gaining valuable business insights. They also stress that data transparency and centralized data procurement can optimize the use of data resources, prevent redundant purchases, and increase the utility value of the data [33]. Therefore, in specific use cases, the demand for business data resources is diverse, but business personnel generally focus on the quality and utility value of the data resources. Collaborative filtering-based recommendation algorithms require a large amount of historical user data to function, making the cold-start problem a significant obstacle in data resource recommendation research. As a result, this paper focuses on studying recommendation algorithms based on evaluations of data resource quality and utility, with an emphasis on scenarios that prioritize the matching and optimization of data quality and utility.

The recommendation algorithm is the core module of the business data resource optimization model for multi-value chain data spaces, and it is crucial for optimizing the matching of data resources. This paper designs a collaborative filtering algorithm based on data quality and another based on data utility. Additionally, it improves and designs a combined recommendation algorithm to address the issues of cold start and data sparsity. Finally, by comparing these three recommendation algorithms, this paper demonstrates the effectiveness of the proposed business data resource optimization technology for multi-value chain data spaces.

5.1. Collaborative Filtering Algorithm Based on Data Quality

The research on the Collaborative Filtering Algorithm Based on Data Quality (CFA-DQ) aims to recommend business data resources with higher data quality to business analysts. Through collaborative filtering based on data quality, analysts can identify data

with high scores in completeness, accuracy, and consistency, ensuring that high-quality data are prioritized for recommendation.

In Section 4.2, a satisfaction evaluation matrix based on data quality factors was constructed, which can serve as input data for CFA-DQ. The construction process of CFA-DQ is as shown in Algorithm 4.

Algorithm 4: Outlines the Main Steps of CFA-DQ.

Input	SA''
Output	High-quality data resource recommendation results in Top-N set
Step 1	Preprocess the data quality satisfaction evaluation matrix SA'' for business data resources; extract the data quality evaluation indicators of business data resources: data quality, data integrity, data accuracy, data availability, and data consistency. Generate the "similar neighbors set": by calculating the similarity of each data quality indicator of all data attributes, form the "similar neighbors set"; that is, calculate the data quality similarity between the target data attribute and other data attributes in SA''. This algorithm uses cosine similarity to calculate the data quality similarity of data attributes.
Step 2	$$SIM(sa_{ik}, sa_{jk}) = \frac{\sum_1^n \left((sa_{ik} - \overline{sa_{\mathbf{I}k}}) \cdot (sa_{jk} - \overline{sa_{\mathbf{I}k}})\right)}{\sqrt{\sum_1^n (sa_{ik} - \overline{sa_{\mathbf{I}k}})^2} \cdot \sqrt{\sum_1^n (sa_{jk} - \overline{sa_{\mathbf{I}k}})^2}}$$
Step 3	Recommendation generation: Based on the generated "similar neighbors set", recommend data attributes with highly similar data quality. Determine the accessible data attribute set from the columns of the "similar neighbors set" in the data quality satisfaction evaluation matrix SA''.
Step 4	Generating high-quality data resource recommendation results in Top-N set; select the top N ranked items that do not belong to the "similar neighbors set".
Step 5	Evaluate the accuracy of the recommendation results; calculate the Root Mean Square Error (RMSE) of the recommendation algorithm.

The main issue with CFA-DQ is the cold-start problem. CFA-DQ calculates the quality similarity of data attributes to search for a "similar neighborhood set" of data quality, thereby recommending high-quality data. However, for data resources that have not yet been evaluated for data quality, this algorithm struggles to make ideal recommendations.

5.2. Collaborative Filtering Algorithm Based on Data

Recommending business data resources to business personnel from the perspective of data utility is a common scenario. This includes evaluations of user satisfaction, timeliness, and security of data resources, leading to better recommendation results. The research on the Collaborative Filtering Algorithm Based on Data Utility (CFA-DU) aims to recommend high-value business data resources to business personnel. The construction process of the data utility evaluation matrix is detailed in Section 4.2, and it can serve as input data for CFA-DU. The construction process of CFA-DU is as shown in Algorithm 5.

Although CFA-DU uses historical usage performance indicators of data resources, to a certain extent, to address the cold-start problem and can filter and recommend individual utility indicator sets, the comprehensive utility evaluation of data is somewhat subjective, and the utility value is effective only in some similar datasets if the recommended data resources have low data.

Algorithm 5: The Main Steps of CFA-DU are Similar to Those of CFA-DQ.

Input	RA''
Output	High-utility value data resource recommendation results in Top-N set
Step 1	Preprocess the data utility satisfaction evaluation matrix RA'' for business data resources; extract the data quality utility indicators of business data resources: data quality, maintenance cost, timeliness, access volume, user satisfaction, data security. Generate the "similar neighbors set"; by calculating the similarity of each data utility evaluation indicator of all data attributes, form the "similar neighbors set"; that is, calculate the utility similarity between the target data attribute and other data attributes in RA''. This algorithm uses cosine similarity to calculate the utility similarity of data attributes.
Step 2	$$SIM(ra_{ik}, ra_{jk}) = \frac{\sum_1^n \left((ra_{ik} - \overline{sa_{I_k}}) \cdot (ra_{jk} - \overline{ra_{I_k}})\right)}{\sqrt{\sum_1^n (ra_{ik} - \overline{sa_{I_k}})^2} \cdot \sqrt{\sum_1^n (ra_{jk} - \overline{ra_{I_k}})^2}}$$
Step 3	Recommendation generation; based on the generated "similar neighbors set", recommend data attributes with highly similar data utility. Determine the accessible data attribute set from the columns of the "similar neighbors set" in the data utility satisfaction evaluation matrix RA''.
Step 4	Generate high-utility data resource recommendation results in Top-N set; select the top N-ranked items that do not belong to the "similar neighbors set".
Step 5	Evaluate the accuracy of the recommendation results; calculate the Root Mean Square Error (RMSE) of the recommendation algorithm.

5.3. Collaborative Filtering Recommendation Algorithm Based on Data Quality-Utility Combined Evaluation

Sections 5.1 and 5.2, respectively, introduced collaborative filtering algorithms based on data quality and data utility. Each of these algorithms has its own advantages and disadvantages. To achieve better data resource recommendations, this paper proposes the Collaborative Filtering Recommendation Algorithm Combining Improved Data Quality and Utility (hereinafter referred to as CFA-DQU). CFA-DQU mainly includes constructing a business data resource evaluation matrix based on data quality and utility, calculating the similarity of data quality and utility, generating a "similar neighbor set" based on data quality and utility, and generating a recommendation set based on data quality and utility.

The design idea of the combined improvement algorithm is to comprehensively consider the evaluation of business data resources in terms of data quality and utility, construct a data quality-utility evaluation matrix, and build a collaborative filtering algorithm based on this to generate recommendation results. The following details are the construction algorithm of the business data resource evaluation matrix based on data quality and utility.

One of the core tasks of CFA-DQU is to solve datasets with similar data attribute qualities and utilities based on the evaluation matrix with the highest satisfaction of data quality-utility factors, thereby generating recommendations. Section 4.3 defines the business data resource evaluation score matrix based on data quality and utility. The flowchart of the business data resource evaluation algorithm based on data quality and utility is shown Figure 3.

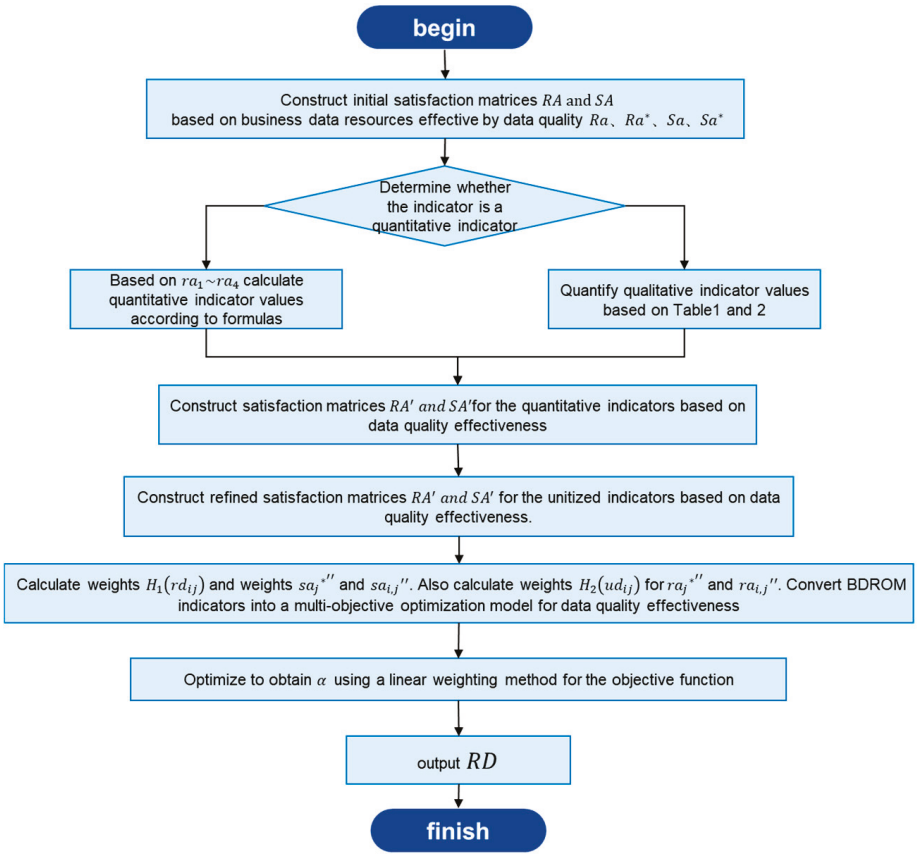


Figure 3. Business data resource evaluation algorithm process based on data quality utility.

The business data resource evaluation process based on data quality and utility is shown in Algorithm 6.

Algorithm 6: The Business Data Resource Evaluation Algorithm Based on Data Quality and Utility.

Input	$RD, RU, Ra, Ra^*; SG, SU, Sa, Ra^*$
Output	Return the evaluation results of business data resources RD.
Step 1	Construct the initial satisfaction matrix based on data utility quality.
Step 2	If the nature of the indicator is quantitative, then quantify the quantitative indicator values according to the calculation formulas $ra_1, \cdots, ra_4$ or else quantify the qualitative indicator values according to formulas (1) and (2); construct the satisfaction matrices RA' and SA' after quantifying the data quality-utility factor indicators.
Step 3	Using Equations (3) and (4) to non-dimensionalize the satisfaction matrix; construct the satisfaction matrices RA'' and SA'' based on the quantification of data quality-utility factor indicators.
Step 4	Using Equations (5) and (6), calculate the weights of $H_1\left(rd_{ij}\right)$, $sa_j{}^{*''}$, and $sa_{i,j}{}''$; similarly, calculate the weights of $H_2\left(ud_{ij}\right)$, $ra_j{}^{*''}$, and $ra_{i,j}{}''$. Transform BDROM into a multi-task optimization model based on data quality-utility factors.
Step 5	Use the entropy weight method to obtain α and form a new objective function, Equation (8), using the linear weighting method.
Step 6	Solve the new objective function to complete the value assessment of business data resources and output the evaluation results RD for business data resources.

At this point, the business data resource evaluation matrix based on data quality and utility is:

$$RD = \begin{bmatrix} rd_{11} & rd_{12} & \cdots & rd_{1j} & \cdots & rd_{1n} & ud_{11} & ud_{12} & \cdots & ud_{1j} & \cdots & ud_{1n} \\ rd_{21} & rd_{22} & \cdots & rd_{2j} & \cdots & rd_{1n} & ud_{21} & ud_{22} & \cdots & ud_{2j} & \cdots & ud_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ rd_{m1} & rd_{m2} & \cdots & rd_{mj} & \cdots & rd_{mn} & ud_{m1} & ud_{m2} & \cdots & ud_{mj} & \cdots & ud_{mn} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ rd_{n1} & rd_{n2} & \cdots & rd_{mj} & \cdots & rd_{nn} & ud_{n1} & ud_{n2} & \cdots & ud_{nj} & \cdots & ud_{nn} \end{bmatrix} \quad (8)$$

Using the business data resource evaluation matrix based on data quality and utility as data input, design the CFA-DQU.

The main steps of CFA-DQU are similar to those of Algorithm 5, as shown in Algorithm 7 below:

Algorithm 7: Outlines the Main Steps of CFA-DQU.

Input	*RD*
Output	High-quality and high-utility value data resource recommendation results in Top-N set.
Step 1	Preprocess the business data resources to form the data quality-utility evaluation matrix; *RD*: extract the data quality-utility indicators of business data resources. Generate a "similar neighbor set"; form the "similar neighbor set" by calculating the similarity of each data quality-utility evaluation indicator for all data attributes; that is, calculate the data quality-utility similarity between the target data attribute and other data attributes in *RD*. This algorithm uses cosine similarity to calculate the utility similarity of data attributes:
Step 2	$$SIM(ra_{ik}, ra_{jk}) = \frac{\sum_1^n \left((ra_{ik} - \overline{sa_{ik}}) \cdot (ra_{jk} - \overline{ra_{ik}})\right)}{\sqrt{\sum_1^n (ra_{ik} - \overline{sa_{ik}})^2} \cdot \sqrt{\sum_1^n (ra_{jk} - \overline{ra_{ik}})^2}}$$
Step 3	Recommendation generation; based on the generated "similar neighbor set", recommend data attributes with highly similar data quality utility. Determine the accessible data attribute set of the "similar neighbor set" from the columns in the data quality-utility satisfaction evaluation matrix *RD*.
Step 4	Generate high-quality, high-utility data resource recommendation results in Top-N set; select the top N results that are ranked highest but do not belong to the "similar neighbor set".
Step 5	Evaluate the accuracy of the recommendation results; calculate the Root Mean Square Error (RMSE) of the recommendation algorithm.

6. Experimental Calculations and Technical Validation

6.1. Dataset and Experimental Setup Description

The dataset for technical validation in this paper is sourced from the Value Chain Collaborative Cloud Platform [34]. It includes 22 business data entities from supply value chain, marketing value chain, parts (operation and maintenance) value chain, and after-sales service value chain, totaling 3486 data attributes. A single business data entity contains up to 397,962 rows of data. The dataset mainly includes records on data quality, user satisfaction, security, and access volume. Sample data are shown in Table 4 below:

Table 4. Sample data content of dataset.

Business Value Chain Domain	Primary Business Process	Secondary Business Process	Business Entity	Business Data Entity	Attribute	Number of Data Rows
Component Supply Value Chain Domain	Purchase Requisition Coordination Order Management	Single Small Batch Pull Production Mode Purchase Requisition Coordination Order Management	Purchase Requisition	Purchase Requisition Information	Purchase Requisition Code	347,599
Component Supply Value Chain Domain	Purchase Requisition Coordination Order Management	Single Small Batch Pull Production Mode Purchase Requisition Coordination Order Management	Purchase Requisition	Purchase Requisition Information	Purchase Requisition Date	347,599
Marketing Value Chain Domain	Sales Plan Management	Cargo, Pickup, Van, and Cargo Truck Franchising Mode Sales Plan Management	Sales Plan	Sales Plan Table	Reviewer	381,149
Marketing Value Chain Domain	Sales Plan Management	Cargo, Pickup, Van, and Cargo Truck Franchising Mode Sales Plan Management	Sales Plan	Sales Plan Table	Review Date	381,149
Spare Parts (Operations and Maintenance) Value Chain Domain	Spare Parts Warehouse Entry Management Process	Commercial Vehicle Driven Production Mode Spare Parts Warehouse Entry Process	Warehouse Entry Order	Warehouse Entry Order Header Information	Entry Type	160,127
Spare Parts (Operations and Maintenance) Value Chain Domain	Spare Parts Warehouse Entry Management Process	Commercial Vehicle Driven Production Mode Spare Parts Warehouse Entry Process	Warehouse Entry Order	Warehouse Entry Order Header Information	Document Creation Date	160,127
After-sales Service Value Chain Field	Warranty Management Process	Integrated Service Model Warranty Management Process for Passenger Vehicles	Warranty Document	Warranty Information	Rear Axle Model ID After Review	366,385
After-sales Service Value Chain Field	Warranty Management Process	Integrated Service Model Warranty Management Process for Passenger Vehicles	Warranty Document	Warranty Information	Rear Axle Code After Review	366,385

Since the multi-value chain data space provides basic preprocessing of business data, this paper focuses on the evaluation of data quality and data utility. The preprocessing is conducted directly according to the calculation instructions for each metric, yielding indicators such as the number of data rows, missing values, out-of-range data, data retrieval counts, data retrieval failure counts, and data inconsistencies, which are used to assess data quality and utility. The computer configuration used for the experiments includes cloud computing resources with a 3.1 GHz processor, 64 GB of memory, and 24 GB of video memory.

In order to directly validate the data resource optimization techniques proposed in this paper, the dataset was preprocessed according to the following steps. Figure 4 illustrates the process of duplicating and joining several data attribute tables of business data resources, constructing the dataset through data integration. The specific construction process is shown in Algorithm 8.

Algorithm 8: The Process of Integrating Datasets

Step 1	Identify the data source $\{DataR^{(1)}, DataR^{(2)}, \cdots, DataR^{(n)}\}$.
Step 2	Analyze the data schema of the data source and define the data model and partial mapping, $L^{i}_{schema} = \langle bs_i \leftarrow db_j \leftarrow entity_k \rangle$.
Step 3	Based on the partial data mapping, construct the semantic model, $DO^{(i)}_{L}$.
Step 4	Perform the copy operation on the attributes of $DO^{(i)}_{L}$ and create a full backup of the extracted attribute data.
Step 5	Use the connect operation to join the similar attributes of $DO^{(i)}_{L}$ mapped from the local data schema L^{i}_{schema}, forming a data entity attribute set $D^{(i)}$.
Step 6	Aggregate the data entity attribute set $D^{(i)}$ and its mapping L^{i}_{schema} to construct the data entity set D.

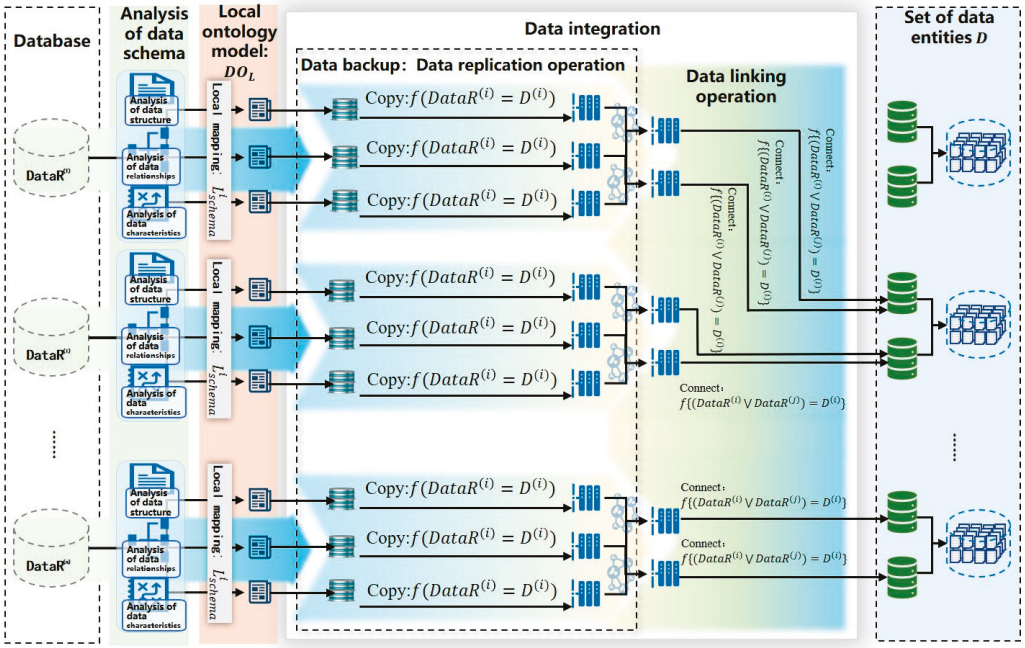


Figure 4. Data preprocessing process based on data integration.

6.2. Business Data Resource Evaluation Calculation

Based on the business data resource evaluation model of data quality and utility, we evaluate the data quality, data utility, and the combination of data quality and utility of business data resources. This provides dimensional support for the construction and research of business data resource recommendation algorithm models.

6.2.1. Data Quality Evaluation Calculation

(1) Preprocess the dataset

First, preprocess the dataset by tallying the missing values, data duplication, and data calls for each business data entity. According to the indicator calculation rules, quantitatively calculate the data quality score. The highest score for each item is taken as the ideal value, converting the data quality score into a satisfaction evaluation matrix based on data quality factors. Finally, dimensionless processing is performed to obtain a dimensionless satisfaction evaluation matrix of data quality factors.

According to the data quality evaluation indicators of business data resources in Table 1, calculate the data integrity, accuracy, availability, and consistency of the dataset and its preprocessing results for business data resources. The calculation results are shown in Table 5.

Table 5. Calculation results of data quality evaluation.

Business Data Entity	Attribute Number	Attribute	Data Integrity	Data Accuracy	Data Availability	Data Consistency
Purchase Request Information	1	Purchase Request Code	1	1	0.942	1
	2	Purchase Request Date	1	1	0.929	1
	3	Salesperson Response	1	1	0.927	1
	4	Remarks	1	1	0.953	1
	5	Supplier Code	1	1	0.891	1
	6	Salesperson Name	1	1	0.885	1
	7	Business Type	1	1	0.901	1
	8	Supplier Response Status	1	1	0.887	1
	9	Confirmation Status	1	1	0.956	1
	10	Publication Status	1	1	0.932	1
Purchase Request Details	1	Purchase Request Code	1	1	0.833	1
	2	Part Code	1	1	0.836	1
	3	Quantity Required	1	1	0.875	1
	4	Supply Guarantee Status	1	1	0.922	1
	5	Supplier Response	1	1	0.878	1
	6	Required Date	1	1	0.815	1
	7	Estimated Arrival Date	1	1	0.927	1
	8	Unit Price	0.979	1	0.933	1
	9	ID	1	1	0.907	1
...	...	...	...	...	...	...
Adjustment Information	12	Auditor Name	0.997	1	0.973	1
	13	Audit Date	1	1	0.901	1
	14	Data Transmission Status	1	1	0.885	1

(2) Data Quality Satisfaction Evaluation Matrix

According to the calculation rules of data quality indicators, compute the data integrity, accuracy, availability, and consistency to obtain the data quality satisfaction evaluation matrix. The last row of the matrix represents the ideal values for data integrity, accuracy, availability, and consistency, with the ideal values being the maximum values of each indicator.

$$SA' = \begin{bmatrix} 1 & 1 & 0.942 & 1 \\ 1 & 1 & 0.929 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0.885 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Based on Formulas (3) and (4), perform dimensionless calculations to obtain the dimensionless data quality satisfaction evaluation matrix.

$$SA'' = \begin{bmatrix} 1 & 1 & 0.854 & 1 \\ 1 & 1 & 0.821 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0.711 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{9}$$

(3) Data Quality Indicator Weight Calculation

Since the data quality evaluation indicators are all quantitative, the entropy weight method can be used to calculate the weights of the indicators.

Based on the data quality satisfaction evaluation matrix, standardize the indicator data using the following formula:

$$P_{ij} = \frac{ra_{i,j}''}{\sum_{i=1}^{m} ra_{i,j}''} \tag{10}$$

Here, P_{ij} represents the standardized data, $ra_{i,j}''$ represents the raw evaluation data of the i-th row and j-th column, and m represents the sample size of the dataset.

Use the entropy formula to calculate the information entropy weight of each indicator. The calculation formula is as follows:

$$g_j = 1 - \left(-\frac{1}{ln(m)} \sum_{i=1}^{m} P_{ij} ln(P_{ij}) \right) \tag{11}$$

Here, g_j represents the information entropy weight of the j-th indicator.

Normalize the weights according to the column distribution of the data quality satisfaction evaluation matrix. The formula for calculating the normalized weights is as follows:

$$\omega_j = \frac{g_j}{\sum_{j=1}^{n} g_j} \tag{12}$$

Through the sequential calculations of Formulas (10)–(12), the weights of the data quality evaluation indicators are obtained as follows:

$$W = [0.0149, 0.8159, 0.1559, 0.0133]$$

(4) Business Data Resource Quality Evaluation Results

Perform a weighted sum of the dimensionless satisfaction evaluation scores of the business data resources to obtain the data quality evaluation results, as follows:

$$BDR_{QScore} = \begin{bmatrix} sa_{1,1}'' \cdots sa_{1,q}'' \cdots sa_{1,n}'' \\ \vdots\vdots\vdots \\ sa_{i,1}'' \cdots sa_{i,q}'' \cdots sa_{i,n}'' \\ \vdots\vdots\vdots \\ sa_{m,1}'' \cdots sa_{m,q}'' \cdots sa_{m,n}'' \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \tag{13}$$

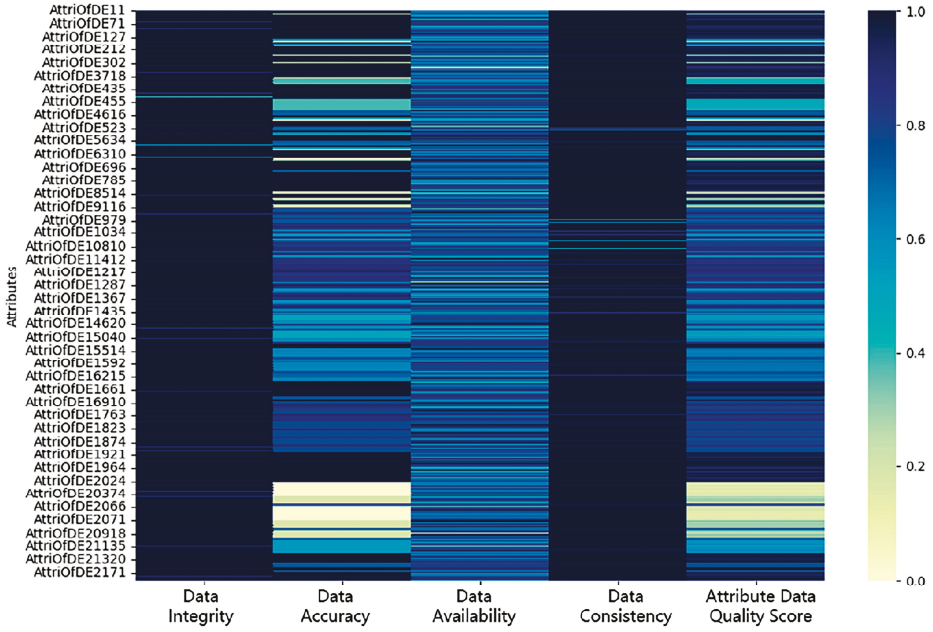
Here, BDR_{QScore} represents the data quality evaluation score matrix.

Substitute the dataset into the above formulas to calculate the data quality scores of the business data resources. The data quality evaluation result of each business data entity is the average of its attributes, as shown in Table 6.

Table 6. Data quality evaluation results of business data resources.

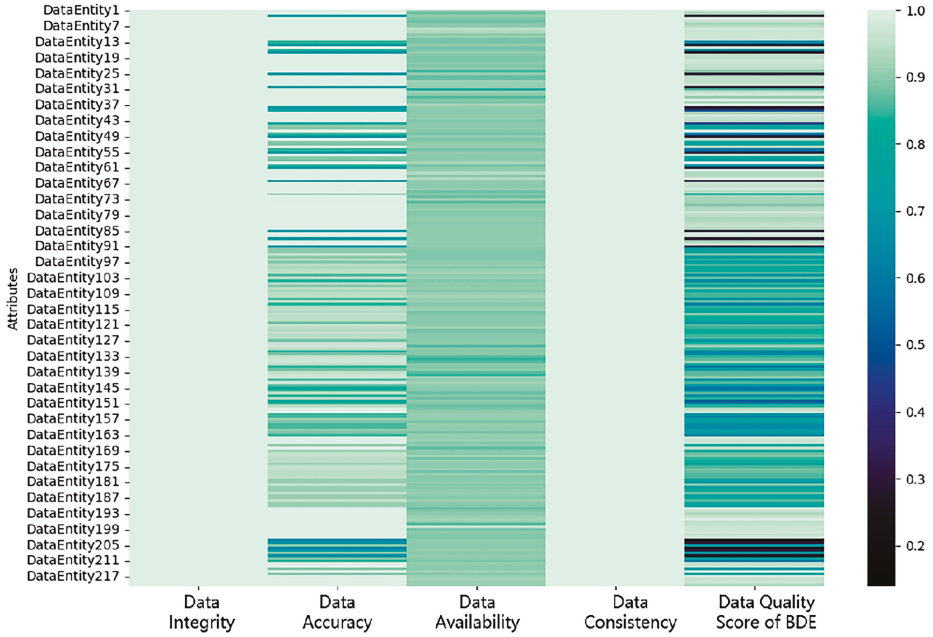
Business Data Entity	Attribute Number	Attribute	Attribute Data Quality Score	Business Data Entity Quality Score
Purchase Request Information	1	Purchase Request Code	0.977	0.977
	2	Purchase Request Date	0.972	
	3	Salesperson Response	0.971	
	4	Remarks	0.982	
	5	Supplier Code	0.957	
	6	Salesperson Name	0.955	
	7	Business Type	0.962	
	8	Supplier Response Status	0.956	
	9	Confirmation Status	0.983	
	10	Publication Status	0.974	
Purchase Request Details	1	Purchase Request Code	0.935	0.935
	2	Part Code	0.936	
	3	Quantity Required	0.951	
	4	Supply Guarantee Status	0.970	
	5	Supplier Response	0.952	
	6	Required Date	0.928	
	7	Estimated Arrival Date	0.971	
	8	Unit Price	0.959	
	9	ID	0.964	
...	...	...	...	...
Adjustment Information	12	Auditor Name	0.988	0.942
	13	Audit Date	0.961	
	14	Data Transmission Status	0.955	

The highest data quality score for business data entity attributes is 1, the lowest score is 0.068, and the average score is 0.747; the highest data quality score for business data entities is 1, the lowest score is 0.140, and the average score is 0.803. Business data resources are expressed through business data entities and their attributes. The distribution of data quality evaluation scores for business data resources is shown in the form of a heatmap, as illustrated in Figure 5.



(a) Heatmap of data quality for data attributes

Figure 5. *Cont.*



(b) Heatmap of data quality for business data entities

Figure 5. Distribution of data quality for business data resources. e.g., (**a**) Heatmap of data quality for data attributes. (**b**) Heatmap of data quality for business data entities.

6.2.2. Data Utility Evaluation Calculation

(1) Data Utility Evaluation Calculation Score

According to the data utility evaluation indicators of business data resources in Table 2, calculate the data quality, maintenance cost, timeliness, and access volume for quantitative indicators, as well as user satisfaction and data security for qualitative indicators of the dataset and its preprocessing results, to obtain the evaluation results of data utility for business data resources. Quantitative indicators can be directly calculated based on the records of the dataset; qualitative indicators, such as user satisfaction and data security, need to be calculated according to Formulas (1) and (2). Table 7 shows the data utility evaluation results of some business data resources.

(2) Satisfaction Evaluation Matrix of Data Utility

According to the calculation rules of data utility indicators, quantitative indicators such as data quality, maintenance cost, timeliness, and access volume, as well as qualitative indicators such as user satisfaction and data security, are calculated. Based on the results of data utility evaluation calculations, the satisfaction evaluation matrix of data utility can be directly obtained.

$$RA' = \begin{bmatrix} 0.977 & 2.8 & 0.097 & 392 & 0.00086 & 0.00201 \\ 0.972 & 2.8 & 0.194 & 50 & 0.00086 & 0.00201 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.955 & 1 & 0.264 & 1273 & 0.00281 & 0.00361 \\ 1 & 0 & 0.694 & 1995 & 0.02194 & 0.20166 \end{bmatrix}$$

Dimensionless calculations are performed based on Formulas (3) and (4), resulting in the dimensionless satisfaction evaluation matrix of data utility.

$$RA'' = \begin{bmatrix} 0.975 & 0.034 & 0.123 & 0.196 & 0.00200 & 0.00889 \\ 0.970 & 0.034 & 0.265 & 0.025 & 0.00200 & 0.00889 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.952 & 0.655 & 0.368 & 0.638 & 0.00361 & 0.01685 \\ 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (14)$$

Table 7. Calculation results of data utility evaluation.

Business Data Entity	Attribute Number	Attribute	Data Quality	Maintenance Cost	Timeliness	Access Volume	User Satisfaction	Data Security
Purchase Request Information	1	Purchase Request Code	0.977	2.8	0.097	392	0.00086	0.00201
	2	Purchase Request Date	0.972	2.8	0.194	50	0.00086	0.00201
	3	Salesperson Response	0.971	2.8	0.111	290	0.00086	0.00201
	4	Remarks	0.982	2.8	0.542	44	0.00086	0.00201
	5	Supplier Code	0.957	2.8	0.444	265	0.00086	0.00201
	6	Salesperson Name	0.955	2.8	0.250	1081	0.00086	0.00201
	7	Business Type	0.962	2.8	0.139	1676	0.00086	0.00201
	8	Supplier Response Status	0.956	2.8	0.444	488	0.00086	0.00201
	9	Confirmation Status	0.983	2.8	0.569	0	0.00086	0.00201
	10	Publication Status	0.974	2.8	0.625	435	0.00086	0.00201
Purchase Request Details	1	Purchase Request Code	0.935	2	0.042	1083	0.00181	0.00232
	2	Part Code	0.936	2	0.125	655	0.00181	0.00232
	3	Quantity Required	0.951	2	0.472	539	0.00181	0.00232
	4	Supply Guarantee Status	0.970	2	0.361	346	0.00181	0.00232
	5	Supplier Response	0.952	2	0.069	780	0.00181	0.00232
	6	Required Date	0.928	2	0.264	256	0.00181	0.00232
	7	Estimated Arrival Date	0.971	2	0.361	216	0.00181	0.00232
	8	Unit Price	0.959	2	0.666	421	0.00181	0.00232
	9	ID	0.964	2	0.444	388	0.00181	0.00232
...	...	...	...	...	...	...	...	...
Adjustment Information	12	Auditor Name	0.988	1	0.5	879	0.00281	0.00361
	13	Audit Date	0.961	1	0.556	710	0.00281	0.00361
	14	Data Transmission Status	0.955	1	0.264	1273	0.00281	0.00361

(3) Calculation of Indicator Weights for Data Utility

Since the evaluation indicators of data utility include both quantitative and qualitative indicators, two algorithms are used for their weight calculations. This section will calculate the indicator weights for data utility based on the weight distribution algorithms for quantitative and qualitative indicators in Section 4.2 (Algorithms 1 and 2):

$$\Psi = \begin{bmatrix} \varphi_1 & \varphi_2 & \varphi_3 & \varphi_4 & \lambda_1 & \lambda_2 \end{bmatrix}$$

$$= \begin{bmatrix} 0.108 & 0.248 & 0.276 & 0.368 & 0.667 & 0.333 \end{bmatrix}$$

(4) Evaluation Results of Data Utility for Business Data Resources

The dimensionless scores from the satisfaction evaluation of data utility for business data resources are weighted and summarized to obtain the evaluation results of data utility, which are:

$$BDR_{UScore} = \begin{bmatrix} ra_{1,1}'' \cdots ra_{1,j}'' \cdots ra_{1,n}'' \\ \vdots\vdots \\ ra_{i,1}'' \cdots ra_{i,j}'' \cdots ra_{i,n}'' \\ \vdots\vdots \\ ra_{m,1}'' \cdots ra_{m,j}'' \cdots ra_{m,n}'' \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \varphi_4 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} \quad (15)$$

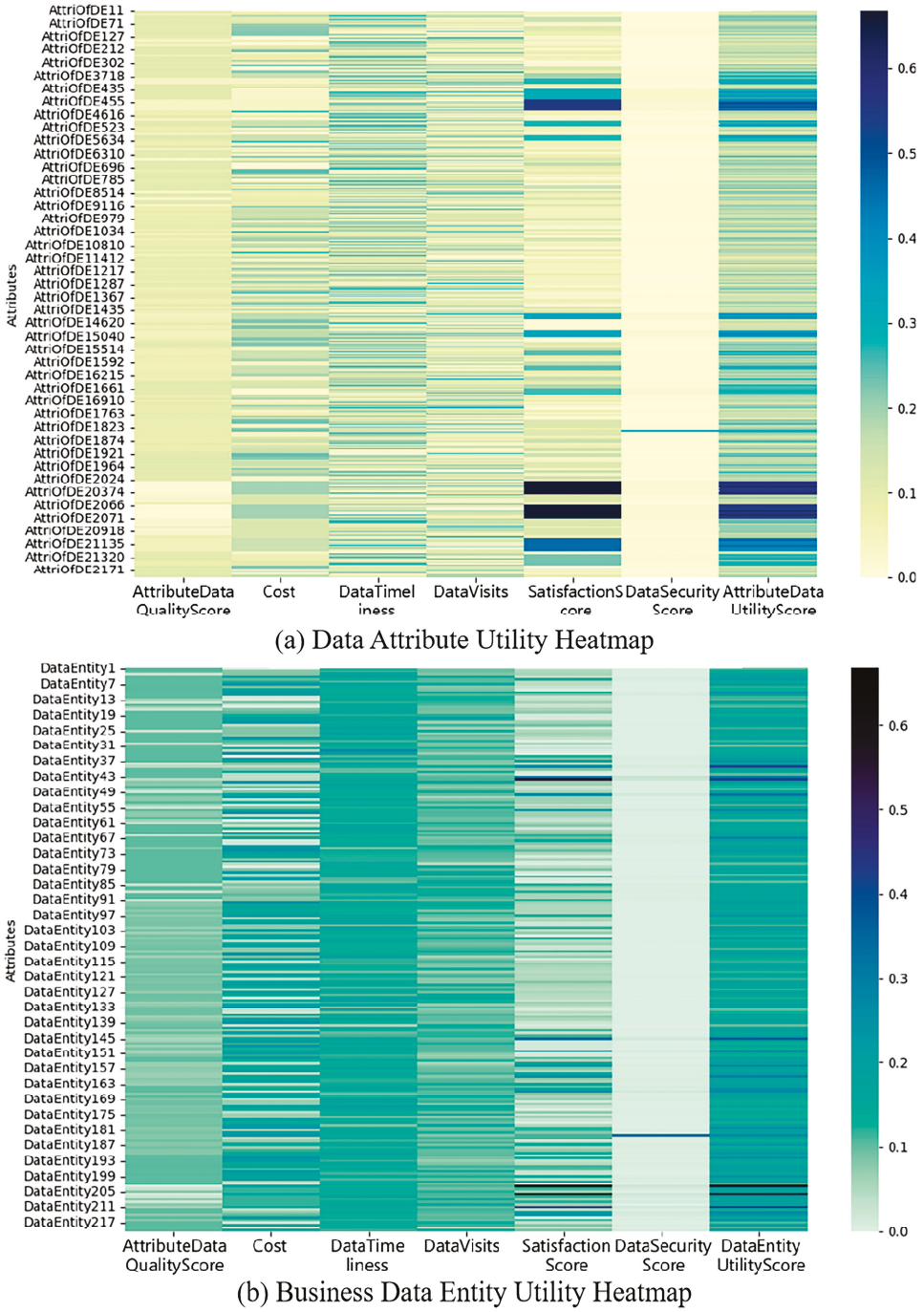
where BDR_{UScore} is the evaluation score matrix of data quality.

Substituting the dataset into the above formulas for calculation, the data utility scores of business data resources are obtained. The evaluation results of data utility for business data entities are the mean values of their attributes. The evaluation results are shown in Table 8.

Table 8. Evaluation results of data utility of business data resources.

Business Data Entity	Attribute Number	Attribute	Attribute Data Utility Score	Business Data Entity Utility Score
Purchase Request Information	1	Purchase Request Code	0.065	0.358
	2	Purchase Request Date	0.053	
	3	Salesperson Response	0.060	
	4	Remarks	0.091	
	5	Supplier Code	0.095	
	6	Salesperson Name	0.129	
	7	Business Type	0.157	
	8	Supplier Response Status	0.110	
	9	Confirmation Status	0.091	
	10	Publication Status	0.127	
Purchase Request Details	1	Purchase Request Code	0.142	0.451
	2	Part Code	0.122	
	3	Quantity Required	0.153	
	4	Supply Guarantee Status	0.128	
	5	Supplier Response	0.125	
	6	Required Date	0.110	
	7	Estimated Arrival Date	0.119	
	8	Unit Price	0.167	
	9	ID	0.140	
...	...	...		
Adjustment Information	12	Auditor Name	0.716	0.610
	13	Audit Date	0.704	
	14	Data Transmission Status	0.689	

The highest data utility score for the attributes of business data entities is 0.663, the lowest score is 0.041, and the average score is 0.219. The highest data utility score for business data entities is 0.581, the lowest score is 0.093, and the average score is 0.177. To visualize the evaluation scores of data utility, a heatmap is used to present the evaluation scores of some business data resources, as shown in Figure 6.



(a) Data Attribute Utility Heatmap

(b) Business Data Entity Utility Heatmap

Figure 6. Data Quality-Utility Evaluation Calculation. e.g., (**a**) Data Attributes Utility Heatmap. (**b**) Business Data Entity Utility Heatmap.

6.2.3. Data Quality-Utility Evaluation Calculation

According to Algorithm 4, the data quality-utility indicators for business data resources are calculated, while the values of ε_q^s and ω_j^o in Equation (5) and the values of η_q^s and σ_j^o in Equation (6) are obtained. Further, the weight of α in Equation (7) is determined based on the entropy weight method. Therefore, the evaluation results of business data resources based on data quality-utility evaluation can be expressed as:

$$H(rd_i) = \alpha * H_1(rd_i) + (1 - \alpha)H_2(rd_i) \tag{16}$$

$$Subject.to\begin{cases}\alpha = 0.661\\ H_1(rd_i) = 1 - \sqrt{\sum_{j=1}^{n}\left(\varepsilon_q^s * {sa_j}^{*''} - \omega_j^o * sa_{i,j}{}''\right)^2}\\ \varepsilon_1^s = 0.25,\ \varepsilon_2^s = 0.25,\ \varepsilon_3^s = 0.25,\ \varepsilon_4^s = 0.25\\ \omega_1^o = 0.015,\ \omega_2^o = 0.816,\ \omega_3^o = 0.156,\ \omega_4^o = 0.013\\ H_2(rd_i) = 1 - \sqrt{\sum_{j=1}^{n}\left(\eta_q^s * {ra_j}^{*''} - \sigma_j^o * ra_{i,j}{}''\right)^2}\\ \eta_1^s = 0.167,\ \eta_2^s = 0.166,\ \eta_3^s = 0.167,\ \eta_4^s = 0.166,\ \eta_5^s = 0.167,\ \eta_6^s = 0.167\\ \sigma_1^o = 0.04,\ \sigma_2^o = 0.089,\ \sigma_3^o = 0.093,\ \sigma_4^o = 0.137,\ \sigma_5^o = 0.241,\ \sigma_6^o = 0.4\end{cases}$$

By substituting the dataset and combining Equations (5), (6), and (16), the data quality-utility evaluation result matrix for business data resources is calculated.

$$RD = \begin{bmatrix} 0.010 & 0.320 & 0.014 & 0.056 & 0.026 & 0.0000006 & 0.027 & 0.024 & 0.026 & 0.027\\ 0.010 & 0.320 & 0.015 & 0.056 & 0.027 & 0.0000006 & 0.026 & 0.027 & 0.026 & 0.027\\ 0.010 & 0.320 & 0.015 & 0.056 & 0.027 & 0.0000006 & 0.027 & 0.025 & 0.026 & 0.027\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ 0.010 & 0.320 & 0.019 & 0.056 & 0.027 & 0.0002091 & 0.025 & 0.018 & 0.022 & 0.027 \end{bmatrix}$$

6.3. Evaluation and Analysis of Algorithm Recommendation Accuracy

Section 5 introduces three algorithms (CFA-DQ, CFA-DU, and CFA-DQU) to recommend business data resources, each capable of generating a Top-N recommendation list. To demonstrate the accuracy of these recommendation algorithms, we designed 10 scenarios to compare the recommendation accuracy of the three algorithms. These 10 scenarios correspond to different values of "similar neighbor set" size *K*, specifically set at 5, 8, 11, 14, 17, 20, 23, 26, 29, and 32. The recommendation accuracy is assessed by comparing the root mean square error (RMSE) values in each scenario.

RMSE is calculated based on the error between the predicted and actual quality/utility values of recommended data resources, with a lower RMSE indicating higher recommendation accuracy.

The RMSE calculation formula is as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{17}$$

where n is the number of data items, x_i is the actual quality-utility value of the i-th data item, and $\hat{x}_i$ is the predicted quality-utility value of the i-th data item.

6.3.1. Experimental Comparison Results

The RMSE of CFA-DQ, CFA-DU, and CFA-DQU under different similar neighbor set values are shown in Table 9.

Table 9. RMSE calculation results of three algorithms.

Values of *K*	CFA-DQ	CFA-DU	CFA-DQU
K = 5	0.01283466	0.01853151	0.00113214
K = 8	0.01415222	0.01935435	0.00124120
K = 11	0.01563750	0.02008541	0.00132608
K = 14	0.01700119	0.02072918	0.00140239
K = 17	0.01836978	0.02152268	0.00147802
K = 20	0.01959997	0.02197381	0.00153857
K = 23	0.02076240	0.02246302	0.00159961
K = 26	0.02189824	0.02287985	0.00166435
K = 29	0.02290673	0.02334594	0.00172426
K = 32	0.02382351	0.02375691	0.00177925

The changes in RMSE are also presented in bar and line charts, as shown in Figure 7.



Figure 7. RMSE trend chart of three algorithms.

6.3.2. Experimental Comparative Analysis

Experimental results indicate that when the similar neighbor set size is 5, the RMSE of CFA-DQU is the lowest (0.001132), followed by CFA-DQ (0.012835) and CFA-DU with the highest RMSE (0.018532). A comparison of the RMSE values for CFA-DQ, CFA-DU, and CFA-DQU across various similar neighbor set sizes (K = 5, 8, 11, 14, 17, 20, 23, 26, 29, and 32) demonstrates that CFA-DQU consistently achieves the highest recommendation accuracy and optimal recommendation performance. Although CFA-DQ is slightly less accurate than CFA-DQU, it still outperforms the CFA-DU algorithm. Further analysis led to the following conclusions:

First, the CFA-DQU algorithm has the lowest RMSE value and the highest recommendation accuracy for all similar neighbor set values.

Second, as the value of "similar neighbor sets" K increases, the RMSE values of the three algorithms rise, indicating that the recommendation accuracy decreases as the number of similar neighbor sets increases.

Third, the RMSE value of CFA-DQU increases the least, indicating that it has better stability when the number of similar neighbor sets increases.

Fourth, the line chart shows that the RMSE value curve of CFA-DQU is relatively flat, indicating that its performance is more stable under different similar neighbor set values, while the RMSE value curves of CFA-DQ and CFA-DU are steeper, indicating that their recommendation accuracy is more affected by the number of similar neighbor sets.

6.4. Performance Analysis of Different Recommendation Algorithms

(1) Design and Calculation of Performance Metrics for Recommendation Algorithms

To analyze the stability and scalability of the recommendation algorithms, we introduce metrics such as precision, recall, and H-mean to evaluate their performance. The corresponding calculation formulas are as follows:

① Precision

$$P = \frac{TP}{TP + FP} \tag{18}$$

② Recall

$$R = \frac{TP}{TP + FN} \tag{19}$$

③ H-mean

$$F_1 = \frac{2PR}{P + R} = \frac{2TP}{2TP + FP + FN} \tag{20}$$

In the formulas, TP represents true positives, where positive cases are correctly predicted as positive; TN represents true negatives, where negative cases are correctly predicted as negative; FP represents false positives, where negative cases are incorrectly predicted as positive (false alarms); and FN represents false negatives, where positive cases are incorrectly predicted as negative (misses). The larger the F_1, the better the corresponding performance.

The experimentally calculated values of performance metrics such as precision, recall, and H-mean for CFA-DQ, CFA-DU, and CFA-DQU under different values of the similar neighbor set are shown in Table 10.

Table 10. Performance metric results for the three algorithms.

Values of K	CFA-DQ			CFA-DU			CFA-DQU		
	Precision	**Recall**	**H-Mean**	**Precision**	**Recall**	**H-Mean**	**Precision**	**Recall**	**H-Mean**
K = 5	0.9840	0.9532	0.9684	0.9689	0.9905	0.9796	0.9553	0.9448	0.9500
K = 8	0.9859	0.9124	0.9477	0.9506	0.9793	0.9648	0.9669	0.9558	0.9611
K = 11	0.9874	0.8716	0.9259	0.9505	0.9762	0.9631	0.9609	0.9503	0.9556
K = 14	0.9883	0.8405	0.9084	0.9505	0.9777	0.9639	0.9607	0.9448	0.9526
K = 17	0.9890	0.8057	0.8880	0.9520	0.9777	0.9647	0.9382	0.9227	0.9304
K = 20	0.9888	0.7762	0.8697	0.9532	0.9714	0.9622	0.9286	0.9337	0.9311
K = 23	0.9898	0.7481	0.8522	0.9531	0.9700	0.9614	0.9378	0.9171	0.9274
K = 26	0.9902	0.7247	0.8369	0.9533	0.9746	0.9638	0.9218	0.9116	0.9167
K = 29	0.9908	0.6920	0.8149	0.9519	0.9746	0.9631	0.9261	0.9006	0.9132
K = 32	0.9917	0.6699	0.7996	0.9503	0.9730	0.9615	0.9209	0.9006	0.9106

(2) Analysis of the Adaptability and Stability of Recommendation Algorithms

This paper assesses the stability and adaptability of the algorithms by calculating the standard deviation and mean values of their performance metrics.

① Adaptability

The adaptability of the algorithm is evaluated using the average values of precision, recall, and H-mean from multiple experimental runs. A higher average indicates better performance across different values of the similar neighbor set, reflecting stronger adaptability. High adaptability means that the algorithm can handle varied datasets more effectively, making it suitable for dynamic or unknown types of datasets.

$$\boldsymbol{Ada} = \frac{1}{\boldsymbol{N}} \sum_{\boldsymbol{i}=1}^{\boldsymbol{N}} \boldsymbol{y_i} \tag{21}$$

In the formula, $\boldsymbol{y_i}$ is the value from the $\boldsymbol{i}$-th experiment.

② Stability

We assess the stability of the algorithm by calculating the standard deviation of each performance metric (precision, recall, and H-mean) across multiple experiments. A smaller standard deviation indicates better stability. If one algorithm's stability metric (standard

deviation) is significantly lower than that of another, it suggests that this algorithm may be better suited to the same dataset.

$$Sta = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \overline{y})^2} \tag{22}$$

where $\overline{y}$ is the mean and y_i is the value of the i-th experiment.

Using the experimental evaluation metrics designed above, we calculate the standard deviation and mean values of all metrics (precision, recall, and H-mean) for each of the three algorithms to reflect each algorithm's adaptability and stability, as shown in Table 11.

Table 11. Evaluation of the adaptability and stability of three recommendation algorithms.

Evaluation Metrics	CFA-DQ			CFA-DU			CFA-DQU		
	Precision	**Recall**	**H-Mean**	**Precision**	**Recall**	**H-Mean**	**Precision**	**Recall**	**H-Mean**
Ada	0.9886	0.7994	0.8812	0.9534	0.9765	0.9648	0.9417	0.9282	0.9349
Sta	0.0023	0.0945	0.0566	0.0056	0.0057	0.0053	0.0177	0.0205	0.0187

Under different values of the similar neighbor set, combinations of data quality, data utility, and data quality-utility formed distinct business data resource usage scenarios. The adaptability and stability performance of the three algorithms in different scenarios are as follows:

Adaptability Analysis of the Algorithms: CFA-DU performs relatively evenly across precision, recall, and H-mean, particularly outperforming the other two algorithms in recall and H-mean, indicating stronger adaptability and the best recommendation performance in scenarios focused on data utility. CFA-DQ performs strongest in Precision but has relatively lower recall and H-mean, indicating slightly weaker adaptability. CFA-DQU's performance falls between the two, suggesting that CFA-DQU better balances recommendations focused on both data quality and utility across different scenarios.

Stability Analysis of the Algorithms: CFA-DU has the smallest standard deviation across all metrics, demonstrating extremely high stability. This indicates that, in different scenarios, the recommendation performance focused on data utility is more stable. CFA-DQ shows higher stability in precision but has greater fluctuations in recall and H-mean, resulting in overall weaker stability. CFA-DQU exhibits moderate stability but is still more stable than CFA-DQ. This suggests that CFA-DQU, while balancing recommendations focused on data quality-utility across different scenarios, also achieves relatively good algorithm stability.

7. Conclusions

The rapid and accurate selection of business data resources is the most time-consuming stage in the preprocessing phase of business analysis. It often requires multiple rounds of data retrieval for exploratory analysis to meet the data requirements for business analysis. Additionally, there are common issues of data sparsity and cold start in the evaluation and recommendation of business data resources.

Therefore, this paper addresses data quality and data utility evaluation by constructing an evaluation indicator system, resulting in an evaluation matrix for business data resources, thus solving the problems of data sparsity and cold start. Furthermore, using a collaborative filtering-based algorithm for business data resources, similar data quality-utility resources are formed into a recommendation set, provided to data analysis users.

Finally, through experimental calculations, evaluation results and recommendation results for business data resources and their attributes were obtained. Additionally, a comparison of three recommendation algorithms demonstrated the excellent performance of the combined improved collaborative filtering recommendation algorithm for data quality utility, leading to conclusions and recommendations. Additionally, this paper

demonstrates the good scalability of the algorithms by calculating the adaptability and stability of the three algorithms across different scenarios.

In the face of fragmented and uncertain business demands, digital transformation has been regarded as a strategic imperative for enterprise development. Optimizing the supply of business data resources is a critical research topic for building data-driven organizations. Business data resources serve as the informational medium for business collaboration. The business data resource optimization framework proposed in this paper, based on a multi-value chain data space, provides enterprise data consumers with data resources that meet their business analysis needs. Moreover, this framework has been technically validated in business domains such as the supply chain, marketing value chain, parts (operations and maintenance) value chain, and after-sales service value chain in the automotive manufacturing industry. Therefore, this framework is adaptable to various value chain business collaboration scenarios within the discrete manufacturing industry. In the next step, we will focus on the study of business data resource models, aiming to solve the problem of synchronizing business data resource evaluation and recommendation in a dynamic data update environment.

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

Funding: The research in this paper has been supported by the National Key R&D Plan, China (No. 2023YFB3308600).

Data Availability Statement: The datasets used and analyzed during the current study are available from the corresponding author on reasonable request.

Acknowledgments: We would like to thank the National Key R&D Program of China for its strong support for this research.

Conflicts of Interest: Author Dong Tan and Meng Pan was employed by the company Chengdu Aerospace Science and Industry Big Data Institute Ltd., CASIC 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.

References

1. Grover, V.; Chiang, R.H.; Liang, T.P.; Zhang, D. Creating strategic business value from big data analytics: A research framework. *J. Manag. Inf. Syst.* **2018**, *35*, 388–423. [CrossRef]
2. Chapman, P.; Clinton, J.; Kerber, R.; Khabaza, T.; Reinartz, T.; Shearer, C.; Wirth, R. *CRISP-DM 1.0: Step-by-Step Data Mining Guide*; SPSS Inc.: Chicago, IL, USA, 2000.
3. Jiang, W.; Wu, X.; Zhuravkov, M.A.; Xue, Z.; Wang, J. A comprehensive review of theories, methods, and techniques for bottleneck identification and management in manufacturing systems. *Appl. Sci.* **2024**, *14*, 7712. [CrossRef]
4. Raj, R.; Kumar, V.; Shah, B. Big data analytics adaptive prospects in sustainable manufacturing supply chain. *Benchmarking Int. J.* **2024**, *31*, 3373–3397. [CrossRef]
5. Faroukhi, A.Z.; El Alaoui, I.; Gahi, Y.; Amine, A. An Adaptable Big Data Value Chain Framework for End-to-End Big Data Monetization. *Big Data Cogn. Comput.* **2020**, *4*, 34. [CrossRef]
6. Peng, S.; Tian, Z.; Siqin, Z.; Xu, X. Construction of data quality evaluation index for manufacturing multi-value chain collaborative data space based on the whole life cycle of data. In Proceedings of the 2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS), Virtual, 7–8 November 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 315–323.
7. Yu, Y.; Sun, L.; Wang, S. Multi-party dynamic data integration scheme of industrial chain collaboration platform in mobile computing environment. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 1550668.
8. Sadiq, S.; Indulska, M.; Brohman, K. A framework for data quality management in emerging digital ecosystems. *Inf. Syst.* **2019**, *84*, 101–120.
9. Zhang, X.; Chen, Y.; Li, D. Evaluating data quality in collaborative information systems: A comprehensive framework and metrics. *Int. J. Inf. Manag.* **2020**, *52*, 102–115.
10. Roy, D.; Dutta, M. A systematic review and research perspective on recommender systems. *J. Big Data* **2022**, *9*, 59. [CrossRef]

11. Adomavicius, G.; Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 734–749. [CrossRef]
12. Fricker, S.A.; Maksimov, Y.V. Pricing of data products in data marketplaces. In Proceedings of the Software Business: 8th International Conference, ICSOB 2017, Essen, Germany, 12–13 June 2017; pp. 49–66.
13. Xiao, Y.; Gu, Y.; Wang, J.; Wu, T. A collaborative multi-modality selection method based on data utility assessment. In Proceedings of the 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, UK, 19–23 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 454–459.
14. Wang, S.; Sun, L.; Zou, Y. Recommendation algorithm of business resources for multi-service value chain. *Comput. Integr. Manuf. Syst.* **2023**, *29*, 2397–2410.
15. Yu, Y.; Sun, L.F.; Ren, C.H.; Han, M. Bilater matching model of business resources for multi-service value chain. *Comput. Integr. Manuf. Syst.* **2021**, *27*, 1397–1409.
16. Chen, H.; Chiang, R.H.; Storey, V.C. Business intelligence and analytics: From big data to big impact. *MIS Q.* **2021**, *45*, 144–155. [CrossRef]
17. Liu, Y.; Zhang, Y. Data-driven analytics and predictive modeling in finance: A critical review. *J. Financ. Data Sci.* **2020**, *2*, 11–28.
18. Kraus, S.; Palmer, C.; Kailer, N.; Kallinger, F.L.; Spitzer, J. Digital entrepreneurship: A research agenda on new business models for the twenty-first century. *Int. J. Entrep. Behav. Res.* **2018**, *24*, 350–375. [CrossRef]
19. Wang, S.; Wan, J.; Zhang, D.; Li, D.; Zhang, C. Towards smart factory for industry 4.0: A self-organized multi-agent system with big data-based feedback and coordination. *Comput. Netw.* **2019**, *101*, 158–168. [CrossRef]
20. Janssen, M.; Van Der Voort, H.; Wahyudi, A. Factors influencing big data decision-making quality. *J. Bus. Res.* **2017**, *70*, 338–345. [CrossRef]
21. Bevilacqua, M.; Ciarapica, F.E. Human factor risk management in the process industry: A case study. *Reliab. Eng. Syst. Saf.* **2018**, *169*, 149–159. [CrossRef]
22. Sun, Z.; Wang, H.; Feng, X. Big data preprocessing technology for machine learning. *J. Data Inf. Sci.* **2019**, *4*, 20–37.
23. Huang, S.; Han, J.; Liu, H. The role of data quality in machine learning applications. *Data Sci. J.* **2020**, *19*, 5–14.
24. Wang, R.Y.; Strong, D.M. Beyond Accuracy: What Data Quality Means to Data Consumers. *J. Manag. Inf. Syst.* **2020**, *36*, 5–33. [CrossRef]
25. Pipino, L.L.; Lee, Y.W.; Wang, R.Y. Data Quality Assessment. *Commun. ACM* **2019**, *45*, 211–218. [CrossRef]
26. Batini, C.; Scannapieco, M. Data Quality: Concepts, Methodologies and Techniques. *Data Knowl. Eng.* **2020**, *121*, 1–15.
27. Ge, M.; Helfert, M. A Framework for Data Quality Management in Business Intelligence and Analytics. *Inf. Syst.* **2021**, *52*, 246–256.
28. Oliveira, P.; Rodrigues, F.; Henriques, P.R. Towards a User-Oriented Data Quality Model. *J. Data Inf. Qual.* **2020**, *12*, 1–33.
29. Olson, A. Data as a Service: Are We in the Clouds? *J. Map Geogr. Libr.* **2010**, *6*, 76–78. [CrossRef]
30. Wang, X.; Wang, H.; Zheng, J. Can Entropy Weight Method Correctly Reflect the Distinction of Water Quality Assessment? *J. Environ. Manag.* **2016**, *180*, 459–466.
31. Wang, Z.; Zhang, Z.; Zhang, H. Principal Component Analysis and Related Methods for Investigating the Dynamics of Biological Macromolecules. *J. Mol. Sci.* **2020**, *5*, 298–317.
32. Chatterjee, S.; Chaudhuri, R.; Kamble, S.; Gupta, S.; Sivarajah, U. Adoption of artificial intelligence and cutting-edge technologies for production system sustainability: A moderator-mediation analysis. *Inf. Syst. Front.* **2022**, *24*, 976–995. [CrossRef]
33. Duan, Y.; Cao, G.; Edwards, J.S. Understanding the impact of business analytics on innovation. *Eur. J. Oper. Res.* **2020**, *281*, 673–686. [CrossRef]
34. Southwest Jiaotong University; Chengdu Guolong Information Engineering Co., Ltd.; University of Electronic Science and Technology of China; China Association of Automobile Manufacturers. ASP/SaaS-Based Collaborative Platform for Manufacturing Industry Value Chain [OL]. Available online: http://www.autosaas.cn/ (accessed on 25 July 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

PMHR: Path-Based Multi-Hop Reasoning Incorporating Rule-Enhanced Reinforcement Learning and KG Embeddings

Ang Ma [1], Yanhua Yu [1,*], Chuan Shi [1], Shuai Zhen [1], Liang Pang [2] and Tat-Seng Chua [3]

[1] College of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China; ang@bupt.edu.cn (A.M.); shichuan@bupt.edu.cn (C.S.); zhenshuai@bupt.edu.cn (S.Z.)
[2] Institute of Computing Technology, University of Chinese Academy of Sciences, Beijing 100190, China; pangliang@ict.ac.cn
[3] Sea-NExT Joint Lab, National University of Singapore, Singapore 119077, Singapore; dcscts@nus.edu.sg
* Correspondence: yuyanhua@bupt.edu.cn

Abstract: Multi-hop reasoning provides a means for inferring indirect relationships and missing information from knowledge graphs (KGs). Reinforcement learning (RL) was recently employed for multi-hop reasoning. Although RL-based methods provide explainability, they face challenges such as sparse rewards, spurious paths, large action spaces, and long training and running times. In this study, we present a novel approach that combines KG embeddings and RL strategies for multi-hop reasoning called path-based multi-hop reasoning (PMHR). We address the issues of sparse rewards and spurious paths by incorporating a well-designed reward function that combines soft rewards with rule-based rewards. The rewards are adjusted based on the target entity and the path to it. Furthermore, we perform action filtering and utilize the vectors of entities and relations acquired through KG embeddings to initialize the environment, thereby significantly reducing the runtime. Experiments involving a comprehensive performance evaluation, efficiency analysis, ablation studies, and a case study were performed. The experimental results on benchmark datasets demonstrate the effectiveness of PMHR in improving KG reasoning accuracy while preserving interpretability. Compared to existing state-of-the-art models, PMHR achieved Hit@1 improvements of 0.63%, 2.02%, and 3.17% on the UMLS, Kinship, and NELL-995 datasets, respectively. PMHR provides not only improved reasoning accuracy and explainability but also optimized computational efficiency, thereby offering a robust solution for multi-hop reasoning.

Keywords: knowledge graphs; knowledge graph reasoning; reinforcement learning; multi-hop reasoning

1. Introduction

Knowledge graphs (KGs) [1] are powerful representations of structured information that capture complex semantic relationships between entities. They have become fundamental components in various applications such as natural language understanding [2,3], recommendation systems [4], and semantic web searches [5]. However, most extant KGs suffer from incompleteness and inaccuracy because they are usually built automatically from a variety of data sources. Incompleteness and inaccuracy are manifested as a significant dearth of crucial relations between the entities, which significantly constrains the applicability and utility of the KGs. Knowledge graph reasoning (KGR) involves inferring new knowledge or relations within a KG by applying various computational

methods and techniques. The aim in KGR is to enhance the completeness, accuracy, and comprehensibility of KGs by revealing hidden information and predicting missing links.

Traditional KGR methods, particularly embedding-based and rule-based methods, have limitations in dealing with noisy or incomplete data and capturing the potential complexity of real-world knowledge. Multi-hop KGR [6–8] is an important KGR technique used to discover complex and multi-level relations between entities in a KG to infer hidden information. Numerous reinforcement learning (RL)-based methods for multi-hop reasoning have emerged recently. Reinforcement methods were first employed to learn reasoning paths in KGs in DeepPath [9], where path finding is treated as a Markov decision process (MDP) and a sophisticated reward function is employed in which accuracy, efficiency, and path diversity are considered simultaneously. This approach provides enhanced control over the path finding process. DeepPath was followed by series of other techniques such as MINERVA [10], AttenPath [11], Reasoning Like Human (RLH) [12], DAPath [13], etc. These techniques offer high efficacy and the valuable characteristic of providing interpretable reasoning paths for KGR. For example, as shown in Figure 1, given a query (Albert Einstein, Nationality, ?), these methods can provide a chain of evidence to draw the conclusion (Albert Einstein, Nationality, Germany), e.g., Albert Einstein $\xrightarrow{\text{Born in}}$ Ulm $\xrightarrow{\text{Located in}}$ Baden-Württemberg $\xrightarrow{\text{Located in}}$ Germany.



Figure 1. An example of a KG. Solid edges are observed and the dashed edge is a query.

However, the existing RL-based multi-hop reasoning methods have certain drawbacks. First, multi-hop reasoning suffers from the problem of sparse rewards, in which no feedback is provided for all the actions during the decision making process except for the final action. Although the problem of sparse rewards can be mitigated by using soft rewards, the majority of RL-based reasoning algorithms still face the problem of spurious paths, which significantly undermines the explainability of RL-based models. Second, the state representation of the agent plays a pivotal role in its ability to make informed decisions and learn effective policies. The state space should contain all the useful variables and information available to the agent in the environment. Third, the efficiency of RL algorithms is a major concern, as most RL algorithms require 10–20 h or more to complete training [14].

In this paper, we present an innovative framework that combines rule-enhanced RL and embeddings for multi-hop reasoning (path based multi-hop reasoning, PMHR). We address how reinforcement learning can effectively combine with embeddings and rules to perform accurate, interpretable multi-hop reasoning. We further explore how this approach leverages both learned representations and logical rules to enhance reasoning efficiency and scalability across complex knowledge graphs. Specifically, PMHR employs the REINFORCE algorithm. To address the issue of sparse rewards and spurious paths, PMHR combines soft rewards and rule-based rewards, which alleviates the problem of

sparse rewards and prompts the agent to discover more reasonable paths. The expressive state space provides the agent with a comprehensive perspective of its environment. To enrich the state representation and provide more guided states for the agent, the state representation is redefined to include not only the immediate inputs but also relevant historical information, graph structure information and query-related guiding information. In addition, PMHR leverages the embedding of entities and relations learned through KG representation learning to initialize the environment and prune the action space, thereby improving the efficiency of RL and reducing the run time.

The key contributions of our work are as follows:

- We propose the PMHR model for KGs in which soft rewards and rule-based rewards are combined in a novel reward function. The use of soft rewards mitigates the sparse rewards problem, whereas rule-based rewards reshaping further addresses the problem of spurious paths.
- We redefine the state representation to provide an informative state, which equips the agent with the necessary tools to make effective decisions. The rich states allow the agent to recognize complex patterns, dependencies, and correlations within the environment. This facilitates more accurate predictions of the consequences of various actions.
- We perform action filtering for large action spaces by removing irrelevant actions. The efficiency of RL is enhanced by leveraging the embeddings of entities and relations learned from the KG representation for environment initialization.

2. Related Works

In this section, we present relevant works pertaining to the study. In addition, we provide a brief comparison of PMHR with previous models to highlight both their differences and connections.

2.1. Embedding-Based Knowledge Graph Reasoning

Embedding-based KGR aims to represent entities and relations in a KG as embeddings in a low-dimensional continuous vector space to perform various reasoning tasks, such as triple classification, relation prediction, and entity prediction. The early models such as TransE [15] and TransD [16] primarily focused on distance-based methods. In bilinear models such as RESCAL [17], DisMult [18], ComplEx [19], it is assumed that there are latent semantics between entities and relations and that triples can be measured by matching the latent semantic spatial similarity between entities and relations. In neural network-based approaches, such as the neural tensor network (NTN) [20], ConvE [21], and ConvKB [22], entities and relations are fed into convolutional neural networks to compute a semantic matching scores. The use of large language models for KGR has also been explored recently. The large language models, which utilize extensive textual information, function as encoders of entities and relations for triplet score calculations (e.g., PKGC [23], SimKGC [24]), generators for the direct generation of tail entities (e.g., GenKGC [25], AutoKG [26]), or rankers for reordering entities returned by traditional embedding-based models (e.g., KICGPT [27]). Despite the powerful representational ability of embedding-based models, their performance is limited by the complexity of multi-hop reasoning because of their difficulty in capturing long-distance or highly complex paths. In addition, it can be challenging for these models to explain their reasoning process, which can be problematic in situations where transparency and interpretability are essential.

2.2. Graph-Based Knowledge Graph Reasoning

Graph neural networks [28] have been widely used in recent years. They aggregate and propagate information by performing convolution on graph structures and perform well in many downstream tasks, such as node classification, graph classification, and link prediction. Graph-based KGR models typically follow the encoder–decoder structure. RGCN [29] is a graph convolutional network (GCN) model for multi-relational graphs. It acts as a graph encoder in which relationship-specific weight matrices are used to model different relationships and convolution operations are used to learn the embeddings of entities. SACN [30] consists of a WGCN encoder [31] and a Conv-TransE decoder. The weight is defined as the strength between two adjacent nodes with the same relation type, and the information in the KG is captured using the node structure, node attributes, and relation types. In order to maintain translation features, ConvE is employed in the Conv-TransE decoder as the semantic matching metric and the reshape step in ConvE 2D convolution is skipped. Differing from the above works, CompGCN [32] is a novel framework for GCNs that simultaneously learns the embeddings of nodes and relations in multi-relational graphs while maintaining effective scalability with the number of relations. Graph-based methods consider the structural information of the graph and, through techniques like graph convolution or graph neural networks, leverage global information to improve reasoning performance. While they capture global structural information well, they often lack interpretability and may miss the specific path information crucial for transparent reasoning.

2.3. Path-Based Knowledge Graph Reasoning

The ability of the path-finding process between entities to be naturally modeled as a sequential decision problem has resulted in the emergence of RL-based multi-hop reasoning methods. DeepPath [9] is the first model in which RL is applied for relational reasoning by using the REINFORCE algorithm to find paths between entities. In MINERVA [10], the query answer is formulated as an RL problem for determining the path that provides an answer based on the input query. Strategies for navigating from the query entity to the answer node are learned by selecting relationships. MINERVA provides interpretable reasoning paths for its predictions. In MultiHopKG [33], pretrained representation learning models such as ConvE are utilized to provide soft rewards for unobserved triples, thereby mitigating the impact of sparse rewards. In AttnPath [11], LSTM and graph-attention mechanism are used as memory components. Unlike the aforementioned methods in which the rewards are manually designed, the reward functions for different datasets in DIVINE [34] are automatically tuned for optimal performance. In GaussianPath [35], a Gaussian distribution is employed to represent entities and relations and to capture reasoning path uncertainty. In the PAAR model [36], multi-hop reasoning is enhanced by leveraging hyperbolic embeddings to capture hierarchical information and using soft rewards to address reward sparsity. A novel reward function was introduced in Path Spuriousness (PS) [14] to estimate the extent of path spuriousness and balance answer accuracy with path reasonableness.

2.4. Differences

Our approach differs significantly from those in previous studies in several key aspects. First, the existing RL-based multi-hop reasoning methods are hindered by the problem of sparse rewards, in which feedback provided only for the final action in the decision making process. Although this issue is partially alleviated by soft rewards, many RL algorithms continue to face the challenge of spurious paths, which undermines the explainability of RL-based models. Our approach combines soft rewards and rule-based

rewards to address both the sparse rewards and spurious paths problems effectively by prompting the agent to discover more reasonable paths. Second, in contrast to the often limited state representation in current algorithms, the importance of comprehensive state representation is emphasized in our approach. We redefine the state representation to include not only the immediate inputs but also the path history, graph structure, and query-related guidance information. This enriched state space equips the agent with a more detailed understanding of its environment to enable better decision making. Third, the efficiency of RL algorithms is a significant concern because most RL algorithms require extensive training. Our framework leverages the embeddings of entities and relations learned through KG representation learning to initialize the environment and prune the action space, thereby significantly improving the efficiency and reducing the run time. In summary, our innovative framework addresses key limitations of existing RL-based methods by effectively handling sparse rewards and spurious paths, enhancing state representation, and improving algorithm efficiency.

3. Preliminaries

We begin this section by presenting the definitions of the key concepts. We subsequently detail the fundamentals of our model by introducing the RL formula.

3.1. Path Based Multi-Hop Reasoning

Knowledge graph: A KG is represented as a directed graph $G = (E, R)$, where E denotes the set of entities and R represents the set of relations. A fact within the knowledge graph G is defined by a triple (e_s, r, e_o), where e_s and e_o are entities belonging to E, and r is a relation in R. The edges within the KG are bidirectional. This enables the graph to be further enriched through the addition of reverse edges.

Path-based Multi-Hop Reasoning: Given query $q = (e_s, r_q, ?)$ and a KG, the tail entity is predicted in path-based multi-hop reasoning by performing effective path searches and discovering one or multiple l-hop paths on the KG (e.g., $e_s \xrightarrow{r_1} e_1 \xrightarrow{r_2} \cdots \xrightarrow{r_l} e_T$) that lead to a potential set of target entities. We let P be a set of paths, where each path p comprises a start entity e_s and passes through a series of relations to reach the target entity e_{T_p}. The set of target entities can be represented as $E_{\text{target}} = \left\{ e_{T_p} \mid p \in P \right\}$, where $\left(e_s, r_q, e_{T_p} \right) \notin G$. The potential reasoning paths serve to elucidate the reasoning process and enhance its interpretability.

For example, we consider the query (Albert Einstein, nationality, ?) shown in Figure 1. One possible three-hop reasoning path is Albert Einstein $\xrightarrow{\text{Born in}}$ Ulm $\xrightarrow{\text{Located in}}$ Baden-Württemberg $\xrightarrow{\text{Located in}}$ Germany.

3.2. RL Formula

The reasoning process can be modeled as an MDP with an agent and a KG environment. The agent selects actions according to the policy network, whereas the KG environment provides the states and rewards. The primary components in MDP are as follows:

States: A state concatenates the current situation with the original query. The state in Step t is represented as a tuple $s_t = (e_t, r_q, h_t, d_q)$, where e_t denotes the current entity, r_q denotes the query relation, h_t denotes the decision history information, and d_q is the query-related guiding information. The state is thoroughly explained in Section 4.3.

Actions: The possible actions at Step t comprise the outgoing edges at the current entity e_t and their associated entities. The action space denotes $A_t = \{(r_{t+1}, e_{t+1}) \mid (e_t, r_{t+1}, e_{t+1}) \in G\}$. The valid actions therefore represent the neighborhoods of the current entity e_t. In line with previous studies, we augment the action space by introducing the self-loop relations r_{sl}. Specifically, at Step t, we add the action (r_{sl}, e_t) to the action space A_t.

Transitions: The agent performs action $a_t = (r_{t+1}, e_{t+1})$, which converts the current state s_t into s_{t+1}. Formally, this transition is denoted as $s_{t+1} = \delta(s_t, A_t) = (e_{t+1}, r_q, h_{t+1}, d_q)$. Note that r_q and d_q represent global information that remains unchanged during the transition.

Rewards: In a KG environment, rewards provide feedback to the agent based on its actions. Typically, the default binary reward is a terminal reward [33], defined as $R_T(e_T) = \mathbb{I}((e_s, r_q, e_T) \in G)$, where $\mathbb{I}(\cdot)$ is an indicator function. If the agent succeeds in locating the target entity e_T at the final step T, i.e., the triple (e_s, r_q, e_T) exists in the KG, the indicator function returns one; otherwise, it returns zero. However, relying solely on terminal rewards often leads to sparse reward scenarios. Even if the path reaches a correct entity, it can be purely coincidental and lack logical relevance to the query. To address this problem, we propose a novel reward function, which is described in detail in Section 4.3.

4. Methodology

In this section, the framework of the PMHR model is described. Following this, we detail the KG representation learning approach for pre-training the entities and relations in the KG and obtaining their representations. The query policy network, which incorporates state representation and action space pruning, is subsequently introduced. Finally, a mixed rules and embedded reward function is proposed.

4.1. Overview

Figure 2 shows an overview of PMHR. PMHR consists of five parts: KGE pre-training, state representation, policy network, action space pruning, and rule-based reward reshaping. Specifically, a KG embedding technique such as ConvE is first utilized for pre-training to obtain representations of the entities and relations in the KG, which are then used to initialize the environment for RL. The state representation is subsequently enriched to better guide the agent in learning and the action space is pruned using the obtained representation to obtain the available candidate actions. Finally, a reward function that combines binary rewards, soft rewards, and rule-based rewards is used to guide the agent in finding a correct and reasonable path.

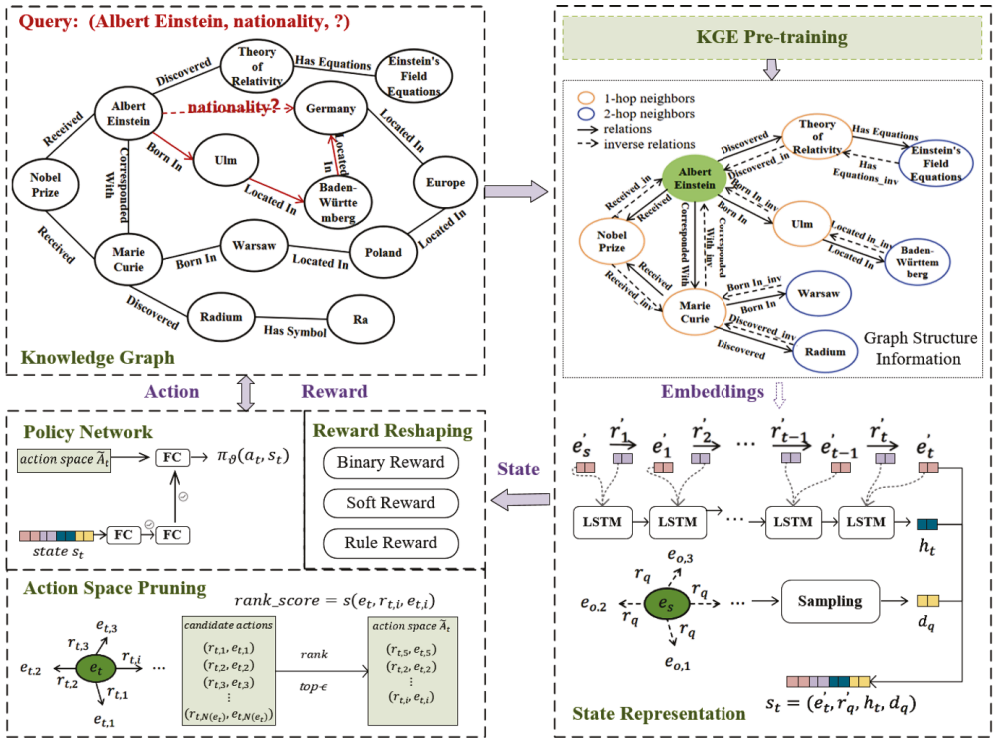


Figure 2. An overview of PMHR model.

At each step, the agent evaluates the current state along with the available candidate actions to determine the move to the next state. This process continues iteratively until the agent reaches the target entity or the maximum exploration length is reached. The exploration trajectory of the agent forms a reasoning path, which is used not only to infer new knowledge but also to provide an explanatory foundation for the reasoning results. The reasoning path aids in elucidating and validating the reasoning results, thereby enhancing their credibility and practical utility.

4.2. KGE Pre-Training

Despite their lack of interpretability, knowledge graph embedding (KGE) models often outperform multi-hop reasoning models across many KGs. KGE models are more resilient due to their independence from KG connectivity [37].

Motivated by this observation, we utilize ConvE [21], a classical KGE method, to perform representation learning of the entities and relations in the KG. The first step in the method involves reshaping and concatenating the entity and relation embeddings. The combined matrix is then fed into a convolutional layer and the resulting feature map tensor vectorized and projected into a k-dimensional space. This projected representation is finally compared with all the candidate object embeddings to obtain the scores of the triples. The score of a triple (e_s, r, e_o) is formally calculated as

$$\psi_r(e_s, e_o) = f(\mathrm{vec}(f([\overline{\mathbf{e}_s}; \overline{\mathbf{r}}] * \omega))\mathbf{W}_1)\mathbf{e}_o, \tag{1}$$

where $\mathbf{e}_s$ and $\mathbf{e}_o \in \mathbb{R}^d$ are the embeddings of entities e_s and e_o, respectively. $\mathbf{r} \in \mathbb{R}^d$ is the embedding of the relation r. $\overline{\mathbf{e}_s}$ and $\overline{\mathbf{r}}$ denote a 2D reshaping of $\mathbf{e}_s$ and $\mathbf{r}$, and $\overline{\mathbf{e}_s}, \overline{\mathbf{r}} \in \mathbb{R}^{d_w \times d_h}$, where $d = d_w d_h$. $\mathbf{W_1}$ is a learnable weight matrix. $*$ represents the convolution operation. Here, the concatenated $[\overline{\mathbf{e}_s}; \overline{\mathbf{r}}]$ is used as an input to a 2D convolutional layer with filters ω. $\mathrm{vec}(\cdot)$ is a vectorization operation that transforms the feature map tensor produced by the convolutional layer into a 1D vector. f denotes the rectified linear unit (ReLU) activation function.

The embeddings of the entities and relations in G are obtained by training with binary cross-entropy loss. These embeddings from the KGE stage are subsequently used as the initial embeddings in the RL stage.

4.3. State Representation

The state representation quality has a significant impact on the RL performance of the agent. A well-defined state representation can capture essential features of the environment, thereby enabling the agent to make more informed and effective decisions. In this section, we explain the various types of information for refining the state representation, comprising the graph structure information, path history information, and query-related guiding information. By improving the state representation, we aim to enhance the agent's ability to understand and navigate complex environments, which ultimately leads to better learning outcomes and more robust policies.

Graph structure information: We explicitly encode the local structure information of the graph to enrich the state representation. Inspired by CompGCN [32], we use GCNs to aggregate the neighbor information of the entities. To be specific, let us denote the current entity as e_t, where $\mathcal{N}(e_t)$ represents the set of neighbors of e_t along its outgoing edges. The representation of e_t after k layers, denoted as $\mathbf{e}_t^{k+1}$, is defined as follows:

$$\mathbf{e}_t^{(k+1)} = f\left(\sum_{(r_{t+1}, e_{t+1}) \in \mathcal{N}(e_t)} \mathbf{W_2}\, \phi\left(\mathbf{r}_{t+1}^k, \mathbf{e}_{t+1}^k\right)\right), \tag{2}$$

where $\mathbf{W}_2 \in \mathbb{R}^{d_o \times d}$ is a learnable weight matrix, ϕ is a non-parametric composition function, such as concatenation $\phi(\mathbf{r}, \mathbf{e}) = [\mathbf{r}; \mathbf{e}]$, addition $\phi(\mathbf{r}, \mathbf{e}) = \mathbf{r} + \mathbf{e}$, or multiplication $\phi(\mathbf{r}, \mathbf{e}) = \mathbf{r} * \mathbf{e}$. The representation of a relation r_t after k layers, denoted as $\mathbf{r}_t^{k+1}$, is defined as follows:

$$\mathbf{r}_t^{k+1} = \mathbf{W}_{\text{rel}}^k \mathbf{r}_t^k, \tag{3}$$

where $\mathbf{W}_{\text{rel}} \in \mathbb{R}^{d \times d}$ is a learnable transformation matrix that maps relations into the entity embedding space so that they can be used in the subsequent GCN layer. We use the entity and relation representations after the kth layer as the final entity and relation representations, which we denote as $\mathbf{e}'$ and $\mathbf{r}'$, respectively.

Path history information: We employ a long short-term memory (LSTM) network to encode the path history information h_t at Step t so that the agent can memorize the path history [14,33]. The LSTM receives the previous history h_{t-1} and the last action $a_t = (r_{t+1}, e_{t+1})$ as input in the following way:

$$\mathbf{h}_t = \text{LSTM}(\mathbf{h}_{t-1}, \mathbf{a}_t). \tag{4}$$

where $\mathbf{a}_t = [\mathbf{r}'_{t+1}; \mathbf{e}'_{t+1}]$.

Query-related guiding information: Given that KGE models outperform multi-hop reasoning models across many KGs, we integrate guiding information from embedding-based models into the state representation to facilitate model learning. Specifically, for the query $(e_s, r_q, ?)$, we utilize the KGE model in Section 4.2 and derive a probability vector denoted as $\mathbf{q} \in \mathbb{R}^{|E|}$. Each dimension i of $\mathbf{q}$ denotes the probability that e_i corresponds to the correct tail entity. We then sample an entity e_i according to the probability distribution $\mathbf{q}$ and denote its vector as $\mathbf{d}_q = \mathbf{e}'_i$.

State representation refinement: After learning the representations of the graph structure information, path history information, and query-related guiding information in the previous stage, the refined state representation is obtained as

$$\mathbf{s}_t = [\mathbf{e}'_t; \mathbf{r}'_q; \mathbf{h}_t; \mathbf{d}_q] \tag{5}$$

where $[;]$ denotes the concatenation operation.

4.4. Policy Network

The parameterized policy network receives the state information and outputs the probability distribution π of the action space.

Action space pruning: The pre-trained KGE model is used to predict the probability of the actions $a_t = (r_{t+1}, e_{t+1})$ for the current entity e_t via the triple scores $\psi_{r_{t+1}}(e_t, e_{t+1})$. To simplify the presentation, we let $A_t = \{a_1, a_2, \ldots, a_m\}$ be the action space at time t, which contains m actions. These actions are sorted based on the values of the scoring function ψ. The sorting can be represented by a permutation ρ such that $f\left(a_{\rho(1)}\right) \geq f\left(a_{\rho(2)}\right) \geq \cdots \geq f\left(a_{\rho(m)}\right)$. The top ϵ actions after sorting are retained as the available candidate actions, which can be represented as the set $\widetilde{A}_t$.

$$\widetilde{A}_t = \left\{a_{\rho(1)}, a_{\rho(2)}, \ldots, a_{\rho(\epsilon)}\right\}. \tag{6}$$

Policy network: Based on the modified state enriched with query-related guiding information, the policy network π_θ outputs the probability distribution $\pi(a_t \mid s_t)$ over all the possible actions in $\widetilde{A}_t$. This process can be expressed as

$$\pi_\theta(a_t \mid s_t) = \sigma\left(\widetilde{\mathbf{A}}_t(\mathbf{W}_3 \,\text{ReLU}(\mathbf{W}_4 \mathbf{s}_t))\right), \tag{7}$$

where σ denotes the softmax operator, $\mathbf{W}_3$ and $\mathbf{W}_4$ are two learnable weight matrices, and $\widetilde{\mathbf{A}}_t \in \mathbb{R}^{\epsilon \times 2d}$ serves as a representation of the action space $\widetilde{A}_t$, formed by concatenating the embeddings of all the actions within it.

4.5. Rule-Based Reward Reshaping

To mitigate the incompleteness of KGs, the reward shaping function R_s was introduced in MultiHopKG [33]. Unlike R_T, which provides only a score of 0 when the answer entity is not present in the KG, R_s provides a score between 0 and 1 as a soft reward in such cases. The score is computed using the pre-trained embedding-based model described in Section 4.2. $R_S(e_T)$ is defined as

$$R_S(e_T) = R_T(e_T) + (1 - R_T(e_T)) \cdot \psi_{r_q}(e_s, e_o). \tag{8}$$

If the answer entity e_T corresponds to a triple (e_s, r_q, e_T) in the KG, and the agent receives a reward, $R_S(e_T) = R_T(e_T) = 1$. If (e_s, r_q, e_T) is not present in the KG, the agent receives a reward $R_S(e_T) = \psi_{r_q}(e_s, e_o)$ computed by the pre-trained embedding-based model. This approach enables the agent to receive informative feedback even when the ground truth is not directly available in the KG.

Although this reward design extends beyond traditional binary rewards and incorporates soft rewards, it does not completely address the issue of spurious paths. Spurious paths can still occur when the agent stumbles upon the correct answer by chance. These paths lack interpretability and fail to provide meaningful insights into the underlying reasoning process.

To mitigate the spurious paths problem and enhance interpretability, additional mechanisms may need to be incorporated into the model. We utilize the KG association rule mining algorithm AnyBURL [38], which is a algorithm for knowledge graph completion within the symbolic space by sampling paths and generalizing them into Horn rules.

The rules mined by AnyBURL are typically Horn clauses in the form of implications. A rule in AnyBURL has the general form of

$$\text{Body} \Rightarrow \text{Head}, \tag{9}$$

where Head is the consequent of the rule, which is typically a single triple pattern, and Body is the antecedent of the rule, which is typically a conjunction of triple patterns. Each rule denoted as H mined by AnyBURL comes with an associated confidence $conf(H)$. The confidence represents the fraction of instances where both the body and head of the rule hold true. We consider a rule that states 'If John is born in Paris, and Paris is located in France, then John's nationality is France'.

$$H_1 : (X, \text{ Born in }, Y) \wedge (Y, \text{ Located in }, Z) \Rightarrow (X, \text{ Nationality, } Z). \tag{10}$$

Here, X, Y, and Z are variables. The resulting path is converted into the form of a Horn rule. For example, the corresponding Horn rule for the path Albert Einstein $\xrightarrow{\text{Born in}}$ Ulm $\xrightarrow{\text{Located in}}$ Baden-Wurttemberg $\xrightarrow{\text{Located in}}$ Germany is

$$H_2 : (X, \text{ Born in }, F) \wedge (F, \text{ Located in }, Y) \wedge (Y, \text{ Located in }, Z) \Rightarrow (X, \text{ Nationality, } Z). \tag{11}$$

If the target entity is reached in a path p, we extract the Horn rule for the path, denoted as H_p, and use the confidence of the rule as the rule reward $R_R(H_p) = conf(H_p)$. In this paper, we use the confidence provided by AnyBURL as the rule reward. The final reward is formally given by

$$R(e_T) = R_T(e_T) \cdot \big(1 + R_R\big(H_p\big)\big) + (1 - R_T(e_T)) \cdot \psi_{r_q}(e_s, e_o). \tag{12}$$

If the answer entity e_T corresponds to a triple (e_s, r_q, e_T) in the KG, i.e., $R_T(e_T) = 1$, the agent receives a reward, $R(e_T) = 1 + R_R(H_p)$. If (e_s, r_q, e_T) is not present in the KG, i.e., $R_T(e_T) = 0$, the agent receives a reward $R(e_T) = \psi_{r_q}(e_s, e_o)$ computed by the pre-trained embedding-based model.

The rule reward scheme can help ensure that the paths generated by the model are logically relevant to the query triple. The interpretability of the model is therefore enhanced by considering the logical relevance of the reasoning paths to the queries during its decision making process.

4.6. Optimization

The policy network is trained to obtain the optimal parameters by maximizing the expected cumulative reward. The objective function can be expressed as

$$J(\theta) = \mathbb{E}_{(e_s, r_q, e_T) \in G} \mathbb{E}_{(a_1, a_2, \cdots, a_T) \sim \pi_\theta} R(e_T), \tag{13}$$

where $J(\theta)$ represents the expected cumulative reward for all the training data following policy π_θ. REINFORCE is utilized to address the optimization problem by iteratively updating θ as follows:

$$\nabla_\theta J(\theta) \approx \sum_{t=1}^{T} R(e_T) \nabla_\theta \log \pi_\theta(a_t \mid s_t), \tag{14}$$

$$\theta = \theta + \eta \nabla_\theta J(\theta), \tag{15}$$

where η is the learning rate. A detailed pseudocode of PMHR is provided in Algorithm 1. Assuming N is the number of episodes, the training set contains M triples, and T represents the maximum number of steps, the time complexity of PMHR simplifies to $O(N \cdot M \cdot T)$.

Algorithm 1: Training Algorithm of the proposed PMHR

Input: KG G, pre-trained embedding model, AnyBURL rules confidence scores, Training Set

Output: parameters of PMHR θ

```
initialize LSTM, Policy network; for episode ← 1 to N do
    for each triple (e_s, r_q, e_T) in Training Set do
        initialize num steps to 0; while num steps < max steps do
            compute the representation of current state with Equation (2),
              Equation (3), Equation (4) and Equation (5); compute the action space
              with Equation (6); compute π_θ(a_t | s_t) with Equation (7) and sample;
              Update the state and increment num steps; if success or num steps =
              max steps then
                break;
            end
        end
        compute the reward with Equation (12); update the parameters with
          Equation (13), Equation (14) and Equation (15);
    end
end
```

5. Experiment

We assess the performance of PMHR for knowledge reasoning tasks using four KG datasets. We first outline the experimental datasets, baseline models, evaluation metrics,

and implementation specifics in the experimental setup. We then conduct a comparative analysis between our model and the baseline models. This is followed by an ablation study aimed at elucidating the effectiveness of the PMHR components. Finally, we present a case study to demonstrate the resulting interpretable reasoning paths.

5.1. Datasets

We utilized four open-access datasets commonly used in KG reasoning comprising UMLS [39], Kinship [33], WN18RR [21], and NELL-995 [9]. UMLS [39] is a comprehensive knowledge base that incorporates general knowledge about various health and biomedical vocabularies. Kinship [33] contains entities and relations representing kinship relationships between family members. WN18RR [21] is a relational dataset derived from WordNet. NELL-995 [9] was obtained from NELL (Never-Ending Language Learning), which is a knowledge base extracted from Web texts for machine learning and natural language processing research. The four datasets contain various types of structural information. The statistics of the datasets are presented in Table 1.

Table 1. Dataset statistics.

Dataset	#Fact			#Ent	#Rel	#Degree	
	#Train	#Valid	#Test			Avg.	Median
UMLS	5216	652	661	135	48	38.6	28
Kinship	8544	1068	1074	104	25	82.2	82
WN18RR	86,835	3034	3134	40,943	11	2.2	2
NELL-995	8747	543	3992	10,105	12	1.6	1

5.2. Baseline Models and Evaluation

We conducted a comprehensive comparison of PMHR with several baseline models comprising embedding-based models, rule-based models, and path-based models.

- The embedding-based models comprise DistMult [18], ComplEx [19], and ConvE [21].
- The rule-based models comprise the following: Neural Logical Programming (NeuralLP) [40] is an end-to-end knowledge base reasoning framework inspired by TensorLog for learning probabilistic first-order logical rules. A neural controller is used in NeuralLP to compose differentiable reasoning operations. AnyBURL [38] is a model for efficiently mining logical rules on KGs.
- The path-based models comprise the following: MINERVA [10] is an automated reasoning model for multi-hop reasoning designed to teach an agent to navigate to the answer entity based on the query. MultiHopKG [33] incorporates embedding-based models as the reward shaping function. MetaKGR [41] uses meta-learning to derive meta parameters from high-frequency relations so that few-shot relations can be quickly adapted to and performance drops for rare relations mitigated. RARL [42] integrates symbolic rules and uses partially random beam search to mitigate sparse reward issues and reduce bias from spurious paths. PAAR [36] is a hyperbolic KGE model with an expanded action space in which KGE scores are utilized as soft rewards, and hyperbolic space metrics are employed as a penalty strategy to address reward sparsity. In Path Spuriousness (PS) [14], Path Spuriousness is introduced to calculate path quality in a model that balances reasoning accuracy and path reasonableness.

The baseline performance metrics shown in Table 2 were obtained directly from the respective studies. We assessed performance using the Hits@k and mean reciprocal rank (MRR) standard metrics. Hits@k represents the proportion of correct tail entities ranked in the top k positions, reflecting the model's ability to prioritize correct answers. MRR

provides the average reciprocal rank of correct answers, indicating how highly the model ranks them overall.

Table 2. Performance comparison of PMHR with baseline models.

	UMLS			**Kinship**			**WN**18RR			**NELL-995**		
	Hit@1	**Hit@10**	**MRR**	**Hit@1**	**Hit@10**	**MRR**	**Hit@1**	**Hit@10**	**MRR**	**Hit@1**	**Hit@10**	**MRR**
DisMult	0.829	0.971	0.877	0.454	0.903	0.593	0.406	0.495	0.434	0.642	0.862	0.730
ComplEx	0.911	0.991	0.943	0.798	0.985	0.873	0.416	0.476	0.436	0.646	0.856	0.729
ConvE	0.926	0.997	0.954	0.815	0.984	0.882	0.398	0.539	0.446	0.673	0.879	0.755
NeuralLP	0.629	0.952	0.766	0.481	0.901	0.620	0.362	0.551	0.423	0.059	0.682	0.349
AnyBURL	0.905	0.986	0.944	0.524	0.922	0.662	0.448	0.558	0.484	0.360	0.545	0.427
MINERVA	0.702	0.924	0.803	0.432	0.920	0.600	0.431	0.528	0.462	0.638	0.825	0.706
MultiHopKG	0.903	0.992	0.940	0.789	0.982	0.866	0.408	0.511	0.442	0.647	0.840	0.720
MetaKGR	0.909	0.994	0.945	0.790	0.986	0.871	0.364	0.502	0.418	0.588	0.834	0.686
RARL	0.762	0.956	0.842	0.613	0.944	0.733	**0.442**	0.533	**0.469**	0.628	0.822	0.704
PAAR	0.895	0.995	0.940	0.727	0.963	0.814	0.406	**0.538**	0.449	0.652	0.839	0.722
Path Spuriousness	0.868	0.992	0.913	0.798	0.982	0.872	0.415	0.513	0.449	0.635	0.840	0.704
PMHR	**0.911**	0.994	**0.947**	**0.807**	**0.987**	**0.878**	0.426	0.525	0.457	**0.684**	**0.849**	**0.747**

Note: The best scores for path-based models are in bold, while those for embedding-based models are underscored.

5.3. Implementation Details

All the experiments were performed using Python 3.7, PyTorch 1.8.1, and the CUDA 11.1. We employed a grid search strategy to identify optimal hyperparameter values. We set the embedding sizes of entities and relations for the four datasets to 200 and utilized a 1-layer GCN as the graph structure encoder and a 3-layer LSTM as the path history encoder. The batch size was set to 512 for the WN18RR dataset, 128 for the Kinship and NELL-995 datasets, and 64 for the UMLS dataset. The embedding dropout rate was set to 0.1 for UMLS, WN18RR, and NELL-995, and to 0.3 for Kinship. The action dropout rate was set to 0.95 for UMLS, 0.9 for Kinship, 0.1 for WN18RR, and NELL-995. In the KGE-pretrained ConvE model, convolutions were performed with a kernel size of 3, and the number of output channels in the convolution layer was set to 32. Adam optimization [43] was employed for parameter refinement. The learning rate was set to 0.001 for UMLS, Kinship, WN18RR, and 0.003 for NELL-995.

To determine the optimal number of hops for multi-hop reasoning, we conducted a statistical analysis of the dataset. Specifically, we utilized the triples in the training dataset to construct a graph and then analyzed the triples in the validation and test datasets. We employed the Breadth-First Search (BFS) algorithm to identify the shortest paths between the head and tail entities and recorded the distribution of the path lengths. Figure 3 shows the distribution of the shortest path lengths in the four datasets. The symbol ∞ indicates that there is no path between the head and tail entities, which means that they are unreachable from one another. As shown in Figure 3, the entities in the UMLS and Kinship datasets can be obtained within two hops, while most of entities in the WN18RR and NELL-995 datasets can be found within three hops. However, some entities in the WN18RR and NELL-995 datasets are unreachable. Therefore, the maximum reasoning hop was set to two for the UMLS and Kinship datasets and three for WN18RR and NELL-995. The values for PMHR in Table 2 were obtained as the averages of the results for three runs.

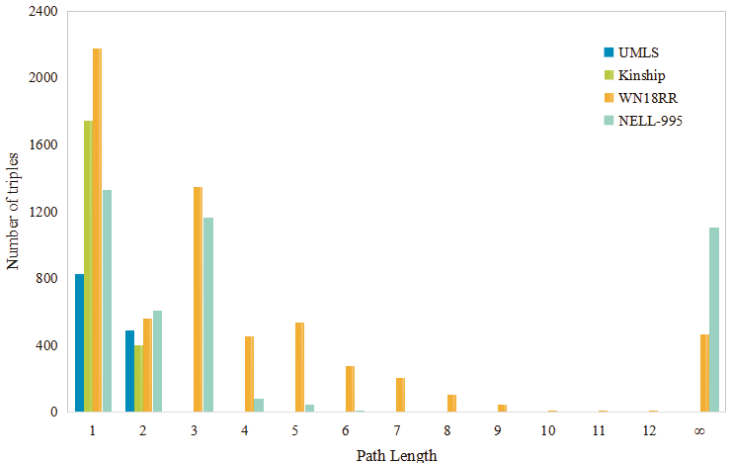


Figure 3. The distribution of the shortest path lengths.

5.4. Results and Analysis

Table 2 shows a comparison between the link prediction results of PMHR and the baseline models on UMLS, Kinship, WN18RR, and NELL-995. The top part shows embedding-based models, the second part shows rule-based models, the third part shows path-based multi-hop reasoning models, and the bottom part shows our proposed method PMHR. The results for these baseline models were obtained by re-running their publicly available open-source code with the original configurations. All experiments were conducted on the same testbed under identical conditions, and we report the mean performance across three independent runs to ensure reliability. For RARL [42] and PAAR [36], as their open-source code was unavailable, we directly cited the results reported in Path Spuriousness [14]. The best scores of the path-based multi-hop reasoning models are highlighted in bold, whereas those of the embedding-based models are underscored.

The performance evaluation metrics in Table 2 indicate that the KGE models outperformed the rule-based models, which exhibited the lowest performance. However, this study emphasizes path-based reasoning, which offers an advantage beyond accuracy: it enables interpretable reasoning paths—a crucial feature often lacking in embedding-based models, especially in applications where understanding the reasoning process is as important as the prediction itself. As shown in Table 2, our model consistently outperformed all the path-based baseline models on three datasets UMLS, Kinship, and NELL-995, and was second only to the state-of-the-art (SOTA) model on WN18RR. These results show that the effectiveness of our model is significantly enhanced by RL through the integration of query-related guiding information into the representation vectors. Although adding a soft reward function helps alleviate the sparse rewards problem in RL-based multi-hop reasoning, it cannot easily resolve the issue of spurious paths. The problem of spurious paths is alleviated in PMHR through rule-based reward reshaping by further modifying the rewards for the paths that correctly reach the target entity.

5.5. Efficiency Analysis

The training time efficiency of our proposed method PMHR was compared against four baseline models (MINERVA [10], MultiHopKG [33], MetaKGR [41], and Path Spuriousness [14]) across four datasets (UMLS, Kinship, WN18RR, and NELL-995). Training time refers to the time required for training the RL agent, excluding the pretraining time for knowledge graph embeddings, which is shared by MultiHopKG [33], MetaKGR [41], Path Spuriousness [14], and PMHR. MINERVA [10] does not include the pretraining process for knowledge graph embedding, which results in a shorter overall running time. Since our analysis focuses on the training efficiency of the RL agent, the pretraining time is excluded for all models. In

Figure 4, the horizontal axis represents the datasets while the vertical axis shows the training time in minutes. Due to the significant differences in training time across datasets, we use a logarithmic scale of base 10 for the vertical axis to better visualize the results.

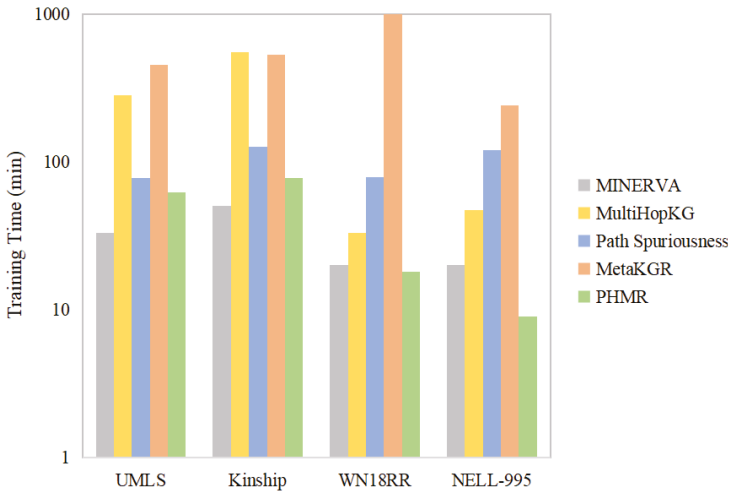


Figure 4. The efficiency comparison between PMHR and path-based models.

The results, as illustrated in Figure 4, highlight the balance our method achieves between training efficiency and effectiveness, as it consistently demonstrates competitive or superior performance in reducing RL agent training time while maintaining better overall results compared to MINERVA [10]. On UMLS, PMHR achieved the second shortest training time, only marginally slower than MINERVA, while delivering superior model performance. On Kinship, PMHR again ranked second in training efficiency, with a slight gap from MINERVA but significantly outperforming MultiHopKG [33], MetaKGR [41], and Path Spuriousness [14], with the latter two showing notably longer training times. On WN18RR, PMHR achieved the shortest training time, narrowly edging out MINERVA [10], while significantly outperforming the other baselines. The RL agent training time of MetaKGR [41] exceeded the vertical axis limit, underscoring its inefficiency in handling larger or more complex datasets. On NELL-995, PMHR demonstrated a more pronounced advantage, achieving the shortest RL agent training time among all models.

In summary, these results highlight the superiority of PMHR, which not only maintains RL agent training efficiency but also delivers better effectiveness compared to baseline models.

5.6. Ablation Study

We performed ablation experiments on the four datasets to study the influence of each model part on the results using the following variants of PMHR:

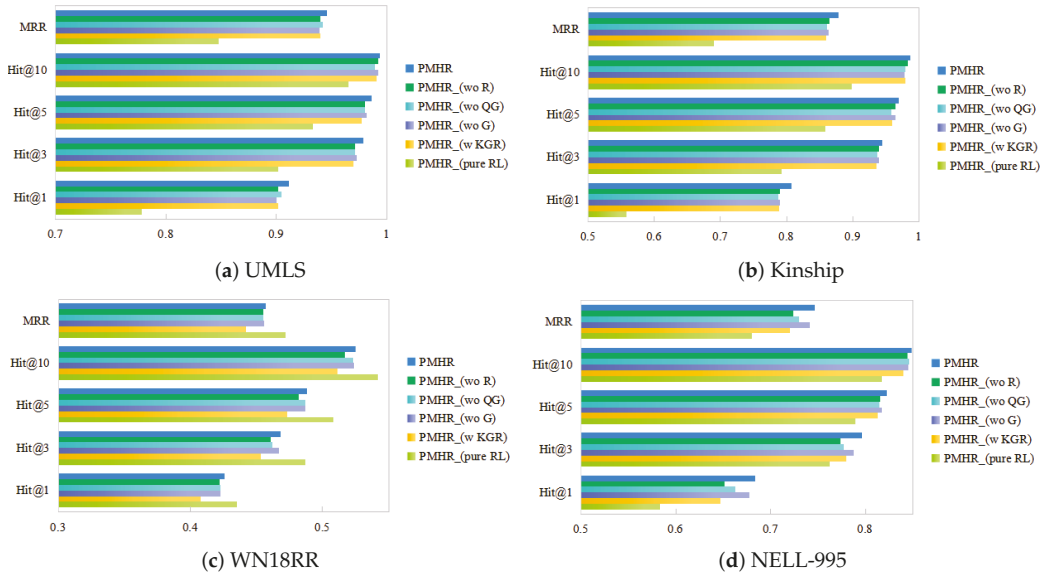
- $\text{PMHR}_{(pure\ RL)}$: A pure RL method is used for KG reasoning without incorporating any enhancements in this variant.
- $\text{PMHR}_{(w\ KGR)}$: The ConvE KG representation learning method is used for soft rewards in this variant, which corresponds to MultiHopKG.
- $\text{PMHR}_{(wo\ G)}$: The graph structure information learning component is omitted in this variant.
- $\text{PMHR}_{(wo\ QG)}$: The query-related guiding information learning component is omitted in this variant.
- $\text{PMHR}_{(wo\ R)}$: The rule-based reward component is omitted in this variant.

The results shown in Figure 5 show that removing any component of the model reduced its effectiveness in most cases. This demonstrates that all parts contribute to the overall performance. Specifically, $\text{PMHR}_{(pure\ RL)}$ is a pure RL method without enhancement. It utilizes

the randomly initialized vectors instead of vectors learned by KG representation learning. Consequently, PMHR$_{(pure\ RL)}$ achieved the worst performance on the UMLS, Kinship, and NELL-995 datasets. An interesting finding is that on the WN18RR dataset, PMHR$_{(pure\ RL)}$ achieved the best performance, which exceeds those of the various variant models, PMHR, and even the SOTA ConvE [21] and DisMult [18] models. Under the PMHR$_{(pure\ RL)}$ method, Hit@1 is 0.435, 0.542 for Hit@10, and 0.472 for MRR, which are better than those of the SOTA models. This is because the reward function is a crucial factor that affects the effectiveness of RL. It directly influences the behavior and decision making process of the agent during the learning process. As shown in Table 2, while the KG representation learning models performed well on other datasets with MRR values above 0.75, their performance on the WN18RR dataset was comparatively poor, with MRR values below 0.5. Thus, representation learning helped mitigate reward sparsity and provided positive guidance on the UMLS, Kinship, and NELL-995 datasets, whereas it interfered with the RL process on the WN18RR dataset. This interference occurred because a high positive reward (close to 1) was assigned to some paths that did not reach the target entity. Consequently, directly using the obtained representations did not have a positive impact on the RL environment and rewards. PMHR$_{(w\ KGR)}$ uses ConvE for KGR and soft rewards. The use of soft rewards helped alleviate the sparse rewards problem in RL and resulted in noticeable improvements on most datasets. A comparison between PMHR and PMHR$_{(wo\ G)}$ shows that the graph structure information helped enrich the state representations. Graph structures offer natural representations of the complex relations between entities. The explicit modeling of these structures allowed the complex interactions within the environment to be captured, thereby enhancing the state representation. A comparison between PMHR and PMHR$_{(wo\ QG)}$ shows that query-related guiding information allowed the agent to make correct choices and improved model performance. A comparison between PMHR and PMHR$_{(wo\ R)}$ shows that adding rule-based rewards improved the model. The agent was further constrained by the rules to find more reasonable paths that lead to the target entity and conform to the rules when it encountered spurious paths that led to the target entity but were unrelated to the query.

5.7. Case Study

In Table 3, we present one triple, two KG reasoning paths generated by PMHR, and three mined rules for the relation 'organizationhiredperson' on the NELL-995 dataset. The inverse of a directional relation r is represented by r^{-1}.



(**a**) UMLS (**b**) Kinship

(**c**) WN18RR (**d**) NELL-995

Figure 5. Ablation study on key parts of PMHR.

Table 3. Case study of PMHR on NELL-995.

Query Triple : (canadian jewish congress, organizationhiredperson, ?) Answer: sylvain_abitbol	
Path	
p_1: canadian jewish congress-organizationhiredperson->E_1-personleadsorganization->E_2-organizationhiredperson->sylvain_abitbol p_2: canadian jewish congress-organizationhiredperson->E_3-personbelongstoorganization->E_4-worksfor^{-1}->sylvain_abitbol	
Rules	Confidence
H_1: organizationhiredperson (X, A), personbelongstoorganization (A, B), worksfor^{-1} (B, Y) =>organizationhiredperson (X, Y)}	0.73
H_2: organizationhiredperson (X, A), coachesteam (A, B), worksfor^{-1} (B, Y) =>organizationhiredperson (X, Y)}	0.38
H_3: organizationhiredperson (X, A), personleadsorganization (A, B), organizationhiredperson (B, Y)=>organizationhiredperson (X, Y)}	0.23

The triple denotes the query triple. The middle section of the table shows the relevant reasoning paths found by the PMHR for the query (canadian jewish congress, organizationhiredperson, ?). The placeholders like 'E_i' are used in replace of the entities to make the presentation simpler, and only the relations are shown. The bottom of the table shows the rules associated with the relation 'organizationhiredperson' and their corresponding confidence values discovered through rule mining with AnyBURL.

As shown in Table 3, for the given query triple (canadian jewish congress, organizationhiredperson, ?), two reasoning paths p_1 and p_2 were identified by PMHR. Because both paths lead to the final target entity, they are assigned equal rewards of 1 by models such as MultiHopKG. However, from a human intuition perspective, the relations in p_1, such as 'personleadsorganization' and 'organizationhiredperson', introduce more intermediate entities. These relations are less clear and less reliable than those in p_2. Consequently, PMHR leverages rule-mining algorithms to obtain query-related rules and their corresponding confidence. The final rewards for each path are adjusted using the confidence of the rules associated with these paths as the rule rewards in such scenarios. For example, the final reward of p_1 is $1 + 0.23$ and that of p_2 is $1 + 0.73$. A more reasonable path has a higher confidence, which results in a larger reward value. This approach encourages the agent to identify logical paths, thereby mitigating the issue of spurious paths.

6. Implications and Conclusions

6.1. Theoretical and Practical Implications

Path-based multi-hop reasoning can be used not only to discover hidden information but also to provide explainable reasoning paths. With the help of KGE and logical rule mining techniques, PMHR addresses the key limitations of existing path-based multi-hop reasoning methods through effective handling of sparse rewards and spurious paths, enhanced state representation, and improved algorithm efficiency. Because of the complexity of multi-hop reasoning, the performance of embedding-based and neural network-based models on complex reasoning tasks is limited by their difficulty in capturing long-distance or highly complex paths. Additionally, it can be challenging for these models to explain their reasoning process. The use of these models in prediction applications that require a high degree of trust such as drug design, disease diagnosis, and fault diagnosis becomes more challenging when users have low confidence in the models. Path-based reasoning can trace relationships between symptoms, diseases, and treatments, leading to more reliable medical diagnoses. Path-based multi-hop reasoning enhances the reliability, interpretability, and overall credibility of KGs by providing clear, traceable reasoning paths for the generated results.

6.2. Conclusions

In this paper, we propose the PMHR framework, which combines KG embeddings and logically rules with RL strategies. To address the issues of sparse rewards and spurious paths, soft rewards and rule-based rewards are integrated into PMHR to mitigate sparse

rewards and encourage the agent to explore more reasonable paths to avoid spurious paths. To enhance state representation and provide more guidance to the agent, the state representation is redefined to incorporate current information, historical information, graph structure information, and query-related guiding information. Furthermore, the embeddings of entities and relations learned through KG representation are leveraged to initialize the environment and prune action space, thereby improving the efficiency of the RL algorithm and reducing the run time. The experimental results on benchmark datasets demonstrate how well PMHR works to increase the accuracy of KG reasoning while maintaining interpretability. On the UMLS, Kinship, and NELL-995 datasets, PMHR obtained Hit@1 increases of 0.63%, 2.02%, and 3.17%, respectively, compared to the state-of-the-art path-based models. In future work, we aim to investigate the alleviation of spurious paths using large language models and explore this area in greater depth.

Author Contributions: Investigation, A.M. and S.Z.; conceptualization, A.M. and Y.Y.; methodology, A.M.; writing—original draft preparation, A.M.; writing—review and editing, Y.Y., C.S. and L.P.; supervision, C.S., T.-S.C. and Y.Y.; funding acquisition, Y.Y. 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 ID U22B2019); National Key Research and Development Program(grant ID 2020YFB2104503).

Data Availability Statement: The data used in this study come from publicly available knowledge graph datasets. For more data information, please visit https://github.com/salesforce/MultiHopKG (accessed on 10 December 2022).

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

Abbreviations

The following abbreviations are used in this manuscript:

KGs	Knowledge graphs
RL	Reinforcement learning
KGR	Knowledge graph reasoning
MDP	Markov decision process
GCN	Graph convolutional network
KGE	Knowledge graph embedding
LSTM	Long short-term memory

References

1. 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.* **2022**, *33*, 494–514. [CrossRef] [PubMed]
2. Xu, W.; Deng, Y.; Zhang, H.; Cai, D.; Lam, W. Exploiting reasoning chains for multi-hop science question answering. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event, 16–20 November 2021; pp. 1143–1156. [CrossRef]
3. Cui, H.; Peng, T.; Han, R.; Han, J.; Liu, L. Path-based multi-hop reasoning over knowledge graph for answering questions via adversarial reinforcement learning. *Knowl.-Based Syst.* **2023**, *276*, 110760. [CrossRef]
4. Heo, Y.J.; Kim, E.S.; Choi, W.S.; Zhang, B.T. Hypergraph transformer: Weakly-supervised multi-hop reasoning for knowledge-based visual question answering. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; pp. 373–390. [CrossRef]
5. Ma, R.; Wang, X.; Cao, C.; Bu, X.; Wu, H.; Zhao, L. GLSEC: Global and local semantic-enhanced contrastive framework for knowledge graph completion. *Expert Syst. Appl.* **2024**, *250*, 123793. [CrossRef]
6. Wang, H.; Song, D.; Wu, Z.; Li, J.; Zhou, Y.; Xu, J. A collaborative learning framework for knowledge graph embedding and reasoning. *Knowl.-Based Syst.* **2024**, *289*, 111505. [CrossRef]
7. Meng, X.; Bai, L.; Hu, J.; Zhu, L. Multi-hop path reasoning over sparse temporal knowledge graphs based on path completion and reward shaping. *Inf. Process. Manag.* **2024**, *61*, 103605. [CrossRef]

8. Shang, B.; Zhao, Y.; Liu, Y.; Wang, C. Attention-based exploitation and exploration strategy for multi-hop knowledge graph reasoning. *Inf. Sci.* **2024**, *653*, 119787. [CrossRef]
9. Xiong, W.; Hoang, T.; Wang, W.Y. Deeppath: A reinforcement learning method for knowledge graph reasoning. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, 9–11 September 2017; pp. 564–573. [CrossRef]
10. Das, R.; Dhuliawala, S.; Zaheer, M.; Vilnis, L.; Durugkar, I.; Krishnamurthy, A.; Smola, A.; McCallum, A. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
11. Wang, H.; Li, S.; Pan, R.; Mao, M. Incorporating graph attention mechanism into knowledge graph reasoning based on deep reinforcement learning. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 2623–2631. [CrossRef]
12. Wan, G.; Pan, S.; Gong, C.; Zhou, C.; Haffari, G. Reasoning like human: Hierarchical reinforcement learning for knowledge graph reasoning. In Proceedings of the 29th International Joint Conference on Artificial Intelligence, Yokohama, Japan 7–15 January 2021. [CrossRef]
13. Tiwari, P.; Zhu, H.; Pandey, H.M. DAPath: Distance-aware knowledge graph reasoning based on deep reinforcement learning. *Neural Netw.* **2021**, *135*, 1–12. [CrossRef] [PubMed]
14. Jiang, C.; Zhu, T.; Zhou, H.; Liu, C.; Deng, T.; Hu, C.; Li, J. Path Spuriousness-aware Reinforcement Learning for Multi-Hop Knowledge Graph Reasoning. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, Dubrovnik, Croatia, 2–6 May 2023; pp. 3173–3184. [CrossRef]
15. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NA, USA, 5–8 December 2013; Volume 26.
16. Ji, G.; He, S.; Xu, L.; Liu, K.; Zhao, J. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*; Association for Computational Linguistics: Beijing, China, 2015; pp. 687–696. [CrossRef]
17. 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, DC, USA, 28 June–2 July 2011; pp. 3104482–3104584.
18. Yang, B.; Yih, W.t.; 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, San Diego, CA, USA, 7–9 May 2015.
19. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex embeddings for simple link prediction. In Proceedings of the 33nd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 2071–2080.
20. Socher, R.; Chen, D.; Manning, C.D.; Ng, A. Reasoning with neural tensor networks for knowledge base completion. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NA, USA, 5–8 December 2013; Volume 26.
21. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2d knowledge graph embeddings. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32, pp. 1811–1818. [CrossRef]
22. Nguyen, D.Q.; Nguyen, T.D.; Nguyen, D.Q.; Phung, D. A novel embedding model for knowledge base completion based on convolutional neural network. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; pp. 327–333. [CrossRef]
23. Lv, X.; Lin, Y.; Cao, Y.; Hou, L.; Li, J.; Liu, Z.; Li, P.; Zhou, J. Do Pre-trained Models Benefit Knowledge Graph Completion? A Reliable Evaluation and a Reasonable Approach. In Proceedings of the Findings of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; pp. 3570–3581. [CrossRef]
24. Wang, L.; Zhao, W.; Wei, Z.; Liu, J. SimKGC: Simple contrastive knowledge graph completion with pre-trained language models. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; pp. 4281–4294. [CrossRef]
25. Xie, X.; Zhang, N.; Li, Z.; Deng, S.; Chen, H.; Xiong, F.; Chen, M.; Chen, H. From discrimination to generation: Knowledge graph completion with generative transformer. In Proceedings of the Companion Proceedings of the Web Conference 2022, Lyon, France, 25–29 April 2022; pp. 162–165. [CrossRef]
26. Zhu, Y.; Wang, X.; Chen, J.; Qiao, S.; Ou, Y.; Yao, Y.; Deng, S.; Chen, H.; Zhang, N. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *arXiv* **2023**, arXiv:2305.13168. [CrossRef]
27. Wei, Y.; Huang, Q.; Kwok, J.T.; Zhang, Y. KICGPT: Large Language Model with Knowledge in Context for Knowledge Graph Completion. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 6–10 December 2023; pp. 8667–8683. [CrossRef]
28. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81. [CrossRef]

29. Chen, J.; Hou, H.; Gao, J.; Ji, Y.; Bai, T. RGCN: Recurrent graph convolutional networks for target-dependent sentiment analysis. In *Knowledge Science, Engineering and Management*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 667–675. [CrossRef]
30. Shang, C.; Tang, Y.; Huang, J.; Bi, J.; He, X.; Zhou, B. End-to-end structure-aware convolutional networks for knowledge base completion. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 3060–3067. [CrossRef]
31. Zhou, L.; Wang, T.; Qu, H.; Huang, L.; Liu, Y. A weighted GCN with logical adjacency matrix for relation extraction. In Proceedings of the 24th European Conference on Artificial Intelligence, Santiago de Compostela, Spain, 29 August–8 September; IOS Press: Amsterdam, The Netherlands, 2020; Volume 325, pp. 2314–2321. [CrossRef]
32. Vashishth, S.; Sanyal, S.; Nitin, V.; Talukdar, P.P. Composition-based Multi-Relational Graph Convolutional Networks. In Proceedings of the 8th International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
33. Lin, X.V.; Socher, R.; Xiong, C. Multi-hop knowledge graph reasoning with reward shaping. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 3243–3253. [CrossRef]
34. Li, R.; Cheng, X. DIVINE: a generative adversarial imitation learning framework for knowledge graph reasoning. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 2642–2651. [CrossRef]
35. Wan, G.; Du, B. Gaussianpath: A bayesian multi-hop reasoning framework for knowledge graph reasoning. In Proceedings of the 35th AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; pp. 4393–4401. [CrossRef]
36. Zhou, X.; Wang, P.; Luo, Q.; Pan, Z. Multi-hop knowledge graph reasoning based on hyperbolic knowledge graph embedding and reinforcement learning. In Proceedings of the 10th International Joint Conference on Knowledge Graphs, Virtual, 6–8 December 2021; pp. 1–9. [CrossRef]
37. Lv, X.; Han, X.; Hou, L.; Li, J.; Liu, Z.; Zhang, W.; Zhang, Y.; Kong, H.; Wu, S. Dynamic anticipation and completion for multi-hop reasoning over sparse knowledge graph. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Online, 16–20 November 2020; pp. 5694–5703. [CrossRef]
38. Meilicke, C.; Chekol, M.W.; Fink, M.; Stuckenschmidt, H. Reinforced anytime bottom up rule learning for knowledge graph completion. *arXiv* **2020**, arXiv:2004.04412.
39. Kok, S.; Domingos, P. Statistical predicate invention. In Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, USA, 20–24 June 2007; pp. 433–440.
40. Yang, F.; Yang, Z.; Cohen, W.W. Differentiable learning of logical rules for knowledge base reasoning. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 2319–2328.
41. Lv, X.; Gu, Y.; Han, X.; Hou, L.; Li, J.; Liu, Z. Adapting meta knowledge graph information for multi-hop reasoning over few-shot relations. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 3374–3379. [CrossRef]
42. Hou, Z.; Jin, X.; Li, Z.; Bai, L. Rule-aware reinforcement learning for knowledge graph reasoning. In Proceedings of the Findings of the Association for Computational Linguistics, Online, 1–6 August 2021; pp. 4687–4692. [CrossRef]
43. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations, 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

Leveraging SNS Data for E-Sports Recommendation: Analyzing Popularity and User Satisfaction Metrics †

Yuanyuan Wang

Graduate School of Sciences and Technology for Innovation, Yamaguchi University, Ube 755-8611, Japan; y.wang@yamaguchi-u.ac.jp

† This paper is an extended version of my paper published in Wang, Y. E-Sports Recommendation Based on Degrees of Popularity and Satisfaction Using SNS Data. In Proceedings of the 2024 18th International Conference on Ubiquitous Information Management and Communication (IMCOM), Kuala Lumpur, Malaysia, 3–5 January 2024; pp. 1–5.

Abstract: The rapid rise of social media and widespread Internet access have contributed significantly to the global popularity of e-sports. However, while popular e-sports attract considerable attention, niche e-sports remain underexplored, limiting user discovery and engagement. This paper proposes a Twitter-based recommendation system that uses advanced data management and processing techniques to address the challenge of identifying and recommending both popular and niche e-sports. The system analyzes social media metadata, including user IDs, followers, followees, engagements, and impressions, to calculate two critical metrics: popularity and satisfaction. Based on the combination of these metrics, the system calculates overall scores for each e-sports and generates two distinct rankings: one for popular and another for niche e-sports. The proposed system reflects the application of data-driven methodologies and social network analysis in creating recommendations that meet diverse user preferences, highlighting the relevance of data processing technologies in personalized content delivery. Experimental evaluations, using a dataset derived from Twitter hashtags (#) representing 30 target e-sports in 2022, demonstrate the system's effectiveness in capturing the emerging dynamics in e-sports and providing actionable insights for diverse user preferences. This study highlights the potential of SNS-based technologies to advance data processing, analysis, and application within the e-sports ecosystem.

Keywords: e-sports recommendation; social media analytics; popularity metrics; user satisfaction metrics; e-sports rankings; Twitter data mining; data-driven ranking systems

1. Introduction

The global rise of electronic sports (e-sports), which merge competitive gaming with various digital platforms, represents a significant development in contemporary entertainment [1–3]. Over the past decade, e-sports have transformed into a thriving industry, capturing a fan base that extends beyond conventional video game enthusiasts [4]. According to Newzoo's Global Esports & Live Streaming Market Report 2022 [5], global e-sports viewers have increased from just 2 million in 2014 to 532 million in 2022. The advancement of live streaming technologies, including platforms such as YouTube and Twitch, and the development of the mobile e-sports market are mainly responsible for this rapid growth. Popular video games such as VALORANT and League of Legends have also contributed to increasing the global interest in e-sports.

In Japan, the e-sports phenomenon is especially pronounced, with active players expected to reach 3.6 million by 2022, underscoring the country's growing role in the

e-sports ecosystem. Out of the various social media platforms, X (formerly Twitter) has emerged as particularly important for e-sports community interaction. On this platform, users can share multimedia content (text, images, videos), participate in events, and discuss strategy in real time, making it essential to building and expanding the reach of the e-sports ecosystem. Unlike other platforms such as Twitch, which focuses on live streaming, or Discord, which specializes in team coordination and private communication, X excels in public engagement, immediate feedback, and viral marketing. Its features, such as hashtags, retweets, and replies, enable large-scale community engagement and rapid dissemination of content, connecting players, fans, and organizations globally. Additionally, X provides insights into user behavior to support personalized marketing and audience growth. By facilitating open interaction and extending the reach of content, X serves as a vital tool for the e-sports community, complementing the more specialized roles of Twitch or Discord [6–8].

Recommendation systems are essential for improving content discovery and personalizing user experience in various domains, including e-sports. Traditional recommendation systems rely on collaborative filtering, content-based filtering, or hybrid approaches to predict user preferences [9]. Collaborative filtering utilizes user–item interaction data to identify patterns in user preferences, while content-based filtering focuses on matching user profiles with item attributes [10]. Hybrid systems combine these methods to address limitations such as data sparsity and the cold-start problem [11].

From a financial perspective, the global e-sports market earned USD 1.38 billion in 2022, of which Asia Pacific contributed almost 42%. Regions such as Southeast Asia, Latin America, and the Middle East are emerging as major players in this global industry, with an increasing number of players, teams, and infrastructure investments. Then, it reflects the growing influence of e-sports in driving not only the gaming industry but also global digital entertainment and sports. While video streaming platforms have contributed to the rise of e-sports, SNS platforms have been important in extending community reach and interaction [12–14]. The vast amount of user-generated content on platforms like Twitter offers a valuable opportunity to analyze user behavior, engagement, and satisfaction across different e-sports genres.

Despite the growth of recommendation systems in various domains, e-sports ranking systems often focus on high-profile games with widespread appeal, overlooking niche e-sports that serve smaller but highly engaged communities. This research addresses this gap by combining theoretical insights from recommendation systems with domain-specific metrics for e-sports. In particular, the dual-metric approach combines popularity and satisfaction into a scoring mechanism, enabling the development of rankings for both popular and niche e-sports. The rankings produced by this study are particularly valuable for stakeholders in the e-sports ecosystem, including players, organizers, marketers, and developers, as they help uncover untapped opportunities in both mainstream and niche markets. This study extends traditional theoretical frameworks by highlighting the importance of domain-specific considerations, such as community engagement and social media impact, in building comprehensive recommendation systems for e-sports.

This study builds on and significantly extends the previous work presented in [15] by introducing innovative scoring methods and extending the analysis to develop a comprehensive ranking system for e-sports. The primary contributions of the previous work were to calculate the degrees of popularity and satisfaction for each e-sports based on user engagement data. Specifically, the original work extracted key metadata such as user IDs, number of followers, number of followees, number of engagements, and number of impressions to measure popularity and satisfaction. Then, the author proposed a classification framework, as depicted in Figure 1, which distinguishes between popular and

niche e-sports based on these calculated popularity and satisfaction degrees. In this extended version, the research goes beyond the foundational work by introducing two critical advancements. First, it develops and implements novel scoring methods that integrate popularity and satisfaction metrics into overall scores, enabling the generation of rankings for both popular and niche e-sports. Second, this extended system provides deeper insights by reflecting the dynamic nature of popular e-sports while providing actionable recommendations tailored to niche e-sports. These enhancements improve the framework's applicability, particularly for personalized recommendations, addressing diverse user preferences and providing a more robust decision-making tool for stakeholders, such as users who want to discover new games and organizers who aim to improve audience engagement. The additional results presented in this paper underscore the utility and validity of these enhancements, marking a significant progression from the preliminary findings in [15].

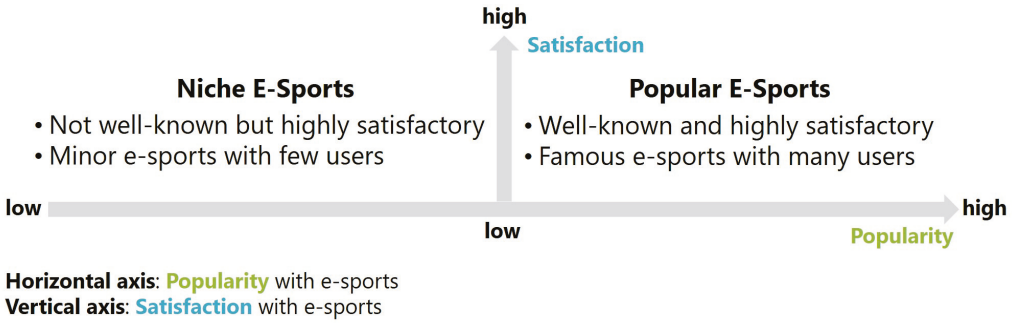


Figure 1. Two types of e-sports can be classified based on popularity and satisfaction metrics.

Popular e-sports are characterized by being well known (high popularity) and highly satisfying to players, attracting large player communities, such as famous e-sports with many users. In the e-sports environment, the popular ranking focuses on e-sports with substantial user bases. On the other hand, niche e-sports refer to e-sports with lower overall popularity but high satisfaction degrees, attracting a small number of dedicated players, such as fewer mainstream genres or e-sports with small but passionate fan bases. This study emphasizes the importance of promoting niche e-sports as they highlight diverse user interests and encourage inclusivity within the e-sports ecosystem. From an industry perspective, niche e-sports can provide a fertile ground for innovation and new market opportunities, giving developers and marketers insight into underexplored segments. Furthermore, players and fans benefit from personalized recommendations that align with their unique preferences, increasing satisfaction and deepening engagement within the e-sports community.

To validate the effectiveness and usefulness of two types of e-sports rankings by the proposed system, the author used a dataset derived from Twitter through hashtags (#) corresponding to the names of the author's randomly selected 30 target e-sports in the global e-sports market in 2022, covering genres such as shooter games (FPS: first-person shooter/TPS: third-person shooter), grappling games, and sports games. This study focuses on Twitter (now known as X) as the primary data source due to its unique role in facilitating large-scale community engagement, real-time interactions, and viral content sharing within the e-sports ecosystem. While social media platforms like Twitch and Discord serve specific purposes, such as live streaming and private team communication, Twitter stands out for its ability to reflect public sentiment and individual user behavior globally. These characteristics make Twitter particularly well suited for analyzing popularity and satisfaction metrics in the e-sports domain.

To sum up, the main contributions of this work are as follows:

- **Dual-Metric Ranking System:** This work develops a Twitter-based e-sports recommendation system that leverages SNS data to generate two rankings—one for popular e-sports and another for niche e-sports—based on popularity and satisfaction metrics.
- **Extension of Previous Research:** This work extends the previous work [15] by proposing new scoring methods to calculate an overall score for each e-sport, effectively combining popularity and satisfaction degrees to provide more comprehensive e-sports rankings than the previous work.
- **Personalized Recommendations:** The dual-metric ranking system improves content discovery by addressing the under-representation of niche e-sports and promoting personalized recommendations, allowing users to explore both popular and niche e-sports, satisfying the diverse preferences within the e-sports community.
- **Actionable Insights for Stakeholders:** The results of this study provide valuable insights for game developers, marketers, and community organizers to design targeted strategies for audience growth and user engagement.

The remainder of this paper is structured as follows: Section 2 reviews related research on e-sports ranking methodologies and the application of social media analytics in understanding e-sports. Section 3 outlines the methodology used to collect and process Twitter data, focusing on calculating popularity and satisfaction degrees. Section 4 details the e-sports recommendation system developed based on the degrees of popularity and satisfaction, providing two rankings of popular and niche e-sports. Section 5 discusses the experiments and validation methods used to assess the effectiveness and usefulness of the proposed dual-metric ranking system. Section 6 explores the implications of the findings, potential applications beyond e-sports, and limitations of the proposed system. Section 7 concludes the study with recommendations for future research directions in e-sports recommendation systems.

2. Related Work

Many studies have been conducted on recommendation systems, particularly in games, due to the growing interest in e-sports and the increasing demand for personalized experiences. This section provides a detailed review of existing research on e-sports recommendation systems, the application of social media analytics in gaming, and the interaction between popularity and satisfaction metrics in e-sports. This work aims to develop a comprehensive e-sports recommendation model that incorporates both popularity and satisfaction metrics from SNS data.

2.1. E-Sports Recommendation Systems

Recommendation systems have been widely utilized in both traditional sports and e-sports to recommend games, opponents, or strategies based on player preferences, performance, and contextual factors [16,17]. In e-sports, recommendation systems are pivotal in enhancing player experience, engagement, and satisfaction, making them essential for managing the diverse and rapidly evolving e-sports ecosystem.

Existing research emphasizes different approaches to recommendation systems, particularly within competitive gaming environments. Kurashima et al. [18] proposed a ranking method that utilizes the contextual information extracted from user reviews on various platforms, such as forums and blogs. Similarly, Oku et al. [19,20] introduced a context-dependent ranking method based on a user value criteria model. This method could provide personalized rankings by analyzing user preferences in different contexts. Thus, it highlights the importance of contextualization in adapting recommendation outputs to

individual user needs. When applied to e-sports, these methods underscore the critical role of context in matching players with suitable competitors or games. For example, Achilleos et al. [21] developed a context-aware recommendation system integrated into the ST76 web platform, pioneering the capability to recommend users most likely to participate in upcoming active sports events. These studies collectively emphasize the growing necessity for personalized recommendations in competitive gaming, particularly given the dynamic nature of user preferences influenced by context and playing history.

While these methods are effective, they focus primarily on ranking e-sports or e-sports players without considering the influence of SNS and incorporating user satisfaction metrics. Moreover, they often prioritize performance optimization over other critical metrics, such as user engagement and satisfaction, which are key drivers of long-term user retention. This paper proposes an innovative approach by integrating social media analytics into e-sports recommendation systems, focusing on Twitter, to bridge the gap between traditional ranking approaches and user-centered design. Twitter is a rich data source that provides insights into user behavior, engagement, and preferences. By leveraging these insights, this study introduces a novel methodology that bridges the gap between traditional ranking approaches and user-centered design. Unlike existing systems, which often rely solely on performance metrics, the proposed approach of this study emphasizes user engagement and satisfaction, creating a more holistic and adaptive recommendation framework tailored to the unique dynamics of the e-sports ecosystem.

2.2. Social Media Analytics in Gaming

In e-sports, social networking platforms have emerged as critical channels for understanding player behavior and preferences, focusing on Twitter, Twitch, and Reddit. Previous work on social media analytics in gaming has shown that these platforms are critical in developing community engagement and the popularity of games [22,23]. The role of social media analytics in the gaming industry was investigated by Jo and Shin [24]. Their research highlights the importance of social media data analysis in understanding player behavior and preferences, which is critical to game development and marketing strategies. This research highlights the impact of real-time player feedback on social media in the design of game updates and the maintenance of player engagement.

In particular, Barney and Pennington [25] synthesized the literature on the use of social media in game research. Three main themes are identified in this review: communicating, collaborating, and computing. It examines how social media advertising influences players, the dynamics within gaming communities on social media, and how social media data are used to assess sentiment and predict gaming outcomes. In addition, Di Virgilio et al. [26] investigated the impact of the COVID-19 pandemic on the engagement of users with e-sports teams through social media. Using content and sentiment analysis of Instagram data, they found significant changes in engagement levels and user sentiment during the pandemic. The influence of social media communication on brand equity and participant engagement during e-sports events was also investigated by Wang et al. [27]. They used structural equation modeling to analyze the relationships between social media communication, brand equity, and user engagement, focusing on the League of Legends World Championship.

In one notable study, Kozaki et al. [28] calculated the degrees of popularity and satisfaction with tourist spots by analyzing metadata, such as the number of views, locations, and posting timestamps of geotagged SNS photos from Flickr. In a similar way, Wang and Yue [29] improved on this method by calculating popularity and satisfaction degrees in tourist spots for individual users using SNS metadata, such as user IDs associated with geotagged SNS photos, the number of views, locations, and posting timestamps.

These investigations introduced methods for generating two kinds of pictorial maps using calculated degrees of popularity and satisfaction.

The novelty of this study lies in its attempt to analyze these two metrics simultaneously by developing a dual ranking system for e-sports. By analyzing engagement rates and user satisfaction derived from tweets and social media interactions, this study not only measures e-sports popularity but also accounts for user satisfaction, a factor largely overlooked in previous studies. It provides a more comprehensive view of player preferences and is critical to understanding how players engage with niche e-sports versus popular e-sports.

2.3. Popularity and Satisfaction Metrics in E-Sports

One of the main limitations of previous studies is that they separated popularity and satisfaction metrics. According to [30], there has been a growing interest in assessing the quality of passive gaming video streaming using both subjective and objective metrics. Viewership statistics, game engagement metrics, and tournament participation are often used to access popularity. Esports Charts [31] provides comprehensive viewership and engagement analysis for various e-sports, highlighting the growing importance of data-driven insights in understanding the expansion of the e-sports industry. According to Statista [32], the number of e-sports users in the United States is expected to double by 2024.

Ito and Yabutani [33] revealed the relationship between players' specialization factors and their sports and video game history. This study uses the concept of recreational specialization as the basis for conducting regression analyses from the perspective of marketing and promotion to elucidate the relationship between e-sports players' specialization and their sports and game backgrounds. In addition, Jenny et al. [34] and Williams et al. [35] conducted surveys on the characteristics of individuals involved in e-sports. Studies such as the one by Himmelstein et al. [36] have focused on analyzing viewership metrics to monitor the popularity of games in tournaments such as League of Legends. While excellent at monitoring player specialization and game popularity, these studies do not address user satisfaction.

On the other hand, research on user satisfaction in e-sports tends to focus on player experience and engagement. It is particularly true for the platforms used to play the games. User satisfaction in e-sports is a critical aspect that can impact community retention and engagement. Many studies have been conducted on video games as a massively mediated communication form to understand changes in user satisfaction and experience [37,38]. In addition, Przybylski et al. [39] explored the factors that influence video game performance, focusing on how the satisfaction of basic psychological needs—competence, autonomy, and relatedness—affects the matchmaking rating (MMR) of games such as Dota 2. This study emphasizes the importance of perceived competence and autonomy for user satisfaction. Moreover, Bányai et al. [40] found significant psychological factors in e-sports, including the process of becoming an e-sports player, mental skills, motivations, and the motivations of e-sports spectators. It provides an insight into the psychological aspects that contribute to user satisfaction in the e-sports domain. In this paper, however, the author takes satisfaction a step further by correlating it with game-specific popularity metrics, thus offering a dual perspective that considers engagement and enjoyment.

In this paper, the core idea of this work is the combined analysis of the popularity and the satisfaction of each e-sports. By utilizing the analysis of tweet metadata, this work provides a more integrated method for recommendation systems to recommend e-sports not only according to their popularity but also according to their satisfaction. This method considers the needs of different users, from casual to highly specialized users, and provides personalized recommendations for both popular and niche e-sports.

2.4. Research Contributions

Although existing work on e-sports recommendation systems and social media analytics provides a strong foundation, it also needs improvement, particularly in combining popularity and satisfaction metrics. In this work, the author builds on previous research by developing a comprehensive model that analyzes both metrics in combination, thus providing a novel method for e-sports recommendation. By focusing on both popular and niche e-sports, this work aims to provide personalized, data-driven recommendations that increase user engagement and satisfaction across the e-sports ecosystem.

3. Analysis of Popularity and Satisfaction Metrics Using Twitter Data

In this section, the author details the methodology used to calculate popularity and satisfaction metrics for e-sports using SNS data from Twitter. This process includes data collection, preprocessing, and analysis. The analysis of various metadata of Twitter data, such as user engagements, impressions, and interaction patterns, allows for a comprehensive understanding of user engagement and sentiment surrounding various e-sports to assess how widely known certain e-sports are (popularity) and how much users enjoy or engage with them (satisfaction).

3.1. Data Collection

Twitter is an attractive data source for analysis for several reasons: it is one of the largest social media platforms in the world; it has a significant impact on many aspects of society (daily conversations, news reports, event announcements, etc.) in many different fields (health, entertainment, business, research, politics, etc.); it makes user posts available to everyone; and it allows a large amount of information to be easily accessed and shared.

Therefore, in this paper, the data collection process involves collecting approximately 300,000 tweets related to 30 target e-sports spanning various genres, including shooter games (FPS/TPS), grappling games, sports games, and others in the global e-sports market, from 6 October 2022 to 30 November 2022. These data were collected using the Twitter API (version 2) with academic research access [41].

For genres such as multiplayer online battle arena (MOBA) games, sandbox games, mobile role-playing games (RPGs), rhythm games, and fighting games, the dataset contains only one or two representative e-sports per genre. These lesser-represented genres are grouped under the category "others" to simplify categorization and maintain analytical clarity. While this study acknowledges that more specific subcategories could be defined for the "others" group, the current grouping reflects the limited representation of these genres in the dataset. It is consistent with the study's focus on analyzing broader e-sports trends.

Table 1 shows a list of 30 target e-sports randomly selected in this paper. These include Street Fighter V and League of Legends, which are popular overseas, and others such as Puzzle & Dragons and Puyo Puyo, which are popular in Japan. The tweets are mainly collected based on the official hashtags (#), which are the names of the e-sports directly added to the e-sports screens when the screens are shared from the respective applications. Additionally, the tweets collected here are limited to those written in Japanese only. As a result, each popular e-sports averages 12,000 tweets, and niche e-sports averages 6000 to 8000 tweets.

In this work, metadata were extracted from each tweet to analyze user interactions and sentiments, categorized into **User-Level Data** and **Tweet-Level Data**, as shown in Figure 2. This categorization ensures a clearer understanding of the role of each data type in calculating **Popularity** and **Satisfaction** metrics.

Table 1. List of 30 target e-sports randomly selected in the global e-sports market of the year 2022.

Shooter Games (FPS/TPS)	Grappling Games	Sports Games	Others
Apex Legends	Card Puzzle	eFootball	League of Legends
Battlefield	DEAD OR ALIVE 6	FIFA World Cup	Minecraft
Call of Duty: Modern Warfare	Puyo Puyo	Gran Turismo Sport	Monster Strike
Fortnite	Puzzle & Dragons	NBA	Taiko no Tatsujin
Overwatch	Shadowverse	Rocket League	Teamfight Tactics
PUBG: BATTLEGROUNDS	Street Fighter V	Zwift	World of Warcraft
Rainbow Six Siege	Super Smash Bros.		
Splatoon	TEKKEN 7		
VALORANT	Tetris		

User-Level Data (Used for **Popularity** Metric)

The following metadata describe the characteristics and network influence of individual Twitter users, making them well suited for calculating the **Popularity** metric.

- **User ID**: The unique identifier for the user who posted the tweet.
- **#Followers**: The number of users following the account, which reflects the user's reach and influence within the network. A higher number of followers indicates greater popularity.
- **#Followees**: The number of accounts the user is following, which may provide context about their level of engagement in the community but does not directly impact popularity.

Popularity depends on the reach and visibility of a user's network, which is directly related to the number of people they follow and the total number of people connected to them.

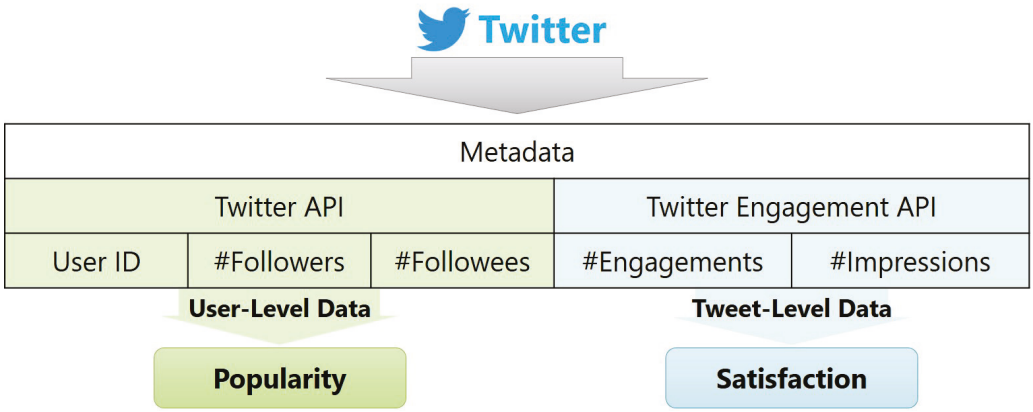


Figure 2. Categorization of metadata collected from tweets for calculating popularity and satisfaction metrics.

Tweet-Level Data (Used for **Satisfaction** Metric)

The following metadata provide insights into the visibility and engagement performance of individual tweets, which are key to calculating the **Satisfaction** metric.

- **#Engagements**: The number of interactions with the tweet, including retweets, likes, replies, URL clicks, hashtag clicks, mention clicks, and media views, covering both organic and promoted contexts. A higher engagement number indicates that users found the content valuable or interesting, which reflects their satisfaction.
- **#Impressions**: The total number of times the tweet has been viewed on Twitter, regardless of whether someone engages with it. Combined with engagement metrics, it can indicate how well the content resonates with its audience.

Satisfaction reflects how users interact with content, making tweet-level metrics ideal for capturing user sentiment and engagement.

The metadata for user-level data, including user ID, number of followers (#followers), and number of followees (#followees), were obtained through the Twitter API (version 2) [42], enabling analysis of user popularity. Engagement-related metadata for tweet-level data, including the total number of engagements (#engagements) and the number of impressions (#impressions), were retrieved from the Twitter Engagement API [43], which allowed for evaluating tweet performance and user satisfaction. In this study, the following preprocessing steps were applied to ensure that the dataset was clean and representative: The first step is to remove non-relevant tweets, bots, spam accounts, and duplicate posts. The next step is to filter for tweets written in Japanese to focus on a specific linguistic and cultural context.

3.2. Calculation of Popularity of E-Sports

In this analysis, popularity measures the degree to which an e-sports is recognized by users on Twitter. The degree of popularity, denoted as $P(a)$ for a given e-sports a, is calculated using the following formula:

$$P(a) = \frac{U(a)}{T(a)} \times \sum_{i=1}^{U(a)} \frac{Fr(a, u_i) + 1}{Fo(a, u_i) + 1} \tag{1}$$

where $U(a)$ represents the number of unique users who post tweets about the e-sports a, while $T(a)$ denotes the total number of collected tweets related to the e-sports a. The functions $Fr(a, u_i)$ and $Fo(a, u_i)$ refer to the number of followers (i.e., those who follow the user u_i) and the number of followees (i.e., those who are followed by the user u_i), respectively, for the user u_i who posted the tweets about a.

This formula denotes the proportion of tweets related to the e-sports a and the influence of the user u_i who posted the tweets, as measured by the user's follower-to-followee ratio. The follower-to-followee ratio is a metric used to evaluate the popularity and influence of a Twitter account [44]. It is the ratio of the number of followers $Fr(a, u_i)$ to the number of followees $Fo(a, u_i)$ of the user u_i, calculated as $\frac{Fr(a,u_i)+1}{Fo(a,u_i)+1}$ in Equation (1), where the addition of 1 to both the numerator and the denominator avoids division by zero and ensures that a user with an equal number of followers and followees will have a ratio of 1. This ratio measures how many users follow the user u_i compared with how many users u_i follows. A high follower-to-followee ratio indicates that the account is influential and well recognized, while a low ratio means that the account lacks engagement, is spammy, or participates in follow-for-follow tactics.

Thus, a higher follower-to-followee ratio indicates that the user has a greater reach and the user's tweets are more likely to contribute to the degree of popularity of e-sports. Therefore, a high degree of popularity can be achieved by having many unique users posting about e-sports, even if they post relatively few tweets. It is especially true if those users have a large number of followers and a small number of followees.

3.3. Calculation of Satisfaction of E-Sports

In this analysis, satisfaction measures the degree to which users enjoy participating in e-sports, as reflected by their interactions on Twitter. The degree of satisfaction, denoted as $S(a)$ for a given e-sports a, is calculated using the following formula:

$$S(a) = \sum_{j=1}^{T(a)} \frac{E(a, t_j)}{I(a, t_j)} \times 100 \tag{2}$$

where $E(a, t_j)$ is the total number of engagements for a tweet t_j related to the e-sports a. Engagements include retweets, likes, replies, URL clicks, hashtag clicks, mention clicks, and media views, reflecting active user interaction. $I(a, t_j)$ is the total number of impressions for the tweet t_j about the e-sports a, representing how many times the tweet was viewed. $T(a)$ is the total number of tweets related to the e-sports a.

This formula calculates the **engagement rate**, expressed as the percentage of engagements relative to impressions for all tweets associated with a specific e-sports. In Equation (2), the engagement rate $\frac{E(a,t_j)}{I(a,t_j)}$ quantifies the proportion of users who engaged with the content out of those who viewed it, providing an indicator of content relevance and user satisfaction. Including a scaling factor (multiplication by 100) converts the engagement rate into a percentage, making the results easier to interpret. A higher engagement rate indicates that users actively interact with the content, suggesting higher satisfaction with the e-sports. For example, if an e-sports has many impressions but relatively few engagements, it may imply that while users are aware of the e-sports, they lack the motivation to engage with it. Conversely, a high engagement rate, achieved through a substantial number of engagements combined with relatively few impressions, indicates high user satisfaction. It is consistent with previous research suggesting that engagement metrics are critical indicators of user loyalty, satisfaction, and content quality [45].

4. E-Sports Recommendation Based on Popularity and Satisfaction Metrics

This section introduces a theoretically grounded recommendation system that ranks e-sports based on two complementary criteria: popularity and satisfaction. The system is designed to bridge the gap between user preferences for widely popular e-sports and highly satisfying but less popular ones, thus ensuring that recommendations cater to diverse user interests. By leveraging these dual metrics, the system accommodates a spectrum of preferences, from mainstream appeal to niche engagement, enriching the user experience in e-sports discovery.

4.1. E-Sports Extraction

The system uses the popularity and satisfaction degrees calculated in Section 3 to classify e-sports into popular and niche types. These classifications are based on the premise that popular e-sports should have both broad appeal (high popularity) and positive user engagement (high satisfaction), while niche e-sports can sustain themselves through high satisfaction despite limited popularity. This theoretical distinction is consistent with the broader goal of capturing varying user expectations.

The system uses thresholds for popularity (α) and satisfaction (β) to classify e-sports. The popularity threshold α is set based on the average of the popularity degrees of all e-sports, and the satisfaction threshold β is set based on the average of the satisfaction degrees of all e-sports. Specifically, α and β serve as benchmarks to identify e-sports that exceed or fall short of typical popularity and satisfaction levels. In this study, the popularity threshold α was set to 0.21, and the satisfaction threshold β was set to 0.22. E-sports that meet or exceed both thresholds ($P(a) \geq \alpha$, $S(a) \geq \beta$) are classified as popular, while those with $S(a) \geq \beta$ but $P(a) < \alpha$ are classified as niche.

The system extracted ten popular e-sports listed in Table 2 and nine niche e-sports listed in Table 3. League of Legends ranked highest in popularity among popular e-sports, while niche e-sports, such as Puzzle & Dragons and NBA, achieved high satisfaction despite lower popularity. Both popularity and satisfaction degrees are higher in other notable e-sports, such as the FIFA World Cup and Shadowverse.

Table 2. Popular e-sports were extracted when $P(a) \geq \alpha$ and $S(a) \geq \beta$.

Popular E-Sports	Popularity ($P(a)$)	Satisfaction ($S(a)$)
Apex Legends	0.36	0.98
FIFA World Cup	0.87	0.51
Fortnite	0.23	0.42
League of Legends	1.00	0.44
Overwatch	0.25	0.63
Puyo Puyo	0.31	0.45
Rocket League	0.41	0.23
Shadowverse	0.62	0.51
Taiko no Tatsujin	0.32	0.82
VALORANT	0.23	0.31
Thresholds	α: 0.21	β: 0.22

Table 3. Niche e-sports were extracted when $P(a) < \alpha$ and $S(a) \geq \beta$.

Niche E-Sports	Popularity ($P(a)$)	Satisfaction ($S(a)$)
Battlefield	0.03	0.83
Minecraft	0.03	0.62
NBA	0.05	1.00
Puzzle & Dragons	0.15	0.99
Splatoon	0.05	0.94
Street Fighter V	0.10	0.41
Super Smash Bros. Ultimate	0.08	0.68
Tetris	0.09	0.62
Zwift	0.00	0.61
Thresholds	α: 0.21	β: 0.22

4.2. E-Sports Ranking

The system ranks popular and niche e-sports separately, using composite scoring methods that combine popularity and satisfaction metrics. This ranking mechanism reflects a theoretical understanding of user preferences: popular e-sports should balance widespread awareness with engagement, while niche e-sports rely on high satisfaction to compensate for limited reach.

As shown in Figure 1, e-sports are classified as popular if their degrees are high in both popularity and satisfaction. These popular e-sports are ranked and recommended to users looking for the most popular and played e-sports. On the other hand, e-sports are classified as niche if they have lower popularity but high satisfaction degrees.

In this paper, the author proposes three distinct methods to calculate the overall score for each e-sports, providing flexible approaches to rank and recommend e-sports tailored to specific user needs.

4.2.1. Popular E-Sports Ranking

It is necessary to balance the influence of these two metrics to recommend popular e-sports in a ranking under high popularity and satisfaction conditions. The ranking of popular e-sports emphasizes their ability to maintain both broad appeal and high engagement. The system captures this dynamic by using the following three methods to calculate the overall score for each e-sport, allowing for a comprehensive ranking:

Method 1: Multiplicative Scoring

This method captures the combined effect of popularity and satisfaction degrees by multiplying the two metrics. The overall score $O(b)$ for each popular e-sports b is calculated by multiplying its popularity degree $P(b)$ in Equation (1) and its satisfaction degree $S(b)$ in Equation (2), using the following formula:

$$O(b) = P(b) \times S(b) \tag{3}$$

where a high overall score $O(b)$ indicates that the popular e-sports b is both widely popular and highly satisfying. Conversely, the overall score will reflect this imbalance if either metric is low. This method suggests that popular e-sports cannot maintain their popularity degrees without satisfying their player base.

Method 2: Relative Strength Scoring

This method utilizes ratios of popularity to satisfaction and satisfaction to popularity to assess the relative dominance of one metric over the other. The overall score $O(b)$ of each popular e-sports b by the ratios of its popularity degree $P(b)$ in Equation (1) to its satisfaction degree $S(b)$ in Equation (2) and its satisfaction degree $S(b)$ to its popularity degree $P(b)$ is calculated using the following formulae:

$$O(b) = \frac{P(b)}{S(b)} \tag{4}$$

$$O(b) = \frac{S(b)}{P(b)} \tag{5}$$

In Equation (4), a high overall score $O(b)$ means that popularity outweighs satisfaction, indicating that the popular e-sports b is well known but may not be as satisfying to its users if the popularity degree $P(b)$ is high. However, the satisfaction degree $S(b)$ is low. Conversely, in Equation (5), a high overall score $O(b)$ indicates high user satisfaction despite low popularity if the satisfaction degree $S(b)$ is high. In contrast, the popularity degree $P(b)$ is low. This method provides insight into whether popular e-sports are driven more by widespread popularity or player engagement, acknowledging that both metrics must exceed certain levels for success.

Method 3: Weighted Composite Scoring

This method combines popularity and satisfaction degrees and weights them to reflect their relative importance. The overall score $O(b)$ for each popular e-sports b is calculated by linearly combining its popularity degree $P(b)$ in Equation (1) and its satisfaction degree $S(b)$ in Equation (2) with varying weights w and $1 - w$, respectively, using the following formula:

$$O(b) = P(b) \times w + S(b) \times (1 - w) \tag{6}$$

where w represents the weight given to the popularity degree $P(b)$ of the popular e-sports b, while $1 - w$ represents the weight given to the satisfaction degree $S(b)$ of the popular e-sports b. By adjusting these weights, the author can prioritize either metric depending on the evaluation context. For example, placing more weight on the popularity degree may help identify trending e-sports, while prioritizing the satisfaction degree could highlight e-sports with strong player loyalty.

These methods provide a comprehensive framework for ranking popular e-sports and offer insights into how the popularity and satisfaction interactions affect their success in the competitive e-sports landscape.

4.2.2. Niche E-Sports Ranking

For recommending niche e-sports with low popularity degrees but high satisfaction degrees in a ranking, it is necessary to adjust the popularity degree $P(c)$ of each niche e-sports c in Equation (1) to reflect low awareness, using $(1 - P(c))$ to align with the low popularity of niche e-sports. This adjustment ensures that this ranking reflects the unique strengths of niche e-sports while maintaining theoretical consistency with the popular ranking framework. The following three methods calculate an overall score for each e-sports based on the same concepts of popular e-sports ranking, allowing for a comprehensive ranking:

Method 1: Multiplicative Scoring

This method emphasizes the interaction between low popularity and high satisfaction by multiplying the adjusted popularity degree $(1 - P(c))$ and the satisfaction degree $S(c)$ in Equation (2). The overall score $O(a)$ for each niche e-sports c is calculated as follows:

$$O(c) = (1 - P(c)) \times S(c) \tag{7}$$

where a high overall score $O(c)$ reflects the niche e-sports c with high satisfaction despite low popularity, emphasizing their uniqueness. This method suggests that niche e-sports can thrive through high player satisfaction, even with limited public awareness.

Method 2: Relative Strength Scoring

This method compares the adjusted popularity degree $(1 - P(c))$ with the satisfaction degree $S(c)$ in Equation (2), and vice versa, to assess the relative strengths of these metrics. The overall score $O(c)$ for each niche e-sports c is calculated using the following formulae:

$$O(c) = \frac{1 - P(c)}{S(c)} \tag{8}$$

$$O(c) = \frac{S(c)}{1 - P(c)} \tag{9}$$

In Equation (8), a high overall score $O(c)$ indicates that the niche e-sports c has low popularity but may be overly dependent on satisfaction to maintain its player base. In contrast, Equation (9) indicates that the niche e-sports c with high satisfaction can succeed in a niche market, even with limited popularity. This method shows how niche e-sports depend on high satisfaction to compensate for low popularity.

Method 3: Weighted Composite Scoring

This method combines the adjusted popularity degree $(1 - P(c))$ and the satisfaction degree $S(c)$ in Equation (2) with assigned weights w and $1 - w$ to reflect their importance. The overall score $O(c)$ for each niche e-sports c is calculated using the following formula:

$$O(c) = (1 - P(c)) \times w + S(c) \times (1 - w) \tag{10}$$

where w represents the weight given to the adjusted popularity degree $(1 - P(c))$ of the niche e-sports c, while $1 - w$ represents the weight given to the satisfaction degree $S(c)$ of the niche e-sports c. By adjusting these weights, the author can emphasize low popularity or high satisfaction, depending on the goal. For example, when looking for e-sports with a highly dedicated fan base, more weight could be given to satisfaction, while a balanced weight could identify e-sports with growth potential in niche markets.

These methods provide a comprehensive framework for ranking niche e-sports and show how low popularity combined with high satisfaction can lead to success, even in a competitive environment.

Therefore, in this study, the proposed ranking methods for popular and niche e-sports provide a theoretically grounded framework for understanding the interplay between popularity and satisfaction and their role in the success of e-sports within different contexts.

5. Experiments and Results

This section outlines the experiments conducted to evaluate the effectiveness and usefulness of the proposed e-sports recommendation system in accurately ranking both "popular" and "niche" e-sports. The experiments consist of four main components: dataset description, accuracy verification for the extracted popular and niche e-sports, quantitative evaluation of their ranking effectiveness, and qualitative evaluation of their ranking usefulness.

5.1. Correct Answer Data

This study uses data from Newzoo's Game Performance Monitor [46] to generate correct answers for "popular" and "niche" e-sports rankings. Newzoo's Game Performance Monitor, formerly known as Newzoo Expert, has provided Activision Blizzard EMEA (Europe, Middle East, and Africa) with approximately 100 million data on player demographics from 1 June 2022 to 30 November 2022. This dataset includes the top game rankings, player counts, player share ratios (%), and total lifetime players for each e-sports within this period.

For this study, e-sports listed in Newzoo's top game rankings serve as candidates for the correct answers. The classification between popular and niche e-sports is determined by calculating an average threshold, γ, based on the ratio of player share ratio (%) $R(x)$ to total lifetime players $L(x)$ for each e-sports x. It is calculated using the following formula:

$$\gamma = \sum_{x=1}^{X} \frac{(R(x) \times 100)/L(x)}{X} \tag{11}$$

where X is the total number of e-sports in Newzoo's top game ranking. The player share ratio $R(x)$ is multiplied by 100 to express it as a percentage (%). If the function $(R(x) \times 100)/L(x)$ in Equation (11) exceeds this average threshold γ, the e-sports are classified as popular e-sports, while those below γ are considered correct answers for niche e-sports.

In addition, the author generates ground truth data for "popular" and "niche" e-sports rankings based on Newzoo's top game ranking using the generated correct answers for "popular" and "niche" e-sports. Table 4 lists the generated correct answers for both "popular" and "niche" e-sports, while Table 5 provides the ground truth data for the top 10 rankings of "popular" and "niche" e-sports, respectively.

Table 4. Lists of generated correct answers for "popular" and "niche" e-sports.

Popular E-Sports	Niche E-Sports
Apex Legends	Clash Royale
Call of Duty: Modern Warfare	Destiny
Counter-Strike: Global Offensive	FIFA World Cup
Dota 2	Granblue Fantasy Versus
eFootball	Hearthstone
Fortnite	Minecraft
HADO	Monster Strike
League of Legends	NBA
PUBG: BATTLEGROUNDS	SAMURAI SPIRITS
Rocket League	Splatoon
Teamfight Tactics	Street Fighter V
TEKKEN 7	Super Smash Bros.
VALORANT	Zwift
World of Warcraft	

Table 5. Ground truth data for the top 10 rankings for "popular" and "niche" e-sports.

Rank	Popular E-Sports	Niche E-Sports
1	League of Legends	Street Fighter V
2	Counter-Strike: Global Offensive	FIFA World Cup
3	Fortnite	Hearthstone
4	Call of Duty: Modern Warfare	Minecraft
5	PUBG: BATTLEGROUNDS	Zwift
6	VALORANT	NBA
7	Rocket League	Clash Royale
8	Apex Legends	Super Smash Bros.
9	Tetris	Destiny
10	World of Warcraft	Splatoon

5.2. Experiment 1: Verification of Accuracy in Popular and Niche E-Sports Extraction

The purpose of Experiment 1 is to verify the accuracy of the proposed recommendation system in extracting popular and niche e-sports, utilizing a combination of popularity and satisfaction metrics. This experiment relies on the lists of correct answers for popular and niche e-sports presented in Table 4 from Section 5.1, which serve as reference datasets for verification.

Accuracy is measured based on Precision, Recall, and F1 Score, calculated using the following formulae:

$$\text{Precision} = \frac{|\text{Correct Answers}| \cap |\text{Popular(Niche) E-Sports}|}{|\text{Popular(Niche) E-Sports}|} \tag{12}$$

$$\text{Recall} = \frac{|\text{Correct Answers}| \cap |\text{Popular(Niche) E-Sports}|}{|\text{Correct Answers}|} \tag{13}$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{14}$$

where "Popular(Niche) E-Sports" refers to popular or niche e-sports extracted according to the method described in Section 4.1.

The results indicate Precision, Recall, and F1 Scores of 50.0%, 35.7%, and 0.417 for popular e-sports, respectively, and 66.7%, 46.2%, and 0.546 for niche e-sports, respectively. These findings highlight the system's stronger performance in identifying niche e-sports than popular e-sports. The lower Recall and F1 Score for popular e-sports suggest that the system struggles to comprehensively capture this category, likely due to the diverse

international fan base of popular titles, which is not fully represented in the Japan-centric dataset used. In contrast, the higher accuracy metrics for niche e-sports reflect their more concentrated fan bases, where engagement patterns are more closely aligned with local data.

These results underscore the sensitivity of the proposed method to regional user behavior, as indicated by social media engagement metrics. The reliance on Japanese language Twitter data introduces a significant bias, skewing the analysis toward Japanese-speaking users and under-representing global trends. Future work should address these limitations by expanding the dataset to include a more diverse selection of e-sports, incorporating multilingual and multi-platform data, and developing dynamic models to adapt to real-time changes in user preferences. While the system shows promise in extracting niche e-sports, improving its generalizability to popular e-sports will require improvements in dataset diversity and analytical methodology.

5.3. Experiment 2: Verification of the Effectiveness of E-Sports Rankings

The purpose of Experiment 2 is to identify the most efficient ranking method for "popular" and "niche" e-sports by combining two key metrics: popularity and satisfaction. This experiment evaluates seven ranking methods, including both baseline and proposed methods, to assess the effectiveness of the newly introduced scoring methods specifically. By comparing these methods against the established top 10 rankings from Newzoo (serving as the ground truth), the author aims to demonstrate the improved accuracy and relevance of the proposed methods in ranking e-sports. The details of this comparison are thoroughly outlined in Table 5 from Section 5.1.

The seven ranking methods include four baseline methods and three proposed methods, which are listed below:

- **Baseline Methods**
  - **B1: Ranking by Popularity Degrees**
    This method ranks e-sports based on their popularity degree $P(a)$ as defined in Equation (1).
  - **B2: Ranking by Satisfaction Degrees**
    This method ranks e-sports based on their satisfaction degree $S(a)$ as defined in Equation (2).
  - **B3: Ranking by Maximum of Popularity and Satisfaction Degrees**
    This method ranks e-sports by their maximum popularity degree $P(a)$ and satisfaction degree $S(a)$, calculated as $\max(P(a), S(a))$.
  - **B4: Ranking by Minimum of Popularity and Satisfaction Degrees**
    This method ranks e-sports by their minimum popularity degree $P(a)$ and satisfaction degree $S(a)$, calculated as $\min(P(a), S(a))$.
- **Proposed Methods**
  - **P1: Ranking by Multiplicative Scoring (Method 1)**
    This method ranks popular e-sports b using the overall score $O(b)$ from Equation (3) and niche e-sports c using the overall score $O(c)$ from Equation (7).
  - **P2: Ranking by Relative Strength Scoring (Method 2)**
    This method ranks popular e-sports b using the overall score $O(b)$ from Equation (4) or (5) and niche e-sports c using the overall score $O(c)$ from Equation (8) or (9).
  - **P3: Ranking by Weighted Composite Scoring (Method 3)**
    This method ranks popular e-sports b using the overall score $O(b)$ from Equation (6) and niche e-sports c using the overall score $O(c)$ from Equation (10).

The effectiveness of each ranking method is evaluated using three of the most popular rank-aware metrics: mean reciprocal rank (MRR), mean average precision (MAP), and

normalized discounted cumulative gain ($nDCG$). These metrics have become standard tools for evaluating recommendation performance, especially for applications requiring ranked outputs that emphasize not only accuracy but also the relevance and position of recommended items, which aligns well with the ranking needs of e-sports and is consistent with the goal of this experiment. Regression-based metrics, such as mean absolute error (MAE) and root mean squared error ($RMSE$), were not used because they do not apply to rank-based evaluations.

Each metric serves a specific purpose:

- ***MRR*** focuses on the highest-ranked relevant item, making it suitable for targeted searches [47].
- ***MAP*** evaluates how highly all relevant items are ranked, aiming to maximize relevant items near the top [48].
- ***nDCG*** is similar to the goal of the MAP metric; it assesses the graded relevance of items and prioritizes their placement within the ranking [49].

5.3.1. Ranking Accuracy Results for Popular E-Sports

Table 6 presents the ranking accuracy results for the top 10 "popular" e-sports based on the seven ranking methods. For Method 3 (P3: Ranking by Weighted Composite Scoring), the weight w in Equation (6) was adjusted from 0.9 to 0.1 in increments of 0.1.

Table 6. Ranking accuracy results for popular e-sports by rank-aware metrics, MRR, MAP, $nDCG@5$, and $nDCG@10$.

	Methods	*MRR*	*MAP*	*nDCG@5*	*nDCG@10*
Baseline	**B1:** $P(b)$ by Equation (1)	0.332	0.609	0.616	0.817
	B2: $P(b)$ by Equation (2)	0.296	0.521	0.339	0.759
	B3: $\max(P(b), S(b))$	0.367	0.664	0.553	0.860
	B4: $\min(P(b), S(b))$	0.183	0.433	0.316	0.623
Proposed	**P1:** $P(b) \times S(b)$ by Equation (3)	0.234	0.497	0.384	0.691
	P2: $P(b)/S(b)$ by Equation (4)	**0.378**	**0.700**	**0.684**	**0.895**
	P2: $S(b)/P(b)$ by Equation (5)	0.326	0.589	0.485	0.708
	P3: $O(b)$ by Equation (6) ($w = 0.9$)	0.332	0.609	0.616	0.817
	P3: $O(b)$ by Equation (6) ($w = 0.8$)	0.326	0.589	0.485	0.806
	P3: $O(b)$ by Equation (6) ($w = 0.7$)	0.321	0.575	0.485	0.798
	P3: $O(b)$ by Equation (6) ($w = 0.6$)	0.334	0.597	0.509	0.816
	P3: $O(b)$ by Equation (6) ($w = 0.5$)	0.334	0.597	0.509	0.816
	P3: $O(b)$ by Equation (6) ($w = 0.4$)	0.367	0.664	0.553	0.860
	P3: $O(b)$ by Equation (6) ($w = 0.3$)	0.317	0.564	0.485	0.792
	P3: $O(b)$ by Equation (6) ($w = 0.2$)	0.307	0.544	0.470	0.777
	P3: $O(b)$ by Equation (6) ($w = 0.1$)	0.301	0.531	0.339	0.767

From the ranking accuracy results for popular e-sports in Table 6, the author analyzed the results for each method and summarized them as follows:

- **Baseline Methods**
  - **B1: Ranking by Popularity Degrees** achieved moderate performance across all metrics, indicating that popularity degrees alone can effectively predict e-sports popularity.
  - **B2: Ranking by Satisfaction Degrees** showed lower performance than the baseline method (B1), particularly in $nDCG@5$, suggesting that satisfaction degrees alone are insufficient for ranking popular e-sports.
  - **B3: Ranking by Maximum of Popularity and Satisfaction Degrees** performed better than using either popularity or satisfaction degrees alone in MRR and

MAP, indicating that taking the maximum of the two metrics captures the strengths of both.

- **B4: Ranking by Minimum of Popularity and Satisfaction Degrees** showed the lowest performance of all methods, indicating that the minimum value does not effectively reflect the overall appeal of e-sports.

- **Proposed Methods**
  - **P1: Ranking by Multiplicative Scoring (Method 1)** showed lower performance, possibly due to the amplification of low values when either metric is low.
  - **P2: Ranking by Relative Strength Scoring (Method 2)**
    - **Popularity/Satisfaction** achieved the highest scores across all metrics, indicating that the popularity-to-satisfaction ratio is a strong predictor of e-sports popularity.
    - **Satisfaction/Popularity** performed moderately but not as well as the inverse ratio, suggesting that the popularity degree plays a more significant role when used as the numerator.
  - **P3: Ranking by Weighted Composite Scoring (Method 3)**
    - As the weight shifts from popularity degree to satisfaction degree (from (Popularity $\times$ 0.9 + Satisfaction $\times$ 0.1) to (Popularity $\times$ 0.1 + Satisfaction $\times$ 0.9)), there is an overall decrease in performance, emphasizing the importance of popularity degrees in the ranking.
    - The combination of (Popularity $\times$ 0.4 + Satisfaction $\times$ 0.6) achieved high scores in MRR, MAP, and $nDCG$@10, comparable to the maximum method (B3).

The author conducted two hypothesis tests to statistically validate the performance differences between ranking methods for popular e-sports. The **Friedman test** was chosen because the author needs to compare multiple related samples (different ranking methods) on the same dataset, and it does not assume a normal distribution of the data [50]. The **Wilcoxon Signed-Rank test** was chosen for pairwise comparisons because it is suitable for comparing matched pairs of observations and is robust against non-normal distributions, which is common in ranking evaluation metrics [51]. These tests were conducted as follows:

- **Friedman Test for Overall Method Comparison**
  - **Null Hypothesis (H_0):** All ranking methods perform equally well.
  - **Alternative Hypothesis (H_1):** At least one method performs differently.
  - **Result:** $\chi^2(6) = 14.92$, $p < 0.021$, indicating that significant differences exist between methods.
- **Wilcoxon Signed-Rank Tests for Pairwise Comparisons**
  Comparing the **P2: Popularity/Satisfaction** method with other methods:
  - P2 vs. B1: $Z = -2.366$, $p = 0.018$
  - P2 vs. B2: $Z = -2.521$, $p = 0.012$
  - P2 vs. B3: $Z = -2.197$, $p = 0.028$
  - P2 vs. P1: $Z = -2.668$, $p = 0.008$
  - P2 vs. P3 (best): $Z = -2.103$, $p = 0.035$

The statistical analysis confirms that the **P2: Popularity/Satisfaction** method significantly outperforms other methods for all metrics ($p < 0.05$ for all comparisons), supporting this conclusion about its effectiveness in ranking popular e-sports.

Based on these findings and supported by the statistical significance tests, the **P2: Popularity/Satisfaction** method is identified as the most effective method for ranking popular e-sports. It achieves the highest scores across all evaluation metrics (MRR: 0.378,

MAP: 0.700, *nDCG*@5: 0.684, *nDCG*@10: 0.895 in bold in Table 6). It indicates that the relative relationship between popularity degree and satisfaction degree is critical in determining the popularity of e-sports, effectively identifying e-sports that both are well known and meet user expectations.

5.3.2. Ranking Accuracy Results for Niche E-Sports

The proposed recommendation system extracted only nine "niche" e-sports, with the ranking accuracy results presented in Table 7. The popularity degree was adjusted using $(1 - P(c))$ as described in Section 4.2.2 to account for the low popularity of niche e-sports. The weight w in Equation (10) was similarly adjusted from 0.9 to 0.1 in increments of 0.1 for Method 3 (P3: Weighted Composite Scoring).

Table 7. Ranking accuracy results for niche e-sports by rank-aware metrics, *MRR*, *MAP*, *nDCG*@5, and *nDCG*@9.

	Methods	*MRR*	*MAP*	*nDCG*@5	*nDCG*@9
Baseline	**B1:** $1 - P(c)$ by Equation (1)	0.356	0.823	0.786	0.905
	B2: $P(c)$ by Equation (2)	0.321	0.696	0.640	0.858
	B3: $\max(P(c), S(c))$	0.329	0.730	0.684	0.910
	B4: $\min(P(c), S(c))$	0.348	0.751	0.684	0.905
Proposed	**P1:** $(1 - P(c)) \times S(c)$ by Equation (7)	0.353	0.691	0.684	0.910
	P2: $(1 - P(c))/S(c)$ by Equation (8)	0.384	0.877	0.530	0.767
	P2: $S(c)/(1 - P(c))$ by Equation (9)	0.235	0.605	0.515	0.747
	P3: $O(c)$ by Equation (10) ($w = 0.9$)	0.344	0.786	0.786	0.900
	P3: $O(c)$ by Equation (10) ($w = 0.8$)	**0.367**	**0.822**	**0.830**	**0.933**
	P3: $O(c)$ by Equation (10) ($w = 0.7$)	0.353	0.775	0.830	0.933
	P3: $O(c)$ by Equation (10) ($w = 0.6$)	0.353	0.775	0.830	0.933
	P3: $O(c)$ by Equation (10) ($w = 0.5$)	0.353	0.775	0.684	0.910
	P3: $O(c)$ by Equation (10) ($w = 0.4$)	0.353	0.691	0.684	0.910
	P3: $O(c)$ by Equation (10) ($w = 0.3$)	0.326	0.719	0.640	0.871
	P3: $O(c)$ by Equation (10) ($w = 0.2$)	0.326	0.719	0.640	0.871
	P3: $O(c)$ by Equation (10) ($w = 0.1$)	0.324	0.711	0.640	0.871

From the ranking accuracy results for niche e-sports in Table 7, the author analyzed the results for each method and summarized them as follows:

- **Baseline Methods**
  - **B1: Ranking by Popularity Degrees** achieved high performance in all metrics, particularly in *MAP* and *nDCG*@9, confirming that using (1-Popularity) could effectively identify niche e-sports, which typically have lower awareness.
  - **B2: Ranking by Satisfaction Degrees** showed relatively lower performance, particularly in *nDCG*@5 and *nDCG*@9, suggesting that satisfaction degrees alone may not adequately rank niche e-sports.
  - **B3: Ranking by Maximum of Popularity and Satisfaction Degrees** and **B4: Ranking by Minimum of Popularity and Satisfaction Degrees** both achieved high scores in *nDCG*@9 (0.910 and 0.905, respectively), but showed average performance in other metrics.
- **Proposed Methods**
  - **P1: Ranking by Multiplicative Scoring (Method 1)** achieved the highest score in *nDCG*@9 (0.910), indicating that the multiplicative combination performs well in identifying the relevance of the top niche e-sports, although performance declined in other metrics.
  - **P2: Ranking by Relative Strength Scoring (Method 2)**

    - **(1-Popularity)/Satisfaction** achieved the highest scores in MRR (0.384) and MAP (0.877) but lower scores in $nDCG@5$ (0.530) and $nDCG@9$ (0.767). It suggests that while the ratio method is effective for top-ranked e-sports, it may have less impact on overall ranking relevance.
    - **Satisfaction/(1-Popularity)** performed poorly across all metrics, indicating that it is not an effective method for ranking niche e-sports.
  - **P3: Ranking by Weighted Composite Scoring (Method 3)**
    - **Weights Biased Towards (1-Popularity)** showed that higher emphasis on (1-Popularity) resulted in improved performance across all metrics with the combination of ((1-Popularity) $\times$ 0.8 + Satisfaction $\times$ 0.2).
    - **Balanced Weights or Bias towards Satisfaction** showed a decline in performance as the weight shifted towards satisfaction degrees, underscoring the critical role of (1-Popularity) in ranking niche e-sports.

To validate the performance differences between ranking methods for niche e-sports, the author used the same statistical approach as for popular e-sports. The **Friedman test** was used to detect overall differences across methods, while the **Wilcoxon Signed-Rank test** was applied for pairwise comparisons. These non-parametric tests were chosen because they are appropriate for ranking data and do not require assumptions about the underlying distribution of the metrics. The tests were conducted as follows:

- **Friedman Test for Overall Method Comparison**
  - **Null Hypothesis (H_0):** All ranking methods perform equally well.
  - **Alternative Hypothesis (H_1):** At least one method performs differently.
  - **Result:** $\chi^2(6) = 12.84$, $p < 0.046$, indicating that significant differences exist between methods.
- **Wilcoxon Signed-Rank Tests for Pairwise Comparisons**
  Comparing the **P3:** $O(c)$ **by Equation (10) (**$w = 0.8$**)** method with other methods:
  - P3 ($w = 0.8$) vs B1: $Z = -1.992$, $p = 0.046$
  - P3 ($w = 0.8$) vs B2: $Z = -2.366$, $p = 0.018$
  - P3 ($w = 0.8$) vs P1: $Z = -2.197$, $p = 0.028$
  - P3 ($w = 0.8$) vs P2: $Z = -2.521$, $p = 0.012$

The statistical analysis confirms that **P3** with $w = 0.8$ significantly outperforms other methods ($p < 0.05$ for all comparisons), validating the choice of this weighted combination for ranking niche e-sports.

Based on these findings and statistical validation, the degree of popularity, particularly (1-Popularity), is crucial for effectively ranking niche e-sports. Statistical analysis confirms that by emphasizing popularity in the scoring method (P3), the system can more accurately identify and recommend these niche e-sports ($p < 0.05$ for all comparative tests). The optimal scoring method is the linear combination of ((1-Popularity) $\times$ 0.8 + Satisfaction $\times$ 0.2), which demonstrated statistically significant improvements over other methods across all evaluation metrics, particularly in $nDCG@5$ (0.830) and $nDCG@9$ (0.933). Although the satisfaction degree has a limited role, it remains essential to ensure that recommended e-sports not only meet niche criteria but also achieve high user satisfaction. This statistically validated balance is critical to increasing user engagement and promoting the overall effectiveness of the recommendation system.

5.4. Experiment 3: Verification of the Usefulness of E-Sports Rankings

In Experiment 3, a questionnaire survey was conducted to verify the usefulness of the proposed recommendation system, which provides two distinct rankings for "popular"

and "niche" e-sports. This survey employed a within-subjects design where the same 31 participants evaluated both popular and niche e-sports using identical criteria (interest and satisfaction). This design is statistically more powerful than a between-subjects design because it controls for individual differences [52]. The insights gained from these rankings have potential real-world applications, such as informing the design of e-sports platforms to increase user engagement and guiding content curation strategies to improve audience satisfaction.

This survey did not require ethics committee approval as it did not involve animal or human clinical trials and did not raise significant ethical concerns. The study adhered to the ethical principles outlined in the Declaration of Helsinki. All participants were informed of the survey's purpose and provided their informed consent before participating. The anonymity and confidentiality of all participants were strictly maintained, and participation was entirely voluntary.

The survey involved 31 university students in their 20s, including 15 participants from Japan and 16 from China. Participants answered the following 11 specific questions in the questionnaire [53] while browsing the "popular" e-sports ranking based on the **P2: Popularity/Satisfaction** method and the "niche" e-sports ranking based on the **P3: Weighted Composite Scoring (with w = 0.8)** method, which achieved the best results in Experiment 2, as described in Section 5.3. This experiment not only tests the practical effectiveness of the newly introduced scoring methods from a technical point of view but also measures their perceived value and user satisfaction in real-world scenarios.

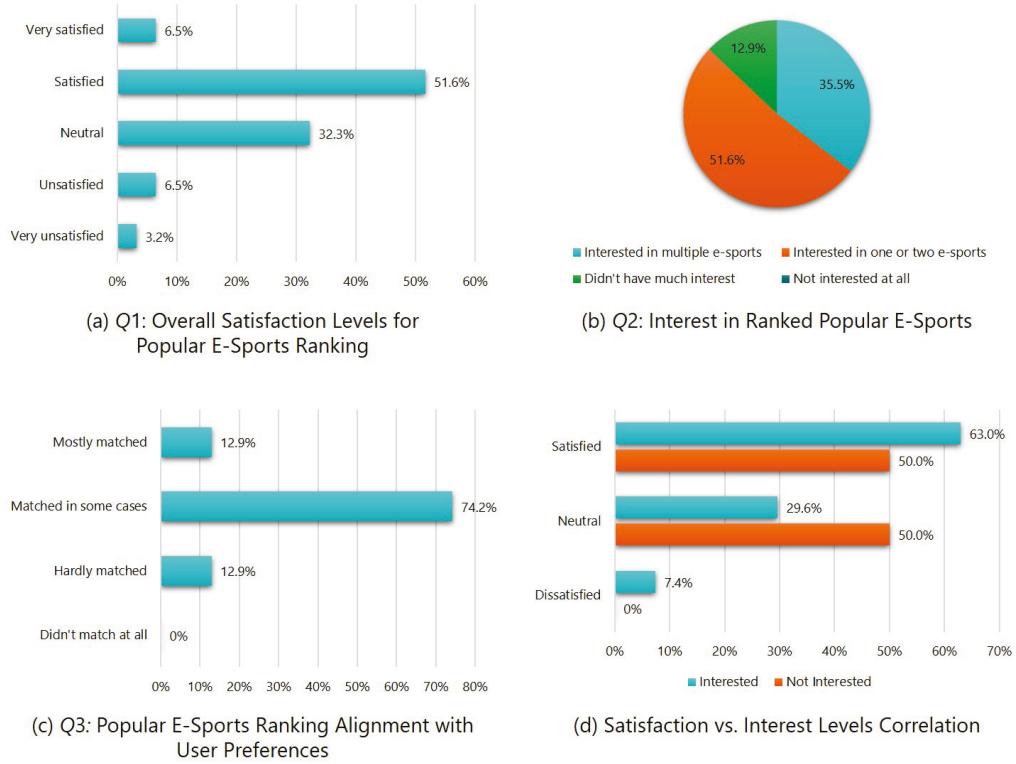
- **Questions Regarding Popular E-Sports Ranking**
  - ***Q*1 :** Please indicate your overall level of satisfaction with the popular e-sports ranking presented.
  - ***Q*2 :** Are you interested in any of the e-sports listed in the popular e-sports ranking?
  - ***Q*3 :** Do the e-sports listed in the popular e-sports ranking match the ones you usually play, watch, or participate in?
  - ***Q*4 :** If there are any e-sports you would like to see in the popular e-sports ranking, please specify.
- **Questions Regarding Niche E-Sports Ranking**
  - ***Q*5 :** Please indicate your overall level of satisfaction with the niche e-sports ranking presented.
  - ***Q*6 :** Are there any e-sports from the niche e-sports ranking that have made you take a new interest in them?
  - ***Q*7 :** Did you discover any new e-sports within the niche e-sports ranking?
  - ***Q*8 :** If there are any e-sports you would like to see included in the niche e-sports ranking, please specify.
- **Overall Feedback**
  - ***Q*9 :** Please indicate your overall satisfaction with both the "popular" and "niche" e-sports rankings.
  - ***Q*10 :** What improvements would you like to see in future rankings?
  - ***Q*11. :** Please feel free to write any additional comments or opinions you may have.

Participants represented a range of e-sports fans with diverse experiences, including playing (17: 54.8%), watching online (live stream) (27: 87.1%), and attending e-sports events in person (11: 35.5%). Their responses provided insights into their satisfaction with the current "popular" and "niche" e-sports rankings, interest levels (high: 18, neutral: 11, low: 2), and how well the rankings aligned with their personal preferences.

5.4.1. Survey Results for Popular E-Sports Rankings

This sub-subsection provides a detailed analysis of survey responses to ***Q1*** through ***Q4*** from 31 participants regarding their experiences with the popular e-sports rankings. The key metrics examined include overall satisfaction, interest in specific popular e-sports, alignment of rankings with user preferences, and participants' suggestions of e-sports for future inclusions in the rankings.

- **Satisfaction with Popular E-Sports Ranking (*Q1*).** As shown in Figure 3a, the majority of the participants (58.1%) expressed satisfaction with the popular e-sports ranking, indicating a generally positive reception. The high level of satisfaction suggests that the rankings effectively meet the participants' expectations for popular e-sports content. However, 32.3% of the participants remained neutral, possibly due to limited engagement with the e-sports currently listed. Only 6.5% of the participants expressed dissatisfaction, and 3.2% reported extreme dissatisfaction, reflecting an overall positive response to the rankings. Further statistical analysis of the satisfaction ratings provided more detailed insights: the mean satisfaction score is 3.71 out of 5 (SD = 0.60), and the 95% confidence interval is [3.49, 3.94]. The distribution shows a consistent pattern, with 16 participants rating 4 (Satisfied), 10 rating 3 (Neutral), and 2 rating 5 (Very Satisfied).



(a) *Q1*: Overall Satisfaction Levels for Popular E-Sports Ranking

(b) *Q2*: Interest in Ranked Popular E-Sports

(c) *Q3*: Popular E-Sports Ranking Alignment with User Preferences

(d) Satisfaction vs. Interest Levels Correlation

Figure 3. Survey results for popular e-sports ranking: (**a**) *Q1*: overall satisfaction levels for popular e-sports ranking, (**b**) *Q2*: interest in ranked popular e-sports, (**c**) *Q3*: popular e-sports ranking alignment with user preferences, and (**d**) correlation between interest and satisfaction levels.

- **Interest in Specific Popular E-Sports (*Q2*).** Figure 3b illustrates that 87.1% of the participants showed interest in popular e-sports within the ranking, with 35.5% interested in multiple e-sports and 51.6% interested in one or two specific e-sports. The remaining 12.9% indicated minimal interest, possibly due to particular e-sports preferences. These findings suggest that the popular e-sports rankings effectively engage most users.

- **Popular E-Sports Ranking Match with Preferred Activities (*Q*3).** In Figure 3c, the majority of the participants (74.2%) felt that the ranking only partially matched their own e-sports preferences and activities. Only 12.9% of the participants felt that their preferences and activities were accurately reflected in the ranking. The remaining 12.9% felt that the ranking barely matched their activities, indicating an apparent disconnect for this subset of users. This disconnect highlights the potential for a broader range of e-sports or customizable rankings to better cater to diverse user preferences.
- **E-Sports Suggested for Inclusion in Popular Ranking (*Q*4).** Participants frequently mentioned e-sports such as Super Smash Bros. (3 mentions), Dota 2 (2 mentions), and eFootball (2 mentions), indicating a strong participant interest in these competitive e-sports. Other suggestions included Counter-Strike 2, PUBG: BATTLEGROUNDS, CSOL, and Hearthstone, demonstrating a desire for a wider variety of e-sports in the rankings. Including these e-sports could further align the rankings with participant preferences and improve overall satisfaction.
- **Correlation between Interest and Satisfaction Levels.** As shown in Figure 3d, the analysis reveals that 63.0% of the participants interested in the popular e-sports ranking reported being satisfied or very satisfied, compared with only 50.0% for those not interested. Neutral responses were higher among those not interested in the popular e-sports ranking (50.0%) than those interested (29.6%), suggesting indifference or less emotional investment among this group. Only 7.4% of the interested participants expressed dissatisfaction, while none of the uninterested participants expressed dissatisfaction, possibly due to lower expectations or a lack of concern about ranking details. These findings are consistent with the Pearson correlation result of 0.309, which indicates a moderate positive relationship between interest and satisfaction with popular e-sports rankings. The results suggest that participants with a strong interest in popular e-sports rankings feel more engaged and satisfied with the rankings provided, although the moderate correlation indicates that other factors may also contribute to user satisfaction.
- **Regional Preferences for Popular E-Sports.** Figure 4 compares regional preferences for popular e-sports between Japanese and Chinese participants as follows:

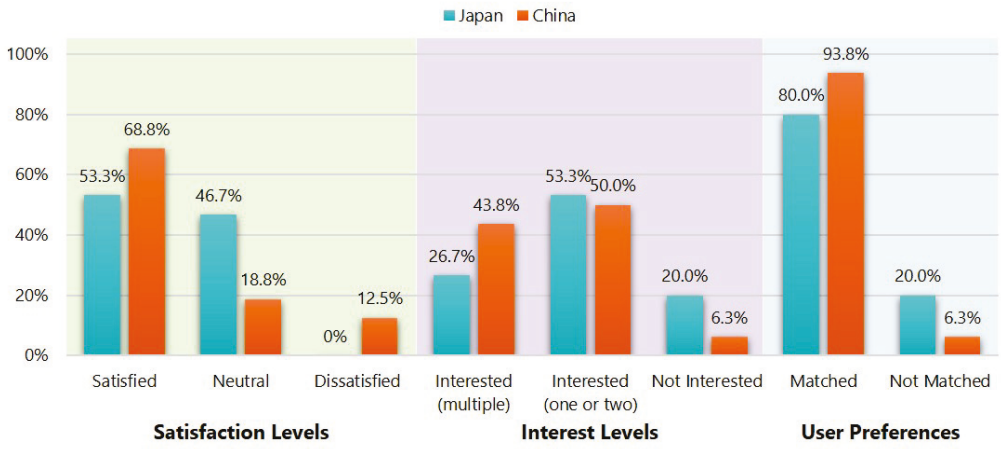


Figure 4. Regional preferences for popular e-sports ranking: Japan vs. China.

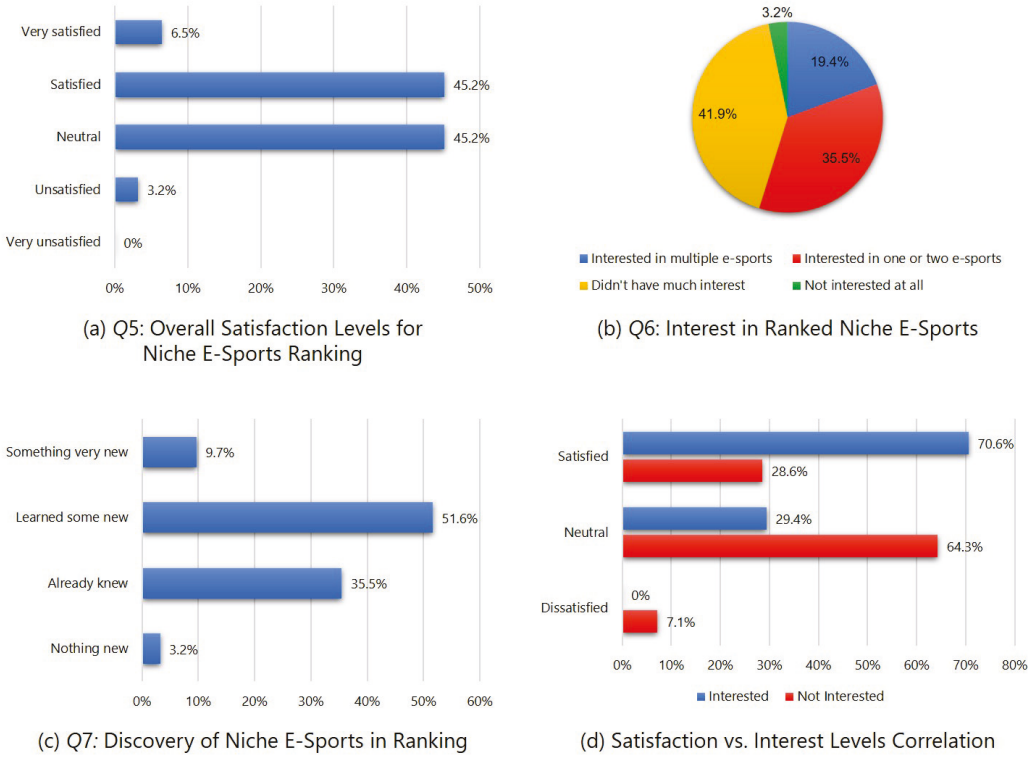
  - **Satisfaction Levels:** Chinese participants reported higher overall satisfaction (68.8%) with fewer neutral responses (18.8%), reflecting a more positive reception of popular e-sports rankings. In contrast, the lack of dissatisfaction among Japanese participants reflects a more balanced but less engaged response.

- **Interest Levels:** Interest in popular e-sports was higher among Chinese participants (43.8%) compared with Japanese participants (26.7%). The significantly lower rate of disinterest among Chinese participants (6.3%) suggests that the ranking of popular e-sports has greater overall appeal in this region.
- **User Preferences:** Both regions strongly aligned with user preferences, with 93.8% of the Chinese participants reporting higher satisfaction with the popular e-sports ranking.

5.4.2. Survey Results for Niche E-Sports Rankings

This sub-subsection provides a detailed analysis of survey responses to questions ***Q*5** through ***Q*8** from 31 participants regarding their experiences with niche e-sports rankings. The key metrics examined include overall satisfaction, interest in niche e-sports, discovery of ranked e-sports, and participants' suggestions of e-sports for future inclusions in the rankings.

- **Satisfaction with Niche E-Sports Ranking (*Q*5).** As shown in Figure 5a, over half of the participants (51.6%) expressed satisfaction with the rankings, suggesting a generally positive reception. However, 45.2% remained neutral, indicating limited engagement likely due to a lack of familiarity with niche e-sports. Only 3.2% of the participants expressed dissatisfaction, with no participants expressing extreme discontent, suggesting that the rankings are widely accepted. A detailed statistical analysis of the niche e-sports satisfaction ratings showed that the mean satisfaction score is 3.60 out of 5 (SD = 0.62), with 95% Confidence Interval: [3.38, 3.82]. The ratings were evenly distributed with 14 participants each rating 3 (Neutral) and 4 (Satisfied) and 2 rating 5 (Very Satisfied).



(a) *Q*5: Overall Satisfaction Levels for Niche E-Sports Ranking

(b) *Q*6: Interest in Ranked Niche E-Sports

(c) *Q*7: Discovery of Niche E-Sports in Ranking

(d) Satisfaction vs. Interest Levels Correlation

Figure 5. Survey results for niche e-sports ranking: (**a**) *Q*5: overall satisfaction levels for niche e-sports ranking, (**b**) *Q*6: interest in ranked niche e-sports, (**c**) *Q*7: discovery of niche e-sports in ranking, and (**d**) correlation between interest and satisfaction levels.

- **Interest in New Niche E-Sports (*Q6*).** Figure 5b shows that approximately half of the participants (54.8%) developed an interest in the niche e-sports listed in the rankings. Among them, 19.4% expressed interest in multiple e-sports, while 35.5% were interested in one or two e-sports. The remaining 45.2% showed little interest, likely due to a lack of familiarity or preference for niche e-sports, indicating potential areas for improvement in introducing these e-sports.
- **Discovery of Niche E-Sports in Ranking (*Q7*).** Figure 5c reveals that 61.3% of the participants found new content through the rankings, with 9.7% discovering entirely new e-sports and 51.6% learning about previously unfamiliar e-sports. However, 35.5% indicated prior familiarity, and 3.2% found nothing new, suggesting that the niche rankings partially achieved their goal of recommending lesser-known e-sports.
- **E-Sports Suggested for Inclusion in Niche Ranking (*Q8*).** Participants frequently mentioned Clash Royale (2 mentions), indicating a specific interest, while Starcraft and MahjongSoul reflect a desire for diverse genres. This feedback supports introducing e-sports from various genres to better align with user preferences.
- **Correlation between Interest and Satisfaction Levels.** As shown in Figure 5d, participants interested in the niche e-sports ranking expressed a significantly higher level of satisfaction (70.6%) compared with those who were uninterested (28.6%). This finding highlights a stronger connection between interest and satisfaction for niche e-sports. A high proportion of uninterested participants (64.3%) expressed neutral feelings, indicating a lack of emotional investment in the niche e-sports ranking. While no dissatisfaction was reported among participants interested in the niche e-sports ranking, a small proportion of uninterested participants (7.1%) expressed dissatisfaction, possibly reflecting unmet expectations or indifference. The strong Pearson correlation result of 0.406 for niche e-sports rankings further supports the conclusion that user satisfaction is more strongly influenced by interest in niche e-sports when compared with popular e-sports. These results emphasize the critical role of interest in determining satisfaction, particularly for niche e-sports where engagement often depends on more dedicated and focused user communities.
- **Regional Preferences for Niche E-Sports.** Figure 6 compares regional preferences for niche e-sports between Japanese and Chinese participants as follows:

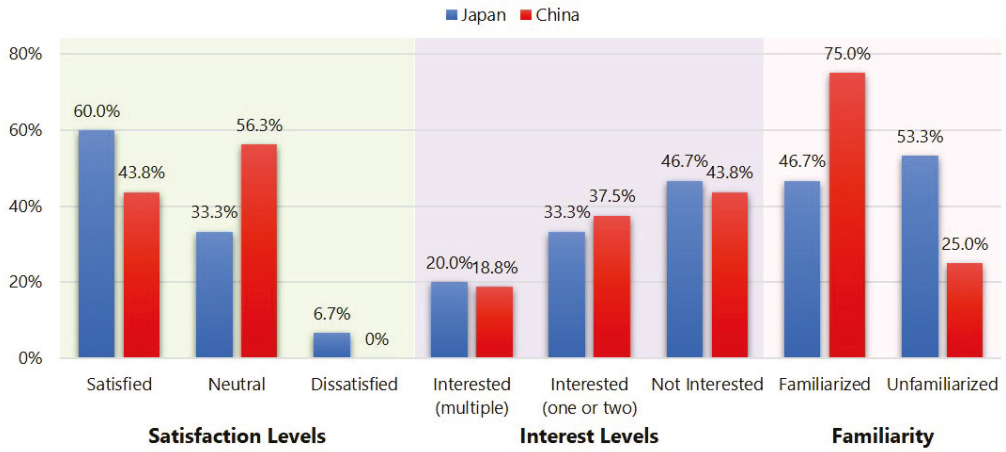


Figure 6. Regional preferences for niche e-sports ranking: Japan vs. China.

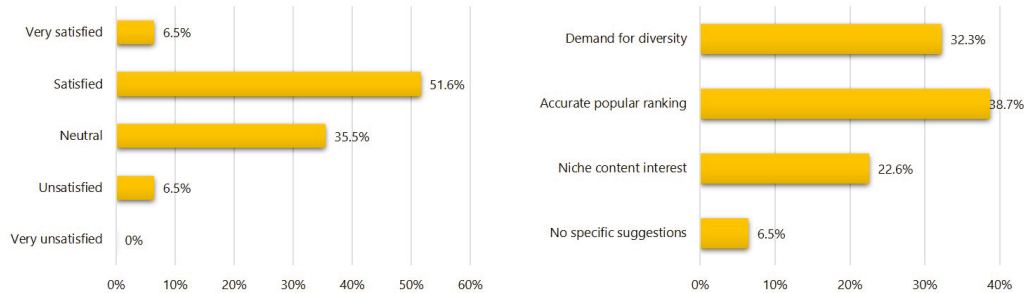
  - **Satisfaction Levels:** Japanese participants (60.0%) reported more satisfaction than Chinese participants (43.8%). However, many Chinese participants (56.3%) expressed neutrality, indicating less emotional engagement with the rankings.

- **Interest Levels:** Interest levels were similar across both regions, though Japanese participants were slightly more engaged in multiple niche e-sports (20.0% vs. 18.8%). Nearly half of the participants from both groups were not interested in niche e-sports.
- **Familiarity:** Chinese participants had a higher discovery rate (75.0%) than Japanese participants (46.7%). It suggests that the niche e-sports rankings may be more effective in introducing new content in China.

5.4.3. Survey Results for Overall Feedback

This sub-subsection analyzes survey responses to questions ***Q*9** through ***Q*11** from 31 participants, focusing on their overall satisfaction with the popular and niche e-sports rankings and their suggestions for improvement.

- **Overview of Satisfaction with Popular and Niche Rankings (*Q*9).** As shown in Figure 7a, 58.1% of the participants expressed satisfaction (either very satisfied or satisfied) with the rankings, indicating a generally positive reception. Nearly one-third of the participants (35.5%) were neutral, suggesting that they may not have strong opinions or expectations for further improvements. Only two participants (6.5%) expressed dissatisfaction, with no participants indicating strong dissatisfaction, highlighting broad acceptance of the rankings. Participants' overall satisfaction was high for both popular and niche e-sports rankings, with mean satisfaction scores of 3.71 (SD = 0.60) and 3.60 (SD = 0.62), respectively, indicating comparable levels of user satisfaction. Further statistical analysis confirmed these findings. The effect size (Cohen's d = 0.19) between satisfaction ratings for popular and niche rankings indicates a slight practical difference, as values less than 0.2 are considered small effect sizes [54]. Combined with 95% confidence [3.49, 3.94] for popular e-sports ranking and [3.38, 3.82] for niche e-sports ranking, this result suggests that participants' satisfaction levels were not significantly influenced by whether the rankings featured popular or niche e-sports. This finding supports the validity of using the same participant group to evaluate both ranking types, as their responses demonstrated consistent engagement and satisfaction across both contexts, reinforcing the robustness of the results.



(a) *Q*9: Overall Satisfaction Levels (b) *Q*10: Improvement Suggestions

Figure 7. Survey results for overall feedback: (**a**) *Q*9: overall satisfaction levels; (**b**) *Q*10: improvement suggestions.

- **Suggestions for Future Improvement (*Q*10).** In Figure 7b, almost one-third of the participants (32.3%) emphasized the need to diversify the rankings by including more genres and styles. Approximately 40% of the participants expressed concerns about improving the accuracy of popular e-sports rankings, indicating a desire for fairness and credibility. Additionally, 22.6% of the participants expressed interest in

a more extensive coverage of niche e-sports, reflecting demand for innovative and lesser-known content.

- **Additional Feedback (*Q*11).** Participants desired more diverse ranking criteria, including factors related to time, platforms, and player experience. Specific feedback included requests for the inclusion of Nintendo-related games, leading to the following suggestions:
  - **Ranking Standards:** Suggestions included classifying rankings by year (e.g., 2023 vs. 2022) and considering gameplay innovation or community impact in the rankings.
  - **Platform-Based Ranking:** Participants expressed interest in rankings by platform, such as mobile, PC, and video games.
  - **E-Sports Inclusion:** It was requested to include Nintendo-related games in the ranking.
  - **Focus on Player Experience:** Many participants preferred to play e-sports rather than watch them.
- **Regional Analysis for Overall Satisfaction and Suggestions.** Figure 8 compares overall satisfaction and suggestions between Japanese and Chinese participants, revealing notable regional differences as follows:
  - **Satisfaction Levels:** Chinese participants showed significantly higher satisfaction (75.0%) than Japanese participants (40.0%). This difference suggests that the current ranking system aligns more closely with the preferences of the Chinese participants. A large proportion of the Japanese participants (46.7%) expressed neutral feelings, while 13.3% reported dissatisfaction, indicating that the ranking system may not fully meet their expectations and that targeted improvements are needed.
  - **Improvement Suggestions:** Both Japanese and Chinese participants emphasized the need for diversity and improved accuracy in popular rankings, reflecting a common desire for a more inclusive and accurate ranking system. A higher proportion of Chinese participants (31.3%) suggested increasing niche e-sports compared with Japanese participants (13.3%), indicating a regional preference for niche content. Japanese participants made specific suggestions for platform-based rankings (e.g., mobile, PC, video games) and the inclusion of Nintendo-related games, while Chinese participants recommended chronological rankings (e.g., annual rankings), highlighting the importance of current gaming trends in China.

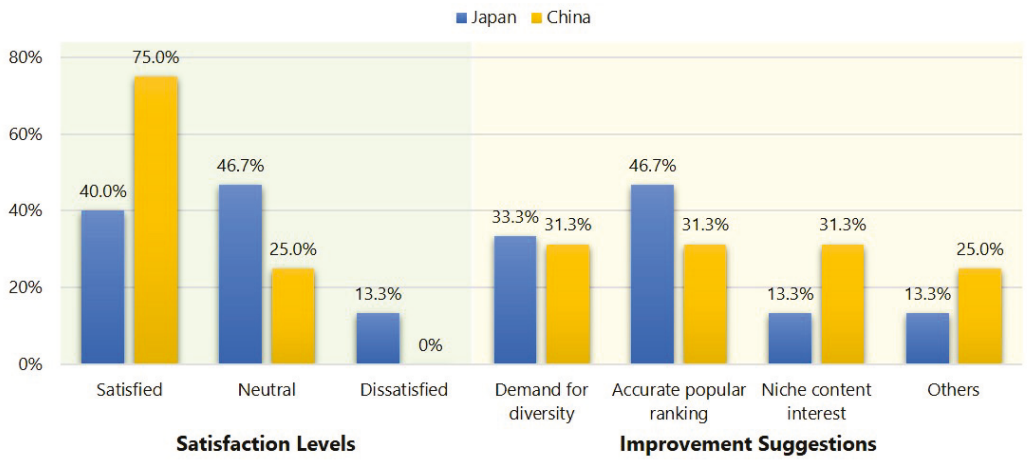


Figure 8. Regional differences in satisfaction levels and improvement suggestions: Japan vs. China.

6. Discussion

This section discusses the factors that influence popular and niche e-sports rankings based on the experimental results presented in Section 5, providing valuable findings about user preferences and the effectiveness of ranking methods. These findings indicate different patterns in how users perceive and interact with popular versus niche e-sports, suggesting improvements and future research directions.

6.1. Popular E-Sports Rankings

The ranking accuracy results showed that popularity is a primary factor in determining the rankings of popular e-sports, consistent with existing literature on user preferences for widely known content. However, this study contributes new insights by demonstrating the effectiveness of the relative strength scoring method (**P2: Popularity-to-Satisfaction Ratio**), which has not been extensively explored in previous work [15] for e-sports recommendations. This ratio effectively highlights how user interest is influenced not only by popularity but also by satisfaction, providing a more nuanced understanding of ranking mechanisms.

Future research could investigate why the popularity-to-satisfaction ratio is so influential, possibly examining underlying psychological or social factors. Additionally, future studies should explore large-scale data integration using distributed computational systems to address scalability and real-world implementation challenges. For example, using cloud-based services to process large datasets of user engagement metrics would provide insight into the robustness and scalability of the system. It would help ensure the applicability of the system in real-world settings.

6.2. Niche E-Sports Rankings

In contrast, the ranking accuracy results showed that niche e-sports rankings depend more on low popularity than satisfaction. It suggested that users associate niche e-sports more with less known than extreme satisfaction. The weighted composite scoring method (**P3: (1-Popularity)** $\times$ 0.8 **+ Satisfaction** $\times$ 0.2), which assigns greater weight to low popularity while moderately weighting satisfaction, achieved the best performance across the evaluation metrics. Unlike previous studies primarily focused on satisfaction with niche content, this study reveals a novel relationship between low popularity and user preferences for niche e-sports. Specifically, the results show that users value the uniqueness and novelty of low-popularity e-sports rather than high satisfaction. This observation highlights an underexplored dimension of e-sports rankings and suggests that perceived novelty plays a larger role in niche markets than previously reported.

These findings suggested that the perceived uniqueness of niche e-sports is more relevant than satisfaction in determining their attractiveness. Interestingly, satisfaction only influences rankings up to a certain threshold, beyond which its influence diminishes. It is consistent with users' perceptions that niche e-sports are valued primarily for novelty and uniqueness. Further enhancement could involve incorporating regional and cultural diversity in niche e-sports data collection, as identified in the regional customization suggestion, ensuring greater inclusivity and cultural relevance.

6.3. Experimental Design

The design of the survey questions, while adjusted for logical sequencing, needs to be more robust due to the novelty of this study and the absence of published frameworks tailored explicitly to the context of e-sports recommendation systems. It has resulted in a questionnaire that may not fully align with both general frameworks for survey design and established theoretical and methodological principles. To address this, the author plans to collaborate with experts in fields such as social psychology and behavioral economics to

adopt existing frameworks for designing more robust and comprehensive questionnaires, thereby improving the overall validity and reliability of the research.

6.4. Suggestions for Improvement

To enhance the ranking system and its applicability in real-world settings, the following technical improvements are recommended:

- **Dynamic and Adaptive Algorithms:** Developing advanced machine learning techniques, such as reinforcement learning and neural networks, to create algorithms that adapt to real-time changes in user behavior and evolving preferences within the e-sports community. Additionally, integrating APIs for retrieving live data will allow the system to respond dynamically to sudden spikes in interest and ensure responsiveness to live trends.
- **Enhance Data Diversity:** Expanding the system to include data from other platforms besides Twitter, such as YouTube, Twitch, and Reddit. It will provide a more complete picture of user preferences and trends.
- **Address Real-World Scalability:** Implementing distributed computing and storage solutions can ensure that the system effectively handles large volumes of data, improving scalability and reliability.
- **Refine Ranking Criteria:** Developing a comprehensive ranking system incorporating dynamic modeling to account for both short-term spikes in interest and long-term growth patterns will improve the system's accuracy and sustainability.

6.5. Practical Implications

The statistical analysis of the survey results from Experiment 3 described in Section 5.4 provides actionable insights for e-sports stakeholders, including platform operators, game developers, and marketers. These findings guide the practical application of ranking systems and user engagement strategies that can also be extended to other digital content domains.

- **Ranking System Design:**
  - The comparable satisfaction levels between popular (Mean = 3.71, SD = 0.60) and niche (Mean = 3.60, SD = 0.62) e-sports rankings, supported by the small effect size (Cohen's $d = 0.19$), suggest that a single, unified platform can effectively serve both markets. Developers should consider integrating rankings for popular and niche e-sports into a centralized, user-friendly dashboard to enhance accessibility and user experience. For example, platforms like Steam or Tencent can integrate both popular and niche e-sports rankings into a single interface, allowing users to toggle between rankings and improving accessibility and user satisfaction efficiently.
  - The overlapping confidence intervals ([3.49, 3.94] for popular e-sports ranking and [3.38, 3.82] for niche e-sports ranking) demonstrate that transparent, data-driven ranking algorithms can ensure consistency and trustworthiness in user evaluations across different categories. Stakeholders should prioritize algorithm transparency to build user confidence by openly communicating the logic behind ranking mechanisms. It can include publishing detailed methodology or algorithmic transparency reports on platform websites and building trust through openness.

  These insights emphasize the potential for creating ranking systems that cater effectively to diverse user preferences while maintaining consistency and accessibility.

- **Market Segmentation Strategy:**
  - The significant regional differences in satisfaction (Chinese participants: 75.0% vs. Japanese participants: 40.0%) highlight the importance of localized content

strategies. For example, platforms targeting Japanese audiences should prioritize Nintendo-related games, while those targeting Chinese users can focus on MOBA e-sports. To operationalize this, platforms like Tencent or Nintendo can develop dedicated landing pages or featured content sections tailored to each region, further enhancing user engagement by focusing on regional interests.
  - Analyzing user demographics when curating rankings can optimize audience reach and engagement. Stakeholders should regularly evaluate regional trends to adapt their offerings dynamically. Platforms like Steam can also run region-specific surveys to gather direct user feedback and optimize content offerings.

  These findings stress the value of demographic analysis in tailoring content strategies to maximize regional user satisfaction and engagement.

- **Content Discovery Optimization:**
  - The high discovery rate among Chinese participants (75.0% vs. 46.7% for Japanese participants) suggests the effectiveness of region-specific content recommendation algorithms. Platforms should invest in machine learning models that leverage regional data to offer personalized content recommendations. For example, platforms like Tencent can train algorithms to recommend MOBA games for Chinese users and Nintendo games for Japanese users based on behavioral trends and preferences specific to each region.
  - Maintaining diverse content categories is critical for sustaining user engagement, as evidenced by the balanced distribution of satisfaction ratings across ranking types. Broadening content diversity should remain a core strategy for platform operators, potentially through partnerships with diverse game publishers, which can ensure a rich catalog of e-sports to attract different user segments.

  These results indicate that region-specific and diverse content discovery mechanisms are key to engaging and retaining users across various demographics.

- **User Engagement Enhancement:**
  - The moderate correlation between interest and satisfaction (Pearson correlation: 0.406 for niche e-sports ranking) highlights the importance of personalized recommendations. Platforms can use predictive analytics, such as collaborative filtering or decision trees, to forecast user preferences and adjust rankings to improve engagement. For example, platforms might analyze users' interactions to recommend niche e-sports that align with their interests and viewing habits.
  - The distribution of user ratings (16 Satisfied, 10 Neutral, 2 Very Satisfied for popular e-sports ranking) provides benchmarks for setting measurable engagement goals. Regularly tracking these metrics and implementing feedback mechanisms will help stakeholders refine user engagement strategies.

  These findings underscore the need for data-driven, personalized engagement strategies to boost user satisfaction and platform loyalty.

These practical implications demonstrate the utility of the findings in enhancing e-sports platforms and provide a foundation for broader applications in digital content curation and user engagement across other industries. By implementing these recommendations, stakeholders can create more effective, inclusive, and user-centric digital ecosystems, ultimately driving growth and satisfaction in the e-sports industry.

6.6. Limitations and Future Research Directions

While this study provides valuable insights, it should acknowledge several limitations to contextualize its findings and guide future research:

- **Sample Size Constraints:**
  - The survey was limited to 31 participants, which is sufficient for preliminary findings but has limitations regarding the generalizability of the results.
  - Future research should involve the following:
    * Expand the sample size using scalable methods such as cloud-based or crowdsourced surveys. It will provide a broader perspective and improve the generalizability of the findings.
    * Incorporate participants from diverse demographic and socio-economic backgrounds to capture a more holistic view of user preferences.
    * Conduct longitudinal studies to capture evolving preferences and trends. For example, multi-year studies can track how user preferences shift with the introduction of new e-sports genres or platforms.
- **Geographic Limitations:**
  - The current study focused exclusively on Japanese and Chinese participants, which limits its applicability to other e-sports markets.
  - Future studies should involve the following:
    * Include participants from other major regions, such as North America, Europe, and Southeast Asia. It will help capture global trends and enhance the applicability of the findings.
    * Investigate cultural and socio-economic factors that shape e-sports preferences in different regions. Understanding these factors can help platforms like Twitch or YouTube tailor content recommendations for specific regions.
    * Analyze regional market dynamics to inform globally relevant strategies. For instance, studying how North American preferences for FPS games differ from East Asian trends in MOBA games can guide platform-specific offerings.
- **Methodological Limitations:**
  - The survey questions may not fully capture user preferences' complex and multifaceted nature.
  - Future research should involve the following:
    * Develop standardized frameworks for measuring user satisfaction and interest in e-sports, enabling more precise comparisons across different studies.
    * Incorporate qualitative research methods, such as interviews or focus groups, to gain deeper insights into user motivations.
    * Validate survey-based findings with experimental or behavioral studies to ensure robustness. For example, controlled experiments can measure user satisfaction more accurately by simulating ranking interfaces.
- **Data Source Limitations:**
  - This study relied solely on Twitter data, which may not fully represent the broader e-sports audience.
  - Future studies should involve the following:
    * Integrate data from multiple platforms, such as YouTube, Twitch, or Reddit, for a more comprehensive view. Cross-platform data can provide deeper insights into user behavior.
    * Develop cross-platform analytics techniques to account for varying user behaviors across platforms.

    - Examine the impact of platform-specific features on user engagement and satisfaction. For example, Twitch's live chat may provide unique insights into real-time user interactions.

- **Technical Implementation Limitations:**
  - The ranking system used in this study does not support real-time adjustments, which limits its adaptability to dynamic user preferences.
  - Future developments should involve the following:
    - Implement advanced machine learning algorithms to enable dynamic, real-time ranking updates. For example, reinforcement learning models can adjust rankings based on real-time user feedback.
    - Design scalable system architectures to accommodate large-scale user data and provide better real-time performance.
    - Develop APIs to facilitate seamless integration with third-party applications and platforms.

By addressing these limitations, future research can build on the current study to advance the understanding of e-sports user behavior and develop more robust, adaptive, and inclusive systems for the industry.

7. Conclusions

The increasing influence of social media and the widespread availability of the Internet have significantly impacted e-sports. While popular e-sports have received considerable attention, niche e-sports still need to be studied more. This study addressed the challenge of identifying and recommending popular and niche e-sports by proposing a Twitter-based ranking system that combines popularity and satisfaction metrics. By utilizing Twitter metadata, including user IDs, followers, followees, engagements, and impressions, the system effectively captured the dynamics of e-sports preferences and provided two distinct rankings for popular e-sports and niche e-sports.

The findings demonstrated the effectiveness of the social media-driven ranking system in addressing diverse user needs in the evolving e-sports environment. Analysis of the experimental results revealed that popularity was the most critical factor for ranking popular e-sports, as higher popularity significantly increased their perceived importance. The popularity-to-satisfaction ratio emerged as a particularly effective metric for reflecting user preferences. For niche e-sports, rankings benefited from prioritizing lower popularity degrees, emphasizing novelty and uniqueness while ensuring that satisfaction met a minimum threshold. These insights highlight the contrasting factors that drive user interest in popular versus niche e-sports.

Despite the promising results, this study has several limitations. The sample size in Experiment 3 was constrained to 31 participants, affecting the findings' generalizability. However, the balanced mix of Japanese and Chinese participants provided preliminary insights into regional differences. Expanding future experiments through cloud-based surveys targeting larger e-sports communities will enhance the reliability of the findings and address scalability issues.

This study also emphasized the significance of regional customization, as preferences varied between Japanese and Chinese participants. This variability highlighted the potential of SNS-based systems to adapt rankings to diverse cultural and regional contexts, promoting broader user engagement. Furthermore, while the study validated the system's effectiveness in a small-scale setting, further discussion on scalability is necessary. Incorporating distributed computing frameworks, such as cloud-based solutions, will

enable the system to handle large-scale datasets, ensuring its practical applicability in real-world environments.

Future studies will include multi-platform analysis to increase the comprehensiveness of this research. By incorporating data from Twitch, Discord, and potentially other platforms, the author aims to account for platform-specific variations—for instance, real-time audience engagement on Twitch and community-driven discussions on Discord. This approach will facilitate a more holistic understanding of e-sports ecosystems and further validate the robustness of the proposed ranking system. In addition, dynamic modeling approaches that account for both short-term spikes in interest and long-term growth patterns could improve the system's predictive accuracy and sustainability.

Future research should also explore additional data dimensions, such as tournament statistics, user engagement metrics, and audience demographics, to refine the ranking methodology further. Enhancing user interaction through feedback mechanisms will be critical for maintaining the relevance and credibility of the system. By addressing these aspects, SNS-based recommendation systems can continue to bridge the gap between popular and niche e-sports, promoting inclusivity, scalability, and growth within the e-sports ecosystem.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this study due to its nature as an anonymous user satisfaction survey, which does not involve personal data collection or invasive procedures by the "Ethical Guidelines for Life Sciences and Medical Research Involving Human Subjects" established by the Japanese Ministry of Health, Labour, and Welfare (MHLW) and the "Regulations for General Research Involving Human Subjects" at Yamaguchi University.

Informed Consent Statement: Informed consent was obtained from all subjects involved in this study.

Data Availability Statement: The data analyzed in this study are subject to the following licenses and restrictions: X Developer Agreement and Policy and Newzoo's License for Research Purposes. Requests to access these datasets should be directed to https://x.com/ (accessed on 6 October 2022) and https://newzoo.com/ (accessed on 28 December 2022). Further inquiries can be directed to the corresponding author.

Acknowledgments: The author would like to thank Yuya Yamaguchi for his assistance with the data collection.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

COVID-19	coronavirus disease 2019
E-Sports	electronic sports
FPS	first-person shooter
MAE	mean absolute error
MAP	mean average precision
MMR	matchmaking rating
MOBA	multiplayer online battle arena
MRR	mean reciprocal rank
nDCG	normalized discounted cumulative gain
RMSE	root mean squared error
RPGs	role-playing games
SNS	social networking services
TPS	third-person shooter

References

1. Yuzyk, M.; Seidner, P. E-Sports Development. In *Developments in Information & Knowledge Management for Business Applications*; Kryvinska, N., Greguš, M., Eds.; Springer International Publishing: Cham, Switzerland, 2022; Volume 5, pp. 615–648. [CrossRef]
2. Tang, D.; Sum, R.K.; Li, M.; Ma, R.; Chung, P.; Ho, R.W. What is esports? A systematic scoping review and concept analysis of esports. *Heliyon* **2023**, *9*, e23248. [CrossRef] [PubMed]
3. Tang, D.; Ma, R.; Chung, P.; Ho, W.; Sum, K.R. Synergistic fields: Unveiling the potential win-win relationship between esports performance and traditional sports participation. *PLoS ONE* **2024**, *19*, e0305880. [CrossRef] [PubMed]
4. Hayday, E.; Collison-Randall, H.; Kelly, S. *Esports Insights*, 1st ed.; Routledge: London, UK, 2022. [CrossRef]
5. Newzoo International B.V. Global Esports & Live Streaming Market Report 2022|Free Version. Available online: https://newzoo.com/insights/trend-reports/newzoo-global-esports-live-streaming-market-report-2022-free-version (accessed on 28 December 2022).
6. Kunz, R.E.; Roth, A.; Santomier, J.P. A perspective on value co-creation processes in eSports service ecosystems. *Sport Bus. Manag. Int. J.* **2022**, *12*, 29–53. [CrossRef]
7. Saiz-Alvarez, J.M.; Palma-Ruiz, J.M.; Valles-Baca, H.G.; Fierro-Ramírez, L.A. Knowledge Management in the Esports Industry: Sustainability, Continuity, and Achievement of Competitive Results. *Sustainability* **2021**, *13*, 10890. [CrossRef]
8. Baltezarević, R.; Baltezarević, V.; Baltezarević, I. The Role of Digital Marketing in the Esports Industry. *Acta Ludologica* **2023**, *6*, 28–45. [CrossRef]
9. Adomavicius, G.; Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 734–749. [CrossRef]
10. Pazzani, M.J.; Billsus, D., Content-Based Recommendation Systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 325–341. [CrossRef]
11. Koren, Y.; Bell, R.; Volinsky, C. Matrix Factorization Techniques for Recommender Systems. *Computer* **2009**, *42*, 30–37. [CrossRef]
12. Li, Y.; Wang, C.; Liu, J. A Systematic Review of Literature on User Behavior in Video Game Live Streaming. *Int. J. Environ. Res. Public Health* **2020**, *17*, 3328. [CrossRef]
13. El Afi, F.; Ouiddad, S. The Rise of Video-Game Live Streaming: Motivations and Forms of Viewer Engagement. In *Proceedings of the HCI International 2021-Posters*; Stephanidis, C., Antona, M., Ntoa, S., Eds.; Springer: Cham, Switzerland, 2021; pp. 156–163. [CrossRef]
14. Ma, Z.; Gu, B. The influence of firm-Generated video on user-Generated video: Evidence from China. *Int. J. Eng. Bus. Manag.* **2022**, *14*, 1–17. [CrossRef]
15. Wang, Y. E-Sports Recommendation Based on Degrees of Popularity and Satisfaction Using SNS Data. In Proceedings of the 2024 18th International Conference on Ubiquitous Information Management and Communication (IMCOM), Kuala Lumpur, Malaysia, 3–5 January 2024; pp. 1–5. [CrossRef]
16. Hong, H.J. eSports: The need for a structured support system for players. *Eur. Sport Manag. Q.* **2023**, *23*, 1430–1453. [CrossRef]
17. Felfernig, A.; Wundara, M.; Tran, T.; Le, V.; Lubos, S.; Polat-Erdeniz, S. Sports recommender systems: Overview and research directions. *J. Intell. Inf. Syst.* **2024**, *62*, 1125–1164. [CrossRef]
18. Kurashima, T.; Bessho, K.; Toda, H.; Uchiyama, T.; Kataoka, R. Ranking Entities Using Comparative Relations. In *Proceedings of the Database and Expert Systems Applications*; Bhowmick, S.S., Küng, J., Wagner, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 124–133. [CrossRef]
19. Oku, K.; Nakajima, S.; Miyazaki, J.; Uemura, S.; Kato, H. A Context-aware Ranking Method Based on User's Value Judgement Model for Information Recommendation. *IPSJ Trans. Databases (TOD)* **2009**, *2*, 57–80. (In Japanese)
20. Oku, K.; Nakajima, S.; Miyazaki, J.; Uemura, S.; Kato, H. A Ranking Method based on Users' Contexts for Information Recommendation. In Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication, Suwon, Republic of Korea, 31 January–1 February 2008; Association for Computing Machinery: New York, NY, USA, 2008; ICUIMC '08, p. 289–295. [CrossRef]
21. Achilleos, A.; Konstantinides, A.; Alexandrou, R.; Markides, C.; Zikouli, E.; Papadopoulos, G.A. A Web Platform and a Context Aware Recommender System for Active Sport Events. In Proceedings of the 21st International Conference on Innovations for Community Services (I4CS), Bamberg, Germany, 26–28 May 2021; Krieger, U.R., Eichler, G., Erfurth, C., Fahrnberger, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2021; pp. 183–197. [CrossRef]
22. Khajeheian, D.; Kolli, S. Digital games get viral on social media: A social network analysis of Pokémon Go on Twitter. *Int. J. Web Based Communities* **2020**, *16*, 262–278. [CrossRef]
23. Mallari, K.; Williams, S.; Hsieh, G. Understanding Analytics Needs of Video Game Streamers. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, Virtual, 8–13 May 2021; ACM: New York, NY, USA, 2021; CHI '21. [CrossRef]

24. Jo, H.; Shin, S.A. Investigating viewer engagement in esports through motivation and attitudes toward metaverse and NFTs. *Sci. Rep.* **2024**, *14*, 19934. [CrossRef]
25. Barney, J.; Pennington, N. An exploration of esports fan identity, engagement practices, and motives. *Telemat. Inform. Rep.* **2023**, *11*, 100081. [CrossRef]
26. Di Virgilio, F.; Soliman, M.; Anwar ul Haq, M.; Fantini, S. Analysing Users' Engagement with eSports Team: Does Covid-19 Matter? In *HR Analytics and Digital HR Practices: Digitalization Post COVID-19*; Mondal, S.R., Di Virgilio, F., Das, S., Eds.; Springer Nature: Singapore, 2022; pp. 243–271. [CrossRef]
27. Wang, X.; You, X.; Xu, Y.; Zheng, J. Online posting intention: Do the social communication and brand equity of esports matter? *Int. J. Sport. Mark. Spons.* **2024**, *25*, 161–187. [CrossRef]
28. Kozaki, Y.; Wang, Y.; Kawai, Y. Generating Pictorial Maps for Tourists using Flickr Photo Data. In Proceedings of the 2018 IEEE 7th Global Conference on Consumer Electronics, Nara, Japan, 9–12 October 2018; GCCE '18, pp. 403–407. [CrossRef]
29. Wang, Y.; Yue, W. Proposal and Evaluation of a Pictorial Map Generation Method based on Degrees of Popularity and Satisfaction for Tourist Spots using SNS Photos. *J. Jpn. Pers. Comput. Appl. Technol. Soc.* **2021**, *16*, 1–10. (In Japanese) [CrossRef]
30. Barman, N.; Zadtootaghaj, S.; Schmidt, S.; Martini, M.G.; Möller, S. An objective and subjective quality assessment study of passive gaming video streaming. *Int. J. Netw. Manag.* **2020**, *30*, e2054. [CrossRef]
31. Esports Charts. Everything About the Esports Popularity. Available online: https://escharts.com/ (accessed on 3 October 2024).
32. Statista. eSports Market Revenue Worldwide from 2020 to 2025. Available online: https://www.statista.com/statistics/490522/global-esports-market-revenue/ (accessed on 3 October 2024).
33. Ito, E.; Yabutani, R. Relationships between Specialization Factors and Sport and Video Game Experiences among E-sport Players. *J. Jpn. Soc. Sport. Ind.* **2021**, *31*, 4_457–4_464. (In Japanese) [CrossRef]
34. Jenny, S.E.; Keiper, M.C.; Taylor, B.J.; Williams, D.P.; Gawrysiak, J.; Manning, R.D.; Tutka, P.M. eSports Venues: A New Sport Business Opportunity. *J. Appl. Sport Manag.* **2018**, *10*, 34–49. [CrossRef]
35. Williams, D.; Yee, N.; Caplan, S.E. Who Plays, How Much, and Why? Debunking the Stereotypical Gamer Profile. *J. Comput.-Mediat. Commun.* **2008**, *13*, 993–1018. [CrossRef]
36. Himmelstein, D.; Liu, Y.; Shapiro, J.L. An Exploration of Mental Skills among Competitive League of Legend Players. *Int. J. Gaming Comput.-Mediat. Simul.* **2017**, *9*, 21. [CrossRef]
37. Chang, J.; Lee, D. Changes in user experience and satisfaction as media technology evolves: The reciprocal relationship between video games and video game-related media. *Technol. Forecast. Soc. Chang.* **2022**, *174*, 121219. [CrossRef]
38. Huang, Y.; Kim, D.; Ko, Y.J. The Effect of Livestreaming Esports Media on Viewer Satisfaction, Flow Experience, and Media Loyalty. *Commun. Sport* **2023**, *12*, 1–4. [CrossRef]
39. Przybylski, A.K.; Rigby, C.S.; Ryan, R.M. A Motivational Model of Video Game Engagement. *Rev. Gen. Psychol.* **2010**, *14*, 154–166. [CrossRef]
40. Bányai, F.; Griffiths, M.D.; Király, O.; Demetrovics, Z. The Psychology of Esports: A Systematic Literature Review. *J. Gambl. Stud.* **2019**, *35*, 351–365. [CrossRef]
41. @XDevelopers. Getting Started with the Twitter API v2 for Academic Research. Available online: https://github.com/xdevplatform/getting-started-with-the-twitter-api-v2-for-academic-research (accessed on 28 October 2022).
42. Twitter Inc. Twitter API (Version 2) Documentation. Available online: https://developer.twitter.com/en/docs/twitter-api (accessed on 15 October 2024).
43. Twitter Inc. Twitter Engagement API Documentation. Available online: https://developer.twitter.com/en/docs/metrics/get-tweet-engagement/overview (accessed on 15 October 2024).
44. Stringhini, G.; Wang, G.; Egele, M.; Kruegel, C.; Vigna, G.; Zheng, H.; Zhao, B.Y. Follow the Green: Growth and Dynamics in Twitter Follower Markets. In Proceedings of the 2013 Conference on Internet Measurement Conference, Barcelona, Spain, 23–25 October 2013; ACM: New York, NY, USA, 2013; IMC '13, pp. 163–176. [CrossRef]
45. Muñoz-Expósito, M.; Oviedo-García, M.Á.; Castellanos-Verdugo, M. How to measure engagement in Twitter: Advancing a metric. *Internet Res.* **2017**, *27*, 1122–1148. [CrossRef]
46. Newzoo International B.V. Game Performance Monitor. Available online: https://newzoo.com/game-performance-monitor (accessed on 28 December 2022).
47. Voorhees, E.M. The TREC-8 Question Answering Track Report. In Proceedings of the 8th Text REtrieval Conference. National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, 17–19 November 1999; Volume 99, TREC-8, pp. 77–82.
48. Baeza-Yates, R.; Ribeiro-Neto, B. *Modern Information Retrieval*; ACM Press Books; ACM Press: New York, NY, USA, 1999.
49. Wang, Y.; Wang, L.; Li, Y.; He, D.; Liu, T.Y. A Theoretical Analysis of NDCG Type Ranking Measures. In Proceedings of the 26th Annual Conference on Learning Theory, Princeton, NJ, USA, 12–14 June 2013; Shalev-Shwartz, S., Steinwart, I., Eds.; PMLR: Cambridge, MA, USA, 2013; Volume 30, pp. 25–54.

50. Friedman, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [CrossRef]
51. Woolson, R.F., Wilcoxon Signed-Rank Test. In *Wiley Encyclopedia of Clinical Trials*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2008; pp. 1–3. [CrossRef]
52. Kim, J.S. Within-subjects design. In *Encyclopedia of RESEARCH Design*; SAGE Publications, Inc.: Thousand Oaks, CA, USA, 2010; pp. 1639–1644. [CrossRef]
53. Wang, Y. User Satisfaction Survey. Available online: https://forms.gle/BAfxE3z2dP7sHMXn8 (accessed on 11 November 2024).
54. Lakens, D. Calculating and reporting effect sizes to facilitate cumulative science: A practical primer for t-tests and ANOVAs. *Front. Psychol.* **2013**, *4*, 863. [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

Hyperledger Fabric-Based Multi-Channel Structure for Data Exchange in Internet of Vehicles

Yiluo Liu [1,*], Yaokai Feng [2,*] and Kouichi Sakurai [2,*]

[1] Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan
[2] Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan
* Correspondence: liuyiluo16@gmail.com (Y.L.); fengyk@ait.kyushu-u.ac.jp (Y.F.); sakurai@inf.kyushu-u.ac.jp (K.S.)

Abstract: The rapid growth of the Internet of Vehicles (IoV) requires secure, efficient, and reliable data exchanges among multiple stakeholders. Traditional centralized database systems can hardly address the challenges associated with data privacy, integrity, and scalability in this decentralized ecosystem. In this paper, we propose a Hyperledger Fabric-Based Multi-Channel Structure to overcome these limitations. By leveraging the blockchain architecture, the system ensures data confidentiality and integrity by segregating data into exclusive channels and enabling different organizations to collaborate. Cross-channel communication ensures security when data are interacted with. Chaincodes automate transactions and enhance trust between participants. Our functional tests and performance tests by using Hyperledger Caliper verified the effectiveness of the system in real-world scenarios, highlighting its advantages over traditional systems in terms of decentralization, transparency, and security. Future work will focus on enhancing the user experience and integrating the system with edge computing. Eventually, attempts will be made to operationalize it in real-world environments.

Keywords: hyperledger fabric; multi-channel blockchain; Internet of Vehicles (IoV); decentralized systems; chaincode

1. Introduction

1.1. Internet of Vehicles (IoV)

IoV, one of the rapidly developing areas of the Internet of Things, is transforming the modern transportation landscape by enabling real-time communication between vehicles, infrastructure, and other entities such as manufacturers and service providers [1]. In IoV, vehicles equipped with sensors and communication systems can relay data about their location, speed, and even the occurrence of potential hazards, such as sudden braking or adverse weather conditions [2]. When this information is shared with other connected vehicles and traffic management systems, it can increase awareness among all the participants in the network, reducing the likelihood of accidents and making more informed decisions. Beyond safety, connected vehicles can revolutionize urban mobility by optimizing traffic flow, reducing congestion, and improving parking resource management [3].

However, IoV faces significant challenges related to the secure and efficient sharing of data between different stakeholders. The current centralized systems create data silos that lead to fragmentation and limited interoperability [4]. In addition, these centralized systems are prone to data breaches that compromise sensitive information such as vehicle status, driver behavior, and insurance records. The lack of a standardized, transparent, and

trustworthy platform further hinders collaboration among stakeholders and reduces the efficiency of operational processes [5]. Therefore, there is a need for an approach to build a secure, transparent, and interoperable data-sharing platform that fosters trust, ensures data privacy, and improves efficiency within the connected car ecosystem.

1.2. Blockchain

Blockchain technology was first introduced in 2008 to support the cryptocurrency Bitcoin and has since attracted significant interest due to its potential to revolutionize industries beyond finance [6]. In a blockchain network, data are stored in consecutive blocks that are linked together to form a chain. Each block contains a set of transactions or messages that are verified and encrypted by a network of distributed participants, often called nodes or miners. Once a block is added to the chain, it becomes immutable, ensuring that the recorded data cannot be modified or deleted, thus providing a tamper-proof and secure ledger [7]. Nowadays, blockchain technology has emerged as a revolutionary solution to the challenge of secure and transparent data sharing across industries. Its decentralized, immutable, and tamper-proof nature makes it particularly suitable for managing sensitive data and ensuring trust between untrusted parties. Blockchain enables seamless information sharing among multiple stakeholders, ensuring data integrity and traceability while protecting privacy [8].

1.3. Hyperledger Fabric (HLF)

HLF is an open-source blockchain framework developed by the Linux Foundation as a part of the Hyperledger Project specifically to meet the needs of enterprise-class blockchain applications [9]. Unlike public blockchains that rely on open participation and resource-intensive consensus mechanisms, HLF is designed to operate as a permissioned blockchain, where only authorized entities are allowed to join the network. This permissioned architecture makes it particularly suitable for environments such as connected car networks, where privacy and controlled participation are critical [10].

One of the main advantages of the HLF is its use of smart contracts, called "chaincode" in Hyperledger terminology. Chaincode defines and automates business logic, ensuring that transactions between participants are executed according to predefined rules [11]. Chaincode is executed in Docker containers, ensuring isolation and security for each instance. It runs on peer nodes and interacts with the ledger via the Fabric chaincode lifecycle. The Fabric chaincode lifecycle provides a well-defined process for managing chaincode, including packaging, installing, approving, and committing. This ensures that chaincode deployment is secure and that all the participating organizations reach a consensus before activation. Users or applications invoke chaincode functions to perform specific operations, such as querying data or submitting a transaction [9,12]. Chaincode is crucial for implementing the business logic for different channels in our multi-channel architecture. Each channel has its own chaincode to handle channel-specific data and policies, ensuring isolated and efficient data operations. This automated and secure contract management process simplifies complex workflows in connected vehicle networks, including the management of vehicle status, insurance policy claims, and parking facilities. This is particularly useful when participants need to establish trust in a decentralized network without relying on a central authority.

In addition, HLF's channel is a defining feature that sets it apart from other blockchain solutions. Channels in HLF can be thought of as private "sub-networks" within a larger blockchain network [13]. Channels enable organizations to share data securely without exposing it to the entire network. Only the members of a channel can access its ledger and participate in transactions. It means that access to a channel's data and chaincode is

restricted to its members. Transactions in one channel have no impact on other channels, ensuring operational independence [9]. In a network of connected vehicles, data generated by sensors, usage patterns, and maintenance logs may require different access rights for different stakeholders. HLF's endorsement policy adds additional security by allowing organizations to specify which members must approve a transaction before it becomes part of the ledger [10].

1.4. Motivation and Challenges

The rapid proliferation of Internet-connected vehicles and the emergence of the IoV have revolutionized the transportation system, enabling vehicles to share data such as traffic conditions, operational status, and environmental information in real time. However, the increasing reliance on data sharing poses significant challenges related to data privacy, security, trust, and scalability that traditional centralized systems have struggled to effectively address [14].

Blockchain technology, with its decentralization, offers promising solutions to these challenges [8]. In particular, HLF has strong scalability while ensuring efficient and secure transactions [10]. By leveraging this technology, our research aims to design a blockchain-based system that can handle the complex requirements of IoV, such as real-time data exchange, privacy protection, and scalability while building trust among different stakeholders, such as manufacturers, insurance companies, and traffic authorities.

Despite the potential of blockchain technology, there are still a number of challenges waiting to be resolved in order to build a robust and efficient system for IoV [15]:

- Data Security: IoV involves sensitive data such as vehicle location, operational status, and user identity. Ensuring that data are shared securely and only with authorized parties without compromising user privacy is a complex challenge.
- Performance: IoV generates large amounts of data in real time, requiring blockchain systems to handle high transaction throughput while keeping latency low. The design must address the trade-off between decentralization and performance.
- Trust Among Stakeholders: Building trust in a decentralized environment with different stakeholders such as manufacturers, insurance companies, and traffic authorities requires strong consensus mechanisms and governance models.
- Data Isolation: The isolation of specific data ensures the rights and interests of different organizations. Blockchain systems need to develop functionality that is tailored to the specific needs of the organization to ensure the proper management of specific data.

1.5. Our Contributions

In this paper, we propose a blockchain-based system architecture for IoV using HLF. Our contributions are summarized as follows:

- Multi-Channel Design: We introduce a multi-channel architecture in HLF specifically tailored for IoV. In the system, different types of data are exchanged across multiple channels involving various organizations. Each channel is dedicated to specific types of data such as vehicle status, insurance policies, and traffic conditions. This approach allows different stakeholders to collaborate efficiently while maintaining strict access control policies.
- Cross-Channel Communication using Chaincode: We develop and implement cross-channel communication using chaincodes, which is a key aspect of our multi-channel approach. This allows for the secure querying and exchanging of data across different channels. For instance, vehicle status data from one channel can be queried by a related insurance policy channel. Chaincode can enhance the system's capability and provide a comprehensive data management solution.

- Functional Testing: Self-defined node parameters ensure that organizations can choose the number of nodes according to their needs. Chaincode deployment and testing ensure stable information exchange between organizations. By implementing and testing our multi-channel system in a real blockchain environment, from deployment to functionality, we demonstrate its scalability, feasibility, and reliability.
- Performance Analysis: We conduct performance tests in different situations, even including high-pressure situations. We test the main functions of the proposed system. The results validate the system's ability to handle high transaction volumes while ensuring stable performance across diverse workloads. At the same time, it is also an important process for our system to approach the real environment.

2. Literature Review

Blockchain-based product traceability systems are receiving increasing attention from industry and academia. Ding et al. [16] proposed a permissioned blockchain-based double-layer framework for a product traceability system that addresses key challenges such as privacy protection, scalability, and regulatory compliance. In their framework, the consortium blockchain serves as the main layer, sharing product traceability information among organizations and allowing consumers to access the data, while a private blockchain is used for sensitive data storage within each enterprise. Unlike traditional single-layer systems, the double-layer framework separates data entry from data sharing. This separation enhances the scalability of the system and ensures better performance when handling a large number of transactions. In addition, the double-layer structure allows regulatory agencies to participate in the traceability process, which enriches traceable information and improves the reliability and credibility of the system.

Hussain et al. [17] proposed a blockchain-enabled secure communication framework aimed at enhancing trust and access control within the IoV ecosystem. The framework includes two core modules: a vehicle registration system and a message alert generation system, which use smart contracts to ensure the secure interaction of autonomous vehicles. A key aspect of their solution is the use of Ethereum's decentralized environment to manage interactions, ensuring that vehicle data integrity and reliability are maintained throughout the communication. The proposed framework introduces a decentralized design that leverages the proof-of-stake (PoS) consensus mechanism to handle transaction validation, which reduces computational overhead compared to proof of work (PoW). This makes the system more energy efficient and suitable for real-time vehicle-to-vehicle communication, an essential feature in IoV scenarios where efficiency and timely data exchanges are critical.

Ndzimakhwe et al. [18] present an innovative approach to solving privacy and data integrity issues within electronic health record (EHR) systems by employing HLF as the foundational technology. HLF allows for granular privacy control, ensuring that only authorized participants have access to particular datasets. This aligns with the privacy requirements critical in healthcare environments. Chaincode in the framework manages user permissions effectively. This contract-driven mechanism supports fine-grained access controls, enabling users to grant and revoke data access permissions dynamically. This system guarantees data immutability, meaning any changes to the health records are securely logged in a transparent yet privacy-preserving manner. The relevance of this study to connected vehicle networks lies in its demonstration of how HLF can manage sensitive data in a secure, distributed manner, with a high emphasis on privacy and selective data sharing.

Novo [19] explores a decentralized access control system leveraging blockchain technology to manage IoT devices. Traditional centralized access control systems are identified as limited in their scalability and adaptability, especially in dynamic IoT scenarios. The

proposed architecture addresses these limitations by eliminating reliance on central authorities. It can ensure mobility, scalability, and transparency in access control. This work highlights a blockchain-based system where IoT device access policies are enforced through a single smart contract. This reduces overhead and enhances policy management. Also, IoT devices remain independent of the blockchain network while managers enforce policies. This ensures scalability and cross-domain adaptability. By using lightweight mechanisms and a management hub, the system caters to constrained IoT devices while maintaining security and data integrity.

3. System Design

The proposed system focuses on secure and trustworthy data exchange between various stakeholders, such as vehicle manufacturers, insurance companies, service stations, and regulatory bodies. We leverage the features of HLF—particularly its channel architecture, consensus mechanisms, and chaincode—to provide the necessary security, privacy, and efficiency in vehicle data management.

3.1. System Architecture

As shown in Figure 1, our HLF-based system is composed of several components that work together to enable a decentralized and secure environment for data exchange.

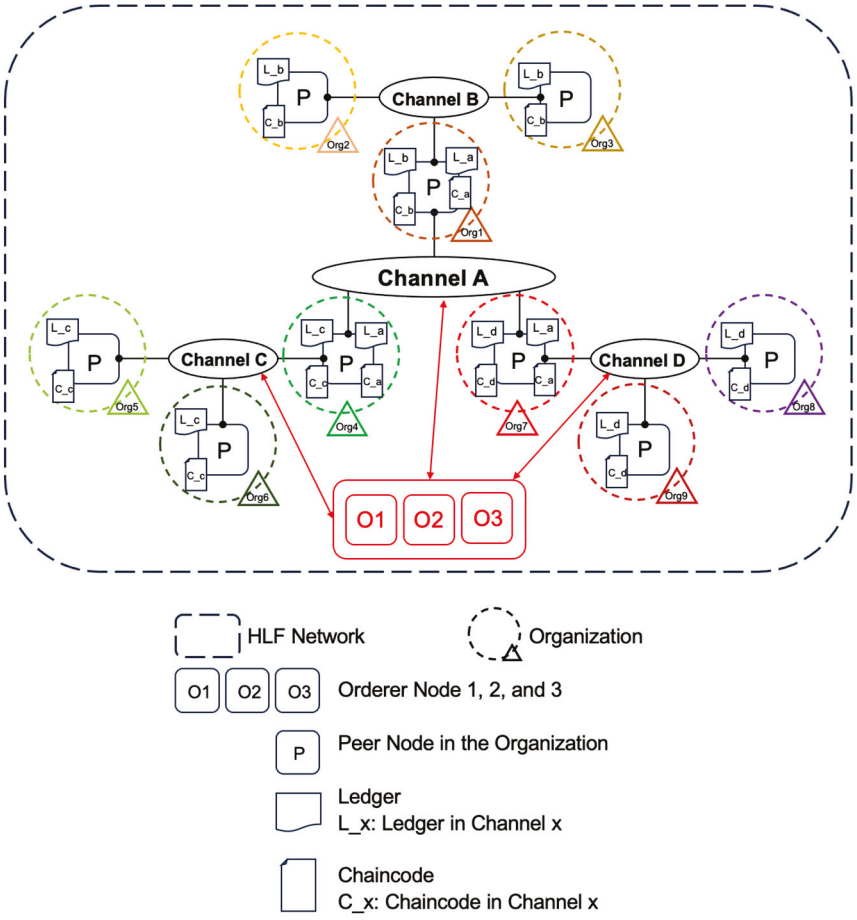


Figure 1. Structure of our proposal.

3.1.1. Organizations and Peer Nodes

In HLF, each stakeholder is represented by an organization [9]. Our system consists of nine organizations. Org1 represents the vehicle manufacturer and holds the running status of the vehicle. Org2 and Org3 are also manufacturers. But, they are below Org1. In other

words, Org2 and Org3 can also be branches of Org1. Org4 stands for a vehicle insurance company. It holds the insurance status of the vehicle. Org5 and Org6 are like Org2 and Org3; they are also branches of Org4. Finally, Org7 stands for the traffic management department. It has information on real-time road conditions and surrounding facilities, such as parking lots, etc. Org8 and Org9 belong to the subordinate departments of Org7. Each organization has its own peer node that is responsible for executing chaincode, endorsing transactions, and maintaining the distributed ledger. Each peer node is linked to its organization's membership service provider (MSP), which ensures proper identity management [10]. Multiple peer nodes can exist within each organization for information storage and sharing. For testing purposes, we set up only one node per organization.

3.1.2. Orderer Nodes

Our system employs a distributed ordering service consisting of three orderer nodes Orderer1, Orderer2, and Orderer3. We use the etcd Raft consensus protocol to ensure fault tolerance and maintain consistency across the network [20]. The orderers are responsible for collecting transactions from different organizations, ordering them, and then creating blocks to be distributed to peers. These orderer nodes are distributed among the stakeholders to avoid centralization, thereby providing resilience to failures and attacks.

3.1.3. Channels

One of the critical aspects of our system is the use of multi-channel architecture to achieve data isolation, privacy, and efficient communication. The channels allow different groups of organizations to share data without exposing it to unauthorized entities. There are four channels in our system.

Channel A serves as the central channel for the entire system. It connects the exchange of information between different organizations. It realizes the sharing of different kinds of information across organizations. For example, the traffic management department can call the insurance information of specific vehicles through Channel A. For security and authority reasons, only main organizations can join Channel A. Branches, such as Org2 and Org3, and subsidiaries cannot join.

Channel B includes Org1, Org2, and Org3. They share information about the real-time status of the vehicle, such as position, speed, etc., within the channel. Figure 2 shows the structure of the real-time state of the vehicle that we set up during the testing phase.

```
type VehicleStatus struct {
    VehicleID string `json:"vehicleID"`
    Location  string `json:"location"`
    Speed     int    `json:"speed"`
    FuelLevel int    `json:"fuelLevel"`
    Battery   int    `json:"battery"`
    Timestamp string `json:"timestamp"`
}
```

Figure 2. Data structure of vehicle status.

Channel C includes Org4, Org5, and Org6. It serves as a conduit between insurance companies, primarily for the sharing of vehicle insurance information. Figure 3 shows the structure of the insurance information.

Channel D includes Org7, Org8, and Org9. It serve as a conduit between the traffic management department and subordinate departments to share information such as real-time road conditions and surrounding facilities. In our testing, we set up real-time information on parking lots. Figure 4 shows the structure.

```
type InsurancePolicy struct {
    PolicyID      string  `json:"policyID"`
    VehicleID     string  `json:"vehicleID"`
    Owner         string  `json:"owner"`
    Provider      string  `json:"provider"`
    CoverageType  string  `json:"coverageType"`
    StartDate     string  `json:"startDate"`
    EndDate       string  `json:"endDate"`
    PremiumAmount float64 `json:"premiumAmount"`
    Status        string  `json:"status"`
}
```

Figure 3. Data structure of insurance.

```
type ParkingFacility struct {
    FacilityID     string  `json:"facilityID"`
    Location       string  `json:"location"`
    TotalSpots     int     `json:"totalSpots"`
    AvailableSpots int     `json:"availableSpots"`
    PricePerHour   float64 `json:"pricePerHour"`
}

type TrafficCondition struct {
    Location      string `json:"location"`
    ConditionType string `json:"conditionType"` // e.g., Congestion, Accident, Roadwork
    Severity      string `json:"severity"`      // e.g., Low, Medium, High
    Timestamp     string `json:"timestamp"`
    Description   string `json:"description"`
}
```

Figure 4. Data structure of parking lot and traffic condition.

3.1.4. Certificate Authority (CA)

CA plays a crucial role in establishing identity management, authentication, and authorization. CA is a fundamental component for managing identities and securing communication in the HLF network. Each organization has its own CA to issue digital certificates and keys, ensuring secure identity management and mutual authentication among the nodes.

The CA issues digital certificates to peers, orderers, and clients, which serve as cryptographic credentials for verifying identities. The CA issues Transport Layer Security (TLS) certificates to peers, orderers, and clients, enabling encrypted communication between network components. This ensures that data transmitted over the network remains secure and prevents unauthorized access. Also, the CA can revoke certificates when necessary, such as when a participant's credentials are compromised. This ensures that revoked participants cannot access or interact with the network.

Here is the workflow of CA. Participants are first registered with the CA, which assigns them identity attributes. During enrollment, the CA generates a key pair and issues a certificate to the participant. HLF's MSP uses CA-issued certificates to manage identities within the blockchain network. MSP ensures that only valid identities interact with the network. TLS certificates secure communication between peers, orderers, and clients. Additionally, multi-channel architecture ensures that only authorized members access specific channel data.

3.1.5. Docker and CouchDB

Our system is deployed using Docker for container orchestration. Docker provides a scalable and manageable environment for a distributed blockchain network. All peers,

orderers, and CAs are deployed as Docker containers. This ensures easy deployment, isolation, and scalability. It allows each organization to manage its resources independently.

Meanwhile, each peer node in the network maintains a ledger and uses CouchDB as the state database. The integration of CouchDB with peer nodes facilitates the efficient querying of vehicle and service-related data.

3.2. Consensus

The consensus protocol used in this system is the etcd Raft consensus protocol. It is a key feature of HLF to ensure the ordering of transactions in a secure and efficient manner [20]. This protocol is particularly well suited for enterprise blockchain applications where resilience and stability are critical.

In the system, etcd Raft achieves consensus through a leader election process. One node is elected as the leader, and it coordinates all the transactions among the other nodes, which act as followers. This leader-based mechanism ensures that the transactions are processed in a sequential manner, minimizing conflicts and ensuring data integrity. The leader is responsible for proposing transaction blocks. Once a majority of the nodes (the followers) agree on the proposed block, it is committed to the blockchain [13].

In our system setup, three orderer nodes are deployed as part of the etcd Raft consensus group. Each orderer node runs an instance of the etcd Raft protocol to replicate data and to ensure that all the nodes have the same view of the ledger. This setup enhances the fault tolerance of the network. It allows the system to continue operating even if one or two of the orderer nodes fail.

3.3. Chaincode

In our system, the chaincode is written in the Go language. The lifecycle of chaincode deployment follows several steps. It begins with packaging and installation. Then, each organization will approve it. Finally, the chaincode will be committed to the channel. This process ensures that the defined logic is agreed upon and trusted by all the participants [9]. Figure 5 shows the structure of chaincodes in our system. Basically, as the central channel, Channel A operates the UnifiedQuery() function to query data from Channels B, C, and D. Meanwhile, Channels B, C, and D operate the Create(), Query(), Update(), and Delete() functions to process the data.

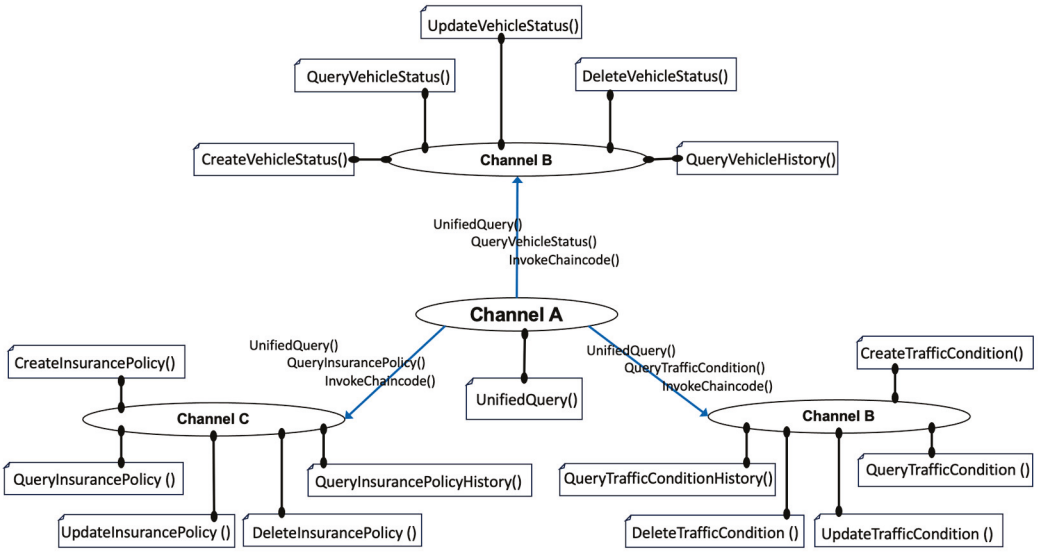


Figure 5. Chaincode structure.

3.3.1. Channel A

In Channel A, the chaincode is designed to facilitate cross-channel queries that enable Channel A to interact with information present in Channels B, C, and D. Before querying, the client's identity and permissions are verified using the MSP of the respective organization. TLS encryption ensures that the communication between the client and the peer nodes is secure. The system leverages HLF's inter-channel capabilities, called InvokeChaincode API, to fetch information about vehicles, insurance policies, traffic conditions, and parking facilities from different channels. The request is forwarded to the chaincode associated with Channel A. Then, the chaincode processes the request and interacts with CouchDB to fetch the required data. The chaincode retrieves the queried data and returns it in a structured format, such as JSON, to the client.

The chaincode includes functions for querying vehicle status and history from Channel B, querying insurance policy information from Channel C, and querying traffic conditions and parking facility data from Channel D. The chaincode is shown in Algorithm 1.

Algorithm 1: Chaincode in Channel A

Input: Transaction Context (ctx), various identifiers (vehicleID, policyID, facilityID, location, localTime)
Output: String representing the result of the cross-channel query or error message

```
function QueryVehicleStatus(ctx, vehicleID)
    args = [ChannelBQuery, vehicleID]
    response = ctx.InvokeChaincode(ChainCodeBName, args, ChannelB)
    if (response is invalid) then
        return response.Message
    end if
    return response.Payload
end function

function UnifiedQuery(ctx, channel, action, id, localTime)
    clientMSP = ctx.GetClientIdentity().GetMSPID()
    if (call channel B but MSP is not Org1MSP) then
        return "Only Org1 can query Channel B"
    else if (call channel C but MSP is not Org4MSP) then
        return "Only Org4 can query Channel C"
    else if (call channel D but MSP is not Org7MSP) then
        return "Only Org7 can query Channel D"
    end if
    switch channel:
        case ChannelB:
            if (action is VehicleStatus) then
                return QueryVehicleStatus(ctx, id)
            else if (action is VehicleStatusHistory) then
                return QueryVehicleStatusHistory(ctx, id)
            end if
```

Algorithm 1: *Cont.*

```
        case ChannelC:
            if (action is InsurancePolicy) then
                return QueryInsurancePolicy(ctx, id)
            else if (action is InsurancePolicyHistory) then
                return QueryInsurancePolicyHistory(ctx, id)
            end if
        case ChannelD:
            if (action is TrafficCondition) then
                return QueryTrafficCondition(ctx, id, localTime)
            else if (action is ParkingFacility) then
                return QueryParkingFacility(ctx, id)
            end if
    end switch
    return error
end function
```

The chaincode related to the query is used to retrieve information from Channels B, C, and D. Their structure is similar, thus we only show QueryVehicleStatus() in Algorithm 1. It accepts vehicleID as input and then invokes the chaincode on Channel B to obtain the data. The output is the current status of the vehicle, such as location. Also, The UnifiedQuery() function acts as a generic entry point that allows users to access multiple query types through a single function call. Based on the input parameters, this function will dynamically decide which specific query function to call. This simplifies user interaction with the system and provides a streamlined method to access multiple datasets across different channels.

3.3.2. Channels B, C, and D

The chaincode in channels B, C, and D offers a set of functions to perform CRUD (Create, Read, Update, and Delete) operations and historical queries. Algorithm 2 shows the chaincode in Channel B. For additions, the implementation of CRUD in Channel C and D is the same as in B. Therefore, we only show the algorithm in Channel B.

Algorithm 2: Chaincode in Channel B

```
Input: VehicleID, Location, Speed, FuelLevel, Battery, Timestamp
Output:
- Confirmation of success or failure
- Requested Vehicle Status or History

function CreateVehicleStatus(vehicleID, location, speed, fuelLevel, battery, timestamp)
    if (this vehicle ID already exists) then
        return "Error: Vehicle already exists"
    end if
    create vehicleStatus
    serialize vehicleStatus to JSON
    store JSON in Ledger[vehicleID]
end function
```

Algorithm 2: *Cont.*

```
function QueryVehicleStatus(vehicleID)
    fetch data from Ledger[vehicleID]
    if (data is NULL) then
        return "Error: Vehicle not found"
    end if
    deserialize data to VehicleStatus
    return VehicleStatus
end function

function UpdateVehicleStatus(vehicleID, location, speed, fuelLevel, battery, timestamp)
    if (vehicle does not exist) then
        return "Error: Vehicle not found"
    end if
    update status
    serialize vehicleStatus to JSON
    store JSON in Ledger[vehicleID]
    return "Vehicle status updated successfully"
end function

function DeleteVehicleStatus(vehicleID)
    if (vehicle does not exist) then
        return "Error: Vehicle not found"
    end if
    delete Ledger[vehicleID]
    return "Vehicle status deleted successfully"
end function

function VehicleStatusExists(vehicleID)
    fetch data from Ledger[vehicleID]
    return data is not NULL
end function

function QueryVehicleHistory(vehicleID)
    fetch history from Ledger[vehicleID]
    if (History is empty) then
        return "No history available"
    end if
    deserialize history to List[VehicleStatus]
    return List[VehicleStatus]
end function
```

These functions ensure the complete lifecycle management of the vehicle status data. CreateVehicleStatus() adds a new vehicle status record to the ledger and validates that the VehicleID does not already exist. Finally, it encodes the data as JSON and writes it to the ledger. QueryVehicleStatus() retrieves the latest status for a given vehicle. It can decode JSON data. UpdateVehicleStatus() updates an existing vehicle record. It can modify the fields and write the updated JSON back to the ledger. DeleteVehicleStatus() deletes a vehicle status record. It can verify the record exists before removing it from the ledger. Chaincodes in Channel C and D have similar structures.

3.4. Use Cases

The proposed system involves diverse stakeholders, such as vehicle manufacturers, insurance companies, and traffic management departments. They need to exchange data in a secure, transparent, and efficient way. The multi-channel blockchain system is designed to address these challenges by enabling isolated yet interconnected communication between these entities. To illustrate the practicality and effectiveness of the proposed architecture, we present two representative use cases: accident management and Parking Assistance. These scenarios demonstrate how the multi-channel design optimizes data flow. The detailed workflows below highlight the advantages of the system and its potential for real-world applications.

3.4.1. Accident Management

Accident management is a critical aspect of IoV, requiring collaboration between vehicle manufacturers, insurance companies, and traffic management departments. Accident data are shared only with relevant parties through their respective channels.

Accident Detection: A vehicle detects an accident using onboard sensors, such as collision detection or sudden deceleration, and records the event. The vehicle's onboard system sends the traffic details through facilities with communication capabilities, such as Roadside Units, to relevant nodes in Channel D, the traffic management department's channel, and the vehicle's status to nodes in Channel B, the manufacturers' channel.

Data Sharing Across Channels: The vehicle manufacturer verifies the accident report and updates the vehicle's status on Channel B. Meanwhile, the traffic management also verifies the report. At this time, the vehicle's status is shared with the traffic management department through Channel A.

Traffic and Emergency Management: Traffic management will take appropriate actions based on the vehicle's status obtained from Channel A and the traffic details in Channel D.

Insurance Processing: Insurance companies can query relevant data through Channel C to verify the accident situation to implement the claims process. Using immutable blockchain records, claims processes can reduce the risk of fraud.

Post-Accident Actions: Once the accident is resolved, all the records can be stored in the blockchain. Immutable accident records enhance trust among all the parties.

3.4.2. Parking Assistance

Efficient parking management is another vital use case for IoV, involving real-time data sharing between vehicles and traffic management departments.

Parking Availability: The traffic management department collects vehicle status and location information through Channel A from vehicle manufacturers and basic parking information in Channel D to generate real-time parking information, including location, total spots, available spots, and pricing.

Parking Request: Traffic management departments can send real-time parking information to drivers through facilities with communication capabilities, such as Roadside Units. Drivers can request this kind of information by using the onboard system.

Parking Information Update: When a vehicle decides on a parking location, the real-time status of the vehicle is uploaded into the system. The traffic management department can update the parking information at any time by obtaining the vehicle status information through Channel A. When the vehicle leaves the parking location, the parking information can also be updated.

Optional Payment Method: The insurance company can also query vehicles' parking records through Channel A while the vehicle decides the parking location. The driver can decide to include the parking bill in the insurance bill and pay it with the insurance.

4. Deployment and Testing

4.1. Deployment

Before testing our system, the deployment of the HLF network and chaincode is an important step.

4.1.1. Deployment of HLF Network

In this section, we outline the step-by-step process for deploying a HLF network to support our proposed multi-channel system. The deployment involves generating cryptographic materials, configuring the network structure, creating channels, and connecting peers from different organizations to their respective channels. Below, we detail each step of the deployment process.

Step 1: We use the *cryptogen* tool to create cryptographic certificates for all the network entities [9]. These materials are critical for creating secure communication and identity verification. By using this tool, we can create MSP directories and TLS certificates for peers and orderers.

Step 2: We use the *configtxgen* tool to define the structure and policies of the network. In total, we create the genesis block, which defines the orderer and consortium configurations, and configuration files for application channels and for updating the anchor peers of each organization in each channel.

Step 3: We submit the channel configuration transactions to the orderer to create channels. In this step, the *peer channel create* command is used for creating each channel.

Step 4: It is necessary to connect peers from various organizations to their respective channels. We need to set the peer's MSP, TLS root certificate, and address and use the *peer channel join* command to associate each peer with the appropriate channel block.

Step 5: The *peer channel update* command is used to update transactions to designate specific peers as anchor peers.

4.1.2. Deployment of Chaincode

This section describes the process of deploying and activating chaincode on the HLF network. Chaincode, which implements the business logic of the system, must be packaged, installed, approved, and committed across all the relevant organizations within their respective channels. This ensures consensus on the chaincode's configuration and enables its functions to be executed securely on the blockchain. The following steps outline the detailed deployment process.

Step 1: We need to package the chaincode into a format suitable for distribution across the network [9]. The chaincode is developed using Go in our system. The *peer lifecycle chaincode package* command is used to package the chaincode into the *.tar.gz* file. This file contains all the chaincode source code and metadata.

Step 2: The packaged chaincode is installed on all the peer nodes of the participating organizations in the respective channel. For example, in Channel D, the packaged chaincode is only installed in Org7, 8, and 9. Therefore, environment variables specific to each organization are set to identify the target peer for installation. The *peer lifecycle chaincode install* command is executed to install the chaincode package on the peer node. Once installed, the chaincode's unique ID is queried and stored for further approval steps.

Step 3: To ensure that all the organizations agree on the chaincode's parameters, each organization must approve the chaincode definition. The *peer lifecycle chaincode approveformyorg* command is used in this step.

Step 4: Once all the organizations have approved, the chaincode is committed to the channel, making it active for use. In this step, the *peer lifecycle chaincode commit* command is used and the *peer lifecycle chaincode querycommitted* command validates the chaincode deployment.

Step 5: Finally, we can test the chaincode by invoking its functions and querying data.

4.2. Testing

In this section, we will show the results of the process of deployment and functional test.

4.2.1. Channel Creation

As shown in Figure 6, the creation of Channel A begins with initializing the endorser and orderer connections. This confirms the communication setup between peers and the orderer. Also, initial attempts to retrieve the channel genesis block resulted in various statuses. NOT_FOUND suggests the block was not yet created and SERVICE_UNAVAILABLE indicates the temporary unavailability of the orderer service. After a series of retries, the genesis block for each channel was successfully retrieved. In a nutshell, the retry mechanism can ensure resilience against transient failures and the genesis block shows successful channel creation.

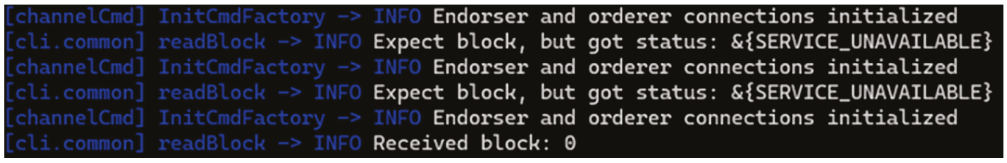


Figure 6. Channel creation test.

4.2.2. Peer Joining Test

As shown in Figure 7, each peer successfully submitted a proposal to join the channel. The successful join operation confirms that the channel block is valid and the peers are configured correctly to participate in the channel. These results confirm the proper configuration of MSPs, certificates, and block files.

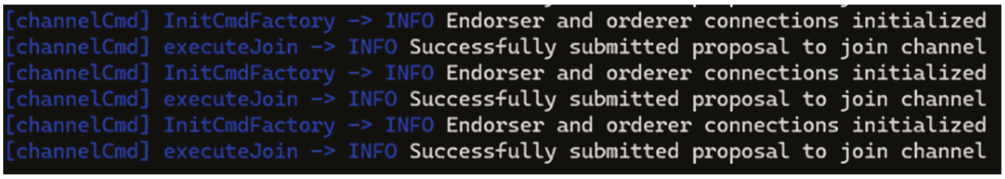


Figure 7. Peer joining test.

4.2.3. Anchor Peer Update Test

Anchor peers for each organization were updated to enable inter-organization communication. As shown in Figure 8, updates to anchor peer configurations were successfully submitted. All the anchor peer updates were processed without errors, indicating robust coordination among the peers, orderer, and the configuration update mechanism.

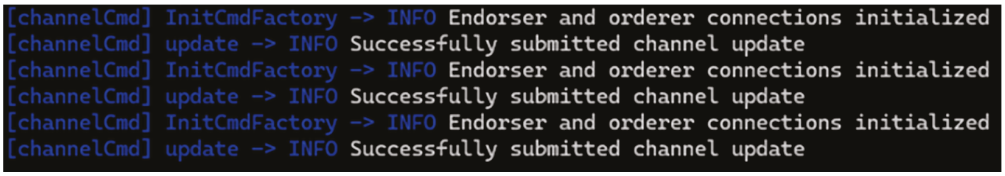


Figure 8. Anchor peer update test.

4.2.4. Query Channel Information Test

It is necessary to query information about the four channels to verify their proper configuration and deployment. The goal is to ensure that all the participating organizations have successfully joined their respective channels and that the ledger state is synchronized across peers. As shown in Figure 9, as we expected, Org1, 4, and 7 joined into Channel

A properly. The consistent ledger height validates proper transaction propagation and block generation across the channel. Also, "_lifecycle" shows that the chaincode lifecycle management process is operational. Due to the excessive length of the image, Figure 9 only shows the information for Org4, omitting the information for Org1 and Org7. The information for Org1 and Org7 is the same as for Org4, except for Endpoint and Identity.

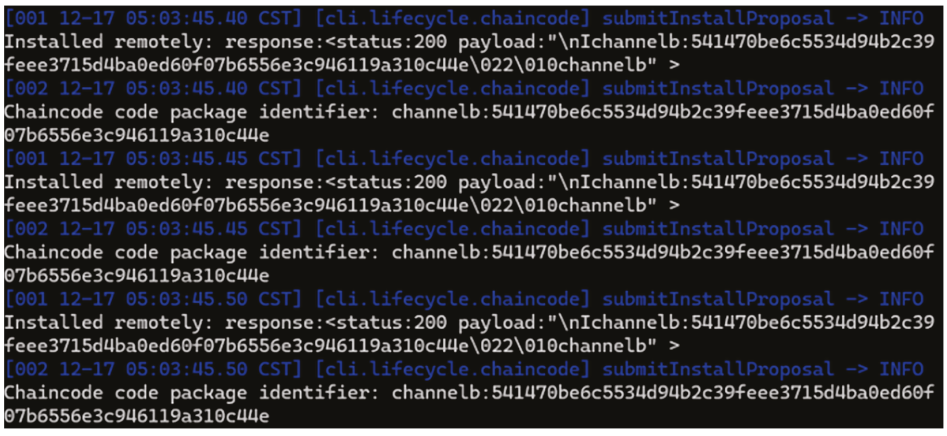


```
#################################################################
#####################   Query channela info   ###################
#################################################################
[
        {
                "MSPID": "Org4MSP",
                "LedgerHeight": 4,
                "Endpoint": "peer0.org4.example.com:10051",
                "Identity": "-----BEGIN CERTIFICATE-----\nMIICKTCC
Ac+gAwIBAgIRAI3SVPeNiBQFFavy7iuG0ocwCgYIKoZIzj0EAwIwczEL\nMAkGA1UE
BhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExFjAUBgNVBAcTDVNhbiBG\ncmFuY2lz
Y28xGTAXBgNVBAoTEG9yZzQuZXhhbXBsZS5jb20xHDAaBgNVBAMTE2Nh\nLm9yZzQu
ZXhhbXBsZS5jb20wHhcNMjQxMjE2MjA0MzAwWhcNMzQxMjE0MjA0MzAw\nWjBqMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEWMBQGA1UEBxMN\nU2FuIEZy
YW5jaXNjbzENMAsGA1UECxMEcGVlcjEfMB0GA1UEAxMWcGVlcjAub3Jn\nNC5leGFt
cGxlLmNvbTBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABJtX+8Y0FdDR\nnoSCMNH9
uFSnkeBEuQbDqm80lDNyiWb1P6YsVaSXYmQH4BgeqzG612/qGZGNYGqj\nC2pBhu19
m6CjTTBLMA4GA1UdDwEB/wQEAwIHgDAMBgNVHRMBAf8EAjAAMCsGA1Ud\nIwQkMCKA
IMBAjlsBiMEiftekTHgfDvHKJjVF+NYaVzFHnN1kUJzBMAoGCCqGSM49\nBAMCA0gA
MEUCIQCxgOaELB6NmKOo2dvcEDJA14irw7mDEBDaSX/doaIPnQIgGjQc\nwbfvLDNg
CvuMng/vu+PMB3N4fWPqNSfzcXaCgQo=\n-----END CERTIFICATE-----\n",
                "Chaincodes": [
                        "_lifecycle"
                ]
        },
```


Figure 9. Query channel information test.

4.2.5. Deployment of Chaincode Test

In this part, we tested the deployment, approval, and functionality of the chaincodes across the respective channels. As shown in Figure 10, the chaincode installation succeeded across all the peers in Channel B. Also, the generated package identifier is the same across all the installations. It confirms successful packaging and propagation.



```
[001 12-17 05:03:45.40 CST] [cli.lifecycle.chaincode] submitInstallProposal -> INFO
Installed remotely: response:<status:200 payload:"\nIchannelb:541470be6c5534d94b2c39
feee3715d4ba0ed60f07b6556e3c946119a310c44e\022\010channelb" >
[002 12-17 05:03:45.40 CST] [cli.lifecycle.chaincode] submitInstallProposal -> INFO
Chaincode code package identifier: channelb:541470be6c5534d94b2c39feee3715d4ba0ed60f
07b6556e3c946119a310c44e
[001 12-17 05:03:45.45 CST] [cli.lifecycle.chaincode] submitInstallProposal -> INFO
Installed remotely: response:<status:200 payload:"\nIchannelb:541470be6c5534d94b2c39
feee3715d4ba0ed60f07b6556e3c946119a310c44e\022\010channelb" >
[002 12-17 05:03:45.45 CST] [cli.lifecycle.chaincode] submitInstallProposal -> INFO
Chaincode code package identifier: channelb:541470be6c5534d94b2c39feee3715d4ba0ed60f
07b6556e3c946119a310c44e
[001 12-17 05:03:45.50 CST] [cli.lifecycle.chaincode] submitInstallProposal -> INFO
Installed remotely: response:<status:200 payload:"\nIchannelb:541470be6c5534d94b2c39
feee3715d4ba0ed60f07b6556e3c946119a310c44e\022\010channelb" >
[002 12-17 05:03:45.50 CST] [cli.lifecycle.chaincode] submitInstallProposal -> INFO
Chaincode code package identifier: channelb:541470be6c5534d94b2c39feee3715d4ba0ed60f
07b6556e3c946119a310c44e
```


Figure 10. Chaincode installation.

Then, as shown in Figure 11, all the participating organizations, Org1, 2, and 3, approved the chaincode definition for Channel B. It means endorsement policies and chaincode parameters were correctly validated and accepted.

```
#################################################################
###########  Org1MSP approve chaincode for channelb   ###########
#################################################################
[001 12-17 05:03:50.79 CST] [chaincodeCmd] ClientWait -> INFO txid
 [c95328fcc107d767c133275d33b978e1a0ee0cd9806cffb89de9943e00b8334c
] committed with status (VALID) at peer0.org1.example.com:7051
#################################################################
#########  Org2MSP approve chaincode for channelb   ############
#################################################################
[001 12-17 05:03:56.05 CST] [chaincodeCmd] ClientWait -> INFO txid
 [af0a914d4ddc28bb506968a1c39c4c0d62b09d63ee6781e2911a5c8f71b863ae
] committed with status (VALID) at peer0.org2.example.com:8051
#################################################################
########  Org3MSP approve chaincode for channelb   ############
#################################################################
[001 12-17 05:04:01.24 CST] [chaincodeCmd] ClientWait -> INFO txid
 [90fd543aafa3a307246f4def68179ed8272de3b24aae74c03d2c4b8d52dfcd31
] committed with status (VALID) at peer0.org3.example.com:9051
#################################################################
############# channelb approve org query  ######################
#################################################################
{
        "approvals": {
                "Org1MSP": true,
                "Org2MSP": true,
                "Org3MSP": true
        }
}
```

Figure 11. Chaincode approval.

After approval, as shown in Figure 12, the chaincode was committed successfully on all the peers in Channel B. The committed version 1.0 and sequence 1 confirm the initial deployment. Finally, it is a functional test.

```
#################################################################
########## Submit the chaincodeB to the channelb   ############
#################################################################
[001 12-17 05:04:09.64 CST] [chaincodeCmd] ClientWait -> INFO txid
[f975a0a964b5e3e31c718c3e405d2fc23a2b043467474208df9a4e7f250073d7]
committed with status (VALID) at peer0.org1.example.com:7051
[002 12-17 05:04:09.64 CST] [chaincodeCmd] ClientWait -> INFO txid
[f975a0a964b5e3e31c718c3e405d2fc23a2b043467474208df9a4e7f250073d7]
committed with status (VALID) at peer0.org3.example.com:9051
[003 12-17 05:04:09.64 CST] [chaincodeCmd] ClientWait -> INFO txid
[f975a0a964b5e3e31c718c3e405d2fc23a2b043467474208df9a4e7f250073d7]
committed with status (VALID) at peer0.org2.example.com:8051
#################################################################
############# query the chaincodeB commit   ####################
#################################################################
Committed chaincode definition for chaincode 'channelb_ledger' on c
hannel 'channelb':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plu
gin: vscc, Approvals: [Org1MSP: true, Org2MSP: true, Org3MSP: true]
```

Figure 12. Chaincode commitment.

As shown in Figure 13, in Channel B, we operated the CreateVehicleStatus() function to create statues in advance and the result shows that chaincode executed the CreateVehicleStatus() function successfully, indicating proper function implementation. Then, by operating the QueryVehicleStatus() function, vehicle status was retrieved successfully with expected values. Next, by operating the UpdateVehicleStatus() function, the vehicle status was updated correctly and the updated values were reflected in the subsequent queries. In

the end, the history query correctly retrieved all the changes to the vehicle status. The basic function of Channels B, C, and D is the same, so only the result of Channel B is shown here.

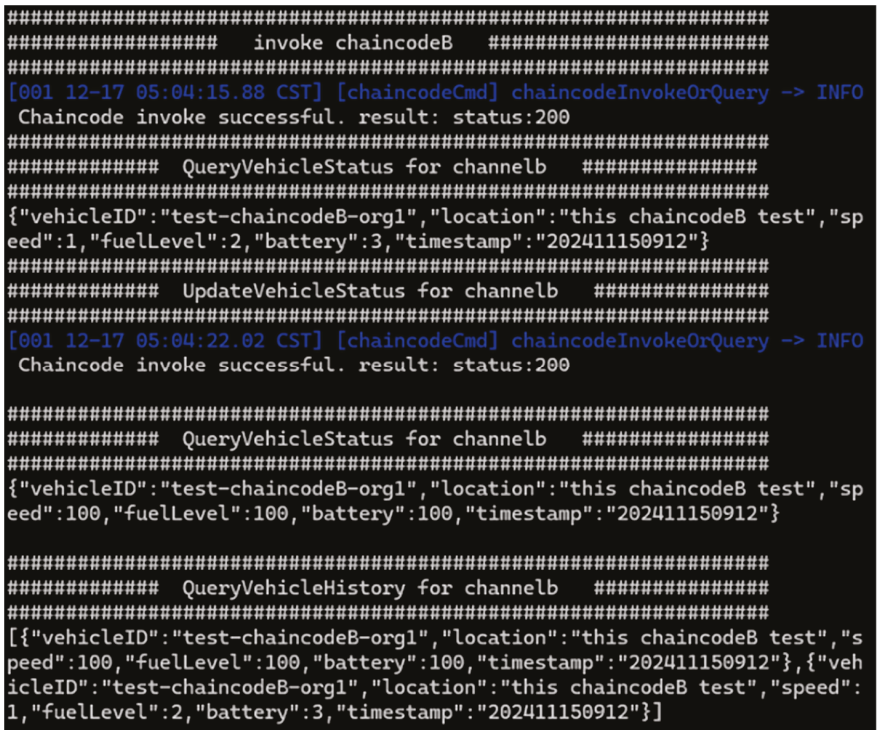


Figure 13. Functional test in Channel B.

For Channel A, which is the central channel of the system, we did a unified query test across channels. As shown in Figure 14, we can see that the unified queries successfully retrieved data across Channels B, C, and D. For Channel B, the query retrieves vehicle operational data. For Channel C, the query fetches insurance policy details. And, for Channel D, the unified query retrieves two types of data, traffic condition data, and parking facility data. Each query is limited to its respective channel, demonstrating the privacy and isolation features of the system. This figure serves as evidence of the system's ability to segregate and query channel-specific data efficiently. It also emphasizes the potential of blockchain-based multi-channel architecture in managing diverse datasets.



Figure 14. Functional test in Channel A.

5. Performance Analysis

Performance analysis is a critical aspect of evaluating the practical feasibility and efficiency of the proposed multi-channel system. In this section, we show the setup of the test environment and results.

5.1. Test Environment

In this work, we use Hyperledger Caliper (Caliper) to monitor the network performance. Caliper is a performance benchmarking tool specifically designed for blockchain platforms, including HLF. It provides a standardized way to evaluate and measure the performance of blockchain networks under various configurations and workloads [21]. The proposed system and performance test are implemented on a local personal computer (PC) which has 10 CPU cores: Intel(R) Core(TM) i9-10900K 3.70 GHz, Ram 16.0 GB. PC runs the Linux distribution Ubuntu 22.04 by using Windows Subsystem for Linux (WSL).

5.2. Performance Measurement

We focus on two indicators to measure the performance of the system. These indicators include the number of transactions per second (TPS), known as throughput, and the latency, which means the overall time the transactions take from issuing to the response.

5.3. Benchmark Setup

Before testing the performance, we need to set up the workspace which is related to benchmark files. The workspace includes three parts: networks, workload, and benchmarks. The network includes a network configuration, which contains compulsory information about the work, such as channels, organizations, etc. The workload is the key part of Caliper. It defines the way to interact with chaincodes. The benchmark is a configuration file for the execution of Caliper. It defines some parameters during the process of the performance test.

To achieve different results during the performance test, our benchmark configuration includes the following parameters:

- Worker Numbers: Worker represents the number of threads. More workers represent more concurrent transaction requests. However, more workers will send more transactions, which may cause transactions to be cached or queued, increasing the load on the system. We set three different values: 5, 25, and 50.
- Number of Transactions (txNumber): It defines the total number of transactions executed during the test period. A larger number allows testing the stability of the system over a long period of time. We set three different values: 1000, 2000, and 5000.
- Transactions Per Seconds (transactionLoad): It defines the number of transactions sent per second. In order to test the latency and maximum throughput in different scenarios, we set three different values: 50, 100, and 200.
- High-Pressure Test: We additionally test the system's performance in a high-pressure environment. We set the number of workers to 50 and the TPS to 400. In particular, we did not use the total number of transactions. We set the total transaction duration (txDuration) to 300 s. The purpose is to perform continuous testing over a long period of time.

5.4. Results and Evaluation

We mainly tested the CRUD function in the system. The CRUD logic in Channels B, C, and D is the same, so we summarize the results in five parts, Create(), Query(), Update(), Delete(), and high-pressure test.

5.4.1. Create()

First, to test the impact of the total number of transactions and TPS on the performance of the system, we fix the number of workers to the default of 5. Then, we change the values of the total number of transactions and TPS, respectively, for testing. The result is shown in Table 1. As the TPS increases (from 50 to 200), both the send rate and throughput improve significantly. At higher TPS (e.g., 200), the throughput approaches the send rate, indicating that the system is efficiently processing transactions and has not yet reached a performance bottleneck. Latency is relatively high for lower TPS (around 2 ms for TPS = 50 and 100). At higher TPS, latency drops significantly (to ~0.6 ms). The reason may be that at higher transaction rates, batching and block generation may optimize the overall performance, reducing per-transaction latency. Increasing txNumber (from 1000 to 5000) while keeping TPS fixed does not significantly affect the send rate or throughput. This suggests that the system maintains consistent performance over larger transaction loads.

Table 1. Performance test of Create() with the fixed number of workers.

Create(): Fixed N. of Workers, Different txNumber and TPS					
N. of Workers	**txNumber**	**TPS**	**Send Rate**	**Throughput**	**Avg Latency**
5	1000	50	11.5	11.2	2.07
		100	20.2	19.3	2.01
		200	111.4	90.5	0.59
5	2000	50	12.5	12.4	1.66
		100	22.7	22.2	1.79
		200	129.3	125.6	0.65
5	5000	50	12.3	12.2	1.75
		100	27.1	26.8	1.71
		200	154.8	147.2	0.56

Then, to test the impact of the number of workers, we fix the number of total transactions and TPS during the test process. We set 2000 transactions and 100 TPS as the fixed parameters. The result is shown in Table 2. Increasing the number of workers leads to a higher send rate and throughput. The reason is that additional workers generate more concurrent transactions, effectively utilizing the network's capacity for parallel processing. Latency increases slightly from 1.79 ms (5 workers) to 2.05 ms (25 workers) but decreases slightly at 50 workers (1.93 ms). The reason may be that at higher concurrency, peer and orderer nodes process transactions more efficiently due to better batching and resource utilization. With 50 workers, the system achieves its highest throughput while maintaining acceptable latency, indicating that the resources are not yet saturated.

Table 2. Performance test of Create() with the fixed number of transactions and TPS.

Create(): Fixed txNumber and TPS, Different N. of Workers					
txNumber	**TPS**	**N. of Workers**	**Send Rate**	**Throughput**	**Avg Latency**
2000	100	5	22.7	22.2	1.79
		25	37.1	35.8	2.05
		50	39.0	38.0	1.93

5.4.2. Query()

The setup is the same as the test of the Create() function. The result of the Query() function with the fixed number of workers is shown in Table 3. As TPS increases from 50 to 200, both the send rate and throughput increase but begin to saturate. With the fixed

number of workers, the system achieves near-optimal performance, as indicated by the minimal differences between the send rate and throughput across various txNumber values. For a larger txNumber (e.g., 5000), the system maintains stable performance with high throughput. The reported latency is essentially negligible (0.00 or 0.01 ms), indicating that query operations are highly efficient. This is expected since queries are typically read-only and do not involve state updates, allowing for faster execution. Therefore, increasing the total number of transactions from 1000 to 5000 does not significantly affect throughput or latency, demonstrating that the system scales well under the given load.

Table 3. Performance test of Query() with the fixed number of workers.

Query(): Fixed N. of Workers, Different txNumber and TPS					
N. of Workers	**txNumber**	**TPS**	**Send Rate**	**Throughput**	**Avg Latency**
5	1000	50	606.4	605.3	0.01
		100	693.5	692.0	0.00
		200	699.3	696.9	0.00
5	2000	50	986.2	985.2	0.00
		100	1042.8	1041.1	0.00
		200	1093.5	1092.3	0.00
5	5000	50	1484.6	1483.7	0.00
		100	1607.2	1605.7	0.00
		200	1518.4	1517.0	0.00

The result of the Query() function with the fixed number of transactions and TPS is shown in Table 4. Contrary to expectations, increasing the number of workers leads to a decrease in both send rate and throughput. For 5 workers, the system achieves a send rate of 1042.8 and a throughput of 1041.1 TPS, which significantly drops to 179.1 and 179.0 TPS, respectively, with 50 workers. Query operations are highly efficient and likely CPU-bound. Adding more workers beyond the system's processing capacity leads to contention for resources, reducing performance. Latency increases slightly (from 0.00 to 0.03 ms) as the number of workers grows. The rise in latency is minor, indicating that the system is still able to process queries quickly, even under higher contention. Therefore, with five workers, the system achieves the best balance between throughput and latency for query operations. Increasing workers to 25 or 50 does not improve performance and instead decreases throughput.

Table 4. Performance test of Query() with the fixed number of transactions and TPS.

Query(): Fixed txNumber and TPS, Different N. of Workers					
txNumber	**TPS**	**N. of Workers**	**Send Rate**	**Throughput**	**Avg Latency**
2000	100	5	1042.8	1041.1	0.00
		25	333.7	333.4	0.03
		50	179.1	179.0	0.03

5.4.3. Update()

The result of the Update() function with the fixed number of workers is shown in Table 5. As TPS increases from 50 to 200, both the send rate and throughput increase significantly. However, the gap between the send rate and throughput widens slightly at higher TPS. This could indicate some resource contention or system queuing. Latency decreases significantly as TPS increases, dropping from ~1.9 ms to ~0.5 ms. Increasing txNumber (from 1000 to 5000) does not significantly affect throughput or latency, demon-

strating that the system handles higher workloads consistently. At TPS = 200, the system achieves its highest throughput and lowest latency, indicating the network's ability to handle high-frequency updates efficiently.

Table 5. Performance test of Update() with the fixed number of workers.

Update(): Fixed N. of Workers, Different txNumber and TPS					
N. of Workers	**txNumber**	**TPS**	**Send Rate**	**Throughput**	**Avg Latency**
5	1000	50	11.4	11.2	1.97
		100	25.4	25.1	1.64
		200	128.8	108.7	0.56
5	2000	50	12.6	12.4	1.82
		100	21.3	20.8	1.93
		200	145.4	127.3	0.45
5	5000	50	13.5	13.4	1.57
		100	27.6	27.3	1.73
		200	138.2	130.6	0.52

The result of the Update() function with the fixed number of transactions and TPS is shown in Table 6. As the number of workers increases, both the send rate and throughput improve. The improvement is linear, indicating the system can effectively handle increased concurrency latency increases slightly from 1.93 ms at 5 workers to 2.02 ms at 50 workers as the number of workers grows. This increase is minor, demonstrating that the system maintains low latency even under higher concurrency. In general, the system performs efficiently with 50 workers, achieving the highest throughput. However, latency begins to plateau, suggesting that further increases in worker count may not yield significant benefits.

Table 6. Performance test of Update() with the fixed number of transactions and TPS.

Update(): Fixed txNumber and TPS, Different N. of Workers					
txNumber	**TPS**	**N. of Workers**	**Send Rate**	**Throughput**	**Avg Latency**
2000	100	5	21.3	20.8	1.93
		25	37.0	35.8	1.96
		50	42.9	41.4	2.02

5.4.4. Delete()

The result of the Delete() function with the fixed number of workers is shown in Table 7. At lower TPS, send rate and throughput are closely aligned, indicating that the system is able to handle this level of workload without issues. Also, the system processes significantly more transactions as the send rate and throughput increase sharply at higher TPS. For txNumber = 5000, the send rate reaches 146.3 TPS, and throughput is close at 137.8 TPS, indicating the system is efficiently handling high rates. At lower TPS (50 and 100), latency is higher due to lower transaction batching efficiency. At TPS = 200, latency drops significantly, indicating the system benefits from higher throughput and better resource utilization. Therefore, the total number of transactions has little impact on performance. Across all the txNumber values (1000, 2000, and 5000), the system maintains consistent throughput and latency.

Table 7. Performance test of Delete() with the fixed number of workers.

Delete(): Fixed N. of Workers, Different txNumber and TPS					
N. of Workers	**txNumber**	**TPS**	**Send Rate**	**Throughput**	**Avg Latency**
5	1000	50	12.6	12.3	1.76
		100	24.8	23.7	1.97
		200	127.3	110.2	0.51
5	2000	50	11.7	11.5	1.79
		100	27.0	26.4	1.68
		200	146.0	137.0	0.56
5	5000	50	13.5	13.7	1.71
		100	23.2	23.0	1.80
		200	146.3	137.8	0.49

The result of the Delete() function with the fixed number of transactions and TPS is shown in Table 8. As the number of workers increases, both the send rate and throughput improve. The system handles concurrency well, with throughput increasing linearly with the number of workers. Latency remains relatively stable across all the worker configurations. This suggests that the system maintains low contention for resources, even with higher concurrency. In general, increasing workers to 50 provides the highest throughput. However, the slight increase in latency indicates that the system is nearing its optimal concurrency limit.

Table 8. Performance test of Delete() with the fixed number of transactions and TPS.

Delete(): Fixed txNumber and TPS, Different N. of Workers					
txNumber	**TPS**	**N. of Workers**	**Send Rate**	**Throughput**	**Avg Latency**
2000	100	5	27.0	26.4	1.68
		25	35.4	34.1	1.9
		50	40.0	38.5	1.86

5.4.5. High-Pressure Test

Table 9 shows the result of the performance test of high pressure. In general, all four functions were completed without any failed transactions under high pressure, indicating excellent system stability. The throughput values are close to their respective send rate values, demonstrating that the system effectively processes most of the submitted transactions within the defined duration.

Table 9. Performance test of high pressure.

High Pressure with N. of Workers: 50, txDuration: 300, TPS: 400					
Function	**Succ**	**Fail**	**Send Rate**	**Throughput**	**Avg Latency**
Create()	105,167	0	361.3	360.7	0.75
Query()	1,011,725	0	3846.4	3486.4	0.06
Update()	131,422	0	451.2	448.0	0.61
Delete()	128,493	0	441.9	440.5	0.62

First, the creation process exhibits relatively higher latency compared to the other operations. This is expected as Create() involves writing new data and updating the blockchain state, which is computationally heavier. Second, Query() achieves the highest throughput among all the operations with minimal latency, as it involves read-only operations that do not require state changes or consensus. Third, the update process is more resource-intensive

than Query() but still achieves high throughput and low latency. The latency (0.61 ms) is slightly better than Create() because updates generally involve modifying the existing records rather than inserting new ones. Fourth, similarly to Update(), Delete() exhibits efficient performance with moderate latency. The throughput and latency are comparable to Update(), indicating consistent resource utilization for state-altering operations.

6. Discussion

This section evaluates the advantages and limitations of the proposed system. The term Pros refers to the system's key strengths and benefits, such as HLF's features, enhanced data privacy, and decentralized management. On the other hand, Cons identify potential challenges and areas where improvements are needed, such as transaction latency and system complexity. By comparing these aspects against traditional database systems, the analysis highlights the practicality and future potential of the proposed architecture.

6.1. Pros of the Proposed System

Unlike computationally intensive Proof of Work (PoW), HLF utilizes the Raft consensus protocol to provide a lightweight yet reliable mechanism to maintain the blockchain's integrity [20]. It enables faster transaction finalization while preserving fault tolerance, making it well suited for IoV environments with high transaction volumes. By leveraging CouchDB as the state database, the system enables efficient and complex queries on chaincode data. Thanks to HLF, our system utilizes TLS for secure communications and encryption for identity management. It can make the system resistant to common cybersecurity threats. Also, access control through MSP ensures that only authenticated and authorized participants can interact with the network.

Also, the decentralized architecture of HLF is the key feature of our system. It can eliminate the need for a central authority and reduce the risk of a single point of failure. Therefore, it ensures that no single organization has any unilateral control over the data. Meanwhile, the use of blockchain ensures that all the transactions are immutable once submitted. The multi-channel structure provides logical isolation to ensure that only authorized parties have access to sensitive data. We find this feature particularly useful in use cases involving multiple departments, especially where data collaboration is required. Finally, in the central channel, Channel A, of our system, unified queries allow seamless integration and interaction between different data domains.

6.2. Cons of the Proposed System

First, during the testing phase, we find that the use of consensus protocols and cryptographic operations introduces latency, particularly under high transaction loads. This may limit the system's applicability for ultra-low latency IoV applications, such as autonomous driving control systems. Second, setting up and maintaining a HLF network requires a significant number of operations. Updating or expanding systems can be challenging compared to conventional systems. Any changes, such as adding new participants, require careful updates to network configurations and chaincode. Third, the size of the ledger grows over time, with higher storage requirements compared to traditional databases. When the system is large enough, it may be inefficient to store large amounts of data on the chain.

6.3. Comparison with Traditional Database Systems

Traditional database systems are typically managed by a single authority or organization. Decisions about the database, such as access rights and data updates, are centralized. For blockchain, data are replicated across multiple nodes in a peer-to-peer network. Each participant holds a copy of the ledger. It can ensure data redundancy and robustness

against single-point failures [22]. Traditional database system relies on user roles, passwords, and firewalls for security. A single breach in the central database can expose the entire dataset [23]. For our system, logical isolation ensures sensitive data are only accessible to authorized parties. Transactions are validated by multiple nodes, making them highly resilient to fraud or cyberattacks. As for performance, traditional database systems can have high throughput. It is optimized for speed, with support for thousands of transactions per second. It is also suitable for applications requiring immediate data updates [24]. However, blockchain systems may have transaction latency due to consensus protocols. It is optimized for trust and focuses on data integrity.

7. Conclusions

This paper presents a multi-channel HLF-based IoV system. Our system leverages the decentralized, secure, and immutable architecture of blockchain to enhance data integrity, privacy, and trust among stakeholders in the IoV.

The key contribution of the system is the multi-channel structure. The separation of data into different channels can ensure data privacy and efficient collaboration among different organizations. The use of chaincodes enables secure and automated transactions across channels. The testing phase validates successful deployment, chaincode operation, and efficient data exchange across channels. The Performance Analysis Section demonstrates the stable performance of the proposed system in different environments.

Several areas remain for further research and enhancement. In the future, first, we will focus on improving user interfaces and developer tools to simplify the interaction and deployment of blockchain-based IoV systems for stakeholders. Moreover, we desire to explore the integration of edge computing for real-time data processing at the vehicle level. It is necessary to reduce the network load and latency associated with blockchain transactions. Finally, we want to conduct extensive scalability tests under real-world scenarios to evaluate system performance with many participants and diverse data streams.

Author Contributions: Conceptualization, Y.L., Y.F. and K.S.; methodology, Y.L. and Y.F.; software, Y.L.; validation, Y.L.; formal analysis, Y.L. and Y.F.; investigation, Y.L. and Y.F.; writing—original draft preparation, Y.L.; writing—review and editing, Y.F. and K.S.; visualization, Y.L.; supervision, Y.F. and K.S.; project administration, Y.L., Y.F. and K.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no funding.

Data Availability Statement: Data are contained within the article.

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

References

1. Meijers, J.; Michalopoulos, P.; Motepalli, S.; Zhang, G.; Zhang, S.; Veneris, A.; Jacobsen, H.A. Blockchain for V2X: Applications and Architectures. *IEEE Open J. Veh. Technol.* **2022**, *3*, 193–209. [CrossRef]
2. Yeong, D.J.; Velasco-hernandez, G.; Barry, J.; Walsh, J. Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors* **2021**, *21*, 2140. [CrossRef] [PubMed]
3. Ang, L.M.; Seng, K.P.; Ijemaru, G.K.; Zungeru, A.M. Deployment of IoV for Smart Cities: Applications, Architecture, and Challenges. *IEEE Access* **2019**, *7*, 6473–6492. [CrossRef]
4. Chen, C.; Quan, S. A Summary of Security Techniques-Based Blockchain in IoV. *Secur. Commun. Netw.* **2022**, *2022*, 8689651. [CrossRef]
5. Hammoud, A.; Sami, H.; Mourad, A.; Otrok, H.; Mizouni, R.; Bentahar, J. AI, Blockchain, and Vehicular Edge Computing for Smart and Secure IoV: Challenges and Directions. *IEEE Internet Things Mag.* **2020**, *3*, 68–73. [CrossRef]
6. Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In Proceedings of the 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017, Honolulu, HI, USA, 25–30 June 2017.

7. Yaga, D.; Mell, P.; Roby, N.; Scarfone, K. *Blockchain Technology Overview*; No. 8202; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2019; pp. 1–57.
8. Mollah, M.B.; Zhao, J.; Niyato, D.; Guan, Y.L.; Yuen, C.; Sun, S.; Lam, K.Y.; Koh, L.H. Blockchain for the Internet of Vehicles towards Intelligent Transportation Systems: A Survey. *IEEE Internet Things J.* **2021**, *8*, 4157–4185. [CrossRef]
9. Github Contributors. Hyperledger Fabric. Available online: https://github.com/hyperledger/fabric (accessed on 11 December 2024).
10. Androulaki, E.; Barger, A.; Bortnikov, V.; Muralidharan, S.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Murthy, C.; Ferris, C.; et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In Proceedings of the 13th EuroSys Conference, EuroSys 2018, Porto, Portugal, 23–26 April 2018.
11. Nasir, Q.; Qasse, I.A.; Abu Talib, M.; Nassif, A.B. Performance analysis of hyperledger fabric platforms. *Secur. Commun. Netw.* **2018**, *2018*, 3976093. [CrossRef]
12. Foschini, L.; Gavagna, A.; Martuscelli, G.; Montanari, R. Hyperledger Fabric Blockchain: Chaincode Performance Analysis. In Proceedings of the ICC 2020–2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020.
13. Pajooh, H.H.; Rashid, M.; Alam, F.; Demidenko, S. Hyperledger fabric blockchain for securing the edge internet of things. *Sensors* **2021**, *21*, 359. [CrossRef] [PubMed]
14. Alalwany, E.; Mahgoub, I. Security and Trust Management in the Internet of Vehicles (IoV): Challenges and Machine Learning Solutions. *Sensors* **2024**, *24*, 368. [CrossRef] [PubMed]
15. Karim, S.M.; Habbal, A.; Chaudhry, S.A.; Irshad, A. Architecture, Protocols, and Security in IoV: Taxonomy, Analysis, Challenges, and Solutions. *Secur. Commun. Netw.* **2022**, *2022*, 113479. [CrossRef]
16. Ding, Q.; Gao, S.; Zhu, J.; Yuan, C. Permissioned Blockchain-Based Double-Layer Framework for Product Traceability System. *IEEE Access* **2020**, *8*, 6209–6225. [CrossRef]
17. Hussain, S.; Tahir, S.; Masood, A.; Tahir, H. Blockchain-Enabled Secure Communication Framework for Enhancing Trust and Access Control in the Internet of Vehicles (IoV). *IEEE Access* **2024**, *12*, 110992–111006. [CrossRef]
18. Ndzimakhwe, M.; Telukdarie, A.; Munien, I.; Vermeulen, A.; Chude-Okonkwo, U.K.; Philbin, S.P. A Framework for User-Focused Electronic Health Record System Leveraging Hyperledger Fabric. *Information* **2023**, *14*, 51. [CrossRef]
19. Novo, O. Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT. *IEEE Internet Things J.* **2018**, *5*, 1184–1195. [CrossRef]
20. Huang, D.; Ma, X.; Zhang, S. Performance Analysis of the Raft Consensus Algorithm for Private Blockchains. *IEEE Trans. Syst. Man. Cybern. Syst.* **2020**, *50*, 172–181. [CrossRef]
21. Github Contributors. Hyperledger Caliper. Available online: https://github.com/hyperledger-caliper/caliper (accessed on 16 January 2025).
22. Stephen, R.; Alex, A. A Review on BlockChain Security. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *396*, 012030. [CrossRef]
23. Muqorobin, M.; Rozaq Rais, N.A. Comparison of PHP Programming Language with Codeigniter Framework in Project CRUD. *Int. J. Comput. Inf. Syst.* **2022**, *3*, 94–98. [CrossRef]
24. Abourezq, M.; Idrissi, A. Database-as-a-Service for Big Data: An Overview. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*, 157–177. [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

Implicit Identity Authentication Method Based on User Posture Perception

Bo Hu [1], Shigang Tang [2], Fangzheng Huang [3], Guangqiang Yin [1] and Jingye Cai [1,*]

[1] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 611730, China; hubo@std.uestc.edu.cn (B.H.); yingq@uestc.edu.cn (G.Y.)
[2] Shenzhen Silver Star Intelligent Technology Co., Ltd., Shenzhen 518110, China; tangshigang@yxrobot.com
[3] Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada; f34huang@uwaterloo.ca
* Correspondence: jycai@uestc.edu.cn

Abstract: Smart terminals use passwords and physiological characteristics such as fingerprints to authenticate users. Traditional authentication methods work when users unlock their phones, but they cannot continuously verify the user's legal identity. Therefore, the one-time authentication implemented by conventional authentication methods cannot meet security requirements. Implicit authentication technology based on user behavior characteristics is proposed to achieve the continuous and uninterrupted authentication of savvy terminal users. This paper proposes an implicit authentication method that fuses keystroke and sensor data. To improve the accuracy of authentication, a neural network-based feature extraction model that integrates keystroke data and motion sensor data is designed. A feature space with dual-channel fusion is constructed, and a dataset collected in real scenarios is built by considering the changes in user activity scenarios and the differences in terminal holding postures. Experimental results on the collected data show that the proposed method has improved the accuracy of user authentication to a certain extent.

Keywords: keystroke data; sensor data; implicit identity authentication

1. Introduction

Identity authentication technology is crucial for the security of user data, preventing unauthorized access by verifying user identity [1–6]. Based on whether the authentication process is perceived by the user, identity authentication technology can be divided into explicit and implicit authentication. Password-based authentication is a type of explicit authentication where users set personal identification numbers (PINs) or pattern-based passwords according to their habits for identity verification. However, numerical and pattern passwords are limited, easily reused, and require additional memory, making them prone to being forgotten. Moreover, passwords are at risk of being leaked, resulting in the poor security of password-based authentication methods [7–16].

To address the drawbacks of password reuse, forgetfulness, and leakage, token-based authentication techniques have been proposed. Token authentication uses hardware devices like smart cards that store private keys to authenticate users. However, since these hardware devices need to be carried around, this method faces issues like loss, theft, and forgery. Passwords and tokens used for authentication do not establish a direct one-to-one link with user identity, leading to weak security in these methods.

Biometric authentication techniques use physiological features such as fingerprints [17], facial recognition [18], voiceprints [19], and iris recognition [20] to identify

users. Currently, fingerprint and facial recognition have shown good performance and are widely used in smartphones. However, fingerprint recognition is affected by finger moisture and fingerprint wear, and fingerprint data can be easily stolen. Facial recognition is influenced by lighting, angle, facial expressions, and makeup, making it limited in practical use. Additionally, explicit authentication techniques can only authenticate the user once when unlocking the phone and cannot continuously verify the user's legitimacy.

In contrast, implicit authentication technology identifies unauthorized users through behavioral characteristics such as gait while carrying the phone, touchscreen patterns during tap or swipe operations, and keystroke dynamics [21]. Since the sensors embedded in phones can directly capture behavioral data without requiring users to perform specific actions, implicit authentication technology can achieve transparent, unobtrusive, and continuous identity verification. It is worth noting that behavioral characteristics are formed through a combination of innate neural, muscular, and skeletal systems and acquired habits, making them difficult to record and mimic. Therefore, implicit authentication technology based on user behavior can achieve high security. Modern smart terminals are equipped with sensors like accelerometers, gyroscopes, magnetometers, orientation sensors, and touchscreens [22], facilitating the collection of user behavior data and improving the performance of implicit authentication technology.

However, there is still room for improvement in the performance of existing implicit authentication methods. On the one hand, the limited representation capability of constructed user behavior features leads to poor authentication accuracy. On the other hand, the performance of existing methods varies across different application scenarios, requiring enhancements in robustness and practical usability. Thus, exploring comprehensive user behavior representation and designing a framework for implicit authentication methods is crucial for achieving better accuracy, robustness, and practical usability, providing continuous and reliable security for user-smart terminal interactions.

This paper proposes an implicit authentication method based on user posture awareness to improve authentication accuracy. It constructs a staged cascade authentication framework that first recognizes the user's posture and then authenticates their identity. Firstly, the motion sensor data acknowledges the user's posture. Then, a two-channel user classification model is designed by fusing the number of keystrokes, enhancing the method's robustness and improving authentication accuracy. The main contributions of this thesis are as follows:

1. An implicit authentication dataset containing two activity scenarios and three holding postures is constructed.
2. An algorithm for recognizing user gestures using sensor data is proposed, which solves the problem of too significant feature differences caused by different operating gestures.
3. An implicit authentication method fusing keystrokes and sensor data is proposed. This method fully utilizes CNN's local spatial feature extraction capability and LSTM's temporal feature extraction capability to capture keystroke data and sensor data's continuity characteristics and unique behavioral features.

2. Related Work

Keystroke characteristics are developed over the long-term use of the same phone, with significant differences in keystroke pressure, frequency, and usage habits of special keys among different users [23,24]. When a user performs keystroke operations, the touchscreen of the phone collects data such as keystroke time, touch point location, and keystroke pressure, reflecting the user's keystroke rhythm and habits. Additionally, the built-in motion sensors can detect changes in the phone's movement during keystrokes. Therefore,

keystroke characteristics can be represented by both touchscreen data and motion sensor data, offering strong discriminative power for identity authentication.

Keystroke authentication can be categorized into fixed-text-based and free-text-based authentication, depending on whether the input text is known. The former refers to scenarios where the input text is fixed, such as entering a numeric password to unlock a phone. The latter refers to scenarios where the input text is unrestricted, such as chatting on social media. In fixed-text-based keystroke authentication, researchers have constructed datasets by collecting touchscreen data during password entry, extracting keystroke features for identity authentication, including time features [25–29], pressure features [26,29], and location features [29]. Buchoux et al. [28] developed an application to collect touchscreen data from 20 users entering passwords and built a feature set by calculating statistical features of inter-keystroke times, achieving a 2.5% FRR. Zahid et al. [26] further divided inter-keystroke time into times between horizontally adjacent keys, horizontally non-adjacent keys, vertically adjacent keys, and vertically non-adjacent keys, introducing key hold time and the number of backspace key uses to represent keystroke characteristics. This achieved a 0% FRR and 2% FAR on a dataset of 25 users.

In addition to time features, Zhang et al. [28] were the first to introduce pixel distance features and speed features of adjacent keys, achieving a 3.5% FAR. Zhang et al. collected time and pressure data from 10 users entering the same password using a self-developed application, achieving a 10.3% false positive rate and 91.7% true positive rate using a radial basis function neural network. Krishnamoorthy et al. [29] collected touchscreen data from 94 users entering passwords, extracting time, pressure, and location features to construct a 155-dimensional feature vector. Using a random forest classifier, they achieved a 98.44% accuracy rate. Besides touchscreen data, motion sensor data during keystrokes can enrich keystroke features. Zheng et al. [30] extracted features such as the amplitude at the time of pressing, the amplitude at the time of release, and the maximum and average amplitude during keystrokes from accelerometer data. Combining these with keystroke pressure, touch area, and duration features to form a feature vector, they determined whether the current sample belonged to an unauthorized user by calculating the distance to training data, achieving a 3.65% EER on a dataset of 85 users.

Fixed-text-based keystroke authentication has demonstrated good performance but is limited to scenarios where users enter passwords. Free-text-based keystroke authentication is more aligned with practical scenarios, but the lack of restrictions on input content increases the difficulty of keystroke authentication. In 2015, Kang et al. [31] studied the impact of different input text lengths on keystroke authentication performance, confirming that as the input text length increases, the EER significantly decreases in both single-handed and dual-handed typing scenarios. In 2018, Kim et al. [32] proposed an adaptive feature extraction method for keystroke characteristics, achieving an EER of 0.44% on a dataset of 150 users. These studies focused on time features of keystroke data. Buschek et al. [33] suggested that combining spatial features of keystroke data could significantly improve authentication performance. By introducing touch point coordinates and keystroke pressure as spatial features, they conducted experiments in single-handed thumb typing, dual-handed thumb typing, and left-hand holding, right-hand index finger typing scenarios. The results showed a 14.3% reduction in EER compared to time-based features alone and a 26.4% reduction when combining spatial and temporal features, confirming the significant role of spatial features in identity authentication.

Additionally, Giuffrida et al. [34] found that motion sensor data during keystrokes show significant variations, with differences in keystroke behavior among users being represented by motion sensor data. Giuffrida et al. [34] collected accelerometer and gyroscope data during the keystroke process from 20 users, achieving an EER of 0.08% on this dataset,

which was significantly better than the 4.97% EER obtained from keystroke time features. Shankar et al. [35] used accelerometer, gyroscope, and magnetometer data to represent keystroke behavior. They input manually extracted time-domain features into a deep autoencoder for feature dimensionality reduction, followed by user classification using a softmax layer, achieving 95% and 97% authentication accuracy in walking and sitting scenarios, respectively. Further, Buriro et al. [36] combined motion sensor and keystroke time features, requiring 10 keystrokes to complete one identity authentication, achieving an authentication accuracy of 85.77% and a FAR of 7.32%. Kim et al. [37] extracted features from an accelerometer, touch point coordinates, and time information to construct a 17-dimensional feature vector. By comparing the distribution of keystroke features, they identified unauthorized users, achieving an optimal performance of 0.45% EER.

The authentication method based on keystroke characteristics can incorporate motion sensor data as input data to enrich the feature representation and further consider the effectiveness of the spatial attributes of keystroke data in representing the user's identity. Both are crucial to improving authentication performance. Since a user's keystroke habits may vary with the user's activity scenario and the grip position of the mobile phone, the authentication method needs to consider the influence of these factors. Current research primarily uses binary classification methods to determine whether it is a legitimate user operation but cannot confirm the exact user identity. In addition, most existing methods extract keystroke features manually, and fewer construct neural networks to complete feature extraction. Designing a neural network structure for keystroke authentication is the next step in improving the method.

3. Data Collection and Preprocessing

User interactions with mobile phones primarily occur in two activity scenarios: stationary activity and walking activity. The behavioral characteristics of phone usage change when the activity scenario shifts. Besides the activity scenario, the differences in phone usage behavior also stem from various holding postures. In stationary scenarios, users typically adopt three holding postures: holding with the dominant hand, holding with both hands, and placing the device flat on a table. In walking scenarios, users generally use two holding postures: holding with the dominant hand and holding with both hands. The characteristics of different holding postures are mainly captured through touchscreen location information and the amplitude variation features of motion sensor data. The data collection details of this study are shown in Table 1.

Table 1. Data field table.

Field	Type	Description
user ID	string	Identifier assigned to the data collector.
action	int	1 for stationary, 2 for walking.
Pose	Int	1 for dominant-hand grip, 2 for two-hand grip, 3 for device lying flat on a table.
Accelerometer	List	Accelerometer data collected during user input. Each set of three represents x-axis, y-axis, and z-axis.
Gyroscope	List	Gyroscope data collected during user input. Each set of three represents x-axis, y-axis, and z-axis.
Gap	List	Represents the press time, release time, and press position of the user clicking the keyboard $\{T_{press}, T_{release}, x, y\}$.

To construct the dataset used in the experiment, this study recruited 60 volunteers, including students and office workers aged 20–35. With each person's consent, sensor

and keystroke data were collected from each volunteer using their smartphones. The smartphones included various brands such as Huawei, Xiaomi, OPPO, and VIVO. The data were collected at a frequency of 100 Hz/s, and each volunteer collected seven rounds of data daily, which took about 10 min a day, for 6740 data points. The experimental collection plan is shown in Table 2.

Table 2. Experimental data collection plan.

Experimental Element	Experimental Arrangement
Number of Participants	60 people.
Experimental Equipment	Personal Android phone.
Activity Scenarios	Stationary, Walking.
Holding Postures	Dominant-hand grip, Two-hand grip, Device lying flat.
Time Span	Each volunteer participates in data collection for 7 rounds, at least one round per day, with each round including data collection in 2 activity scenarios and 3 holding postures.

3.1. Data Collection

Keystroke Data Collection. For each keystroke action, the data collection module should record $\{T_{press}, T_{release}, x, y\}$, where T_{press} is the time the screen is pressed, and $T_{release}$ is the time the screen is released. T_{press} and $T_{release}$ can be used to calculate dwell time and inter-key time, which reflect the user's screen tapping rhythm habits. x, y represent the horizontal and vertical positions of the touch point on the screen.

Sensor data collection. The accelerometer and gyroscope sensors built into smartphones can measure the micro-movements and rotational rates caused by user interaction with the phone. These sensors are commonly equipped in most smartphones today, and this study utilizes them to model the phone's motion modes. For each keystroke action, the data collection module needs to record the sampled data sequences of the accelerometer and gyroscope along the X, Y, and Z axes during the duration of the behavior.

3.2. Data Preprocessing

Since the raw data recorded by the data collection module usually contains information not required for the authentication method, such as screen touch data and multi-finger operation data, it is necessary to perform a series of data cleanings and processing to make it usable by the feature extraction module. Specifically, the main functions of the data preprocessing module are as follows:

Segmentation of Continuous Keystroke Behaviors: When users input a piece of text, they often pause for thinking or other reasons, causing discontinuities in text input. To facilitate the extraction of keystroke features, the data preprocessing module segments the keystroke data of a text input into several "continuous keystroke behaviors". If the inter-key time between two keystrokes is less than or equal to a set threshold, the two keystrokes belong to the same continuous keystroke behavior; if the interval time between two keystrokes is greater than the set threshold, the two keystrokes belong to different continuous keystroke behaviors.

Extraction of Motion Sensor Data Sequences: To facilitate the extraction of motion mode features during the duration of continuous keystroke behaviors, for each continuous keystroke behavior, the data preprocessing module extracts the sampled data sequences of the accelerometer and gyroscope on the X, Y, and Z axes for the duration of each continuous keystroke behavior. A total of six sensor data sequences are extracted for each duration.

Handling Missing Values: The raw data recorded by the data collection module may contain various missing values, such as the lack of press time or release time data for a keystroke. These missing values need to be deleted in the data preprocessing module.

4. Experimental Analysis

This paper considers the impact of changes in user activity scenarios and differences in holding postures. It constructs a two-stage cascading authentication framework that first identifies user posture and then authenticates user identity, as shown in Figure 1. Initially, user posture is recognized through motion sensor data. Then, a dual-channel user classification model is designed by integrating keystroke data, which enhances the robustness of the method while improving authentication accuracy.

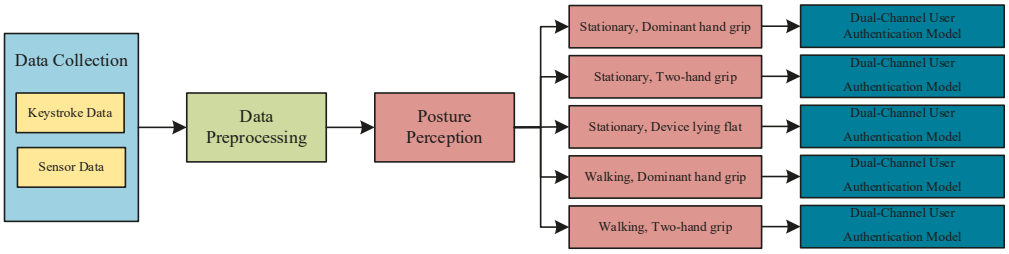


Figure 1. Implicit identity authentication framework integrating keystroke data and sensor data.

4.1. Posture Perception Model

Implicit identity authentication performance can be affected by changes in user activity scenarios and holding postures. Identifying the current activity scenario and holding posture can help improve authentication performance. Posture perception can be constructed as a five-classification problem to recognize the following states: a stationary state with dominant-hand holding, a stationary state with both hands holding, a stationary state with the device lying flat, a walking state with dominant-hand holding, and a walking state with both hands holding. Considering that the touchscreen data records information such as the time of user operation, touch point location, and contact area, which cannot reflect the user's movement state, the posture perception model only uses motion sensor data as input data.

In terms of model construction, the designed CNN model consists of three convolutional layers, with a batch normalization (BN) layer and a ReLU activation layer connected after each convolutional layer. The BN and ReLU layers can speed up the training and convergence of the network and prevent gradient disappearance or explosion. Considering that the channels of gait signals are all time-series signals, the convolutional layer of the CNN model only uses a one-dimensional convolution operation, which is conducive to the Bi-LSTM module for extracting temporal features. The size and number of convolutional kernels are essential parameters that affect feature extraction performance. As the depth of the network increases, smaller convolutional kernels can extract richer features due to the increase in the receptive field. Therefore, the size of the convolutional kernel is reduced, and its number is increased. The bidirectional long short-term memory module contains 64 hidden units, uses the tanh activation function, and uses dropout to prevent overfitting. The hierarchical structure and parameter settings of the CNN-LSTM model are shown in Table 3.

Table 3. CNN-LSTM model hierarchical structure.

Layer	Convolution Kernel Size	Number of Convolution Kernels
Convolution Layer	1×5	12
Batch Normalization Layer	/	/
Activation Layer	/	/
Batch Normalization Layer	1×5	24
Activation Layer	/	/
Batch Normalization Layer	/	/
Batch Normalization Layer	1×5	36
Activation Layer	/	/
Batch Normalization Layer	/	/
Bi-directional LSTM Module	/	/

4.2. Dual-Channel User Authentication Model

When users interact with the smartphone, the touchscreen and motion sensors reflect different aspects of the interaction. The touchscreen records the contact time and position of the user's fingertip, while motion sensors detect changes in the smartphone's movement during the operation. To extract effective features maximally from sensor data, the paper designs a dual-channel feature extraction network. Each channel consists of a Convolutional Neural Network (CNN) and a Long Short-Term Memory network (LSTM). The outputs from both channels are then fused and fed into a fully connected layer for final user identity decision-making. The designed user classification model is illustrated in Figure 2.

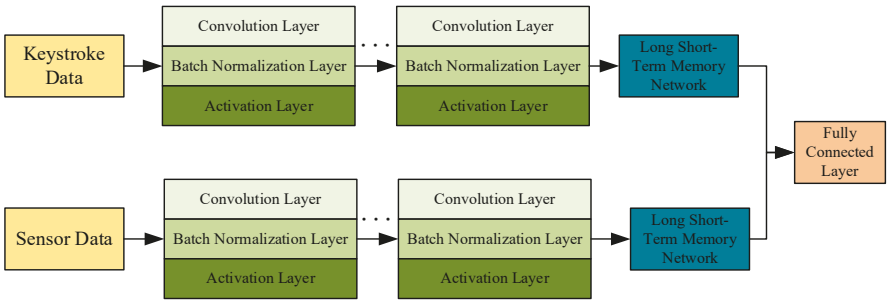


Figure 2. Dual-channel user authentication model.

At the forefront of the dual-channel network, the CNN model directly receives pre-processed keystroke signals and motion sensor signals. The CNN model consists of two convolutional layers, utilizing one-dimensional convolutional kernels to better preserve the inherent temporal features of the data. The advantage of convolutional processing lies in its capability to extract long-range temporal features, which is beneficial for facilitating the effectiveness of the LSTM. After LSTM processing, the features from both channels need to be fused together to construct a unified user classifier. Assuming that the features extracted from the touchscreen signal channel are denoted as T, and those from the motion sensor signal channel are denoted as O, the fused feature F serves as the input to the fully connected layer, represented as:

$$F = \begin{bmatrix} T \\ O \end{bmatrix} \tag{1}$$

4.3. Experimental Environment

The experimental environment configuration for this experiment is shown in Table 4.

Table 4. Experimental environment configuration.

Type	Configuration
Operating System	Ubuntu 18.04 LTS
Python	3.8
CUDA	11.2
NVIDIA Driver	460.8
CPU	2×Intel(R)Xeon(R)Silver4216CPU@2.10GHz
Memory	8 × 32 GB
GPU	8×NVIDIAQuadroRTX8000

4.4. Experimental Result

Performance of the posture perception model. In actual usage scenarios, users may use their phones while they are stationary or walking. When users use their phones while stationary, the sensor data are mainly influenced by human interaction with the device. When users use their phones while walking, the sensor data are primarily influenced by movement. Therefore, data collection during walking activities involves many interferences, which somewhat weakens the ability to represent user behavior characteristics.

In actual usage scenarios, users may use their phones with a dominant-hand grip, a two-hand grip, or with the device lying flat. When users operate their phones with a dominant-hand grip, the contact points with the touchscreen are mainly concentrated on the side of the dominant hand. If users need to tap touchpoints on the opposite side, it can cause the significant rotation of the phone. Given the large screen sizes of most smartphones on the market, this phenomenon is common. Thus, data collection under a dominant-hand grip involves many interferences, which somewhat weakens the ability to represent user behavior characteristics. Under a two-hand grip, the phone's motion state is more stable, and the behavior data collected are less interfered with, better representing the user's behavior characteristics. When the device is placed flat, the phone's dynamic state is most stable, best representing user behavior characteristics.

Table 5 compares the authentication performance of users operating phones in different postures (dominant-hand grip, two-hand grip, or device lying flat) in both stationary and walking activity scenarios. It can be observed that in stationary scenarios, the highest authentication accuracy is achieved when the device is lying flat. In both stationary and walking scenarios, the two-hand grip achieves higher authentication accuracy than the dominant-hand grip, and the same posture in stationary scenarios achieves higher authentication accuracy than in walking scenarios. There is significant variation in the frequency of use of the dominant-hand grip and the two-hand grip among different users. To better reflect actual usage scenarios, the evaluation of the method's performance should consider not only the high authentication accuracy of the two-hand grip posture but also the operational data under the dominant-hand grip posture.

Table 5. Posture perception model performance results.

Activity Scenario	Holding Posture	Accuracy	FAR	FRR	EER
Stationary	Dominant-hand grip	81.64%	0.1782	0.1904	0.1880
	Two-hand grip	86.78%	0.0910	0.0916	0.0924
	Device lying flat	95.83%	0.0624	0.0623	0.0624
Walking	Dominant-hand grip	76.28%	0.1927	0.2018	0.1996
	Two-hand grip	83.24%	0.1237	0.1224	0.1276

Comparison of Authentication Performance Across Different Sensors. This paper integrates keystroke data and motion sensor data as input data to represent various operations.

Keystroke data provides the time of screen touch, release time, and touch coordinates, reflecting the user's temporal and spatial characteristics during operation. Motion sensors, on the other hand, capture the changes in the phone's motion state during user operations. When authenticating user identity, keystroke data and motion sensor data play different roles, as shown in Table 6.

Table 6. Authentication performance comparison of different sensors.

User Posture	Data Type	Accuracy
Stationary, Dominant-hand grip	Motion sensor	78.32%
	Motion sensor + Keystroke	81.64%
Stationary, Two-hand grip	Motion sensor	83.91%
	Motion sensor + Keystroke	86.78%
Stationary, Device lying flat	Motion sensor	93.76%
	Motion sensor + Keystroke	95.83%
Walking, Dominant-hand grip	Motion sensor	71.53%
	Motion sensor + Keystroke	76.28%
Walking, Two-hand grip	Motion sensor	78.42%
	Motion sensor + Keystroke	83.24%

Keystroke data are generated only when the user's finger touches the screen, while motion sensor data are generated at a fixed sampling frequency, providing a comprehensive representation of the phone's motion state before, during, and after screen touches. Keystroke data records the touchpoint coordinates during typing, which can be influenced by thinking time, operational habits, and typing proficiency, leading to longer intervals between data points and limiting the representational capacity of touchscreen data for keystroke operations.

As indicated by the performance comparison results in Table 6, motion sensor data alone can achieve good authentication performance, and the integration of keystroke data results in a certain degree of performance improvement. This indicates that in authentication scenarios, motion sensors play a primary role in representing user identity information. Integrating keystroke data can slightly enhance the representation of user information, with keystroke data demonstrating a stronger user representation capability in walking scenarios.

LSTM models have a significant advantage when applied to datasets with strong temporal correlations. However, the length of the click data samples is relatively short, making it difficult for LSTM to play a role. The convolutional operation of a CNN can be used to extract relevant spatiotemporal features. Table 7 compares the performance of CNN, stacked LSTM, and CNN-LSTM models on the dataset. The CNN consists of three convolutional layers, with kernel sizes of 1×5, 1×5, and 1×3 for each layer. LSTM contains 64 hidden units and uses the tanh activation function and dropout to prevent overfitting. The CNN in the CNN-LSTM model consists of two convolutional layers, with kernel sizes of 1×5 and 1×5, respectively, and a batch normalization layer and a nonlinear activation layer follow each convolutional layer. The batch size is 32, and the initial learning rate is 1×10^{-4}. Under these training conditions, the authentication results shown in the table below are obtained. The CNN-LSTM model achieves a higher authentication accuracy than the other two models.

Table 7. Authentication performance comparison of feature extraction models.

User Posture	Model Type	Accuracy
Stationary, Dominant-hand grip	CNN	74.84%
	stacked LSTM	77.38%
	CNN-LSTM	81.64%
Stationary, Two-hand grip	CNN	81.24%
	stacked LSTM	83.75%
	CNN-LSTM	86.78%
Stationary, Device lying flat	CNN	91.59%
	stacked LSTM	93.75%
	CNN-LSTM	95.83%
Walking, Dominant-hand grip	CNN	68.88%
	stacked LSTM	71.42%
	CNN-LSTM	76.28%
Walking, Two-hand grip	CNN	75.69%
	stacked LSTM	79.12%
	CNN-LSTM	83.24%

The stacked LSTM shows significant performance differences between stationary and walking activity scenarios. In stationary scenarios, the variations in motion sensor data are solely due to user interactions, and the spatial and temporal features in the data have a strong discriminative power for user identity. In walking scenarios, the variations in motion sensor data are the combined result of changes in the user's movement state and user interactions. The changes caused by the movement state are more pronounced, weakening the ability to represent user operational characteristics. Thus, the temporal features extracted by the stacked LSTM have strong user identity representation capabilities in stationary scenarios but are limited in walking scenarios, leading to poor performance stability across different activity scenarios.

In walking scenarios, the deep features extracted from the keystroke data channel have better discriminative power for user identity. For keystroke data with shorter sample lengths and relatively weak temporal correlations, the spatial feature representation capability extracted by the CNN through convolution operations is stronger. Therefore, in walking scenarios, the CNN is chosen as the feature extraction model, while in stationary scenarios, the combination of the CNN and LSTM models achieves better authentication performance.

Comparison with Existing Methods. The authentication performance of implicit authentication methods based on operational features is closely related to sensor data, data preprocessing methods, feature extraction and classifier construction, the consideration of activity scenario changes, and the impact of holding postures. This paper compares recent research outcomes on authentication methods based on operational features from these aspects, as shown in Table 8.

In terms of data sources and feature extraction, some studies only use motion sensor data to reflect the characteristics of user interactions with the phone. The touchscreen, being that it is the sensor directly involved in user interactions, records time, location, and pressure characteristics of the operations. This paper integrates motion sensor and keystroke data, and extracts deep features through the construction of deep learning networks, enriching the input information for representing user identity.

Considering usage scenarios, the user's activity scenario and phone holding posture affect behavioral characteristics during phone operations. Some research works consider different activity scenarios during data collection but do not account for the impact of holding postures, thus not effectively mitigating the interference caused by scenario changes.

This paper constructs a two-stage authentication framework comprising activity scenario recognition and user identity classification, considering the effects of dominant-hand holding, two-hand holding, and the device lying flat on authentication performance.

Table 8. Comparison with existing authentication methods.

Method	Data Source	Classifier	Activity Scenario	Holding Posture	Experimental Performance
Reference [38]	Ac, Gy, Ma	Scale Manhatten	Stationary/Walking	-	EER: Stationary-10.05%, Walking-7.16%
Reference [39]	Ac	MLP	Walking	-	ACC: 99.11% EER: 1%
Reference [40]	Ac, Gy, Ma	CNN	Stationary/Walking	-	ACC: 97.8%
Reference [41]	Screen, Ac, Gy, Ma	CNN	Stationary/Walking	-	ACC: 88% EER: 15%
Ours	Screen, Ac, Gy	CNN-LSTM	Stationary/Walking	Dominant-hand grip/Two-hand grip/Device lying flat	ACC: 85.76%

5. Conclusions

This paper proposes an implicit identity authentication method based on user posture perception. The paper considers the impact of changes in user activity scenarios and differences in terminal holding postures. It constructs a staged cascaded authentication framework that first senses the user's posture and then authenticates the user's identity. First, a user posture perception model was designed using sensor data. Then, a two-channel user classification model was developed by fusing keystroke data, improving authentication accuracy, and enhancing the method's robustness. However, this research also faces certain limitations. First, the current data scale is limited, and more samples must be added to improve generalization ability. Second, there is room for improvement in authentication accuracy in walking scenarios. In the future, multimodal data such as pressure sensing and gait information can be combined to optimize model performance. In addition, this method is mainly aimed at smartphone devices. In the future, wearable and IoT applications can be explored to improve environmental adaptability.

From an application perspective, this method has essential potential in several fields. For example, bank authentication can be used in mobile banking applications to achieve contactless authentication and improve transaction security. It can be used in hospital access control in the medical industry to prevent unauthorized personnel from entering sensitive areas. In addition, in terms of Internet of Things (IoT) devices, this method can be combined with wearable devices to achieve smart home authentication and improve device security.

Author Contributions: B.H.: Conceptualization, methodology, software; S.T.: formal analysis, validation; F.H.: investigation, supervision; G.Y.: writing—original draft, writing—review editing; J.C.: data curation, supervision. 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: Author Shigang Tang was employed by the company Shenzhen Silver Star Intelligent Technology Co., Ltd. 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.

References

1. Khan, S.; Nauman, M.; Othman, A.T.; Musa, S. How Secure is Your Smartphone: An Analysis of Smartphone Security Mechanisms. In Proceedings of the 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), Kuala Lumpur, Malaysia, 26–28 June 2012; pp. 76–81.
2. Naji, Z.; Bouzidi, D. Deep learning approach for a dynamic swipe gestures based continuous authentication. In Proceedings of the 3rd International Conference on Artificial Intelligence and Computer Vision (AICV2023), Marrakesh, Morocco, 5–7 March 2023; Springer: Cham, Switzerland, 2023; pp. 48–57.
3. Stylios, I.; Chatzis, S.; Thanou, O.; Kokolakis, S. Continuous authentication with feature-level fusion of touch gestures and keystroke dynamics to solve security and usability issues. *Comput. Secur.* **2023**, *132*, 103363. [CrossRef]
4. Al-Saraireh, J.; AlJa'afreh, M.R. Keystroke and swipe biometrics fusion to enhance smartphones authentication. *Comput. Secur.* **2023**, *125*, 103022. [CrossRef]
5. Chao, J.; Hossain, M.S.; Lancor, L. Swipe gestures for user authentication in smartphones. *J. Inf. Secur. Appl.* **2023**, *74*, 103–450. [CrossRef]
6. Sejjari, A.; Moujahdi, C.; Assad, N.; Abdelfatteh, H. Dynamic authentication on mobile devices: Evaluating continuous identity verification through swiping gestures. *SIViP* **2024**, *18*, 9095–9103. [CrossRef]
7. Li, Z.; Han, W.; Xu, W. A Large-Scale Empirical Analysis of Chinese Web Passwords. In Proceedings of the 23rd USENIX Security Symposium (USENIX Security 14), San Diego, CA, USA, 20–22 August 2014; pp. 559–574.
8. Mazurek, M.L.; Komanduri, S.; Vidas, T.; Bauer, L.; Christin, N.; Cranor, L.F.; Kelley, P.G.; Shay, R.; Ur, B. Messuring Password Guessability for an Entire University. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, Berlin, Germany, 4–8 November 2013; pp. 173–186.
9. Yang, W.Q.; Wang, M.; Zou, S.H.; Peng, J.H.; Xu, G.S. An Implicit Identity Authentication Method Based on Deep Connected Attention CNN for Wild Environment. In Proceedings of the 2021 9th International Conference on Communications and Broadband Networking, Shanghai, China, 27 February 2021; pp. 94–100.
10. Kokal, S.; Vanamala, M.; Dave, R. Deep Learning and Machine Learning, Better Together Than Apart: A Review on Biometrics Mobile Authentication. *J. Cybersecur. Priv.* **2023**, *3*, 227–258. [CrossRef]
11. Sandhya, M.; Morampudi, M.K.; Pruthweraaj, I.; Garepally, P.S. Multi-instance cancelable iris authentication system using triplet loss for deep learning models. *Vis. Comput.* **2022**, *39*, 1571–1581. [CrossRef]
12. Zeroual, A.; Amroune, M.; Derdour, M.; Bentahar, A. Lightweight deep learning model to secure authentication in Mobile Cloud Computing. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 6938–6948. [CrossRef]
13. Cao, Q.; Xu, F.; Li, H. User Authentication by Gait Data from Smartphone Sensors Using Hybrid Deep Learning Network. *Mathematics* **2022**, *10*, 2283. [CrossRef]
14. Delgado-Santos, P.; Tolosana, R.; Guest, R.; Vera-Rodriguez, R.; Deravi, F.; Morales, A. GaitPrivacyON: Privacy-preserving mobile gait biometrics using unsupervised learning. *Pattern Recognit. Lett.* **2022**, *161*, 30–37. [CrossRef]
15. Stragapede, G.; Delgado-Santos, P.; Tolosana, R.; Vera-Rodriguez, R.; Guest, R.; Morales, A. Mobile Keystroke Biometrics Using Transformers. In Proceedings of the 2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG), Waikoloa Beach, HI, USA, 5–8 January 2023.
16. Zheng, Y.; Wang, S.; Chen, B. Multikernel correntropy based robust least squares one-class support vector machine. *Neurocomputing* **2023**, *545*, 126–324. [CrossRef]
17. Peralta, D.; Galar, M.; Triguero, I.; Paternain, D.; García, S.; Barrenechea, E.; Benítez, J.M.; Bustince, H.; Herrera, F. A Survey on Fingerprint Minutiae-based Local Matching for Verification and Identification: Taxonomy and Experimental Evaluation. *Inf. Sci.* **2015**, *315*, 67–87. [CrossRef]
18. Huang, P.; Gao, G.; Qian, C.; Yang, G.; Yang, Z. Fuzzy Linear Regression Discriminant Projection for Face Recognition. *IEEE Access* **2017**, *5*, 4340–4349. [CrossRef]
19. Dovydaitis, L.; Rasymas, T.; Rudžionis, V. Speaker Authentication System based on Voice Biometrics and Speech Recognition. In Proceedings of the International Conference on Business Information Systems, Leipzig, Germany, 6–8 July 2016; pp. 79–84.
20. Ignat, A.; Luca, M.; Ciobanu, A. Iris Features Using Dual Tree Complex Wavelet Transform in Texture Evaluation for Biometrical Identification. In Proceedings of the 2013 E-Health and Bioengineering Conference (EHB), Iasi, Romania, 21–23 November 2013; pp. 1–4.
21. Alzubaidi, A.; Kalita, J. Authentication of Smartphone Users Using Behavioral Biometrics. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1998–2026. [CrossRef]

22. Liu, J.; Yuan, Q.; Li, Y.; Lv, X. Recognition of Human Motion Based on Built-in Sensors of Android Smartphones. *J. Integr. Technol.* **2014**, *3*, 61–67.
23. McLoughlin, I.V. Keypress Biometrics for User Validation in Mobile Consumer Devices. In Proceedings of the 2009 IEEE 13th International Symposium on Consumer Electronics, Kyoto, Japan, 25–28 May 2009; pp. 280–284.
24. Wu, J.S.; Lin, W.C.; Lin, C.T.; Wei, T.E. Smartphone Continuous Authentication based on Keystroke and Gesture Profiling. In Proceedings of the 2015 International Carnahan Conference on Security Technology (ICCST), Taipei, Taiwan, 21–24 September 2015; pp. 191–197.
25. Buchoux, A.; Clarke, N.L. Deployment of Keystroke Analysis on a Smartphone. In Proceedings of the 6th Australian Information Security Management Conference, Perth, Australia, 1 December 2008; pp. 29–39.
26. Zahid, S.; Shahzad, M.; Khayam, S.A.; Farooq, M. Keystroke-based User Identification on Smart Phones. In Proceedings of the International Workshop on Recent Advances in Intrusion Detection, Saint-Malo, France, 23–25 September 2009; pp. 224–243.
27. Kambourakis, G.; Damopoulos, D.; Papamartzivanos, D.; Pavlidakis, E. Introducing Touchstroke: Keystroke-based Authentication System for Smartphones. *Secur. Commun. Netw.* **2016**, *9*, 542–554. [CrossRef]
28. Zhang, H.; Yan, C.; Zhao, P.; Wang, M. Model Construction and Authentication Algorithm of Virtual Keystroke Dynamics for Smartphone Users. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016; pp. 171–175.
29. Krishnamoorthy, S.; Rueda, L.; Saad, S.; Elmiligi, H. Identification of User Behavioral Biometrics for Authentication Using Keystroke Dynamics and Machine Learning. In Proceedings of the 2018 2nd International Conference on Biometric Engineering and Applications (ICBEA'18), Amsterdam, The Netherlands, 16–18 May 2018; pp. 50–57.
30. Zheng, N.; Bai, K.; Huang, H.; Wang, H. You Are How You Touch: User Verification on Smartphones via Tapping Behaviors. In Proceedings of the 2014 IEEE 22nd International Conference on Network Protocols, Raleigh, NC, USA, 21–24 October 2014; pp. 221–232.
31. Kang, P.; Cho, S. Keystroke Dynamics-based User Authentication Using Long and Free Text Strings from Various Input Devices. *Inf. Sci.* **2015**, *308*, 72–93. [CrossRef]
32. Kim, J.; Kim, H.; Kang, P. Keystroke Dynamics-based User Authentication Using Freely Typed Text based on User-Adaptive Feature Extraction and Novelty Detection. *Appl. Soft Comput.* **2018**, *62*, 1077–1087. [CrossRef]
33. Buschek, D.; De Luca, A.; Alt, F. Improving Accuracy, Applicability and Usability of Keystroke Biometrics on Mobile Touchscreen Devices. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, Seoul, Republic of Korea, 18–23 April 2015; pp. 1393–1402.
34. Giuffrida, C.; Majdanik, K.; Conti, M.; Bos, H. I Sensed It Was You: Authenticating Mobile Users with Sensor-Enhanced Keystroke Dynamics. In Proceedings of the 11th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Egham, UK, 10–11 July 2014; pp. 92–111.
35. Shankar, V.; Singh, K. An Intelligent Scheme for Continuous Authentication of Smartphone Using Deep Auto Encoder and Softmax Regression Model Easy for User Brain. *IEEE Access* **2019**, *7*, 48645–48654. [CrossRef]
36. Buriro, A.; Crispo, B.; Gupta, S.; Frari, F.D. Dialerauth: A Motion-Assisted Touch-based Smartphone User Authentication Scheme. In Proceedings of the eighth ACM Conference on Data and Application Security and Privacy, Tempe, AZ, USA, 19–21 March 2018; pp. 267–276.
37. Kim, J.; Kang, P. Freely Typed Keystroke Dynamics-Based User Authentication for Mobile Devices Based on Heterogeneous Features. *Pattern Recognit.* **2020**, *108*, 107556. [CrossRef]
38. Sitová, Z.; Šeděnka, J.; Yang, Q.; Peng, G.; Zhou, G.; Gasti, P.; Balagani, K.S. HMOG: New Behavioral Biometric Features for Continuous Authentication of Smartphone Users. *IEEE Trans. Inf. Forensics Secur.* **2015**, *11*, 877–892. [CrossRef]
39. Lamiche, I.; Bin, G.; Jing, Y.; Yu, Z.; Hadid, A. A Continuous Smartphone Authentication Method based on Gait Patterns and Keystroke Dynamics. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 4417–4430. [CrossRef]
40. Centeno, M.P.; Guan, Y.; Moorsel, A.V. Mobile based Continuous Authentication Using Deep Features. In Proceedings of the 2nd International Workshop on Embedded and Mobile Deep Learning, Munich, Germany, 15 June 2018; pp. 19–24.
41. Volaka, H.C.; Alptekin, G.; Basar, O.E.; Isbilen, M.; Incel, O.D. Towards Continuous Authentication on Mobile Phones Using Deep Learning Models. *Procedia Comput. Sci.* **2019**, *155*, 177–184. [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

Efficient Ensemble of Deep Neural Networks for Multimodal Punctuation Restoration and the Spontaneous Informal Speech Dataset

Homayoon Beigi [1,2,3,*] and Xing Yi Liu [4]

[1] Recognition Technologies, Inc., South Salem, NY 10590, USA
[2] Department of Mechanical Engineering, Columbia University, New York, NY 10027, USA
[3] Department of Electrical Engineering, Columbia University, New York, NY 10027, USA
[4] Cheriton School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada; xy39liu@uwaterloo.ca
* Correspondence: beigi@recotechnologies.com

Abstract: Punctuation restoration plays an essential role in the postprocessing procedure of automatic speech recognition, but model efficiency is a key requirement for this task. To that end, we present *EfficientPunct*, an ensemble method with a multimodal time-delay neural network that outperforms the current best model by 1.0 F1 point while using less than a tenth of its network parameters for inference. This work further streamlines a speech recognizer and a *BERT* implementation to efficiently output hidden layer acoustic embeddings and text embeddings in the context of punctuation restoration. Here, forced alignment and temporal convolutions are used to eliminate the need for attention-based fusion, greatly increasing computational efficiency and improving performance. *EfficientPunct* sets a new state of the art with an ensemble that weighs *BERT*'s purely language-based predictions slightly more than the multimodal network's predictions. Although *EfficientPunct* shows great promise, from a different perspective, to date, another important challenge in the field has been the fact that punctuation restoration models have been evaluated almost solely on well-structured, scripted corpora. However, real-world ASR systems and postprocessing pipelines typically apply to spontaneous speech with significant irregularities, stutters, and deviations from perfect grammar. To address this important discrepancy, we also introduce *SponSpeech*, a punctuation restoration dataset derived from informal speech sources, which includes punctuation and casing information. In addition to publicly releasing the dataset, the authors have contributed by providing a filtering pipeline that can be used to generate more data. This filtering pipeline examines the quality of both the speech audio and the transcription text. A challenging test set is also carefully constructed, aimed at evaluating the models' ability to leverage audio information to predict, otherwise grammatically ambiguous, punctuation. *SponSpeech* has been made available to the public, along with all code for dataset building and model runs.

Keywords: speech recognition; punctuation restoration; multimodal learning; punctuation restoration; speech recognition postprocessing; dataset; corpus

1. Introduction

Transcription of audio into text via automatic speech recognition (ASR) warrants a wide range of downstream postprocessing tasks. This transformation into natural language text enables applications such as machine translation and voice assistants. However, raw ASR outputs lack punctuation and hence do not convey the full meaning of the text.

Therefore, punctuation marks need to be restored, in order to facilitate the above and related tasks. To illustrate the importance of punctuation, consider how the meaning of the sentence, "I have a favorite, family." differs drastically from the unpunctuated version, "I have a favorite family". This example illustrates the importance of punctuation for the intelligibility of the transcribed speech and accuracy of conveyed message.

Following the standards of punctuation restoration, we focus on three key punctuation marks, which most commonly occur and play critical roles in the language, namely commas (,), full stops or periods (.), and question marks (?). We also consider no punctuation (NP) as a fourth class for the sake of completing the space.

1.1. Related Work

Many architectures have been proposed and are devoted to restoring the punctuation of ASR transcriptions. As such, two main research categories have emerged: (1) considering only text output from the ASR, and (2) considering both the text output of an ASR and the original audio.

Most proposed systems only consider text, effectively forming a natural language processing task. They usually train and evaluate on the benchmark textual datasets from *IWSLT 2011* and *2012*. Researchers have studied a wide variety of methods, including *n*-gram models [1], recurrent neural networks [2–4], adversarial models [5], contrastive learning [6], and transformers [7,8]. Conditional random fields [9–12] had particularly notable success. Direct fine-tuning of *BERT* models [13] has also proven effective, which we demonstrate in Section 4.1 as a constituent part of our multimodal approach.

In another approach, both the audio and text modalities are considered. Earlier techniques involved statistical models such as finite state machines [14], but unsurprisingly, more recently, we have seen the exploration of neural networks [15,16] and the re-purposing of existing models to take an audio-based input and predict the punctuation [17,18]. The current state-of-the-art models take on two separate paths: one to tokenize and process text and the other to process raw audio waveforms. They then use an attention mechanism [19] to fuse text and acoustic embeddings [20,21].

1.2. Significance of Multimodal Approach

Despite research in multimodal punctuation restoration being far less numerous than the text-only category, [17] explicitly demonstrated the value of added acoustic information. Intuitively, audio provides more diverse features available for models to learn [22]. As a simple example, long pauses in speech are definitive indicators of the presence of a full stop (.). Similarly, shorter pauses may indicate a comma (,), and rising pitch is often associated with question marks (?).

The substantial benefit of involving both the transcribed text and the original speech audio is that, in practical applications, one may design a highly streamlined system for restoring punctuation. Speech can first be transcribed into text through a forward pass of the audio signal through an ASR network. However, it is possible to preserve latent representation from an intermediate layer of the ASR network (a speech embedding) for further usage as an input, along with the text embeddings derived from the transcribed text, to a separate punctuation model. Then, the concatenated input would embed not only textual information, but also the acoustic content, including prosody.

The work presented here is precisely motivated by this potential for high-speed punctuation labeling after receiving the ASR output. We present *EfficientPunct* (Github page: https://github.com/GitHubAccountAnonymous/PR (accessed on 24 February 2025)), a model that surpasses state-of-the-art performance while requiring far fewer parameters, enabling practical and pragmatic implementation.

1.3. Database Issues

The field of punctuation restoration has made great progress recently, from proposals of recurrent neural networks [2] and end-to-end approaches [23] to adaptations of transformers [21] and time-delay neural networks [24]. However, an emerging problem is the singular type of datasets used for evaluating English punctuation restoration models.

As mentioned previously, in the English language domain, research projects on text-only punctuation models such as [6,8,25,26] almost always utilize the *IWSLT 2011* and *2012* datasets, both derived from the transcription of TED talks. Models derived from text and acoustics such as [21,23,24,27] have very often used the *MuST-C* dataset [28], which is also derived from TED talks. These models have steadily presented improvements in the performance and have indeed advanced the field. However, the dataset on which their insights are based is limited in scope.

TED talks are scripted monologues that are carefully practiced by the presenters before their formal delivery. Samples in the *IWSLT* and *MuST-C* datasets, therefore, exhibit high uniformity in style. Audio and transcriptions derived from this source exhibit a high quality, but they lack many natural traits of typical and spontaneous speech. For example, stutters, hesitations, and other spoken imperfections should optimally present themselves often in the punctuation restoration data, in order for trained models to be robust with respect to such disturbances. Although very few conversations between TED hosts and presenters are available, the *IWSLT* and *MuST-C* datasets predominantly feature flawless, formal monologues.

To address this limitation in the data, for thetraining and evaluation of punctuation restoration models, we have created the *SponSpeech* dataset (https://storage.googleapis.com/sponspeech/sponspeech.tar.gz (accessed on 24 February 2025)). The generation of this dataset has focused on extracting spontaneous, informal speech, which more accurately reflects natural conversations, rather than formal scripted speech that rarely occurs in daily life. In the process, podcasts were used as the source of the dataset, since they offer considerable stylistic diversity. They not only contain stints of discussion on focused subjects, but they also produce interactive dialogue among multiple speakers. Importantly, spoken imperfections are preserved in podcast conversations.

In addition to releasing *SponSpeech* as a punctuation restoration dataset of varied style, our contribution also provides another angle, namely the emphasis on utterances with punctuation ambiguity. A significant value of performing punctuation restoration is to improve readability of the text [29]. Partially, this involves eliminating ambiguity. For example, the text

i have eight boys

contains **punctuation ambiguity** , since "*I have eight boys*". suggests that the speaker has eight sons, whereas, "*I have eight, boys.*" suggests that the speaker is indicating to their boys that they possess eight objects. On the other hand, punctuation restoration has much less value in a text, such as,

i have an apple

as the only way to perform the punctuation is, "*I have an apple*". There is no punctuation ambiguity in the latter case.

In this dataset, we aim to include a significant number of utterances with punctuation ambiguity. In particular, we create two test sets, one with slightly more ambiguous cases than the other. Such a dataset will better evaluate the models' ability to leverage audio patterns to resolve punctuation ambiguities. As a simple example, a longer pause, in certain contexts, may distinguish the absence or presence of a comma.

1.4. Related Databases

To train punctuation restoration models, only the corpora which include ground-truth punctuation information are sought. This is particularly noteworthy because many ASR transcription datasets are case-insensitive and lack any punctuation information, and hence are not suitable for this purpose. In the case of text-only models, most printed text would be acceptable since punctuation marks are integral to printed text. However, this work focuses on the domain that considers both text and acoustic information. Our prior research has convincingly confirmed the added value of incorporating acoustics information [17], and ASR systems can certainly be effortlessly streamlined to output audio embeddings to be used for training punctuation prediction [24].

The *IWSLT* pure-text dataset has approximately 2.4 million words, which corresponds to a rough estimate of 267 h of "average" speech [30]. Table 1 presents some of the most prevalent datasets which contain both speech audio and punctuation information.

Table 1. Summary of speech datasets suitable for punctuation restoration. Fisher corpus refers to the punctuated section used by [20].

Dataset	Hours	Style
SponSpeech (ours)	665	Podcasts
Libriheavy [31]	56,389	Reading
NSC [32]	2170	Reading
LibriTTS [33]	585	Reading
MuST-C v1 [28]	438	Monologues
Fisher [34]	432	Telephone
Switchboard-1 [35]	260	Telephone

The largest dataset, Libriheavy, impressively contains over 50,000 h of audio and punctuation/casing information. A dataset of book recordings, its speech content has a scripted and uniformly flawless style. Similarly, NSC and LibriTTS involve readers verbalizing prior-written texts. As discussed, *MuST-C* comprises formal monologues in the form of TED talks.

Among the most popularly used datasets, only the Fisher and Switchboard-1 corpora feature spontaneous speech. Unfortunately, they require a paid membership to the Linguistic Data Consortium or a non-member licensing fee to access. We release a reasonably abundant 665 h of speech data with punctuated transcripts under a Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0) license, allowing all researchers to make use of our resource for free.

2. Methodology

The following describes our punctuation problem formulation. The spoken audio signal, $\mathbf{a} = (a_1, a_2, \ldots, a_S)$, and the transcribed words, $\mathbf{t} = (t_1, t_2, \ldots, t_W)$, are provided. Here, S represents the number of samples in the audio and W is the corresponding number of words. The goal is to predict the punctuation labels, $\mathbf{y} = (y_1, y_2, \ldots, y_W)$, that follow each word, for each $y_i \in \{\text{`,'}, \text{`.'}, \text{`?'}, \texttt{NP}\}$.

As illustrated in Figure 1, *EfficientPunct* begins in two branches which separately process the audio signal **a** and the transcription text **t**. Details are provided below.

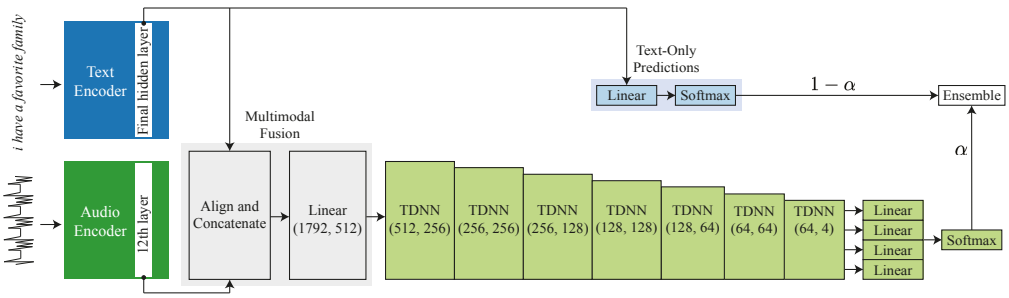


Figure 1. The *EfficientPunct* framework. The top branch predicts using text only, while the bottom branch predicts using text and audio.

2.1. Text Encoder

First, the text sequence, **t**, is passed through the default WordPiece tokenizer used by *BERT*. Then, using a pre-trained *BERT* model which we have fine-tuned for predicting the four previously described punctuation classes, we obtain the final hidden layer text embeddings,

$$H_t = \text{BERT}(\mathbf{t}). \tag{1}$$

H_t is a matrix whose columns are 768-dimensional vectors and represent token embeddings. These text embeddings contain each token's context-aware information about grammar and linguistics.

2.2. Audio Encoder

To process the raw spoken audio waveforms and to obtain meaningful acoustic embeddings, we use a pre-trained model, built using the *Kaldi* speech recognition toolkit [36]. This is directly analogous to previous works' usage of wav2vec 2.0 [37] as their pre-trained audio encoder. *Kaldi*'s TED-LIUM 3 [38] framework first extracts Mel frequency cepstral coefficients (MFCCs) [22], pitch features [39], and identity vectors (i-vectors), which are then passed to a time-delay neural network (*TDNN*) for speech recognition. We extract the embeddings from the 12th layer of the speech recognition model, to be used in the punctuation model,

$$H_a = \text{KaldiTedlium12}(\mathbf{a}). \tag{2}$$

H_a is a matrix whose columns are 1024-dimensional acoustic embedding vectors. The number of columns is equal to the number of frames in the original audio. The frames are 25 milliseconds long with a 10 millisecond shift, producing a 15ms overlap.

2.3. Alignment and Fusion

The first step in fusing the 768-dimensional textual embedding vectors, H_t, and the 1024-dimensional acoustic embedding vectors, H_a, is to find the correspondence between columns in each matrix. In other words, we must align the text token being spoken, with the corresponding frames of audio. This is performed through forced alignment. Depending on the alignment results, we concatenate the textual and acoustic embeddings related to the same frame, repeating the textual embedding for audio frames which correspond to the same token. This process results in 1792-dimensional embedding vectors corresponding to each frame. We further use a linear layer to learn affine transformations of embeddings which would be useful for punctuation restoration.

Many related works opt for attention-based fusion of the two modalities, but we have found forced alignment and a simple linear layer to be the most parameter-efficient and

competitive approach. Through experiments, we have determined that more sophisticated fusion methods were counterproductive.

2.4. Time-Delay Neural Network (TDNN)

Next, the fused embeddings are passed through a *time-delay neural network (TDNN)* [40]. It contains a series of $1 - D$ convolutional layers to capture temporal properties of the features, with a gradually decreasing number of channels. At the last convolution layer, there are 4 channels, with each channel corresponding to a punctuation class. The channels are passed through two linear layers with weights and biases shared among them to output 4 values for the softmax activation.

2.5. Ensemble Method

In completion of the *EfficientPunct*, we created an ensemble of the main *TDNN* and predictions using *BERT*'s text embeddings only. We pre-trained *BERT* using the dark-blue and light-blue modules in Figure 1, which can still be used at the inference time to obtain a set of predictions that only consider text, grammar, and linguistics. The second set of predictions obtained from the *TDNN* considered both text and audio.

Let us denote $\alpha \in [0, 1]$ to be the weight assigned to the *TDNN* predictions and $1 - \alpha$ to be the weight assigned to the *BERT* predictions. Our final predicted punctuation will then be

$$f(\mathbf{a}, \mathbf{t}, \alpha) = \arg\max[\alpha y_a + (1 - \alpha) y_t], \tag{3}$$

where y_a represents the *TDNN* softmax values and y_t represents the *BERT* softmax values. Essentially, if either of the *TDNN* or the *BERT* outputs a maximum class probability much lower than 1, then the other model may help resolve the ambiguity in predicting a punctuation mark.

3. Experiments with Public Datasets

3.1. Public Datasets

The primary public dataset used here was *MuST-C v1* [41], which is the same as the dataset used by UniPunc [21] for the sake of a fair comparison. This dataset was compiled using TED talks. We also use the same training and test set splits as the original authors, whose information is available on our GitHub page (Github Page: https://github.com/GitHubAccountAnonymous/PR (accessed on 24 February 2025)). We further split the original training set into 90% for training and 10% for validation. Please see Table 2 for full detailed information.

Table 2. Training, validation, and test set information.

Set	Number of Samples	Total Duration (h)
Training	92,723	392.0
Validation	10,301	43.5
Test	490	2.8

Each sample is an English audio piece of approximately 10 s to 30 s with the corresponding transcription. In the *Kaldi* ASR implementation, we use a frame duration of 25 ms with a shift of 10 ms for the 40-dimensional MFCC computation. In addition to the MFCC features, 3 pitch features, as described in [39], are prepended to each feature vector to account for prosodic variations, important to punctuation prediction. These pitch features include the NCCF-based probability of voicing, the mean-normalized log of pitch,

and the raw log value of the pitch [39]. We also compute 100-dimensional identity vectors (i-vectors) and append them to 3 consequently appended feature vectors, amounting to a 229-dimensional feature that feeds the TDNN. Then, the 12th layer of the neural network model is used to extract the acoustic embeddings. We follow the procedure described in Section 2.3 to generate a matrix of aligned embeddings for each data sample. Then, to obtain examples for training and inference, we consider segments of 301 frames, or 3 s, wherein the exact middle frame is the point of transition from one text token to the next. The resulting example will thus be labeled with the punctuation following the prior token, which occurs at the middle frame. We use a context window of 3 s. This duration should be sufficient to capture all the acoustic and prosodic information relevant to a punctuation mark, such as pauses and pitch variations. At the same time, this duration is not so long as to include much extraneous information, such as extensions into adjacent words.

For the entire dataset, punctuation label distributions are provided in the Table 3. Due to the highly imbalanced nature of the dataset, we sampled less prevalent classes more frequently for training such that, in effect, all class counts are equalized and the network avoids learning just the prior probability distribution.

Table 3. Punctuation label distributions.

Label	Number of Examples	% of Total
No punctuation (NP)	3,567,572	86.9%
Comma (,)	280,446	6.8%
Full stop (.)	238,213	5.8%
Question mark (?)	20,897	0.5%

Moreover, since *BERT* was already pre-trained on massive corpora, we fine-tune it for punctuation prediction using the National Speech Corpus [32] of Singaporean English, in addition to *MuST-C*.

3.2. Training

To fine-tune *BERT* and to pre-train the text encoder, we place two linear layers above the base network which is *BERT*'s last uncased hidden layer, geared for a 4-class classification problem. For the pre-trained audio encoder, we use the TED-LIUM 3 [38] framework, implemented in *Kaldi*.

Our main *TDNN* module for punctuation restoration comprises seven 1-dimensional convolution layers, with said dimension spanning across time. Figure 1 shows the number of input and output channels of each layer. The kernel sizes were as follows: 9, 9, 5, 5, 7, 7, and 5, alternating between no dilation and a dilation of 2. The stride was kept at 1 in all layers. Additionally, we applied the ReLU activation function and batch normalization [42] to the output of each layer. We trained the module using stochastic gradient descent [43] with a learning rate of 1×10^{-5} and a momentum of 0.9, instead of the typically used Adam optimizer [44]. This allowed for greater generalizability, but still provided reasonable training speeds [45].

To experiment with our ensemble architecture, we explored the effect of varying α, the weight assigned to the *TDNN* for final predictions. Recall that $1 - \alpha$ was the weight assigned to *BERT*. In Section 5.2, we report the results associated with a range of $\alpha = 0.3 \rightarrow \alpha = 0.7$ in 0.1 increments.

We used a standard Linux computing environment hosted on the Google Cloud Platform with a single NVIDIA Tesla P100 GPU. Training took roughly 2 days, and inference could be performed on CPU-only machines 50 times faster than real time, or in about 0.02 s per second of audio.

4. Creation of the SponSpeech Database

We have created and released the *SponSpeech* dataset with four standard subsets, in order to facilitate the reproducibility of future research in punctuation restoration. The four subsets are *train*, *dev* (validation), *test*, and *test-amb*. The last subset, *test-amb*, is an evaluation set with slightly more cases of punctuation ambiguity. We expect models to perform slightly worse on *test-amb* than on the ordinary *test* set, as resolving ambiguities presents an extra challenge. Special insights must be drawn from acoustic information when multiple versions of punctuation are possible in a grammatical sense. At the same time, we do not desire for *test-amb* to drastically differ from utterance population norms. A dataset and its subsets should accurately reflect the state of the standard distribution of nature [46], a fact that we strived to balance, through creating a challenging, second evaluation set.

Table 4 lists the relevant statistics on each subset. We source our data from YouTube videos published under the *Creative Commons Attribution 4.0 International License*, rather than standard YouTube licenses. This respects the rights of all uploaders, while allowing our creation of *SponSpeech* as an adapted work, with the appropriate attribution. In Table 4, the fourth column reports the number of videos involved in each subset. Videos are sliced to create utterances.

The *train*, *dev*, *test*, and *test-amb* sets make up approximately 70%, 12%, 9%, and 9% of the entire dataset, respectively. The overall minimum, average, and maximum utterance durations are 1.6s, 11.6s, and 45.0s, respectively.

Table 4. Statistics on each subset of *SponSpeech*.

Subset	Hours	Utterances	Videos
train	469	147,209	1736
dev	79	25,253	277
test	58	17,697	205
test-amb	60	16,473	397

The following subsections describe the dataset creation process.

4.1. Data Source

Each YouTube video has a unique video ID. To obtain an initial pool of YouTube video IDs, we specify a filter code in the search query URL (in this case, `sp=EgQoATAB`) that only allows videos with a Creative Commons license and subtitles to be returned. The resulting list of videos merely serves as candidates for inclusion in *SponSpeech*, with careful evaluation still needed for each video to ensure that the criteria for desirable content are met. Then, a series of five filters are applied to eliminate any unacceptable candidates.

We use the `yt-dlp` tool to download the videos, their subtitles, and audio content, as well as any metadata.

4.2. Filters

Figure 2 provides an illustration of the filtering pipeline. Each filter acts on the pool of candidate video IDs. Since text can be more computationally efficient to analyze than audio, the first three filters use text as the basis of their evaluation. The fourth and fifth filters evaluate the audio.

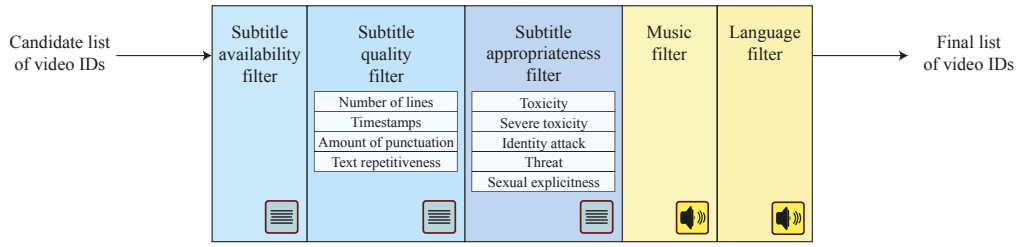


Figure 2. Filtering pipeline used to create *SponSpeech*. Blue indicates a text-based filter, and yellow indicates an audio-based filter, also shown by the bottom-right icons. The evaluation criteria are represented by the sub-boxes within the subtitle, quality, and appropriateness filters.

In the first stage, the **subtitle availability filter** simply detects whether manually uploaded English transcriptions exist for each candidate video. YouTube's system automatically generates subtitles. However, in producing *SponSpeech*, we are not interested in automatically generated transcriptions. Hence, the filter requires the content creators to have provided human generated transcripts to be used as the ground-truth text.

At the second stage of the data selection, the **subtitle quality filter** assesses four criteria for evaluating the content of the subtitles. Subtitle files follow a standard format, with a header block of metadata to begin. This is followed by roughly alternating lines of timestamp ranges and the corresponding text. By default, subtitle files also provide accurate alignments to serve as the basis for the extraction of individual utterances. These timestamps are either provided manually by the uploader alongside the transcription, or by YouTube's alignment algorithm, depending on the uploader's choice.

1. **Number of lines:** Each video subtitle file is required to have at least 20 lines. This is a heuristic selected on the basis of manual examination and erring on the side of caution. The subtitle file has a header block with the metadata. As such, videos with substantial content, at the very minimum, have 20 lines of subtitles.
2. **Timestamp formatting:** The subtitle files must contain timestamps that follow a standard format, common to the vast majority of videos. Unfortunately, a small number of files have highly irregular formatting, such as those with mistakenly littered HTML tags. These are difficult to consistently interpret and are hence filtered out.
3. **Amount of punctuation:** Some subtitle files are poorly written and have very few punctuation marks. This presents a problem for our dataset, which is designed for punctuation restoration. As a result, this criterion checks for a minimum of one punctuation mark every ten subtitle lines.
4. **Text repetitiveness:** Some subtitle files have a considerable amount of overlapping text between different lines. In other words, the same phrase may appear repetitively in error, with the corresponding audio not featuring any repetition. Every line has a distinct corresponding timestamp range; the same text appearing on multiple lines may indicate alignment issues. Videos with textual repetitiveness of this type (and not a truly repetitive nature) are filtered out.

The third stage is the **subtitle appropriateness filter**, which is designed to eliminate videos with inappropriate language. The types of inappropriate language targeted include toxicity, severe toxicity, identity attack, threat, and sexual explicitness. While these samples do exist in the natural population of speech data, we do not wish for such language to be further propagated through a public dataset. We pass each subtitle file through the `Detoxify` model, available in Python's `detoxify` library [47,48]. The probability of inappropriateness is computed and compared to a threshold to determine the videos to be excluded.

At the fourth stage of the process, a **music filter** is used to ensure that only speech content is accepted into the dataset. Even though the search keyword, "podcast", was applied to obtain the candidate videos, YouTube contains both speech-dominant and music-dominant content. We use a *wav2vec 2.0*-based speech/music classifier to determine the amount of speech, music, and simultaneous content in each video [37,49]. Those files with more music than speech, measured based on a probability heuristic, were filtered out. Of course not all the videos with music were eliminated because, for example, speech overlaid with quiet music has significant validity as an utterance to train for punctuation restoration.

The fifth and last stage is the **language filter**, which allows for only English content to pass through, as this dataset concentrates on English content. A *Whisper*-based language identification model evaluates the speech from each video [50,51]. In future versions, we will be addressing other languages and multilingual conditions.

Community-trained machine learning models were also used with caution. Only those models with convincing test set results and those fine-tuned from established foundation models were used. Any performance drawbacks due to these models could be compensated by using highly cautious threshold values for filtering decisions. Threshold selection (including in sub-filter criteria) were chosen quite conservatively, in order to favor eliminating unproblematic videos, rather than failing to eliminate problematic ones. The aim is to maximize recall and somewhat overlook precision in the context of a positive class representing videos to be eliminated. The rationale is that an abundance of candidate videos would allow for a more selective process.

Videos that pass all the filters have made the basis of the presented *SponSpeech* dataset.

4.3. Utterance Creation

Each unit of the data sample in *SponSpeech* is an utterance. For each video that survived all the described filters, select sentence delimiters were used to split the video into utterances. As it has become standard in punctuation restoration research to consider only the dominant punctuation marks full stop (.), comma (,), and question mark (?), we use full stops and question marks as partition boundaries.

Standard ASR datasets use terminal punctuation marks as natural endings for utterances in the dataset [31]. However, examples with full stops and question marks are also made available within some utterances, so that punctuation restoration models may be trained robustly on the presented *SponSpeech* dataset.

The utterance creation step provides a pool of utterances with audio extracted using the YouTube-provided alignment information. The results are utterances distributed in the `wav` format with single-channel linear PCM audio at a 16 kHz sampling rate. The collection is split into the described four data subsets: *train*, *dev*, *test*, and *test-amb*. As previously mentioned, *test-amb* is a special test set with an increased number of utterances portraying high punctuation ambiguity. The following subsection describes the creation of the *test-amb* portion of the dataset.

4.4. Creating the Ambiguous Test Set

To create the *test-amb* portion of the dataset, the main consideration involved the determination of whether the text of each utterance contained some form of punctuation ambiguity, effectively forming a binary classification task. The null hypothesis [22] is "punctuation ambiguity exists", with the alternative hypothesis [22] being "no punctuation ambiguity". For a description and an example of punctuation ambiguity, please see Section 1.

The framework with which we determine punctuation ambiguity is *ELECTRA* [52], fine-tuned on a small number of samples. To accomplish this, we manually labeled 100 samples for training and 20 samples for validation, with a few examples shown in Table 5. For each unpunctuated textual utterance input, "label A" is associated with the list of all correct punctuation variations, separated by newline characters. "Label B" denotes binary information, with 0 representing "no punctuation ambiguity" (i.e., only one item in label A) and 1 representing "punctuation ambiguity" (i.e., more than one item exists in label A).

Table 5. Data samples used to fine-tune *ELECTRA* for creating *test-amb*. Label A is manually created and lists all correct versions of punctuation for the input. Label B is binary and indicates whether punctuation ambiguity exists.

Input	Label A	Label B
and i liked the fact…	And I liked the fact…	0
…go there but you...	…go there, but you… …go there. But you…	1
…then education is…	…then education is… …then, education is…	1

ELECTRA utilizes a generator–discriminator architecture. We leverage both components for binary classification as follows:

1. **Generator tuning for text generation.** The unpunctuated textual utterance input and label A are used to fine-tune *ELECTRA*'s generator for text generation. The model is prompted to generate all correct versions of punctuation based on the input text. This step's task is more complex than ultimately needed, since only a final binary prediction is used to create *test-amb*. However, this precursor step is designed to guide the foundation model toward reasoning about the punctuation. Directly predicting label B without this step may obfuscate whether the task relates to punctuation at all.
2. **Generator for ambiguity classification.** After performing step 1, the generator is used to obtain the last hidden layer embeddings for the input text. Along with label B, these embeddings are used to train a multilayer perceptron (MLP) for binary classification. The results of this step—predictions of whether utterances contain punctuation ambiguity—are directly used in the creation of *test-amb*.
3. **Discriminator for ambiguity classification.** Similar to step 2, the discriminator's embeddings from the last hidden layer for the input text and label B are used to train an MLP for binary classification. Since the discriminator is not designed for text generation, its usage does not involve the precursor fine-tuning step as it did in the generator.

In accordance with an optimal performance evaluation on the validation set, a weight of $\alpha = 0.8$ is used for the generator's sigmoid output probabilities, and hence, a weight of $1 - \alpha = 0.2$ is assigned to the discriminator. As expected, the generator, tuned with the precursor generation step, is more useful. The weighted sum of the probability values are used for the final predictions of punctuation ambiguity on all *SponSpeech* utterances. A threshold of $\beta = 0.6$ was similarly selected based on the validation results, with prediction probabilities $p \geq \beta$ being classified as "having punctuation ambiguity".

A formal evaluation of the fine-tuned models was not deemed necessary, since the generator, the discriminator, and the MLPs only needed to give a general suggestion as to which utterances were likely to have had punctuation ambiguity. Unlike typical machine learning tasks, maximizing accuracy was less important, since classification mistakes do not deter the overall formation of *test-amb*. Moreover, any additional samples labeled for a

potential test set would, anyway, much better serve to enlarge the scant fine-tuning and validation data subsets.

In the end, the *test-amb* set was created with a greater number of utterances predicted to have punctuation ambiguity. The effectiveness of this procedure is proven in the results presented below.

5. Results

The results reported in Table 6 include a comparison of our different models with the current state-of-the-art (SOTA) and best-performing models, MuSe [20] and UniPunc [21]. The reporting of our *EfficientPunct*'s results is given in three categories:

1. *EfficientPunct-BERT* considers text only, which is equivalent to the fine-tuned *BERT* model.
2. *EfficientPunct-TDNN* considers text and audio via our *TDNN*.
3. *EfficientPunct* is an ensemble of predictions from categories (1) and (2) with $\alpha = 0.4$, the best-performing weight, as reported in Section 5.2.

Categories (1), (2), and (3) are reported in the third, fourth, and fifth rows of Table 6, respectively.

As is standard in punctuation restoration research, we report the F1 scores of commas, full stops (periods), and question marks. The "overall" F1 score aggregates all classes, while we considered varying the number of examples to account for the class-count imbalance. The number of parameters for each model is also provided, which is indicative of the computational efficiency of the method of interest.

Table 6. F1 scores of *EfficientPunct* and its various submodules on each punctuation type, compared against existing state-of-the-art (SOTA) models. *EfficientPunct-BERT* considers text only, *EfficientPunct-TDNN* considers text and audio, and *EfficientPunct* predicts using an ensemble of *Efficient-BERT* and *EfficientPunct-TDNN*.

	Model	Embedding Type(s) Used	,	.	?	Overall	No. of Params.
SOTA	MuSe [1,2]	BERT,wav2vec 2.0	73.2	83.6	79.4	77.9	1.7×10^8
	UniPunc [1]	BERT,wav2vec 2.0	74.2	83.7	80.8	78.5	2.5×10^8
Ours	EfficientPunct-BERT	BERT	73.4	83.9	84.7	78.4	1.1×10^8
	EfficientPunct-TDNN	BERT,TEDLIUM3	74.3	83.6	85.8	78.5	1.2×10^8
	EfficientPunct (Ensemble)	BERT,TEDLIUM3	**75.4**	**84.3**	**86.5**	**79.5**	1.2×10^8

[1] Statistics taken directly from the UniPunc paper due to the public inaccessibility of certain models hindering our ability to run them. Fairness of comparison is ensured, since we use the exact same training and test sets as the UniPunc authors. [2] The number of parameters in MuSe was conservatively estimated from information provided in the original paper.

5.1. EfficientPunct and Submodules

Our main *EfficientPunct* model achieves an overall F1 score of 79.5, outperforming all current state-of-the-art frameworks by 1.0 or more points. We also achieve the highest F1 scores for each individual punctuation mark, with the most significant improvement occurring for question marks. These results were accomplished with *EfficientPunct* using less than half of *UniPunc*'s total number of parameters, which achieved the previously reported best results on the same data. This significant improvement in recognizing question marks may be attributed to our audio encoder, *Kaldi*'s TED-LIUM 3 framework, aiming explicitly at phone recognition, as we had expected. In this process, the acoustics surrounding question marks may be more pronounced in the embedding representation than other acoustics models.

Our *EfficientPunct-BERT* and *EfficientPunct-TDNN* models are even more lightweight. *EfficientPunct-BERT* is simply a concatenation of two linear layers and a softmax layer on top of *BERT*. With the incorporation of audio features, we observe that *EfficientPunct-TDNN* indeed performs slightly better.

These results validate the strength of the *TDNN* models for punctuation restoration, traditionally used in speech and speaker recognition. *UniPunc* and *MuSe* both use attention-based mechanisms for fusing text and acoustic embeddings, but alignments learned as such rely on trainable attention weights. Our forced alignment strategy likely generated more precise temporal matching between text and audio. Combined with a *TDNN* architecture, we achieved a significantly more efficient model.

5.2. Ensemble Weights

In this section, we observe the effect of ensemble weights on the *EfficientPunct*'s performance. Equation (3) details the role of α in weighing the predictions made by the *TDNN* and *BERT* models, with $\alpha = 0$ indicating only *BERT* predictions, and $\alpha = 1$ indicating only those of the *TDNN* model.

Table 7 reports the effect of α on the model performance. When both *BERT* and *TDNN* models play an approximately equal role in the ensemble, a fair voting mechanism is enabled, and the highest F1 score is achieved. However, a weight of $\alpha = 0.4$, which considers the *BERT* predictions slightly more strongly than those of the *TDNN*, achieves the maximum overall F1 score. This gain comes mostly from better comma predictions, which are notoriously difficult due to varying grammar and styles of composition. We believe that the superiority of the $\alpha = 0.4$ case might be attributed to a stronger reliance on *BERT*'s language modeling perspective, yielding more linguistically adequate punctuation, as agreed upon by countless composition contributions to *BERT*'s large training corpus.

Table 7. F1 scores for different α weights.

α	Comma	Full Stop	Question	Overall
0.3	75.0	84.1	86.3	79.2
0.4	75.4	84.3	86.5	**79.5**
0.5	75.0	84.0	86.5	79.1
0.6	75.0	83.8	86.2	79.0
0.7	74.8	83.8	85.8	78.9

The strength of our ensemble strategy is that, in cases of uncertain predictions by either model (such as approximately equal softmax probabilities over all classes), the other model can provide guidance in clarifying the ambiguity. As indicated by the last two rows of Table 6, this process demands very few additional parameters applied on the input, but greatly improves the state-of-the-art performance.

5.3. Parameter Breakdown

In order to show the specific modules in which we attain superior efficiency, we further break down the parameters count from the last column of Table 6. Table 8 provides details of the number of parameters used by each model for extracting embeddings and inferring those embeddings to make the punctuation decisions.

Our *EfficientPunct* requires much less computational cost in both embedding extraction and inference stages. Our usage of *Kaldi*'s TED-LIUM 3 model brought massive efficiency gains compared to *MuSe* and *UniPunc*'s utilization of the more costly wav2vec 2.0. Moreover, our inference module uses *less than a tenth* of those of *UniPunc* (which had achieved the previous state of the art) at the same stage.

Table 8. Number of parameters required in various stages of each model.

Model	Embedding Network	Inference Network	Total
MuSe	1.6×10^8	4.3×10^6	1.7×10^8
UniPunc	2.0×10^8	4.8×10^7	2.5×10^8
EfficientPunct	$\mathbf{1.1 \times 10^8}$	$\mathbf{3.0 \times 10^6}$	$\mathbf{1.2 \times 10^8}$

5.4. SponSpeech Evaluation

To evaluate the quality of our *SponSpeech*, we follow a procedure similar to that used for *MuST-C v2* [41]. An indication of the dataset's quality is achieving reasonable results when cross-testing with another dataset. This is in the following forms: (1) training on another dataset and testing on *SponSpeech*, and (2) training on *SponSpeech* and testing on another dataset. We use *MuST-C* for this purpose, since it is currently the most frequently used multimodal dataset (with text and acoustics) for punctuation restoration. The models used were *BERT* [13] and *EfficientPunct*, which are among the recent top-performing models in the text-only and multimodal domains.

As shown in Table 9, both models trained on *MuST-C* achieve overall F1 scores of around or above 70% when tested on our *SponSpeech* dataset. By comparison against the *SponSpeech* dataset results in Tables 10 and 11, *MuST-C*-trained and *SponSpeech*-tested results appear very reasonable. Namely, not training on the *SponSpeech* population itself only decreases the overall F1 score on the *SponSpeech test* and *test-amb* sets by about 2.3% to 4.0%. The high quality of the SponSpeech *test* and *test-amb* sets is hence confirmed. As expected, results for the *test-amb* set are lower than for the *test* set, demonstrating the effectiveness of our *ELECTRA* tuning process in creating a *test* set with more *punctuation ambiguity*.

The results of training on *SponSpeech* and testing on *MuST-C* are shown in Table 12 and they indicate that the overall F1 scores are well above 70%. This demonstrates the quality of our *SponSpeech* training and validation sets. Together, the two sets of cross-testing results show that *SponSpeech* is of very high quality and is very robust.

Table 9. Precision (P), recall (R), and F1 score (F1) achieved by models trained on *MuST-C* and tested on *SponSpeech*.

Subset	Model	Full Stop			Comma			Question			Overall		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
test	EfficientPunct	76.9	84.4	80.5	54.6	76.1	63.6	70.5	79.77	4.8	63.2	79.7	70.5
	BERT	75.5	85.3	80.1	60.7	69.3	64.7	69.4	80.0	74.3	67.1	76.3	71.4
test-amb	EfficientPunct	76.6	80.6	78.6	55.1	76.2	63.9	67.4	77.7	72.2	63.1	78.1	69.8
	BERT	75.9	81.7	78.7	61.0	69.6	65.0	66.9	79.2	72.6	67.2	75.0	70.9

Table 10. Precision (P), recall (R), and F1 score (F1) achieved by models on *SponSpeech*'s test set.

Model	Full Stop			Comma			Question			Overall		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
EfficientPunct	79.3	85.6	82.3	56.9	82.6	67.4	74.0	80.4	**77.0**	65.2	83.8	73.3
UniPunc	70.5	83.2	76.3	46.9	75.9	58.0	71.8	71.2	71.5	55.8	78.7	65.3
BERT	82.4	82.3	**82.4**	66.5	75.0	**70.5**	76.1	77.6	76.8	73.0	78.1	**75.4**
GPT-4 Turbo	87.0	70.0	77.5	47.7	85.0	61.1	67.5	81.9	74.0	58.0	78.7	66.8

Table 11. Precision (P), recall (R), and F1 score (F1) achieved by models on *SponSpeech*'s test-amb set.

Model	Full Stop			Comma			Question			Overall		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
EfficientPunct	78.0	82.9	**80.4**	56.5	81.6	66.7	72.1	78.8	**75.3**	64.4	82.0	72.1
UniPunc	70.6	79.8	74.9	47.0	74.5	57.7	69.9	70.0	70.0	55.7	76.5	64.5
BERT	81.3	79.2	80.3	65.5	73.5	**69.3**	74.6	75.5	75.0	71.9	75.9	**73.8**
GPT-4 Turbo	85.5	66.8	75.0	48.8	84.3	61.8	65.8	80.0	72.2	58.4	77.0	66.4

Table 12. Precision (P), recall (R), and F1 score (F1) achieved by models trained on *SponSpeech* and tested on *MuST-C*. The test set used is the same as in [21,24].

Model	Full Stop			Comma			Question			Overall		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
EfficientPunct	79.3	81.8	80.5	64.4	76.7	70.0	82.9	77.6	80.1	70.7	78.9	74.5
BERT	82.0	76.7	79.3	69.1	65.7	67.3	85.3	73.4	78.9	75.0	70.5	72.7

5.5. SponSpeech Baselines

To initiate punctuation restoration research using *SponSpeech*, we provide several notable models for the baseline results, shown in Table 10 for the *test* set and Table 11 for the *test-amb* set. Training and validation are performed using the *SponSpeech* standard data subsets. The chosen models are as follows:

- **EfficientPunct**, an ensemble with a time-delay neural network to process *BERT* [13]- and *Kaldi*-derived [36] embeddings, picked for its state-of-the-art performance and efficiency. It considers text and acoustics [24].
- **UniPunc**, an attention-based architecture with a coordinate bootstrapper that allows for missing audio. We apply the same *BERT* and *Kaldi* embeddings as *EfficientPunct* for fair comparison of the architectures. It considers text and acoustics [21].
- **BERT**, a bidirectional transformer-based foundation language model that can be fine-tuned for a wide variety of downstream tasks with very few additional layers. It considers text only [13].
- **GPT-4 Turbo**, a generative pre-trained transformer large language model created by OpenAI, picked due to its immense popularity in the AI community and the public through ChatGPT and its origin model, GPT-4. It considers text only [53]. We evaluate the zero-shot performance of this model.

The best performance for all types of punctuation marks in both evaluation sets is dominated by *BERT* and *EfficientPunct*.

For the *test* set, *BERT* achieves the highest F1 scores for *full stops (periods)*, *commas*, and overall, while *EfficientPunct* achieves the highest performance for *question marks*. *BERT*'s language modeling capabilities for punctuation restoration outmatch all other models. However, *EfficientPunct*'s outperformance on question marks may be attributed to the consideration of acoustics, which can often signal question marks with rising pitch, as we had expected at the onset of our research. *BERT* may suffer from the lack of pitch/acoustic information in this case.

For the *test-amb* set, *BERT* achieved the highest F1 scores for *commas* and overall, while *EfficientPunct* achieved the highest for *full stops (periods)* and *question marks*. All models performed worse on the *test-amb* set compared to the the *test* set, as expected.

An important finding from the baseline results is that although the multimodal models proposed thus far fall short of *BERT* in overall performance, experiments on *MuST-C*'s highly structured TED talks have repeatedly shown the proposed models' outperformance

of *BERT*. This highlights the importance of testing on diverse datasets, warranting the creation of our *SponSpeech* dataset and the ensemble versions of our *EfficientPunct* models.

6. Conclusions

We explored the application of time-delay neural networks to punctuation restoration, which proved to be more computationally efficient than previous approaches, yet as effective. Combined with *BERT* in an ensemble, *EfficientPunct* establishes a strong, new state of the art with a fraction of the parameters used by the previous state of the art. A key factor of the success of our models is the removal of the need for attention-based fusion of text and audio features. In previous approaches, multiple attention heads added extraordinary overhead at the punctuation prediction (inference) stage. We demonstrated that forced alignment of text and acoustic embeddings, in conjunction with temporal convolutions, rendered attention unnecessary.

EfficientPunct requires a far lower number of parameters and hence enables the post-processing of punctuation to be performed in real time on limited-resource and edge devices. Also, the postprocessing of an ASR result should not require more effort than the ASR itself. The significant reduction in the number of parameters required for inference for *EfficientPunct* proves to be significantly smaller than the effort in performing the ASR.

Additionally, we studied the effect of different weightings of constituent models in an ensemble architecture. We found that a slightly stronger weighting of *BERT* models against the multimodal *TDNN* model optimized performance by emphasizing language rules associated with punctuation. The ensemble technique, however, allowed for a tuning parameter (the weight) to handle different domains, depending on the importance of the acoustics vs. the stylistic and grammatical content.

To deal with the database inadequacies, *SponSpeech* (https://storage.googleapis.com/sponspeech/sponspeech.tar.gz (accessed on 24 February 2025)) was created and made available. This is a new dataset for punctuation restoration with spontaneous speech in the form of podcast conversations. Many ASR datasets, especially scripted ones, lack natural tendencies in speech like stutters, random pauses, grammatical imperfections, and interruptions among multiple speakers. We remedy this dearth of important speaking characteristics in datasets by contributing *SponSpeech* as a public, free resource, licensed under *CC BY-NC 4.0*.

Several considerations are made for future work. One is examination of the effectiveness of jointly training the ensemble weights and the *TDNN*, which could allow for the learning of an optimal ensemble. Jointly training with the text and audio encoders may also be considered, but this procedure should not inhibit the generalizability of the encoders for purposes other than punctuation restoration. Another very important provision would be to explore the applicability of *EfficientPunct* (Github Page: https://github.com/GitHubAccountAnonymous/PR (accessed on 24 February 2025)) in more languages and a similar framework for other postprocessing tasks of speech recognition. This will include special punctuation provisions in languages such as Spanish where exclamation and interrogatory phrases are preceded as well as succeeded by special opening and closing marks. Eventually, we would like to expand multilingually and try to adapt our techniques, as well the dataset to also include code-switching scenarios. Finally, we would also like to increase the number of supported punctuation marks to include exclamation, semicolon, color, quotation marks, etc.

Further, with regard to data adequacy, it should be noted that certain punctuation decisions are stylistic, and oftentimes, two or more ways of punctuating are readable, reasonable, and grammatically correct, all without altering the intended meaning. Such subjectivities slightly influence punctuation labels and, in turn, the evaluation results.

Measures of punctuation accuracy are hence coarser than one may initially assume. Also affected is our labeling of the training and validation set for *ELECTRA* fine-tuning. Especially in the cases of spontaneous speech, when grammatical rules are defied, flawless and objective punctuation labeling proves even more difficult. Our classification of whether select utterances are ambiguous may somewhat suffer from this linguistic phenomenon.

As for potential future work in relation to the dataset, multimodal model performance falling short of *BERT* on *SponSpeech* in certain categories serves as a strong motivation for further research in multimodal punctuation restoration models. It has been repeatedly shown that multimodal approaches have significant advantages over text-only models [17]. We can, therefore, conclude that there exist effective, undiscovered methods of considering acoustics for punctuation restoration that would enhance results in all punctuation categories. The most likely candidates involve heavily weighing the grammatical insights of language models while performing selective acoustic leveraging to extract fine-grained information.

Author Contributions: Conceptualization, H.B. and X.Y.L.; methodology, H.B. and X.Y.L.; data curation, H.B. and X.Y.L.; formal analysis, H.B. and X.Y.L.; project administration, H.B.; supervision, H.B.; software, X.Y.L. and H.B.; validation, X.Y.L. and H.B.; writing—original draft, X.Y.L. and H.B.; writing—review, revision, editing, H.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Code and data have been made available through public access links and the links have been provided in the manuscript.

Conflicts of Interest: Homayoon Beigi was employed by Recognition Technologies, Inc. and Columbia University and his work was conducted in his capacity at Recognition Technologies, Inc. Xing Yi Liu was a doctoral student at the University of Waterloo. This work is published with express permission of Recognition Technologies, Inc. There is no conflict of interest with Columbia University and University of Waterloo.

References

1. Gravano, A.; Jansche, M.; Bacchiani, M. Restoring punctuation and capitalization in transcribed speech. In Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009; pp. 4741–4744.
2. Tilk, O.; Alumäe, T. Bidirectional Recurrent Neural Network with Attention Mechanism for Punctuation Restoration. In Proceedings of the Interspeech, San Francisco, CA, USA, 8–12 September 2016; pp. 3047–3051.
3. Kim, S. Deep Recurrent Neural Networks with Layer-wise Multi-head Attentions for Punctuation Restoration. In Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing, Brighton, UK, 12–17 May 2019; pp. 7280–7284. [CrossRef]
4. Salloum, W.; Finley, G.; Edwards, E.; Miller, M.; Suendermann-Oeft, D. Deep learning for punctuation restoration in medical reports. In Proceedings of the BioNLP 2017, Vancouver, BC, Canada, 4 August 2017; pp. 159–164.
5. Wang, W.; Liu, Y.; Jiang, W.; Ren, Y. Making Punctuation Restoration Robust with Disfluency Detection. In Proceedings of the 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design, Hangzhou, China, 4–6 May 2022; pp. 395–399.
6. Huang, Q.; Ko, T.; Tang, H.L.; Liu, X.; Wu, B. Token-level supervised contrastive learning for punctuation restoration. In Proceedings of the Interspeech, Brno, Czechia, 30 August–3 September 2021; pp. 2012–2016.
7. Courtland, M.; Faulkner, A.; McElvain, G. Efficient automatic punctuation restoration using bidirectional transformers with robust inference. In Proceedings of the 17th International Conference on Spoken Language Translation, Online, 9–10 July 2020; pp. 272–279.
8. Alam, T.; Khan, A.; Alam, F. Punctuation restoration using transformer models for high-and low-resource languages. In Proceedings of the 2020 EMNLP Workshop W-NUT: The Sixth Workshop on Noisy User-generated Text, Online, 19 November 2020; pp. 132–142.
9. Lu, W.; Ng, H.T. Better punctuation prediction with dynamic conditional random fields. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, MA, USA, 9–11 October 2010; pp. 177–186.

10. Nguyen, T.B.; Nguyen, Q.M.; Nguyen, T.T.H.; Do, Q.T.; Luong, C.M. Improving Vietnamese Named Entity Recognition from Speech Using Word Capitalization and Punctuation Recovery Models. In Proceedings of the Interspeech, Shanghai, China, 25–29 October 2020; pp. 4263–4267.
11. Uyen, H.T.T.; Tu, N.A.; Huy, T.D. Vietnamese Capitalization and Punctuation Recovery Models. In Proceedings of the Interspeech, Incheon, Republic of Korea, 18–22 September 2022; pp. 3884–3888.
12. Ueffing, N.; Bisani, M.; Vozila, P. Improved models for automatic punctuation prediction for spoken and written text. In Proceedings of the Interspeech, Lyon, France, 25–29 August 2013; pp. 3097–3101.
13. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
14. Christensen, H.; Gotoh, Y.; Renals, S. Punctuation annotation using statistical prosody models. In Proceedings of the ITRW on Prosody in Speech Recognition and Understanding, Red Bank, NJ, USA, 22–24 October 2001.
15. Tilk, O.; Alumäe, T. LSTM for punctuation restoration in speech transcripts. In Proceedings of the Interspeech, Dresden, Germany, 6–10 September 2015; pp. 683–687.
16. Klejch, O.; Bell, P.; Renals, S. Sequence-to-sequence models for punctuated transcription combining lexical and acoustic features. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, New Orleans, LA, USA, 5–9 March 2017; pp. 5700–5704.
17. Yi, J.; Tao, J. Self-attention Based Model for Punctuation Prediction Using Word and Speech Embeddings. In Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing, Brighton, UK, 12–17 May 2019; pp. 7270–7274.
18. Klejch, O.; Bell, P.; Renals, S. Punctuated transcription of multi-genre broadcasts using acoustic and lexical approaches. In Proceedings of the 2016 IEEE Spoken Language Technology Workshop, San Diego, CA, USA, 13–16 December 2016; pp. 433–440.
19. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
20. Sunkara, M.; Ronanki, S.; Bekal, D.; Bodapati, S.; Kirchhoff, K. Multimodal Semi-supervised Learning Framework for Punctuation Prediction in Conversational Speech. In Proceedings of the Interspeech, Shanghai, China, 25–29 October 2020; pp. 4911–4915.
21. Zhu, Y.; Wu, L.; Cheng, S.; Wang, M. Unified Multimodal Punctuation Restoration Framework for Mixed-Modality Corpus. In Proceedings of the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing, Singapore, 23–27 May 2022; pp. 7272–7276.
22. Beigi, H. *Fundamentals of Speaker Recognition*; Springer: New York, NY, USA, 2011.
23. Nozaki, J.; Kawahara, T.; Ishizuka, K.; Hashimoto, T. End-to-End Speech-to-Punctuated-Text Recognition. In Proceedings of the Interspeech. ISCA, Incheon, Republic of Korea, 18–22 September 2022; pp. 1811–1815.
24. Liu, X.Y.; Beigi, H. Efficient Ensemble for Multimodal Punctuation Restoration using Time-Delay Neural Network. In Proceedings of the International Conference on Ubiquitous Information Management and Communication, Kuala Lumpur, Malaysia, 3–5 January 2024; pp. 1–6.
25. Che, X.; Wang, C.; Yang, H.; Meinel, C. Punctuation prediction for unsegmented transcript based on word vector. In Proceedings of the International Conference on Language Resources and Evaluation. ELRA, Portorož, Slovenia, 23–28 May 2016; pp. 654–658.
26. Lai, V.D.; Salinas, A.; Tan, H.; Bui, T.; Tran, Q.; Yoon, S.; Deilamsalehy, H.; Dernoncourt, F.; Nguyen, T.H. Boosting punctuation restoration with data generation and reinforcement learning. In Proceedings of the Interspeech. ISCA, Dublin, Ireland, 20–24 August 2023; pp. 2133–2137.
27. Kim, H.; Seo, S.; Lee, L.; Baek, S. Improved Training for End-to-End Streaming Automatic Speech Recognition Model with Punctuation. In Proceedings of the Interspeech. ISCA, Dublin, Ireland, 20–24 August 2023; pp. 1653–1657.
28. Di Gangi, M.A.; Cattoni, R.; Bentivogli, L.; Negri, M.; Turchi, M. MuST-C: A multilingual speech translation corpus. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). ACL, Minneapolis, MN, USA, 2–7 June 2019; pp. 2012–2017.
29. Tündik, M.Á.; Szaszák, G.; Gosztolya, G.; Beke, A. User-centric evaluation of automatic punctuation in ASR closed captioning. In Proceedings of the Interspeech. ISCA, Hyderabad, India, 2–6 September 2018; pp. 2628–2632.
30. Brigance, W.N. How fast do we talk? *Q. J. Speech* **1926**, *12*, 337–342. [CrossRef]
31. Kang, W.; Yang, X.; Yao, Z.; Kuang, F.; Yang, Y.; Guo, L.; Lin, L.; Povey, D. Libriheavy: A 50,000 h ASR corpus with punctuation casing and context. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Seoul, Republic of Korea, 14–19 April 2024; pp. 10991–10995.
32. Koh, J.X.; Mislan, A.; Khoo, K.; Ang, B.; Ang, W.; Ng, C.; Tan, Y.Y. Building the singapore english national speech corpus. In Proceedings of the Interspeech, Graz, Austria, 15–19 September 2019; pp. 321–325.
33. Zen, H.; Dang, V.; Clark, R.; Zhang, Y.; Weiss, R.J.; Jia, Y.; Chen, Z.; Wu, Y. LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech. In Proceedings of the Interspeech 2019, Graz, Austria, 15–19 September 2019; pp. 1526–1530.

34. Cieri, C.; Miller, D.; Walker, K. The Fisher corpus: A resource for the next generations of speech-to-text. In Proceedings of the International Conference on Language Resources and Evaluation, ELRA, Lisbon, Portugal, 26–28 May 2004; pp. 69–71.
35. Godfrey, J.J.; Holliman, E.C.; McDaniel, J. SWITCHBOARD: Telephone speech corpus for research and development. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, San Francisco, CA, USA, 23–26 March 1992; pp. 517–520.
36. Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlicek, P.; Qian, Y.; Schwarz, P.; et al. The Kaldi speech recognition toolkit. In Proceedings of the IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, Waikoloa, HI, USA, 11–15 December 2011.
37. Baevski, A.; Zhou, Y.; Mohamed, A.; Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 2–6 December 2020; pp. 12449–12460.
38. Hernandez, F.; Nguyen, V.; Ghannay, S.; Tomashenko, N.; Esteve, Y. TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In Proceedings of the 20th International Conference on Speech and Computer, Leipzig, Germany, 18–22 September 2018; pp. 198–208.
39. Ghahremani, P.; BabaAli, B.; Povey, D.; Riedhammer, K.; Trmal, J.; Khudanpur, S. A Pitch Extraction Algorithm Tuned for Automatic Speech Recognition. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014.
40. Phoneme recognition using time-delay neural networks. In Proceedings of the IEEE Transactions on Acoustics, Speech, and Signal Processing, Online, 6 March 1989; pp. 328–339.
41. Cattoni, R.; Di Gangi, M.A.; Bentivogli, L.; Negri, M.; Turchi, M. MuST-C: A multilingual corpus for end-to-end speech translation. In *Computer Speech & Language Journal*; Elsevier: Amsterdam, The Netherlands, 2020.
42. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
43. Robbins, H.; Monro, S. A stochastic approximation method. *Ann. Math. Stat.* **1951**, 400–407. [CrossRef]
44. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2015**, arXiv:1412.6980.
45. Gupta, A.; Ramanath, R.; Shi, J.; Keerthi, S.S. Adam vs. SGD: Closing the generalization gap on image classification. In Proceedings of the OPT2021: 13th Annual Workshop on Optimization for Machine Learning, Virtual, November 2021.
46. Ribeiro, M.T.; Singh, S.; Guestrin, C. 'Why should I trust you?' Explaining the predictions of any classifier. In Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
47. Hanu, L.; Unitary Team. Detoxify. GitHub. 2020. Available online: https://github.com/unitaryai/detoxify (accessed on 24 February 2025).
48. Hanu, L.; Thewlis, J.; Haco, S. How AI is learning to identify toxic online content. *Sci. Am.* **2021**. Available online: https://www.scientificamerican.com/article/can-ai-identify-toxic-online-content/(accessed on 27 February 2025).
49. Demirkıran. Hugging Face. 2023. Available online: https://huggingface.co/FerhatDk/wav2vec2-base_music_speech_both_classification (accessed on 24 February 2025).
50. Radford, A.; Kim, J.W.; Xu, T.; Brockman, G.; McLeavey, C.; Sutskever, I. Robust speech recognition via large-scale weak supervision. In Proceedings of the International Conference on Machine Learning. PMLR, Honolulu, HI, USA, 23–29 July 2023; pp. 28492–28518.
51. Gandhi, S. Hugging Face. 2023. Available online: https://huggingface.co/sanchit-gandhi/whisper-small-ft-common-language-id (accessed on 24 February 2025).
52. Clark, K.; Luong, M.T.; Le, Q.V.; Manning, C.D. ELECTRA: Pre-training text encoders as discriminators rather than generators. In Proceedings of the International Conference on Learning Representations. ICLR, Addis Ababa, Ethiopia, 30 April 2020; pp. 1–18.
53. OpenAI. *GPT-4 Technical Report*; OpenAI: San Francisco, CA, USA, 2023.

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

Automatic Image Distillation with Wavelet Transform and Modified Principal Component Analysis

Nikolay Metodiev Sirakov [1,*,†] and Long H. Ngo [2,†]

[1] Department of Mathematics, East Texas A & M University, Commerce, TX 75429, USA
[2] Smile Inc., 92600 Paris, France; long.ngo@smile.fr
* Correspondence: nikolay.sirakov@tamuc.edu; Tel.: +1-903-450-6212
† These authors contributed equally to this work.

Abstract: Data distillation is an emerging research area, attracting the attention of machine learning (ML) and big data scientists and experts. The main goal of a distillation approach is to generate a compact dataset that preserves the essential characteristics of a larger one. In our study, we considered an initial large set of images and developed a novel method for distilling images from the initial set. The method combined discrete wavelet transform (DWT) and modified principal component analysis (M-PCA). Hence, our method first transforms images into vectors of low-band (LL) wavelet coefficients and then applies M-PCA to modify and reduce the number of vectors rather than their dimensionality. This distinguishes our approach from the traditional PCA method. We implemented the new method in Python 3.10 and validated it on public image databases, including Extended YaleB, digit-MNIST, and the ISIC2020. We demonstrated that creating a dictionary from a small set of distilled images and training a sparse representation wavelet-based classifier (SRWC) provides higher accuracy if compared to a classification when the SRWC method is trained with the entire initial training set of images.

Keywords: images; wavelets; principle components; distillation; sparse representation; classification; machine learning

1. Introduction

An important meta parameter in machine learning (ML) is the training set. For deep ML classifiers, extensive training datasets are required to achieve robust statistical results. If the large training sets also contain relatively high-resolution images, then the training process becomes extremely time-consuming. The training time could be significantly decreased if a smaller training dataset hosting the same features, qualities, statistical properties, and training abilities, is used.

Therefore, methods capable of generating or extracting, from a given large training set, a smaller one with the above abilities has become a subject of strong interest to the ML society. Early works in this field emerged around the middle of the last decade [1,2], with significant developments following in the subsequent years.

Methods for reducing training data while preserving data quality and model performance can be broadly categorized into two approaches: methods based on ML models [3–5] and data pruning (erasing) methods [6–8].

There is another concept gaining popularity among ML society, namely, transfer learning. It aims to develop methods and approaches that transfer knowledge across different tasks, for example, between image segmentation NN models and classification models [4].

Since knowledge transfer methods do not necessarily decrease the training time of the receiving model, we do not include them into the nomenclature of time reducing methods.

ML models for distillation use as a subject of their application knowledge and/or data. Knowledge distillation methods transfer meta parameters from teacher to student neural networks (NNs) [9–11]. Teacher NNs are generally large and complex, while student NNs are smaller and simpler as architecture. These teacher–student NN models significantly reduce the training time of student NNs, preserving the original training data and the statistical outcomes. They can solve problems in the fields of objects classification, detection, and segmentation in a variety of application areas in medicine and industry.

Data distillation methods keep the complexity and the architecture of the NN model the same but generate (distil) new data from the existing data. The distilled data has a smaller cardinality than the cardinality of the original training data, preserving its information quality and outcome statistics [5,12]. Note that in some cases, the newly distilled (generated) small-sized training set may contain images that do not visually resemble the original images at all [13]. Despite this dramatic change in the visual image features and the image theme, the distilled small image sets train the NN such that it provides the same or sometimes even better statistical measures. However, the visual form and the training ability of the ML distilled images depend on the number of epochs used in the process of distillation [14]. Note that this number could be high for some databases and may require more computer resources and longer run times, especially if the original images in the dataset are relatively sizable [14].

A comprehensive survey on the recent methods of data distillation, including the application of NNs, is given in [4]. The authors provide a detailed description of multiple data distillation techniques based on the concepts of "meta-model", "gradient", "distribution", "trajectory", and "factorization" matching.

On the other hand, data pruning methods are designed to remove less important and/or ill-defined samples from the original training set [6,7,15]. For the process of removal, pruning methods use optimization techniques to determine a sample's importance [7]. Hence, optimization-based pruning methods can determine the smallest efficient training set from the original samples.

In 2024, ref. [8] published a new method capable of distilling vectors. This method was based on a modification of the principal components analysis (PCA) approach [16,17] and proved itself efficient for distilling vectors with relatively small dimensions. It improved the classification statistics of ML methods trained by sets augmented with distilled vectors. A disadvantage of the modified PCA distillation method is that training an ML classifier only with distilled vectors provides poor statistical outcomes.

In the present study, we developed a new method of image distillation that modifies training images and reduces the cardinality of the training set yet improves the classification accuracy of the ML classifier trained with this set. The new method does not apply ML techniques but was developed to reduce the training sample sets required in ML methods using, for example, sparse representation classification (SRC) [18,19]. These kinds of classifiers use a dictionary in the process of classification and the dictionary is designed by a set of atomic samples selected from the set of training samples.

The first part of the new distillation method consists of discrete wavelet transform (DWT) [20,21], which represents the image with the help of its low band (LL) wavelet coefficients. Further, we describe every image as a vector of LL wavelet coefficients and apply an adapted version of the modified PCA method [8] to generate a modified version of the vectors created by the image wavelet LL coefficients. Out of the modified vectors, we select the top m vectors from every modified class (matrix) and name them distilled

vectors. They are used to build the distilled training dictionary of an SRC method. A block diagram that describes the new data distillation method is shown in Figure 1.

Note that m is a user-selected integer that should be smaller than the cardinality of the smallest class in the training image dataset. Even though we verified the new image distillation method by creating a distilled wavelet dictionary (DWD) for the sparse representation wavelet classification (SRWC) [19] method, the distilled images could be used for training in other ML methods including NNs.

The main contribution of the present study is the development of the DWD with the help of the new image distillation method, which implements the DWT, low-band wavelet (LL) coefficients (features), and a modified PCA method (M-PCA). The advantage one may obtain, if using our DWT-LL-M-PCA method, is a reduced set of training images that possess the same training capabilities as the entire original training set.

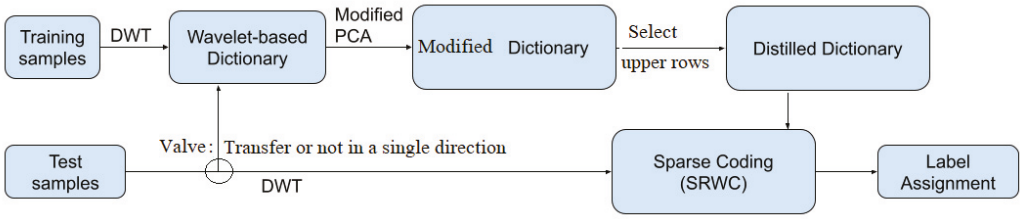


Figure 1. The main components of our data distillation method and the flow of its image classification.

Further, this paper is organized as follows: Section 2 presents the DWT, shows an example of image wavelet decomposition, and describes the development of the training dictionary for the wavelet vectors of images; Section 3 describes the development of the new approach to modify the wavelet matrix of each training class with the help of the M-PCA method and the creation of the DWD; Section 4 comprises five subsections that describe the SRWC method, the public image databases YaleB [22] and MNIST (digit and fashion) [23,24], and ISIC2020 [25] used for the experiments, the metrics implemented to measure the effectiveness of the methods, the experimental scheme, and the results of image classification. This paper end with Sections 5 and 6, which present the discussion and conclusions, where we comment on the comparisons between classifications with original and distilled training sets, list the contributions of the present study and its advantages, and outline the future work.

2. Wavelet-Based Training Dictionary Design

2.1. Image Decomposition to Wavelet Sub-Bands

In the context of two-dimensional signals, let I represent an image wherein $I(u, v)$ indicates the grey-level value at the pixel (u, v). Implementing a single-level two-dimensional wavelet transformation for an image can be achieved utilizing the technique that employs the Haar kernels, introduced in [26]. Therefore, the half-band high-pass $G_H(k)$ and the half-band low-pass $G_L(k)$ filters can be characterized, in the spatial domain, by the Haar wavelet kerne, as shown below [26]:

$$G_H(k) = \begin{cases} 1, & 0 \leqslant k < 1/2 \\ -1, & 1/2 \leqslant k < 1; \\ 0, & otherwise \end{cases} \quad G_L(k) = \begin{cases} 1, & 0 \leqslant k < 1 \\ 0, & otherwise \end{cases}. \tag{1}$$

Note that the high-pass filters $G_H(k)$ attenuate the low-band image features while passing the high-band image features. On the other hand, the low-pass filters $G_L(k)$ attenuate the high-band features while passing the low-band features. In the wavelet

domain, the low-band features describe image overall properties, and structures as well as large-scale patterns. The high-band wavelet features describe details, like, corners, edges, and noise.

Further, 2D discrete wavelet transform (DWT) [20] decomposes an image into four distinct sub-bands: approximation (LL), vertical (HL), horizontal (LH), and diagonal (HH) components. Out of the four sub-bands, the following three are considered high-band features: LH, HL, and HH, while LL is considered as describing low-band features. An example of image decomposition to the four sub-bands is shown in Figure 2.

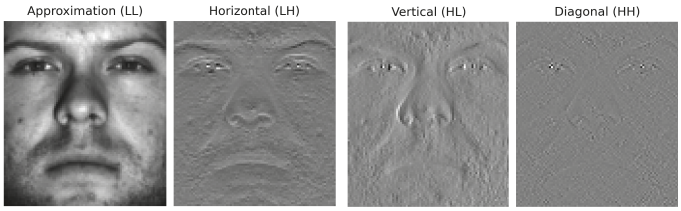


Figure 2. Decomposition of a grey-level image from YaleB [22] to four wavelet sub-band images.

2.2. Training Wavelet Dictionary Design

Consider a set of total N_k training images distributed in k different classes. Denote the i-th class of the training images with ξ_i and the label of this class with l_i, for $i = 1, \ldots, k$. The images in class ξ_i we indicate with I_{ij}, where $I_{ij} \in R^{h_{ij} \times w_{ij}}$ and h_{ij} denote the number of rows (height) in the image, while w_{ij} denotes the number of columns (width) for $j = 1, \ldots, n_i$, where n_i denotes the number of images in class ξ_i. In connection with our further study, we resize the entire set of training images such that $I_{ij} \in R^{h \times w}$ for $i = 1, \ldots, k$ and $j = 1, \ldots, n_i$.

In the next step, using the 2D DWT, we decompose every image $I_{ij} \in \xi_i$ for $i = 1, \ldots, k$ to the four wavelet coefficients LL, HL, LH, and HH. Following the idea introduced in [19], we flatten every image to a vector, where, for every image pixel, there is an LL wavelet value. Hence, we map every image $I_{ij} \in R^{h \times w}$ to a vector of the wavelet coefficient $\mathbf{v_{ij}} \in R^{wh}$, which is a vector with wh entries and the label l_i. Therefore, the class ξ_i can be presented as the matrix $\xi_i = (\mathbf{v_{i1}} \ldots \mathbf{v_{in_i}})^T \in R^{n_i \times hw}$, with its extended form being $(\xi_i | \mathbf{L_i})$, where $\mathbf{L_i} = (l_i, l_i, \ldots, l_i)^T \in R^{n_i}$, and $l_i \in R$ for $i = 1, \ldots, k$.

If we vertically concatenate all training wavelet matrices ξ_i we receive the matrix $\mathbf{D_t} = (\xi_1 \ldots \xi_k)^T \in R^{N_k \times hw}$. Note that $N_k = n_1 + n_2 + \ldots + n_k$, where n_i for $i = 1, \ldots, k$ is the cardinality of the i-th training class. We call matrix $\mathbf{D_t}$ the training wavelet dictionary (TWD); its label's extended form is $(\mathbf{D_t} | \mathbf{L})$, where the vector of labels $\mathbf{L} = (\mathbf{L_1}, \ldots, \mathbf{L_k})^T \in R^{N_k}$.

3. Distilling a Dictionary from the TWD

3.1. Modifying the Wavelet-Based Class Matrices

Consider TWD $\mathbf{D_t} = (\xi_1 \ldots \xi_k)^T \in R^{N_k \times hw}$, constructed by the vectors of wavelet coefficients $\mathbf{v_{ij}} \in R^{wh}$, $i = 1, \ldots, k$ and $j = 1, \ldots, n_i$. Denote $\mathbf{v_{ij}} = (a_{ij}^1, \ldots, a_{ij}^{hw})$, where $a_{ij}^p \in R$ represents a wavelet coefficient, which we call a feature.

Analogously to [8], we construct the p-th feature vector of the i-th class with $\mathbf{f_i^p} = (a_{i1}^p, \ldots, a_{in_i}^p)$, such that $\mathbf{f_i^p} \in \mathbf{R^{n_i}}$ and $p = 1, \ldots, hw$. We then denote the mean of the feature vectors $\mathbf{f_i^p}$ with μ_i^p and the rank of the matrix ξ_i with $r_i \leq min\{n_i, hw\}$. Then, we normalize every wavelet coefficient LL as follows:

$$a_{iu}^p = a_{iu}^p - \mu_i^p, \quad for \quad i = 1, \ldots, k, \quad u = 1, \ldots, n_i, \quad and \quad p = 1, \ldots, hw. \tag{2}$$

Further, we calculate the covariance matrix, which measures the linear relationship between the features in the matrix (class) ξ_i [17] for $i = 1, \ldots, k$. For this purpose, we use

the equation below and calculate every entry of the covariance matrix of the $i-th$ class, which we denote by $\mathbf{F^i} = \{F^i_{qj}\}, q = 1, \ldots, hw, j = 1, \ldots, hw$ [8]:

$$F^i_{qj} = \frac{1}{n_i - 1} \sum_{p=1}^{n_i} (a^p_{ij} - \mu^p_i)(a^p_{iq} - \mu^p_i). \tag{3}$$

In the next stage, we determine the Eigen vectors $\mathbf{u^i_j} \in R^{hw}$ and the Eigen values λ^i_j of $\mathbf{F^i}$ using the equation below, where $\mathbf{I} \in R^{hw \times hw}$ is the unit matrix:

$$\mathbf{F^i u^i_j} = \lambda^i_j \mathbf{u^i_j}, \quad |\mathbf{F^i} - \lambda \mathbf{I}| = 0, \quad (\mathbf{F^i} - \lambda \mathbf{I})\mathbf{u^i_j} = \mathbf{0}. \tag{4}$$

Let us consider the pairs $(\lambda^i_j, \mathbf{u^i_j})$, where $j = 1, \ldots, hw$. Recall that hw is the number of features in ξ_i, while $r_i \leq min\{n_i, hw\}$ is its rank. Recall that the number of classes in $\mathbf{D_t}$ is k. We now select the integer $m \leq r_i$ and find the m largest Eigen values $\lambda^i_{j_1} \geq \ldots \geq \lambda^i_{j_m}$ as well as their corresponding Eigen vectors $\mathbf{u^i_{j_1}}, \ldots, \mathbf{u^i_{j_m}}$. Then, we construct the matrix $\mathbf{T^i} \in \mathbf{R^{hw \times m}}$ according to decreasing Eigen values from right to left:

$$\mathbf{T^i} = [\mathbf{u^i_{j_1}}, \mathbf{u^i_{j_2}}, \ldots, \mathbf{u^i_{j_m}}], \tag{5}$$

where every Eigen vector $\mathbf{u^i_{j_1}} \in R^{hw}$ and m are the user-selected integer. The idea behind the construction of matrix $\mathbf{T^i}$ is the concept that the larger the Eigen value, the stronger the influence of its Eigen vector.

Recall that the class matrix $\xi_i \in R^{n_i \times hw}$. Therefore, the matrix product below,

$$\xi_i \mathbf{T^i} = \mathcal{D}^\mathbf{i}, \tag{6}$$

maps the matrix of the $i-th$ class to a transformed one $\mathcal{D}^\mathbf{i} \in R^{n_i \times m}$ for $i = 1, \ldots, k$. The next left product with the matrix $(\mathbf{T^i})^\mathbf{T} \in \mathbf{R^{m \times hw}}$ maps the transformed matrix $\mathcal{D}^\mathbf{i}$,

$$\mathcal{D}^\mathbf{i}(\mathbf{T^i})^\mathbf{T} = \mathbf{\Delta^i}, \tag{7}$$

into the original space such that $\mathbf{\Delta^i} \in \mathbf{R^{n_i \times hw}}$. Note that unless $\mathbf{T^i}(\mathbf{T^i})^\mathbf{T} = \mathbf{I}$, where $\mathbf{I} \in R^{hw \times hw}$ is the unit matrix, the new matrix $\mathbf{\Delta^i} \neq \xi_i$. Hence, we call the matrix $\mathbf{\Delta^i}$ the modified matrix of the i-th training class.

Please note that the modified version of the PCA method described above decreases the number of vectors in a set, while the original PCA decreases the dimension of the vectors in the set.

3.2. Creating a Distilled Wavelet Dictionary

Recall that we vertically concatenated the original classes of wavelet vectors to receive the training wavelet dictionary (TWD) $\mathbf{D_t} = (\xi_1, \ldots, \xi_k)^T = (\mathbf{v_{11}}, \ldots, \mathbf{v_{1n_1}}, \ldots, \mathbf{v_{k1}}, \ldots, \mathbf{v_{kn_k}})^T \in R^{N_k \times hw}$. If we do the same with the modified training matrices (classes) $\mathbf{\Delta^i}$ for $i = 1, \ldots, k$, we receive the modified training wavelet dictionary

$$\mathbf{\Delta_t} = (\mathbf{\Delta^1}, \ldots, \mathbf{\Delta^k})^T = (\mathbf{v^d_{11}}, \ldots, \mathbf{v^d_{1n_1}}, \ldots, \mathbf{v^d_{k1}}, \ldots, \mathbf{v^d_{kn_k}})^T \in R^{N_k \times hw}. \tag{8}$$

Note that in the present notation of vectors, the superscript $\mathbf{d}$ denotes that the vectors are modified.

Recall also that the modified training wavelet matrices $\mathbf{\Delta^i}$ are obtained by two left multiplications (see Equations (6) and (7)) of the original training wavelet matrices ξ_i for $i = 1, \ldots, k$. Since the multiplication of matrices does not change the order of the rows, it follows that the rows in $\mathbf{\Delta_t}$ are modified versions of the rows in $\mathbf{D_t}$.

To create the training distilled wavelet dictionary (DWD), we select the upper m vector rows from every modified matrix (class) of training wavelet vectors $\mathbf{\Delta^i}$ for $i = 1, \ldots, k$. Concatenating the k selected sets of m modified vectors, we receive the distilled wavelet dictionary (DWD):

$$\mathbf{\Delta_t^d} = (\mathbf{\Delta_d^1}, \ldots, \mathbf{\Delta_d^k})^T = (\mathbf{v_{11}^d}, \ldots, \mathbf{v_{1m}^d}, \ldots, \mathbf{v_{k1}^d}, \ldots, \mathbf{v_{km}^d})^T \in R^{km \times hw}. \quad (9)$$

Note that $\mathbf{\Delta_d^i}$ contains only the selected m upper rows (vectors) from the matrix of the modified training wavelet class $\mathbf{\Delta^i}$. Thus, the number of training vectors (rows) in the DWD $\mathbf{\Delta_t^d}$ is $mk << N_k$ and the number of vectors in the modified training wavelet dictionary is $\mathbf{\Delta^i}$. Also, the selected vectors $\mathbf{v_{ij}^d} \in R^{h \times w}$, which complies with $I_{ij} \in R^{h \times w}$.

4. Experimental Validation

In the present section, we first introduce the classifier sparse representation wavelet-based classification (SRWC), which we used to measure statistics when conducting training with original and distilled sets of the training dictionary. Recall that the set of distilled training images is smaller than the set of original images.

Next, we introduce the image databases and metrics that were used for the purpose of measurement. In the last subsection, we describe the experimental procedure and the results obtained by the SRWC method when using original TWD and DWD.

4.1. Sparse Representation Wavelet-Based Classification

The sparse representation wavelet-based classification (SRWC) method [19] was implemented in the wavelet domain to construct a feature dictionary derived from training samples. Paper [19] demonstrates that utilizing features exclusively from low-pass sub-band (LL) wavelet coefficients can significantly enhance image classification performance. Hence, we used the low-pass sub-band coefficients LL to generate the TWD.

The preprocessing techniques for the TWD design applied several steps, including the following: labeling the k training classes and conducting principal component analysis (PCA) to reduce the dimensionality of each vectorized sample, named "atom". As shown in Section 2.2, for each class i, the vectorized training atoms $\mathbf{v_{ij}}, j = 1, \ldots, n_i$, creating a sub-dictionary matrix $\xi_i = (\mathbf{v_{i1}}, \ldots, \mathbf{v_{in_i}})^T \in R^{n_i \times hw}$, where hw represents the dimension of each vectorized training atom and n_i denotes the number of atoms within the class. Aggregating N_k atoms from k categories (classes), the comprehensive walet training matrix of original vectors was constructed as $\mathbf{D_t} = (\xi_1, \ldots, \xi_k)^T$ and modified to $\mathbf{\Delta_t}$ with Equation (7), where every modified class $\mathbf{\Delta^i} = (\mathbf{v_{i1}^d}, \ldots, \mathbf{v_{in_i}^d})^T \in R^{n_i \times hw}$ was distilled to $\mathbf{\Delta_d^i} = (\mathbf{v_{i1}^d}, \ldots, \mathbf{v_{im}^d})^T \in R^{m \times hw}$.

Based on the assumption of sparse representation, a new unknown test atom $\mathbf{x^T} = (\mathbf{x_{i,1}}, \ldots, \mathbf{x_{i,hw}})^T \in \mathbb{R}^{hw \times 1}$ was associated with a specific class i if represented as a sparse linear combination of atoms associated with this class:

$$\mathbf{y} = \mathbf{x}_{i,1}\mathbf{v}_{i,1}^d + \mathbf{x}_{i,2}\mathbf{v}_{i,2}^d + \ldots + \mathbf{x}_{i,m}\mathbf{v}_{i,m}^d, \quad (10)$$

where $\mathbf{x}_{i,j} \in \mathbb{R},\ j = 1, 2, \ldots, m$ constitutes the sparse vector coefficients corresponding to class i. The sparse classification methodology centered on identifying the most appropriate set of training atoms for best approximation of $\mathbf{y}$. This was accomplished by determining a sparse approximation vector $\hat{\mathbf{x}}$ with non-zero entries associated with the i^{th} class, resolved through the following Lasso optimization formulation [19]:

$$\hat{\mathbf{x}} = \min_{\mathbf{x}} \left\| \mathbf{y} - \mathbf{\Delta_t^d x} \right\|_2^2 + \lambda_0 \|\mathbf{x}\|_1. \quad (11)$$

The classification of the unknown vector $\mathbf{x}$ was ultimately determined by analyzing the non-zero coefficients within $\hat{\mathbf{x}}$. The final classification was executed by minimizing the discrepancy between the sparse sample and the original representation:

$$class_{pred}(\mathbf{y}) = \underset{i}{\operatorname{argmin}} \left\| \mathbf{y} - \boldsymbol{\Delta}_{\mathbf{t}}^{\mathbf{d}} \delta_i(\hat{\mathbf{x}}) \right\|_2, \tag{12}$$

where $\hat{\mathbf{x}} \in \mathbb{R}^m$, δ_i is a characteristic function that selects elements in $\hat{\mathbf{x}}$ that are associated with class i.

4.2. Original Datasets

To experimentally validate the new image distillation method based on the DWT, LL wavelet coefficients, and M-PCA methods, we used the public image databases YaleB [22], digit-MNIST [24], and ISIC2020 [25].

The YaleB [22] database contains 2414 frontal-face images of size 192 × 168 taken from 38 faces. Therefore, the database consists of 38 classes and every class contains 64 pictures taken under different illumination conditions.

The public image database digit-MNIST [24] consists of 60,000 hand-written images of digits for training and 10,000 hand-written digit images for testing. Both the training and testing sets are split into 10 distinct classes. Every image has a size 28 × 28 pixels.

The most recent skin lesion image database is ISIC2020 [25]. It contains of 33,126 training images and 10,982 unlabeled benign and malignant images for testing. The training samples are split into 32,542 benign and 584 malignant. Experienced clinicians and dermatologists labeled the training images. The original image sizes vary between [480, 6000] for the number of rows and between [640, 6000] for the number of columns. To standardize the images, we resized them to 256 × 192 and 332 × 590 pixels. In Figure 3, we show an original training sample from ISIC2020, resized from 6000 × 4000 to 590 × 332 pixels. On the right side of the same figure, we show original training samples from digit-MNIST.

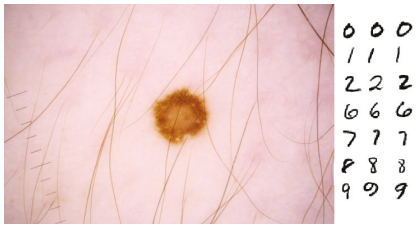


Figure 3. On the left side of the figure, we show resized, original training skin lesion image from ISIC2020 [25]. On the right side, we show original training samples from digit-MNIST [24].

4.3. Classification Metrics

The notations TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives usually describe the outcomes from a classification method. Also, they are implemented in the calculation of accuracy, the most often used metrics to evaluate the effectiveness of classification.

- Accuracy (AC): measures the overall correctness of a model's predictions:

$$\mathrm{AC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}.$$

4.4. Experimental Schemes

Our experimental scheme consisted of two major stages. In the first one, we distilled m number of images applying the DWT [20], LL, and M-PCA [8] methods. In the next stage, we trained the SRWC method with original images and classified original testing images.

Afterward we train the same classifier under the same conditions and parameters but with distilled samples and classify distilled training images (Case 1) or original training images (Case 2 and Case 3, defined below).

If the training or testing was conducted with the entire original or distilled training/testing image dataset, a single classification was performed. But, if a smaller number of training or testing original or distilled images was used, we implemented the Monte Carlo scheme [27] to select the necessary set for conducting the experiments. This scheme increased the statistical objectivity of the classification results and consisted of n runs of the classifier under the same conditions and parameters but with different sets of training samples. At the end, the average of the experimental outcomes was calculated and used for comparisons. In most cases, $n = 5$ or $n = 10$ was named 5-fold or 10-fold cross-validation, respectively.

4.5. Experimental Results

To validate our new method for training image dataset distillation, based on the DWT [20], LL wavelet coefficients, and M-PCA [8], we implemented it in Python alongside the SRWC method [19]. Employing the code, we conducted three kinds of experiments with the public image databases described in Section 4.2.

In the first kind of experiment, namely, Case 1, we distilled images from the union of the training and testing vectors (recall that every image was presented as a vector of LL wavelet coefficients) of every class. Then, we split the distilled vectors (images) into training and testing sets for every class. Further, we created the DWD by concatenating the distilled training vectors from all classes. Similarly, we merged the distilled testing vectors from all classes to form the test set for each experiment. The experimental results, which provide four different number of distilled vectors (images) from the image database YaleB, are presented in Table 1. An image from this database and a sample from its DWD Δ_t^d are shown in Figure 4.

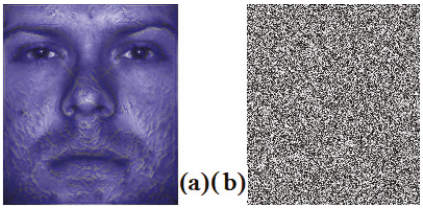


Figure 4. (**a**) Image from YaleB [22]; (**b**) distilled image in DWD Δ_t^d of the image database YaleB.

Table 1. Accuracy (Ac) results obtained by the SRWC method [19] applied to the original YaleB image database and classifications results with distilled images from each of the 38 classes of YaleB implementing Case 1 and Case 2. In the Ac column the highest value is in bold.

	Training per Class	**Total for 38 Classes**	**Train/Test per Class**	**Ac%**
Original images YaleB-38 classes	64	2432	30/34	98.06% [28]
Case 1 setup for YaleB-38 classes				
Distilled images 1st experiment	57	2166	30/27	98.9%
Distilled images 2nd experiment	44	1672	30/14	98.72%
Distilled images 3rd experiment	38	1444	30/8	**99.29**%
Distilled images 3rd experiment	32	1216	30/2	98.7%
Case 2 setup for YaleB-38 classes				
Distilled images	18	684	18/26	91.5%

In the second kind of experiment, named Case 2, we distilled training vectors (images) from the original training and testing sets of every class. Further, we used the distilled vectors to construct the DWD, which we applied to train the SRWC [19] method and classify a randomly selected subset of original testing vectors (images). The selection was conducted with the Monte Carlo scheme [27]. The experimental results obtained with distilled images when applying the present distillation Case 2 are presented at the bottom of Table 1, where we show the classification results when training with distilled images from Yale B.

In the distillation Case 3, we distilled training (images) only from the original training set of every class. Then, we created the DWD, trained the SRWC method, and classified the original testing vectors (images). The experimental results with the present setup and the image databases digit-MNIST and ISIC2020 are shown in Tables 2 and 3.

Table 2. Accuracy (Ac) results obtained by SRWC [19] for the images distilled from digit-MNIST implementing the distillation described in Case 3. In the Ac column the highest value is in bold.

	Training per Class	**Total for 38 Classes**	**Train/Test per Class**	**Ac %**
Original images digit-MNIST-10 classes	600	6000	600/1000	11%
Original images digit-MNIST-10 classes	1000	10,000	1000/1000	14.19%
Case 3 setup for digit-MNIST-10 classes				
Distilled images 1st experiment	157	1570	157/1000	**95.25%**
Distilled images 2nd experiment	94	940	94/1000	93.84%

Table 3. The 2nd and most left columns show results of classifying, by the SRWC method, the 10,982 original testing ISIC2020 images. The 3rd column shows results when the original training and testing images are sharpened by Laplace operator. All results are in "score" metric evaluated in the ISIC2020 web page [25]. The highest value is in bold.

Training Samples (Malignant/Benign)	**Original Images**	**Sharpened with Laplacian**	**Distilled Under Case 3 Setup**
292/584	score = 0.7734 [28]	score = 0.7804 [28]	score = 0.7336
523/7704	-	-	score = **0.8115**

5. Discussion

Examining the results in Table 1 for the Case 1 distillation and classification, we observe that the accuracy (Ac) achieved with the four distilled sets was higher than the Ac obtained with the original dataset. Note that the number of distilled training images for all experiments was the same (30), while the set of distilled testing images decreased. The largest Ac = 99.29% was achieved in the 3rd experiment, when we distilled 1444 images out of the 2432 original images.

These results suggest that distilling from the combined training and testing sets improved classification accuracy, even when the total number of images used was reduced. However, this approach was feasible only if the distilled training images could be identified, as demonstrated with the SRWC method.

Recall that in Case 2, we distilled both training and testing images. Then, we used the distilled images for training and tested a set of original images. The experiment in this case with the Yale B dataset is presented at the bottom of Table 1. Please note that the 18 training images per class were distilled from the 64 original images in the same class. Then, 26 original images for testing per class were selected, with the Monte

Carlo scheme [27], from the original images in the class of consideration. The obtained $Ac = 91.5\%$ was lower than the accuracies in Table 1 achieved for Case 1. The reason for this drop is that we used 40% less distilled images for training and trained the SRWC method with distilled images but classified originals.

In Case 3, we conducted two classifications with distilled images from digit-MNIST. In the first experiment, we distilled 157 training images per class from the 6000 original images in the same class. Further, applying the Monte Carlo scheme [27], we selected 1000 original images from the digit-MNIST testing set and classified them employing the DWD dictionary. Hence, we obtained Ac = 95.25%. In the second experiment, we repeated the above but distilled 94 images per class and obtained an Ac = 93.84%.

The above two results are many times greater than the results obtained when 600 and 1000 original images per class were used for training. The poor SRWC classification performance with the original digit-MNIST images was likely due to the small size of the original images (28×28), which did not exhibit sufficient image features. The distilled images were of the same sizes but were "filled up" with more and condensed information, which led to higher classification accuracy.

Further, we conducted additional experiments distilling images from the training set of ISIC2020 [25] under the conditions of Case 3, described in Section 4.5. In the first experiment, we distilled 584 benign and 292 malignant skin lesion images from the original 32,542 benign and 584 malignant training images, respectively. In the second experiment, we distilled 7704 benign and 523 malignant images from the ISIC2020 training set. In the two experiments, we trained the SRWC method with the DWDs, created with the distilled sets, and classified the 10,000 original images in the ISIC2020 testing set. The classification "scores" are present in Table 3.

The measure score was provided by the International Skin Imaging Collaboration (ISIC) in [25]. It calculated the area under the ROC (receiver operating characteristic) after we uploaded our classification results to the ISIC [25] web page.

Comparing, using the digit-MNIST image database, the new DWT-LL-M-PCA distillation method with the ML-based distillation methods presented in [4,5], one may see that the latter methods provide higher accuracy than the former one. For example, the Kip ScatterNet from [4] has an Ac = 99% if trained with 10 images distilled from every digit-MNIST class and an Ac = 99.4% if trained with 50 images distilled from every class.

At the same time, the SRWC [19] gives an Ac = 95.25% and Ac = 93.24% if trained with 157 and 94 images distilled by the DWT-LL-M-PCA method from every class, respectively. Note that the latter method takes a single pass through the original data to create the DWD and a single pass through the training data to conduct the classification with the DWD.

Please note, the Kip ScatterNet takes up to 2000 epochs (passes through the data 2000 times) to distill 50 images from every digit-MNIST class [14]. This is a time-consuming procedure and requires large computer resources, even if the image sizes are very small (28×28 pixels in digit-MNIST). But, if the images to distill from are large, as in the case of ISIC2020 [25], the needed computer resources may increase exponentially. In [29], the authors used V100 GPUs. Later, this number was reduced with the method presented in [14]. However, the latter reference does not provide details regarding the computational complexity or the run time for distillation, while our new method distilled 157 images per class, from a total of 1570 images, in about 600 seconds using a regular desktop.

6. Conclusions

The main novelty of this study consists in the development of a new image distillation method that utilizes discrete wavelet transform (DWT), low-band wavelets (LL) coefficients, and a modified principal component analysis method (M-PCA). The DWT-LL-M-PCA

method generates the DWD, which trains the SRWC method and can be used for training with any other machine learning (ML) method. Thus, from a larger original training image dataset, a smaller one is distilled, which provides the advantage of increasing the classification accuracy.

Compared to the method presented in [8], which distills vectors, the newly proposed DWT-LL-MPCA method distills images. Also, the images distilled by the new method are used alone for training and produce the DWD, which provides an accuracy higher than the accuracy achieved if the entire original training set is used. On the other hand, the method published in [8] distills vectors that are not efficient if used alone for training in ML methods. They provide high and competitive classification accuracy only if they are augmented to the original set of vectors.

Studying the classification results in Tables 1–3, one may observe that the user-selected number m of vectors (images) to be distilled affects the classification statistics. Finding the optimal m is the subject of a full-length investigation and will be developed in a new publication. Other future avenues of research are to integrate k-means or another ML technique to enhance the distillation from the modified training wavelet dictionary $\mathbf{\Delta_t}$ and investigate the distillation behavior of our method when applied to databases with noisy, degraded, blurred, or sharp images and/or images with distorted or sheared objects.

Author Contributions: Conceptualization, N.M.S.; methodology, N.M.S.; software, L.H.N.; validation, L.H.N.; formal analysis, N.M.S. and L.H.N.; resources, N.M.S. and L.H.N.; data curation, L.H.N.; writing—original draft preparation, N.M.S.; writing—review and editing, N.M.S. and L.H.N.; supervision, N.M.S.; project administration, N.M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The code for the SRWC method used for image classification is given at the website below in Software Downloads. The links to the image databases ISIC2020 [25], YaleB [22], and digit-MNIST [24], are located in Databases Downloads, Links to Original Databases: https://www.tamuc.edu/projects/augmented-image-repository/?redirect=none (accessed on 23 March 2025).

Conflicts of Interest: Author Long H. Ngo was employed by the Smile company. He did the present resecrh during his free of work time and has no conflict of interest. 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

The following abbreviations are used in this manuscript:

ML	Machine learning
LL	Low-band wavelet coefficients
M-PCA	Modified principle component analysis
SRWC	Sparse representation wavelet-based classification
DWT	Discrete wavelet transform
TWD	Training wavelet dictionary
DWD	Distilled wavelet dictionary

References

1. Zhou, L.; Pan, S.; Wang, J.; Vasilakos, A.V. Machine learning on big data: Opportunities and challenges. *Neurocomputing* **2017**, *237*, 350–361.
2. Leevy, J.L.; Khoshgoftaar, T.M.; Bauder, R.A.; Naeem, S. A survey on addressing high-class imbalance in big data. *J. Big Data* **2018**, *5*, 1–30. [CrossRef]

3. Nguyen, T.; Chen, Z.; Lee, J. Dataset meta-learning from kernel ridge-regression. In Proceedings of the International Conference on Learning Representations, ICLR, Vienna, Austria, 4 May 2024.
4. Sachdeva, N.; McAuley, J. Data Distillation: A Survey. *Trans. Mach. Learn. Res.* **2023**, 7, 1–21.
5. Radosavovic, I.; Dollár, P.; Girshick, R.; Gkioxari, G.; He, K. Data Distillation: Towards Omni-Supervised Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4119–4128.
6. Yang, Z.; Yang, H.; Majumder, S.; Cardoso, J.; Gallego, G. Data Pruning Can Do More: A Comprehensive Data Pruning Approach for Object Re-identification. *Trans. Mach. Learn. Res.* **2024**, 2835–8856. Available online: https://openreview.net/forum?id=vxxi7xzzn7 (accesse on 27 December 2024).
7. Yang, S.; Xie, Z.; Peng, H.; Xu, M.; Sun, M.; Li, P. Dataset Pruning: Reducing Training Data by Examining Generalization Influence. In Proceedings of the 2023 International Conference on Learning Representations, ICLR, Kigali, Rwanda, 1–5 May 2023.
8. Sirakov, N.M.; Shahnewaz, T.; Nakhmani, A. Training Data Augmentation with Data Distilled by the Principal Component Analysis. *Electronics* **2024**, *13*, 282. [CrossRef]
9. Yu, X.; Teng, L.; Zhang, D.; Zheng, J.; Chen, H. Attention correction feature and boundary constraint knowledge distillation for efficient 3D medical image segmentation. *Expert Syst. Appl.* **2025**, *262*, 125670, ISSN 0957-4174. [CrossRef]
10. Zhang, W.; Zhang, Y.; Huang, F.; Qi, X.; Wang, L.; Liu, X. Negative-Core Sample Knowledge Distillation for Oriented Object Detection in Remote Sensing Image. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 56486182. [CrossRef]
11. Li, X.; Huang, G.; Cheng, L.; Zhong, G.; Liu, W.; Chen, X.; Cai, M.Cross-domain visual prompting with spatial proximity knowledge distillation for histological image classification. *J. Biomed. Inform.* **2024**, *158*, 104728, ISSN 1532-0464. [CrossRef]
12. Zhao, B.; Mopuri, K.R.; Bilen, H. Dataset Condensation with Gradient Matching. In Proceedings of the International Conference on Learning Representations-ICLR, Vienna, Austria, 4 May 2021.
13. Wang, T.; Zhu, J.-Y.; Torralba, A.; Efros, A. Dataset Distillation. *arXiv* **2020**, arXiv:1811.10959v3.
14. Vinaroz, M.; Park, M. Differentially private kernel inducing points using features from scatternets (DP-KIP-scatternet) for privacy preserving data distillation. *arXiv* **2024**, arXiv:2301.13389. Available online: https://arxiv.org/pdf/2301.13389v2 (accessed on 15 January 2025).
15. Cazenavette, G.; Wang, T.; Torralba, A.; Efros, A.A.; Zhu, J.Y. Dataset distillation by matching training trajectories. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4750–4759.
16. Hotelling, H. Analysis of a complex of statistical variables into principa. *J. Educ. Psychol.* **1933**, *24*, 417. [CrossRef]
17. Abdi, H.; Williams, L.J. Principal component analysis. *Wires Comput. Stat.* **2010**, *2*, 433–459. [CrossRef]
18. Ngo, L.H.; Luong, M.; Sirakov, N.M.; Viennet, E.; Le-Tien, T. Skin lesion image classification using sparse representation in quaternion wavelet domain. *Signal Image Video Process.* **2022**, *16*, 1721–1729.
19. Ngo, L.H.; Luong, M.; Sirakov, N.M.; Le-Tien, T.; Guerif, S.; Viennet, E. Sparse Representation Wavelet Based Classification. In Proceedings of the International Conference in Image Processing, ICIP2018, Athens, Greece, 7–10 October 2018; IEEE XPlore: Piscataway, NJ, USA , 2018; pp. 2974–2978. [CrossRef]
20. Mallat, S. *Wavelet Tour of Signal Processing: The Sparse Way*; Academic Press: Cambridge, MA, USA, 2008.
21. Zou, W.; Li, Y. Image classification using wavelet coefficients in low-pass bands. In Proceedings of the International Joint Conference on Neural Networks, IJCNN, Orlando, FL, USA, 12–17 August 2007; IEEE Xplore: Piscataway, NJ, USA, 2007; pp. 114–118.
22. Georghiades, A.S.; Belhumeur, P.N.; Kriegman, D. From few tomany: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 643–660. [CrossRef]
23. Kadam, S.S.; Adamuthe, A.C.; Patil, A.B. CNN model for image classification on MNIST and fashion-MNIST dataset. *J. Sci. Res.* **2020**, *64*, 374–384. [CrossRef]
24. General Site for the MNIST Database. Available online: http://yann.lecun.com/exdb/mnist (accessed on 12 September 2024).
25. International Skin Imaging Collaboration. SIIM-ISIC 2020 Challenge Dataset. Internat. Skin Imaging Collaboration. Available online: https://challenge2020.isic-archive.com/ (accessed on 17 May 2023).
26. Guo, T.; Mousavi, H.S.; Vu, T.H.; Monga, V. Deep wavelet prediction for image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Workshops, Honolulu, HI, USA, 21–26 July 2017.

27. Dubitzky, D.; Granzow, M.; Berrar, D.P. *Fundamentals Data Mining Genomics Proteomics*; Springer: Boston, MA, USA, 2007.
28. Sirakov, N.M.; Bowden, A.; Chen, M.; Ngo, L.H.; Luong, M. Embedding vector field into image features to enhance classification. *J. Comput. Appl. Math.* **2024**, *44*, 115685, ISSN 0377-0427.
29. Nguyen, T.; Novak, R.; Xiao, L.; Lee, J. Dataset distillation with infinitely wide convolutional networks. In Proceedings of the NeurIPS 2021, 35 Annual Conference on Neural Information Processing Systems, Virtual Conference, 7–10 December 2021; Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P.S., Vaughan, J.W., Eds.; Curran Associates Inc.: Red Hook, NY, USA; Volume 34, pp. 5186–5198.

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

Time Series Forecasting via an Elastic Optimal Adaptive GM(1,1) Model

Teng Li [1], Jiajia Nie [2], Guozhi Qiu [2], Zhen Li [1,*], Cun Ji [3] and Xueqing Li [2]

[1] Office of Information Technology, Shandong University, Jinan 250100, China; liteng@sdu.edu.cn
[2] School of Software, Shandong University, Jinan 250101, China; 202335384@mail.sdu.edu.cn (J.N.); 202415306@mail.sdu.edu.cn (G.Q.); xqli@sdu.edu.cn (X.L.)
[3] School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China; jicun@sdnu.edu.cn
* Correspondence: jens@sdu.edu.cn

Abstract: The GM(1,1) model is a well-established approach for time series forecasting, demonstrating superior effectiveness with limited data and incomplete information. However, its performance often degrades in dynamic systems, leading to obvious prediction errors. To address this impediment, we propose an elastic optimal adaptive GM(1,1) model, dubbed EOAGM, to improve forecasting performance. Specifically, our proposed EOAGM dynamically optimizes the sequence length by discarding outdated data and incorporating new data, reducing the influence of irrelevant historical information. Moreover, we introduce a stationarity test mechanism to identify and adjust sequence data fluctuations, ensuring stability and robustness against volatility. Additionally, the model refines parameter optimization by incorporating predicted values into candidate sequences and assessing their impact on subsequent forecasts, particularly under conditions of data fluctuation or anomalies. Experimental evaluations across multiple real-world datasets demonstrate the superior prediction accuracy and reliability of our model compared to six baseline approaches.

Keywords: stationarity test; elastic adjustment; optimal adaptation; GM(1,1)

1. Introduction

As a fundamental research direction in data mining [1,2], time series analysis is considered one of the top ten technical challenges of the 21st century [3–5]. Time series research encompasses multiple core domains, specifically including key technical directions such as time series forecasting, time series classification, time series clustering, and time series anomaly detection [6–8]. At its core, time series forecasting leverages historical data to uncover temporal patterns, constructing predictive models that provide quantitative foundations for trend anticipation, risk early warning, and evidence-based decision-making [9–11]. Research indicates that effective time series forecasting not only reveals the evolution mechanisms of systems [12–14] but also offers technological support for industrial upgrading, thereby facilitating the implementation of cross-industry sustainable development strategies [15–17].

Among various forecasting methods [18–20], the GM(1,1) model has garnered significant attention due to its unique capability in small-sample modeling [21,22]. As an important tool in gray system theory, this model demonstrates significant advantages in system behavior modeling and evolution analysis through specialized data generation mechanisms with limited information [23,24]. Its core features are (1) modeling with as

few as four data points, (2) no preset distribution assumptions, and (3) a combination of computational efficiency and verifiability. These distinctive characteristics have facilitated its widespread adoption across diverse domains. For instance, Cai et al. [25] employed an enhanced GM(1,1) model to predict economic losses caused by marine disasters, while Ding [26] used a self-adaptive intelligent gray model to forecast natural gas demand.

However, the classical GM(1,1) model exhibits notable limitations when applied to dynamic systems. In [27,28], the following points are acknowledged: first, the traditional accumulation generation mechanism struggles to capture temporal fluctuation characteristics; second, the fixed-length modeling window fails to adapt to changes in data distribution; third, the parameter optimization process lacks dynamic error correction mechanisms [29]. Although existing studies have improved model performance through refinements of background values, construction of gray derivatives, and parameter optimization [30,31], most methods still assume system stability and do not effectively address data fluctuation issues in dynamic environments.

To overcome the challenges encountered by the traditional GM(1,1) model in dynamic system forecasting, we propose an innovative elastic optimal adaptive GM(1,1) model (EOAGM). This model enhances performance through three key innovations: first, it replaces the traditional cumulative generation sequence with an adaptive sequence generation method, strengthening the influence of recent observations and enhancing the model's ability to capture dynamic system characteristics. Second, it introduces a statistical stationarity test framework to establish a dynamic adjustment mechanism for sequence length, enabling a transition from fixed windows to elastic ones. Lastly, it constructs a candidate sequence evaluation system that incorporates predicted values for back-validation and optimal sequence selection, forming a closed-loop optimization system for error self-correction, effectively improving the model's predictive performance. These improvements significantly enhance the model's adaptability and accuracy in dynamic systems, achieving superior simulation and forecasting outcomes. The contributions of our work could be summarized as follows:

1. Adaptive sequence generation mechanism: We improve the architecture of the gray system model by employing an adaptive sequence generation method that integrates the accumulation of historical data with the timing of recent observations. This enhanced framework effectively captures and characterizes the inherent complex temporal patterns in the system's evolution by strengthening the dynamic representation of temporal features.
2. Elastic modeling window: We integrate stationarity testing into the gray system to dynamically adjust the sequence length used for prediction within the gray system. This reduces the adverse impact of changes in data distribution on prediction accuracy.
3. Candidate sequence evaluation system: We propose an optimal sequence selection method to evaluate whether replacing actual values with predicted values can improve the accuracy of subsequent predictions. The optimal sequence is then chosen to dynamically correct the prediction errors.
4. Applications and validation: The proposed model is applied to forecast dynamic systems, including China's GDP and indigenous thermal energy consumption. Comparative analysis against other models reveals that our approach delivers superior predictive accuracy for dynamic systems.

2. Related Work

The gray forecasting model GM(1,1), introduced by Deng in 1982 [32], provides a robust framework for time series forecasting under incomplete information conditions. Renowned for its high precision and computational simplicity, the GM(1,1) model has been

extensively adopted across various domains. It operates on the principle of interactive computation, leveraging limited and incomplete data during both model construction and parameter optimization. This involves formulating a first-order differential equation derived from the accumulated generating sequence, capturing the system's development trends, and performing parameter estimation using statistical techniques like least squares. The model's mechanism comprises three key steps:

1. Accumulated generating operation (AGO): The original dataset undergoes an accumulated transformation to yield a new sequence, emphasizing the system's development trend and rendering the data's evolution more discernible.
2. A first-order linear differential equation is constructed based on the accumulated sequence, describing the data's progression. Parameters are estimated using statistical methods, such as the least squares method.
3. Optimization focuses on estimating the development coefficient and other parameters critical for the model's accuracy.

Researchers have worked extensively to refine the GM(1,1) model by optimizing background values, gray derivatives, and parameter estimation processes [33–35]. Enhancements in the background value calculation have significantly improved prediction precision. For instance, Zhan et al. [36] developed a multi-parameter background value approach for nonlinear optimization, while Ma and Wang [37] proposed background value optimization based on improved algorithms. Similarly, the approximation of piecewise linear and generalized functions can be used for constructing background values to improve accuracy. Gray derivative calculations, crucial for capturing trends in data, have also seen advancements [38–40]. Parameter optimization remains a cornerstone for ensuring the GM(1,1) model's robustness [41]. Despite these advancements, the GM(1,1) model's assumption of system stability often limits its performance when applied to datasets characterized by high volatility or dynamic behavior [42].

Adaptive parameter estimation frameworks have emerged to mitigate these limitations, enabling real-time parameter adjustment based on incoming data [43]. However, such approaches often overlook the role of sequence stability in improving predictions [44]. Sequence stationarity, a fundamental prerequisite in time series forecasting, is essential for ensuring reliable and interpretable predictions. Stationarity detection methods, such as the augmented Dickey–Fuller (ADF) test [45,46], the Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test [47,48], and the Phillips–Perron (PP) test [49,50], are widely used to evaluate unit roots or trends in datasets. Correctly addressing non-stationarity through techniques like differencing or detrending enhances a model's ability to capture underlying patterns and produce robust forecasts. Despite these efforts, the fixed-length sequence framework of traditional gray systems, which retains recent data and discards older data, often fails to maintain sequence stability in dynamic environments. To overcome these limitations, we propose the EOAGM. This innovative model dynamically adjusts the data sequence length, discarding outdated information while integrating new data, thereby reducing the influence of irrelevant historical data. A stationarity test is incorporated to detect sequence fluctuations and dynamically modify sequence length, enhancing robustness against data volatility. Furthermore, parameter optimization is refined by integrating predicted values into the candidate sequence, enabling the evaluation of their efficacy in subsequent forecasts. This strategy improves the model's adaptability and precision in capturing patterns within dynamic systems, achieving superior simulation and forecasting performance.

3. Materials and Methods

3.1. Traditional GM(1,1) Model

The traditional GM(1,1) model (TGM) operates through the following key steps: let the time series $X^{(0)}$ have n observations, $X^{(0)} = \{X^{(0)}(1), X^{(0)}(2), \ldots, X^{(0)}(n)\}$; by AGO, we generate a new sequence, $X^{(1)} = \{X^{(1)}(1), X^{(1)}(2), \ldots, X^{(1)}(n)\}$, where

$$X^{(1)}(K) = \sum_{i=1}^{k} X^{(0)}(i), k = 1, 2, 3, \ldots, n. \tag{1}$$

The adjacent mean generation sequence $Z^{(1)}$ of $X^{(1)}$ is computed as follows:

$$Z^{(1)}(K) = 0.5X^{(1)}(K) + 0.5X^{(1)}(K-1), k = 2, \ldots, n. \tag{2}$$

Then, the model establishes the first-order linear differential equation:

$$\frac{dX^{(1)}}{dt} + \alpha X^{(1)} = \mu. \tag{3}$$

The discretization method of Equation (3) is $X^{(0)}(K) + \alpha Z^{(1)}(K) = \mu$, where α is the developmental gray number, and the μ is the endogenous control gray number. The parameters α and μ are estimated using the least squares method:

$$\begin{Bmatrix} \alpha \\ \mu \end{Bmatrix} = (B^T B)^{-1} B^T Y_n, \tag{4}$$

where

$$Y_n = \begin{bmatrix} X^{(0)}(2) \\ X^{(0)}(3) \\ \vdots \\ X^{(0)}(n) \end{bmatrix}, B = \begin{bmatrix} -Z^{(1)}(2) & 1 \\ -Z^{(1)}(3) & 1 \\ \vdots & \\ -Z^{(1)}(n) & 1 \end{bmatrix}.$$

The prediction model derived from the differential equation is as follows:

$$\hat{X}^{(1)}(k+1) = \left[\hat{X}^{(0)}(1) - \tfrac{\mu}{\alpha}\right] e^{-\alpha k} + \frac{\mu}{\alpha}, k = 0, 1, 2, \ldots, n. \tag{5}$$

From this, the predicted values for the original sequence are obtained as follows:

$$\hat{X}^{(0)}(k+1) = \hat{X}^{(1)}(k+1) - \hat{X}^{(1)}(k), k = 0, 1, 2, \ldots, n. \tag{6}$$

3.2. The Elastic Optimal Adaptive GM(1,1) Model

The elastic optimal adaptive GM(1,1) model (EOAGM) enhances the prediction accuracy of gray systems through parameter optimization methods. As illustrated in Figure 1, the EOAGM model is structured into three main steps:

1. Sequences construction: This step involves generating two types of sequences, the original sequence and the candidate sequence, to ensure robust prediction capability.
2. Stationarity detection: The EOAGM model improves on the AGM by using the ADF test to evaluate the stability of sequences. Stationarity ensures the model's parameters accurately represent the system's behavior, leading to better prediction accuracy.
3. Optimal sequence selection: This step identifies the sequence that provides the best prediction accuracy by evaluating the performance of both original and candidate sequences.

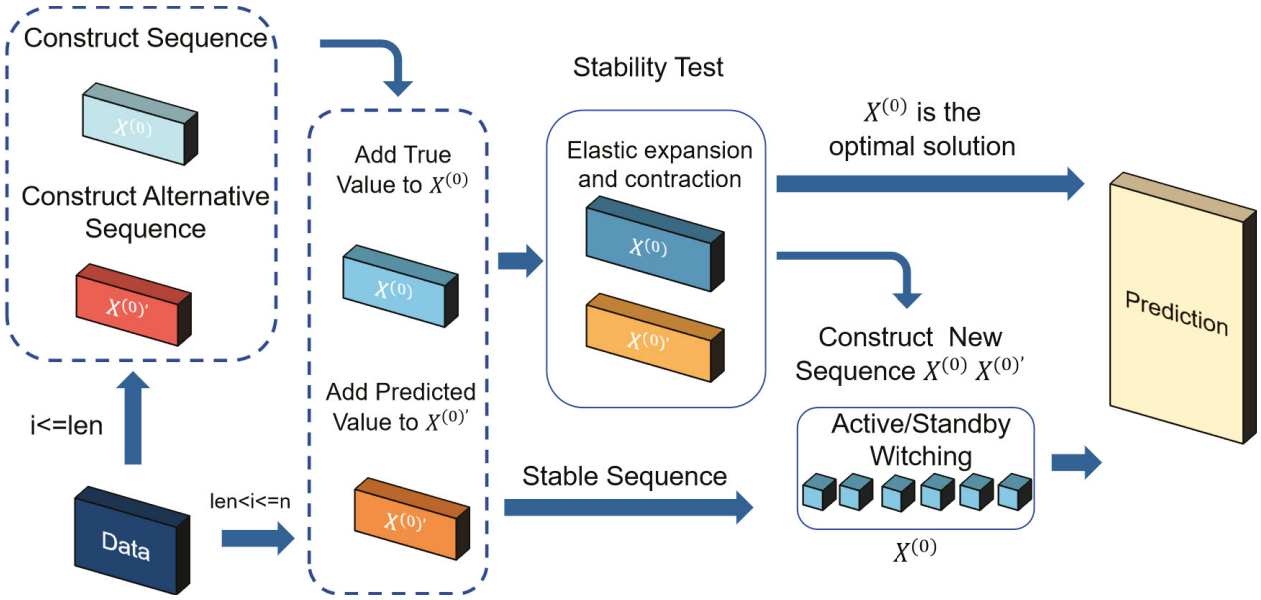


Figure 1. General framework.

3.2.1. Sequences Construction

The process of sequence construction involves generating the original sequence and the candidate sequence, each designed to optimize prediction performance. The variable *len* (length) is initialized to represent the number of elements in the sequence. The steps of sequence construction include the following:

1. Defining sequence length: The optimal length *len* is set for both the original and candidate sequences.
2. Data insertion: When the sequence length is not more than *len*, new data points are directly added to both sequences. When the sequence length exceeds *len*, the observed value is appended to the original sequence, while the predicted value is added to the candidate sequence.
3. Maintaining sequence length: To keep the sequence length fixed, the oldest value is discarded from both sequences.
4. Temporary storage of discarded values: Discarded values are stored temporarily, allowing the model to revert to previous states during the stationarity detection phase if instability is detected. This structured approach ensures the sequences are robust, adaptable, and ready for subsequent stationarity detection and optimization steps.

3.2.2. Stationarity Detection

In the AGM, the length of the data sequence used to calculate the model parameters α and μ is fixed. However, even with operations that involve discarding outdated data and incorporating new data, the sequence stability may still degrade, leading to poor model performance and inaccurate predictions. To address this limitation, we introduce the ADF test method to assess the stability of the sequence within the model.

The ADF test is a statistical method used to determine whether a time series has a unit root, which indicates non-stationarity.

The procedure for calculating ADF statistics to test for the presence of unit roots is shown in Equation (7).

$$Y_t = c + \beta t + \alpha Y_{t-1} + \phi \Delta Y_{t-1} + e_t, t = 0, 1, 2, \ldots, n. \quad (7)$$

In the traditional DF test, the value of the time series at time t is denoted as Y_t, the first-order difference of the series at time $t-1$ is denoted as ΔY_{t-1}, and the difference term

is e_t. In traditional DF detection, an enhancement is made, transforming it as shown in Equation (8).

$$\phi \Delta Y_{t-1} = \phi_1 \Delta Y_{t-1} + \phi_2 \Delta Y_{t-2} + \ldots + \phi_{p-1} \Delta Y_{t-(p-1))}, t = 0, 1, 2, \ldots, n. \quad (8)$$

When conducting an ADF test, we typically pay attention to the test statistic (T_{value}), the probability of the sequence being non-stationary (P_{value}), and the critical values (C_{value}) associated with the confidence interval [51].

The ADF test indicates whether the sequence has a unit root. A T_{value} smaller than the C_{value} implies stationarity. P_{value} measures the probability of the sequence being non-stationary, with a $P_{value} < 0.05$ indicating stationarity. The ADF test results provide C_{values} at different confidence levels (e.g., 1%, 5%, and 10%), which are used to compare with the T_{value}. In the ADF test, the critical values of the confidence interval help us determine whether the time series is stationary, rather than directly providing a confidence interval for a parameter.

Modern statistical software packages, including including R (4.3.0+), Python (3.7), employ automated algorithms to select optimal lag orders for Dickey–Fuller tests based on predefined distribution criteria. In econometric analysis, the lag order specification in the augmented Dickey–Fuller (ADF) test is pivotal to ensure the validity of stationarity conclusions. For this study, the lag order was algorithmically determined via the Akaike information criterion (AIC) using Python's statsmodels library, which generated the C_{values} and P_{value}. The AIC optimizes model parsimony by penalizing over-parameterization, thereby mitigating risks of underfitting (insufficient lags) and overfitting (redundant lags). Furthermore, the AIC's systematic optimization ensures that the selected lags adequately account for serial correlation in the residuals, which is critical for valid inference.

Using the ADF test, we assess whether adaptive adjustments to the sequence maintain its stability. Based on the test results, decisions can be made to modify the sequence length if necessary. If the adjusted sequence fails to achieve stationarity, the original sequence length is preserved.

The elastic adjustment process, as outlined in Algorithm 1, incorporates several parameters: c_f, representing the confidence level (e.g., 1%, 5%, or and 10%, corresponding to 99%, 95%, and 90% probabilities of stationarity, respectively); d_f [52], indicating the differencing order, which can be first-order or second-order differencing for non-stationary time series data [53]. The ADF test method is applied to the sequence, yielding a result array containing T_{value}, C_{value} and P_{value}, which collectively determine stationarity. To evaluate this, the stationarity method examines the array: if $P_{value} < 0.05$ and $T_{value} < C_{value}$, the sequence is deemed stationary; otherwise, it is considered non-stationary. For non-stationary sequences, adjustments involve creating two new sequences (one by adding a data point to the beginning of the original sequence and the other by removing the oldest data point) and applying the ADF test to both. The sequence with the smaller P_{value} is selected for further evaluation using the stationarity method. If the sequence achieves stationarity, it is used for subsequent modeling; otherwise, the original sequence is retained. This iterative approach ensures the balance and robustness of the sequence for accurate forecasting. In the revised methodology, it is adjusted up to two times during a single prediction cycle, and the window length is reset to its original baseline before initiating the next detection cycle.

Algorithm 1: Elastic adjustment process.

Input: $X^{(0)}, c_f, d_f$
Output: $X^{(0)}$

p_{array} = adftest($X^{(0)}, c_f, d_f$)
if *stationarity*(p_{array}) **then**
  return $X^{(0)}$
end
else
  $X^{(0)}_{add} = d_{old} + X^{(0)}$
  $X^{(0)}_{remove} = X^{(0)}[1:]$
  p_{add} = adftest($X^{(0)}_{add}, c_f, d_f$)
  p_{remove} = adftest($X^{(0)}_{remove}, c_f, d_f$)
  if $p_{add}[2] < p_{remove}[2]$ **then**
    $S_{sequence} = X^{(0)}_{add}$
    $T_{result} = p_{add}$
  end
  else
    $S_{sequence} = X^{(0)}_{remove}$
    $T_{result} = p_{remove}$
  end
  if *stationarity*(T_{result}) **then**
    return $S_{sequence}$
  end
  else
    return $X^{(0)}$
  end
end

3.2.3. Optimal Sequence Selection

Stationarity detection adjusts the sequence length by adding or removing data from the sequence's head to achieve a stable state. However, the sequence accepts all newly added values, including outliers or values with significant fluctuations, which would reduce the accuracy of the forecast result. Therefore, we employ the optimal sequence selection method to enhance prediction accuracy. The optimal sequence selection method is described in Algorithm 2. In Algorithm 2, we add a candidate series. Let the original series be a time series $X^{(0)}_{k=i}$ with i observations. At the moment $i+1$, we obtain $X^{(0)}(i+1)$, which is added to the original series and discard the oldest data to obtain a new series $X^{(0)}_{k=i+1}$, which serves as a prediction sequence, which is consistent with the AGM. In addition, we add the candidate series $X^{'(0)}_{k=i+1} = \{X^{(0)}(2), \ldots, X^{(0)}(n), y_p\}$.

At the moment $i+2$, we compare the true values with the predicted values of the two series, select a series with a small relative error, and then add new data, discard the old data, and generate a new candidate series right on top of the selection sequence.

The most useful thing about the candidate series is that the better data among the predicted data and the real data are selected for prediction, which leads to better prediction results.

Algorithm 2: Optimal sequence selection.

Input: $X_{k=i}^{(0)}, X_{k=i}^{'(0)}, X^{(0)}(i+1), X^{(0)}(i+2), y_{next}, y_{next}^{'}$
Output: $X_{k=i+1}^{(0)}, X_{k=i+1}^{'(0)}, y_{next}, y_{next}^{'}$

if y_{next} **then**
  $r_{next} = abs((X^{(0)}(i+1) - y_{next})/X^{(0)}(i+1))$
  $r_{next}^{'} = abs((X^{(0)}(i+1) - y_{next}^{'})/X^{(0)}(i+1))$
  if $r_{next} > r_{next}^{'}$ **then**
    $X_{k=i+1}^{(0)} = X_{k=i+1}^{'(0)}$
  end
  else
    $X_{k=i+1}^{'(0)} = X_{k=i+1}^{(0)}$
  end
end
$y_p = gm11(X_{k=i}^{(0)})$
$X_{k=i+1}^{(0)}, X_{k=i}^{'(0)} = updateQueue(X_{k=i}^{(0)}, X^{(0)}(i+1), y_p)$
$y_{next}, y_{next}^{'} = gm11List(X_{k=i+1}^{(0)}, X_{k=i+1}^{'(0)})$
return $X_{k=i+1}^{(0)}, X_{k=i+1}^{'(0)}, y_{next}, y_{next}^{'}$

4. Experimental Results and Discussion

In this study, we first present the datasets, baseline methods, and evaluation metrics. Subsequently, we compare the prediction accuracy across different baseline approaches. Next, we conduct a sensitivity analysis to assess model robustness. Finally, we perform an ablation experiment to evaluate the contribution of key components.

Datasets: To validate the superiority of our algorithm, we conducted experiments on three distinct datasets: (1) China's annual GDP data from 2008 to 2023, (2) district heating data from Jinan City, China, and (3) the annual cargo and mail throughput data of Jinan Yaoqiang Airport.

For the GDP dataset, obtained from the National Bureau of Statistics of China (https://data.stats.gov.cn (accessed on 22 May 2024)), we utilized observations from the years 2008–2014 as the training set to predict subsequent values for 2015–2021. For the heating dataset, we selected daily measurements from six residential buildings between 1–13 January. The model was trained on data from 1–7 January and tested on its ability to forecast heating demand for 8–13 January. The raw district heating data are presented in Table 1. Column 1 represents observation dates, and columns 2–7 represent daily thermal energy consumption (kWh) for buildings 11–16. For the cargo and mail throughput data of Jinan Yaoqiang Airport (JYA-CMTD), obtained from the website (https://www.huaon.com/channel/industrydata/1060968.html (accessed on 5 January 2025)), we utilized observations from 2011–2016 as the training set to predict subsequent years' values for 2017–2021.

To promote transparency and reproducibility in scientific research, the original datasets and supplementary materials associated with this article were deposited in a public repository. The resources are freely accessible via Gitee and the link is https://gitee.com/sduliteng/gm.git (accessed on 10 May 2025).

Table 1. Raw data of thermal energy consumption.

Date	No. 11	No. 12	No. 13	No. 14	No. 15	No. 16
01-01	13,900.29	3326.23	2536.82	2295.93	4926.74	17,017.67
01-02	13,556.13	3396.63	2530.54	2304.05	4882.54	16,663.45
01-03	13,209.10	3333.60	1914.24	2267.04	4844.14	16,070.78
01-04	12,385.61	3059.22	2964.36	2118.25	4544.05	14,970.55
01-05	12,313.99	3054.51	2338.64	2106.83	4535.83	14,551.11
01-06	13,223.57	3160.68	2366.73	2150.19	4633.16	15,105.55
01-07	13,267.33	3090.40	2349.92	2117.3	4585.94	15,458.17
01-08	12,532.19	2840.63	2218.29	1980.02	4277.26	14,559.78
01-09	13,145.06	3111.99	2423.01	2164.07	4703.21	15,530.5
01-10	12,972.85	3166.90	2475.54	2218.12	4793.15	15,543.7
01-11	13,024.71	3128.9	2441.32	2189.76	4729.36	14,660.22
01-12	13,868.68	3292.88	2509.36	2255.72	4882.86	15,947.28
01-13	14,165.04	3382.87	2591.21	2312.53	5029.03	16,464.54

Baseline Method: In addition to the traditional GM(1,1) (TGM), this study comprehensively evaluates five advanced gray models and one non-gray model proposed in recent studies.

1. Cumulative GM(1,1) Model (CGM) [54]: The CGM recalculates α and μ iteratively by incorporating new data. While this method improves adaptability by considering growing datasets, the increasing sequence length may introduce noise and irrelevant information, reducing prediction precision over time.
2. Adaptive GM(1,1) Model (AGM) [22]: The AGM dynamically adjusts the data sequence by discarding older data and incorporating new observations. This approach enhances forecasting accuracy for dynamic systems by reflecting real-time changes. However, removing older data may destabilize predictions, and incorporating new data (including anomalies) may reduce prediction reliability.
3. Simultaneous Gray Model (SimGM) [42]: This model can improve the algorithm for calculating α and μ [55]. Traditional GM(1,1) employs ordinary least squares (OLS) to estimate α and μ. However, since real-world systems are governed by interconnected and evolving factors, a single differential equation may inadequately capture these relationships. To address this limitation, the SimGM has been proposed, which significantly improves prediction accuracy compared to conventional single-equation models.
4. Nonlinear Gray Model (NonlGM) [56]: The NonlGM proposes an enhanced GM(1,N) model incorporating nonlinear optimization techniques to improve forecasting accuracy and robustness. The background value in the TGM(1,1) model is defined as $Z^{(1)}(K) = 0.5X^{(1)}(K) + 0.5X^{(1)}(K-1), k = 2, \ldots, n$. In the process of background value optimization, the fixed weight coefficient (0.5) can be optimized. In the NonlGM, $Z^{(1)}(K) = aX^{(1)}(K) + (1-a)X^{(1)}(K-1)$, where a is optimized.
5. Improved Gray Model (ImGM) [37]: This model is an improved optimized background value determination method for the GM(1,1) model. In this method, background value optimization can also be achieved through the exponential background value and dynamic adaptive background value using data characteristics methods.
6. Lagged [57]: This model employs a non-gray modeling approach for data forecasting, specifically utilizing a quantile regression framework with lagged and asymmetric effects.

Evaluation Metrics: In the experiment, relative error and mean error are used as test indicators. The formula for calculating relative error is as follows:

$$w_k = \left| \frac{\hat{X}^{(0)}(k) - X^{(0)}(k)}{X^{(0)}(k)} \right|, \tag{9}$$

where $\hat{X}^{(0)}(k)$ is the predicted value of the k-th data point, and $X^{(0)}(k)$ is the true value of the k-th data point. Assuming n is the number of predicted values, then the average error of the prediction is as follows:

$$w_{avg} = \frac{\sum_{n}^{k=1} w_k}{n}. \tag{10}$$

Common Parameter Settings: In the experiments, the baseline models incorporate several shared parameters to ensure consistency and comparability. For the GM(1,1) model, the initialization parameters are uniformly set as gm11.alpha = 0.5, gm11.convolution = True, and gm11.stride = 1 across all datasets. These configurations aim to balance model flexibility and computational efficiency. For the ADF algorithm, parameters regression = "c" (constant term included) and autolag = "AIC" (automatic lag selection via the Akaike information criterion) are adopted universally to standardize stationarity testing and enhance robustness in time series analysis.

Dataset-Specific Configurations: The sequence length in GM(1,1) is optimized based on empirical results: a length of seven minimizes prediction errors for forecasting China's GDP and Jinan's thermal energy consumption, while six is optimal for the annual cargo and mail throughput data of Jinan Yaoqiang Airport. In the ADF algorithm, differencing strategies vary: thermal energy and cargo throughput data use first-order differencing, whereas GDP data employs first-order differencing for sequences shorter than six and second-order differencing for longer sequences to address non-stationarity. Elastic scaling ranges also differ: thermal energy and cargo throughput are assigned a range of 1, while GDP uses 2 to accommodate its higher volatility.

Rationale for Optimality: Parameter selections are empirically validated. For GM(1,1), a sequence length of seven optimally captures temporal patterns in GDP and thermal energy data, balancing underfitting and overfitting risks. A shorter length of six for cargo throughput aligns with its lower seasonal complexity. In ADF, differencing orders are tailored to data characteristics: higher-order differencing for longer GDP sequences mitigates trend persistence, while first-order suffices for other datasets. Elastic ranges reflect inherent volatility, i.e., GDP's broader range of 2 accommodates macroeconomic fluctuations, whereas thermal and cargo data's narrower range of 1 suits their smoother trends. These settings collectively ensure model generalizability and precision across diverse datasets.

4.1. Performance Comparison with Base Lines

4.1.1. National GDP Prediction

To demonstrate the general applicability of our proposed model, we performed a comprehensive forecasting of China's GDP from 2015 to 2021 using eight gray models: TGM/CGM/AGM/SimGM/NonlGM/ImGM/Lagreg and EOAGM.

The errors (%) of each model for different years, as well as the mean errors, are presented in Table 2. As can be observed from Table 2, our algorithm achieves the smallest values in both mean error and maximum error.

As Table 2 shows, error differences of <1 % can be seen between EOAGM and AGM for four of the GDP years. We provide a paired Wilcoxon signed-rank test over all fore-

casting horizons to show significance. The test was performed on the paired error data (before = AGM errors and after = EOAGM errors) from 2015 to 2024, yielding a p-value of 0.0499. Since this value is below the significance threshold of 0.05, we reject the null hypothesis and conclude that the improvement in EOAGM over AGM is statistically significant.

Furthermore, a line chart (Figure 2) was employed to illustrate the stability of the algorithms. The yellow line represents the error trend of our proposed algorithm, demonstrating its exceptional stability.

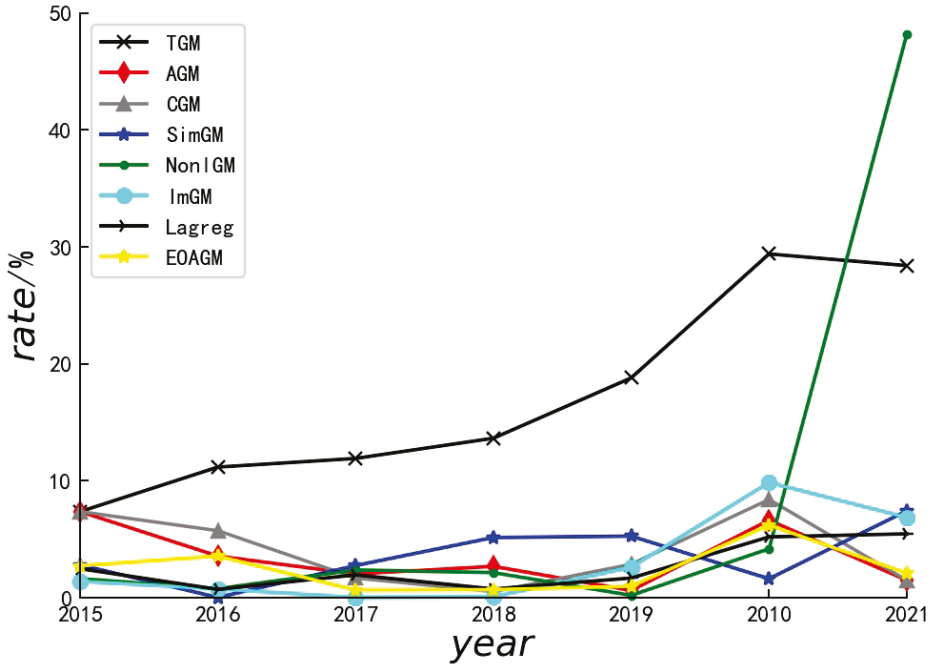


Figure 2. Algorithmic errors in GDP.

Table 2. Comparison of algorithmic errors in GDP analysis.

Errors (%)	Algorithms							
	TGM	**AGM**	**CGM**	**SimGM**	**NonlGM**	**ImGM**	**Lagreg**	**EOAGM**
w_{2015}	7.36	7.36	7.36	2.8	1.63	1.40	2.48	2.72
w_{2016}	11.17	3.56	5.73	0.02	0.77	0.749	0.712	3.56
w_{2017}	11.91	2.03	1.68	2.75	2.37	3.27×10^{-5}	1.95	0.65
w_{2018}	13.64	2.68	0.52	5.15	2.14	0.078	0.746	0.69
w_{2019}	18.8	0.62	2.86	5.26	0.20	2.60	1.68	0.98
w_{2020}	29.4	6.63	8.40	1.62	4.18	9.85	5.2	6.21
w_{2021}	28.4	1.49	1.48	7.43	48.2	6.86	5.46	2.09
w_{avg}	17.241	3.48	4.53	3.57	8.50	3.07	2.60	2.41

4.1.2. Indigenous Thermal Energy Prediction

TGM/CGM/AGM and EOAGM were applied to forecast indigenous thermal energy prediction for six buildings over the period from 8 to 13 January. The raw data presented in Table 1 were processed using Equation (9) to calculate daily prediction errors. For example, for the 8 January prediction error, the thermal load data on that date was predicted using either the preceding 6-day or 7-day sequences (with sequence length adjusted based on balance test results), yielding the predicted value $\hat{X}^{(0)}(k)$. The observed value $X^{(0)}(k)$, corresponding to 8 January, was directly extracted from the 8 January entry in Table 1. Figure 3 explicitly visualizes these errors from 8 to 13 January through line charts, en-

abling quantitative comparison of algorithmic accuracy (mean absolute error) and stability (temporal error variance).

As shown in Figure 3, both AGM and EOAGM outperform TGM and CGM. Furthermore, EOAGM consistently surpasses the adaptive model in prediction accuracy.

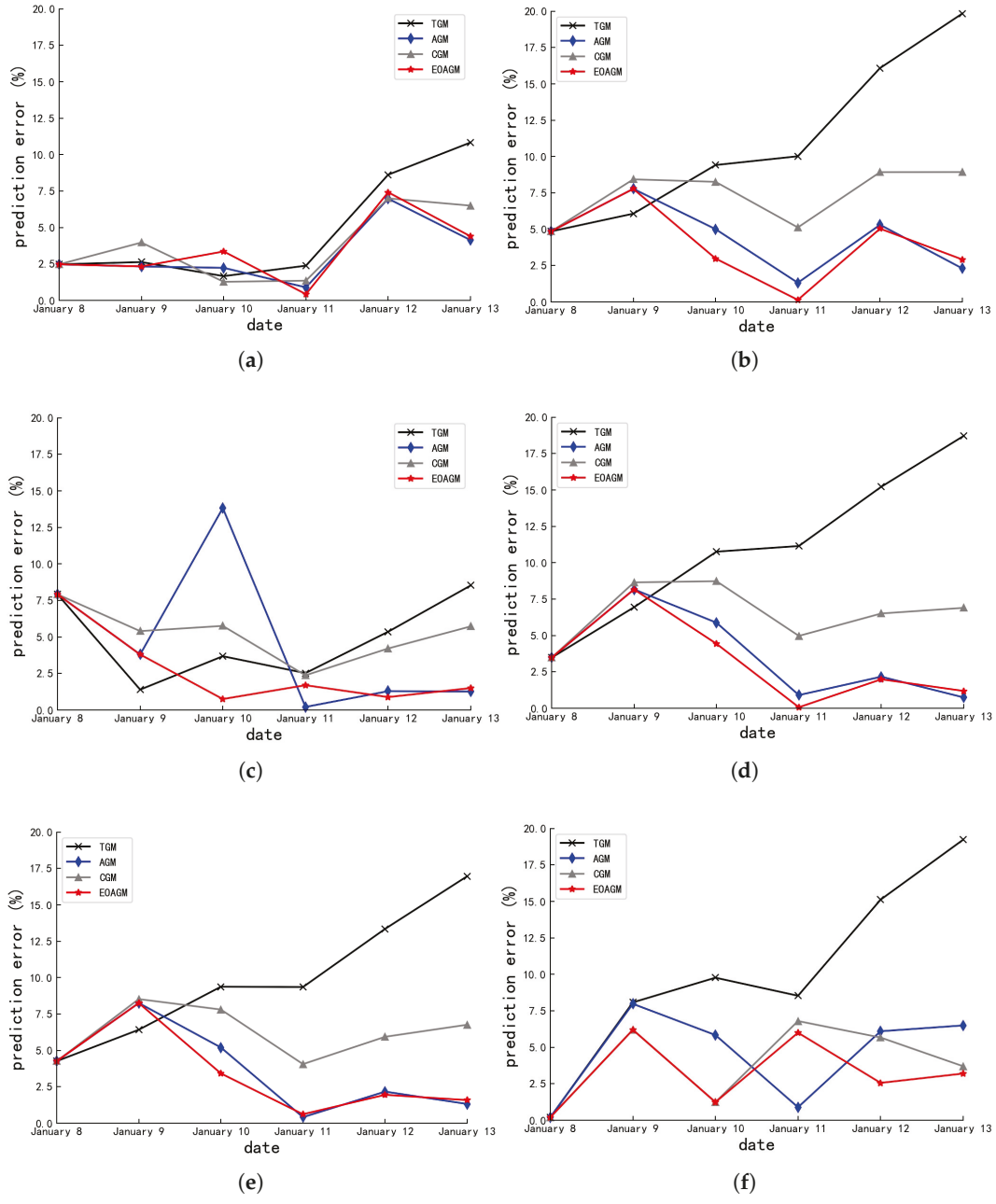


Figure 3. (**a**) represents the indigenous thermal energy prediction error for building No. 11 over the period from 8 to 13 January. (**b**) represents the indigenous thermal energy prediction error for building No. 12 over the period from 8 to 13 January. (**c**) represents the indigenous thermal energy prediction error for building No. 13 over the period from 8 to 13 January. (**d**) represents the indigenous thermal energy prediction error for building No. 14 over the period from 8 to 13 January. (**e**) represents the indigenous thermal energy prediction error for building No. 15 over the period from 8 to 13 January. (**f**) represents the indigenous thermal energy prediction error for building No. 16 over the period from 8 to 13 January.

Using the mean error as a test indicator for the four algorithms, as shown in Table 3, except for building 11, the average error of the EOAGM is the smallest in the other five build-

ings. Moreover, if the data fluctuate or there are anomalous data, the prediction accuracy of the EOAGM will be significantly improved.

Table 3. Comparison of average error results.

Algorithms	Average Errors (%)					
	No. 11	**No. 12**	**No. 13**	**No. 14**	**No. 15**	**No. 16**
TGM	4.76 ± 3.92	11.04 ± 5.82	4.902.89	11.04 ± 5.49	9.95 ± 4.61	10.16 ± 6.53
AGM	3.17 ± 2.13	2.31 ± 3.92	4.71 ± 5.26	3.55 ± 2.9	3.60 ± 2.90	3.97 ± 3.22
CGM	3.76 ± 2.51	7.42 ± 3.92	1.90 ± 1.84	6.54 ± 2.06	6.23 ± 1.83	4.58 ± 2.74
EOAGM	3.39 ± 2.37	3.95 ± 2.57	3.06 ± 2.75	3.22 ± 2.88	3.34 ± 2.74	3.22 ± 2.45

4.1.3. JYA-CMTD Prediction

To validate the universal applicability of our proposed model, we conducted comprehensive predictions of JYA-CMTD from 2017 to 2021 using TGM/CGM/AGM and EOAGM.

The errors (%) of each model for different years, as well as the mean errors, are presented in Table 4. As can be observed from Table 4, our algorithm achieves the smallest values in both mean error and maximum error, improving precision by at least 2.6% compared to baseline methods.

Furthermore, a line chart (Figure 4) was employed to illustrate the stability of the algorithms. The red line represents the error trend of our proposed algorithm, demonstrating its exceptional stability.

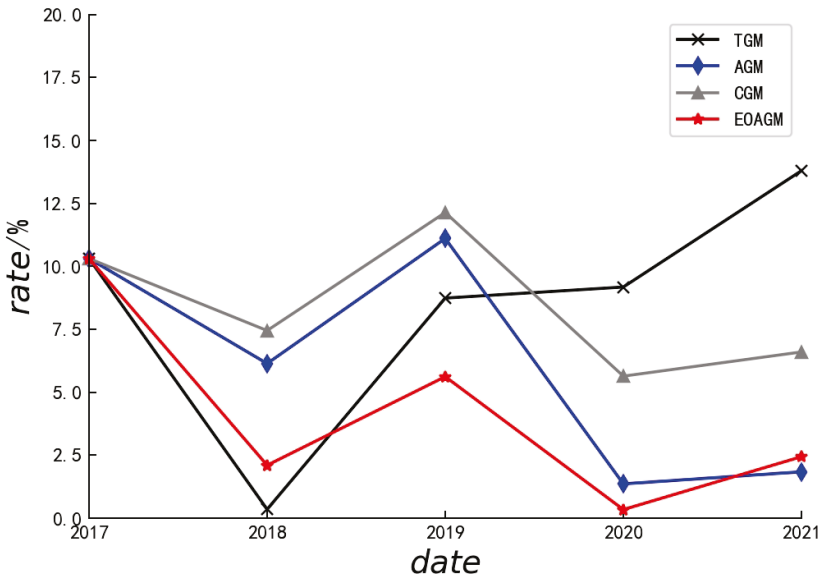


Figure 4. Algorithmic errors in JYA-CMTD.

Table 4. Comparison of algorithmic errors in JYA-CMTD analysis.

Errors (%)	Algorithms			
	TGM	**AGM**	**CGM**	**EOAGM**
w_{2017}	10.31	10.31	10.31	10.31
w_{2018}	0.35	6.14	7.45	2.10
w_{2019}	8.74	11.11	12.14	5.62
w_{2020}	9.18	1.36	5.64	0.34
w_{2021}	13.80	1.84	6.60	2.44
w_{avg}	8.47	6.84	8.42	4.16

4.1.4. On Model Comparison

As demonstrated in Tables 2–4, the proposed EOAGM model exhibits superior performance compared to the seven baseline methods. This enhanced performance can be attributed to the following key innovations:

Stability-Aware Adaptive Forecasting: The model incorporates stationarity testing and dynamic sequence length adjustment, effectively mitigating prediction errors caused by temporal fluctuations. This ensures robust performance under non-stationary conditions.

Predictive Parameter Optimization: A novel parameter optimization strategy is implemented by embedding predicted values into candidate sequences. This approach allows for the direct evaluation of their predictive efficacy in subsequent forecasting steps, thereby enhancing both adaptability and precision in dynamic system modeling.

4.2. Ablation Experiment

In this section, we perform necessary ablation studies by removing specific components to generate three ablation models.

1. EOAGM-A: We replace the dynamic adaptive adjustment of sequences with a cumulative aggregation mode for data prediction.
2. EOAGM-O: We remove optimal sequence selection for data prediction.
3. EOAGM-E: We omit stationarity detection for data prediction.

As can be seen from Table 5, removing one of the three components affects the accuracy of the algorithm's prediction.

Table 5. Variants on benchmark datasets.

Dateset	Average Errors (%)			
	EOAGM-A	**EOAGM-O**	**EOAGM-E**	**EOAGM**
w_{gdp}	4.01	2.43	3.08	2.41
w_{avg11}	3.48	3.17	3.39	3.39
w_{avg12}	6.85	4.28	4.16	3.95
w_{avg13}	4.87	4.53	3.21	3.06
w_{avg14}	6.35	3.55	3.22	3.22
w_{avg15}	5.91	3.60	3.34	3.34
w_{avg16}	4.27	3.68	3.54	3.22

5. Conclusions

To overcome challenges related to prediction instability and noise sensitivity in the TGM, this study proposes the EOAGM, an enhanced version of the AGM. Firstly, we replace the traditional cumulative generation sequence with an adaptive sequence generation method, which enhances the model's ability to manage dynamic time series. Secondly, a stationarity detection method is introduced to adjust the sequence length dynamically, shifting from fixed-length to elastic sequences to reduce the impact of data fluctuations. Finally, the parameter optimization process is improved by incorporating predicted values into the candidate sequence. This allows for evaluating whether the inclusion of predicted values in subsequent forecasts improves performance, particularly in the presence of data volatility or anomalies. By selecting the optimal sequence, the proposed model significantly improves both simulation accuracy and predictive precision. The EOAGM was empirically validated in the Jinan Municipal Thermal Energy System, demonstrating robust performance in addressing nonlinear dynamics and operational uncertainties. Furthermore, the model exhibits inherent versatility for domains requiring high-fidelity time series analysis. For instance, in airline operations, the EOAGM enables accurate

forecasting of monthly passenger volumes, which is critical for optimizing fleet scheduling, crew allocation, and revenue management under seasonal variations or exogenous disturbances. Similarly, as for power grid systems, high-frequency financial trading, and pattern prediction systems [58–60], it can predict monthly electricity generation patterns to mitigate supply–demand imbalances, enhance grid resilience, and facilitate renewable energy integration. These applications leverage the model's capacity to disentangle complex temporal dependencies while adaptively incorporating external covariates.

Author Contributions: Conceptualization, T.L. and Z.L.; methodology, T.L. and Z.L.; software, T.L. and J.N.; validation, G.Q., T.L., and C.J.; formal analysis, X.L. and Z.L.; investigation, X.L. and Z.L.; resources, X.L. and Z.L.; data curation, T.L.; writing—original draft, C.J.; writing—review and editing, G.Q.; visualization, T.L.; supervision, J.N. and Z.L.; project administration, T.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 62276155 and 62206157.

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

Acknowledgments: The authors are grateful to the editors and reviewers for their constructive comments, which have significantly improved this work.

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

References

1. Alqahtani, A.; Ali, M.; Xie, X.; Jones, M.W. Deep time-series clustering: A review. *Electronics* **2021**, *10*, 3001. [CrossRef]
2. Hu, Y.; Zhan, P.; Xu, Y.; Zhao, J.; Li, Y.; Li, X. Temporal representation learning for time series classification. *Neural Comput. Appl.* **2021**, *33*, 3169–3182. [CrossRef]
3. Liu, F.; Chen, L.; Zheng, Y.; Feng, Y. A prediction method with data leakage suppression for time series. *Electronics* **2022**, *11*, 3701. [CrossRef]
4. Wei, Y.; Wang, Y.; Du, M.; Hu, Y.; Ji, C. Adaptive shapelet selection for time series classification. In Proceedings of the International Conference on Computer Supported Cooperative Work in Design, Rio de Janeiro, Brazil, 24–26 May 2023; pp. 1607–1612.
5. Hu, Y.; Ji, C.; Jing, M.; Ding, Y.; Kuai, S.; Li, X. A continuous segmentation algorithm for streaming time series. In Proceedings of the Collaborate Computing: Networking, Applications and Worksharing: 12th International Conference, CollaborateCom 2016, Beijing, China, 10–11 November 2016; Proceedings 12; Springer: Berlin/Heidelberg, Germany, 2017; pp. 140–151.
6. Ji, C.; Hu, Y.; Liu, S.; Pan, L.; Li, B.; Zheng, X. Fully convolutional networks with shapelet features for time series classification. *Inf. Sci.* **2022**, *612*, 835–847. [CrossRef]
7. Hu, Y.; Ren, P.; Luo, W.; Zhan, P.; Li, X. Multi-resolution representation with recurrent neural networks application for streaming time series in IoT. *Comput. Netw.* **2019**, *152*, 114–132. [CrossRef]
8. Hu, Y.; Jiang, Z.; Zhan, P.; Zhang, Q.; Ding, Y.; Li, X. A novel multi-resolution representation for streaming time series. *Procedia Comput. Sci.* **2018**, *129*, 178–184. [CrossRef]
9. Liu, F.; Yin, B.; Cheng, M.; Feng, Y. n-Dimensional chaotic time series prediction method. *Electronics* **2022**, *12*, 160. [CrossRef]
10. Wan, R.; Tian, C.; Zhang, W.; Deng, W.; Yang, F. A multivariate temporal convolutional attention network for time-series forecasting. *Electronics* **2022**, *11*, 1516. [CrossRef]
11. Kim, M.; Lee, S.; Jeong, T. Time series prediction methodology and ensemble model using real-world data. *Electronics* **2023**, *12*, 2811. [CrossRef]
12. Yuan, H.; Liao, S. A time series-based approach to elastic kubernetes scaling. *Electronics* **2024**, *13*, 285. [CrossRef]
13. Li, G.; Yang, Z.; Wan, H.; Li, M. Anomaly-PTG: A time series data-anomaly-detection transformer framework in multiple scenarios. *Electronics* **2022**, *11*, 3955. [CrossRef]
14. Zhong, Y.; He, T.; Mao, Z. Enhanced Solar Power Prediction Using Attention-Based DiPLS-BiLSTM Model. *Electronics* **2024**, *13*, 4815. [CrossRef]
15. Ma, X.; Chang, S.; Zhan, J.; Zhang, L. Advanced Predictive Modeling of Tight Gas Production Leveraging Transfer Learning Techniques. *Electronics* **2024**, *13*, 4750. [CrossRef]
16. Dai, L.; Liu, J.; Ju, Z. Attention Mechanism and Bidirectional Long Short-Term Memory-Based Real-Time Gaze Tracking. *Electronics* **2024**, *13*, 4599. [CrossRef]

17. Hu, Y.; Liu, M.; Su, X.; Gao, Z.; Nie, L. Video moment localization via deep cross-modal hashing. *IEEE Trans. Image Process.* **2021**, *30*, 4667–4677. [CrossRef]
18. Zeng, B.; Luo, C.; Liu, S.; Bai, Y.; Li, C. Development of an optimization method for the GM(1,N) model. *Eng. Appl. Artif. Intell.* **2016**, *55*, 353–362. [CrossRef]
19. Masini, R.P.; Medeiros, M.C.; Mendes, E.F. Machine learning advances for time series forecasting. *J. Econ. Surv.* **2023**, *37*, 76–111. [CrossRef]
20. Molnár, A.; Csiszárik-Kocsir, Á. Forecasting economic growth with v4 countries' composite stock market indexes—A granger causality test. *Acta Polytech. Hung.* **2023**, *20*, 135–154. [CrossRef]
21. Wang, Z.X.; Li, Q.; Pei, L.L. A seasonal GM(1,1) model for forecasting the electricity consumption of the primary economic sectors. *Energy* **2018**, *154*, 522–534. [CrossRef]
22. Fan, G.; Li, B.; Mu, W.; Ji, C. The Application of the Optimal GM(1,1) Model for Heating Load Forecasting. In Proceedings of the 2015 4th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering, Xi'an, China, 12–13 December 2015; Atlantis Press: Paris, France, 2015.
23. Ren, Y.; Xia, L.; Wang, Y. An improved GM(1,1) forecasting model based on Aquila Optimizer for wind power generation in Sichuan Province. *Soft Comput.* **2024**, *28*, 8785–8805. [CrossRef]
24. Prakash, S.; Agrawal, A.; Singh, R.; Singh, R.K.; Zindani, D. A decade of grey systems: Theory and application–bibliometric overview and future research directions. *Grey Syst. Theory Appl.* **2023**, *13*, 14–33. [CrossRef]
25. Cai, L.; Wu, F.; Lei, D. Pavement condition index prediction using fractional order GM(1,1) model. *IEEE Trans. Electr. Electron. Eng.* **2021**, *16*, 1099–1103. [CrossRef]
26. Ding, S. A novel self-adapting intelligent grey model for forecasting China's natural-gas demand. *Energy* **2018**, *162*, 393–407. [CrossRef]
27. Han, H.; Jing, Z. Anomaly Detection in Wireless Sensor Networks Based on Improved GM Model. *Teh. Vjesn.* **2023**, *30*, 1265–1273.
28. Javanmardi, E.; Liu, S.; Xie, N. Exploring the challenges to sustainable development from the perspective of grey systems theory. *Systems* **2023**, *11*, 70. [CrossRef]
29. Zhang, K.; Yuan, B. Dynamic change analysis and forecast of forestry-based industrial structure in China based on grey systems theory. *J. Sustain. For.* **2020**, *39*, 309–330. [CrossRef]
30. Cheng, M.; Liu, B. Application of a novel grey model GM (1,1, exp× sin, exp× cos) in China's GDP per capita prediction. *Soft Comput.* **2024**, *28*, 2309–2323. [CrossRef]
31. Wang, Z.X.; Wang, Z.W.; Li, Q. Forecasting the industrial solar energy consumption using a novel seasonal GM(1,1) model with dynamic seasonal adjustment factors. *Energy* **2020**, *200*, 117460. [CrossRef]
32. Julong, D. Introduction to grey system theory. *J. Grey Syst.* **1989**, *1*, 1–24.
33. Li, J.; Feng, S.; Zhang, T.; Ma, L.; Shi, X.; Zhou, X. Study of Long-Term Energy Storage System Capacity Configuration Based on Improved Grey Forecasting Model. *IEEE Access* **2023**, *11*, 34977–34989. [CrossRef]
34. Delcea, C.; Javed, S.A.; Florescu, M.S.; Ioanas, C.; Cotfas, L.A. 35 years of grey system theory in economics and education. *Kybernetes* **2025**, *54*, 649–683. [CrossRef]
35. Chia-Nan, W.; Nhu-Ty, N.; Thanh-Tuyen, T. Integrated DEA Models and Grey System Theory to Evaluate Past-to-Future Performance: A Case of Indian Electricity Industry. *Sci. World J.* **2015**, *2015*, 638710.
36. Zhan, T.; Xu, H. Nonlinear optimization of GM(1,1) model based on multi-parameter background value. In Proceedings of the International Conference on Computer and Computing Technologies in Agriculture, Beijing, China, 29–31 October 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 15–19.
37. Ma, Y.; Wang, S. Construction and application of improved GM(1,1) power model. *J. Quant. Econ.* **2019**, *36*, 84–88.
38. Chen, F.; Zhu, Y. A New GM(1,1) Based on Piecewise Rational Linear/linear Monotonicity-preserving Interpolation Spline. *Eng. Lett.* **2021**, *29*, 3.
39. Wang, X.; Qi, L.; Chen, C.; Tang, J.; Jiang, M. Grey System Theory based prediction for topic trend on Internet. *Eng. Appl. Artif. Intell.* **2014**, *29*, 191–200. [CrossRef]
40. Liu, S.; Tao, L.; Xie, N.; Yang, Y. On the new model system and framework of grey system theory. In Proceedings of the 2015 IEEE International Conference on Grey Systems and Intelligent Services (GSIS), Leicester, UK, 18–20 August 2015; pp. 1–11.
41. Tang, L.; Lu, Y. An Improved Non-equal Interval GM(1,1) Model Based on Grey Derivative and Accumulation. *J. Grey Syst.* **2020**, *32*, 77.
42. Cheng, M.; Cheng, Z. A novel simultaneous grey model parameter optimization method and its application to predicting private car ownership and transportation economy. *J. Ind. Manag. Optim.* **2023**, *19*, 3160–3171. [CrossRef]
43. Zeng, X.; Xu, M.; Hu, Y.; Tang, H.; Hu, Y.; Nie, L. Adaptive edge-aware semantic interaction network for salient object detection in optical remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–16. [CrossRef]
44. Yuhong, W.; Jie, L. Improvement and application of GM(1,1) model based on multivariable dynamic optimization. *J. Syst. Eng. Electron.* **2020**, *31*, 593–601. [CrossRef]

45. Gianfreda, A.; Maranzano, P.; Parisio, L.; Pelagatti, M. Testing for integration and cointegration when time series are observed with noise. *Econ. Model.* **2023**, *125*, 106352. [CrossRef]
46. Chang, Y.; Park, J.Y. On the asymptotics of ADF tests for unit roots. *Econom. Rev.* **2002**, *21*, 431–447. [CrossRef]
47. Alam, M.B.; Hossain, M.S. Investigating the connections between China's economic growth, use of renewable energy, and research and development concerning CO_2 emissions: An ARDL Bound Test Approach. *Technol. Forecast. Soc. Change* **2024**, *201*, 123220. [CrossRef]
48. Hassan, M.K.; Kazak, H.; Adıgüzel, U.; Gunduz, M.A.; Akcan, A.T. Convergence in Islamic financial development: Evidence from Islamic countries using the Fourier panel KPSS stationarity test. *Borsa Istanb. Rev.* **2023**, *23*, 1289–1302. [CrossRef]
49. Lin, J.X.; Chen, G.; Pan, H.S.; Wang, Y.C.; Guo, Y.C.; Jiang, Z.X. Analysis of stress-strain behavior in engineered geopolymer composites reinforced with hybrid PE-PP fibers: A focus on cracking characteristics. *Compos. Struct.* **2023**, *323*, 117437. [CrossRef]
50. Asgari, H.; Moridian, A.; Havasbeigi, F. The Impact of Economic Complexity on Income Inequality with Emphasis on the Role of Human Development Index in Iran's Economy with ARDL Bootstrap Approach. *J. Dev. Cap.* **2024**, *9*, 35–56.
51. Yung, Y.F.; Bentler, P.M. Bootstrap-corrected ADF test statistics in covariance structure analysis. *Br. J. Math. Stat. Psychol.* **1994**, *47*, 63–84. [CrossRef]
52. Tian, W.; Zhou, H.; Deng, W. A class of second order difference approximations for solving space fractional diffusion equations. *Math. Comput.* **2015**, *84*, 1703–1727. [CrossRef]
53. Guo, Z.; Yu, J. The existence of periodic and subharmonic solutions of subquadratic second order difference equations. *J. Lond. Math. Soc.* **2003**, *68*, 419–430. [CrossRef]
54. Liu, S.F.; Forrest, J. Advances in grey systems theory and its applications. In Proceedings of the 2007 IEEE International Conference on Grey Systems and Intelligent Services, Nanjing, China, 18–20 November 2007; pp. 1–6.
55. Romano, J.L.; Kromrey, J.D.; Owens, C.M.; Scott, H.M. Confidence interval methods for coefficient alpha on the basis of discrete, ordinal response items: Which one, if any, is the best? *J. Exp. Educ.* **2011**, *79*, 382–403. [CrossRef]
56. Fu, Z.; Yang, Y.; Wang, T. Prediction of urban water demand in Haiyan county based on improved nonlinear optimization GM (1, N) model. *Water Resour. Power* **2019**, *37*, 44–47.
57. Dawar, I.; Dutta, A.; Bouri, E.; Saeed, T. Crude oil prices and clean energy stock indices: Lagged and asymmetric effects with quantile regression. *Renew. Energy* **2021**, *163*, 288–299. [CrossRef]
58. Han, Y.; Hu, Y.; Song, X.; Tang, H.; Xu, M.; Nie, L. Exploiting the social-like prior in transformer for visual reasoning. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024; Volume 38, pp. 2058–2066.
59. Tang, H.; Hu, Y.; Wang, Y.; Zhang, S.; Xu, M.; Zhu, J.; Zheng, Q. Listen as you wish: Fusion of audio and text for cross-modal event detection in smart cities. *Inf. Fusion* **2024**, *110*, 102460. [CrossRef]
60. Hu, Y.; Wang, K.; Liu, M.; Tang, H.; Nie, L. Semantic collaborative learning for cross-modal moment localization. *ACM Trans. Inf. Syst.* **2023**, *42*, 1–26. [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

Preserving Clusters in Synthetic Data Sets Based on Correlations and Distributions

Lucija Petricioli *, Luka Humski and Mihaela Vranić

University of Zagreb Faculty of Electrical Engineering and Computing, Unska 3, 10000 Zagreb, Croatia; luka.humski@fer.unizg.hr (L.H.); mihaela.vranic@fer.unizg.hr (M.V.)
* Correspondence: lucija.petricioli@fer.unizg.hr

Abstract: The rising popularity of machine learning has resulted in quality data becoming increasingly valuable. However, in some cases, the data are too sparse to effectively train an algorithm or the data cannot be disclosed to unaffiliated researchers due to privacy concerns. The sparsity of data may also affect various data analyses that require a certain volume of data to be accurate. One possible solution to the aforementioned problems is data generation. However, to be a viable solution, data generation must simulate real-life data well. To this end, this paper tests whether a previously presented iterative data generation method that generates synthetic data sets based on the attribute distributions and correlations of a real-life data set can faithfully reproduce a clustered data set. The approach is shown to be ineffective for the proposed application, and consequently, a new method is introduced that might preserve the clusters present in the real-life data set. The new method is demonstrated to not only preserve the clusters within the synthetic data set, but also improve the similarity of the attribute correlations of the synthetic data set and the real-life data set.

Keywords: clusters; correlations; distributions; iterative algorithm; synthetic data generation

1. Introduction

Synthetic data are often utilized to thoroughly test and evaluate the robustness of a model or solution. The tests could require, for example, incorporating numerous edge cases, utilizing large volumes of data, testing with data that fulfill certain criteria, etc. When testing solutions, the focus is often on robustness—if a method is proved to be robust, it is deemed to be accurate enough. However, the similarity of the synthetic data to the real-life data can easily be overlooked. Any solution that will, once deployed, process real-life data should be tested with synthetic data that are as similar as possible in volume, type, and relationships to those the solution will ultimately be processing. Generating quality synthetic data is not a simple task for multiple reasons, such as privacy concerns [1], data sparsity, or data volume. However, if researchers can obtain an amount of real-life data relevant for their model, these data should be used to generate synthetic data sets that are large enough for testing.

Synthetic data prove especially useful when any kind of model is being developed with identifying data. Since synthetic data can retain valuable features without sacrificing privacy [2], they can enable other experts that do not have direct access to a real-life data set to analyze patterns and provide solutions without endangering the privacy of those whose data the data set contains [3]. Synthetic data are especially valuable in healthcare [4] and finance [5]. However, both fields require data of high quality to create models that are of high quality.

Two important questions arise when the goal is generating quality synthetic data that reproduce any of the characteristics of a real-life data set being used as a starting point:

- Which parameters are needed to generate a quality synthetic data set that reproduces the features of the real-life data set?
- How can the quality of a synthetic data set be measured or even determined?

Neither of the questions have a definite answer. The answer to the first question depends on either the researchers who are generating the synthetic data set or those who are providing the parameters for data generation. A perfect reproduction of the real-life data set may not only lead to data duplication, since the synthetic data set would be the same as the real-life data set, but also privacy concerns, which is why a level of variation in the generated data is welcome. This variation simulates naturally occurring variation present in real-life data. Each additional parameter that defines the features that should be reproduced in the synthetic data could lead to overfitting, which is why a precise number of parameters cannot be defined outright. Similarly, the quality of a synthetic data set is dependent on the task the data set will be used for. Nonetheless, if the inter-attribute correlations or attribute distributions of a real-life data set are not reproduced in a synthetic data set, the synthetic data cannot be considered a good substitute for the real-life data and should not be used as such. This reasoning led to interest in the Ruscio–Kaczetow method [6], which reproduces the aforementioned measures well. The method also reproduces non-normal distributions, an important aspect when simulating real-life data, since they are often not distributed normally. However, the method directly generates a synthetic data set from real-life data, an unsuitable characteristic when working with real-life data sets that are not open to the public or that have sparse anomalies. Therefore, a modification enabling synthetic data generation based on inter-attribute correlations and attribute distributions was proposed in [7], wherein the authors show those parameters are enough to generate a synthetic data set that reproduces inter-attribute correlations and attribute distributions well.

Another important aspect that should be reproduced in synthetic data are the clusters or classes present in a real-life data set (if the data set contains clusters or classes), which is why this paper expands on previous research by exploring how clusters can be maintained within the synthetic data. The method described in [7] is thoroughly tested alongside a further modified approach that should reproduce clusters. Additional measures and evaluations on downstream tasks are also provided to ensure the synthetic data are similar to real-life data. It is shown that the method described in this paper reproduces clusters in synthetic data, unlike the method described in previous research.

This paper is organized as follows: Section 2 defines the terms frequently used in the paper and provides related work; Section 3 describes the method presented in this paper; Section 4 describes the data set used as the basis for data generation; Section 5 shows the results of the conducted tests; Section 6 analyzes the results; and Section 7 concludes the paper and describes future work.

2. Basic Term Definitions and Related Work

Terms coined for or frequently used in this paper are defined or clarified in the next subsection. The definitions are followed by an overview of various researched data generation methods, with a focus on modern, deep-learning-based generative methods. The methods preceding the method presented in this paper are also examined.

2.1. Basic Term Definitions and Explanations

The term **real-life data** (or **real-life data set**) is used to denote data that can be observed, measured, or obtained from real-world sources or events. They are the opposite of the data

in a **synthetic data set**; a data set constructed by an algorithm or a method that can create data. The terms **generated synthetic data set** and **generated data set** are also used for the aforementioned definition. A **starting data set** is the input required for a data generation method that creates data according to the characteristics of another data set (real-life or synthetic). Furthermore, the method that produces a synthetic data set is called a **synthetic data generator** or simply a **data generator**.

Please note that **cluster** and **class** are used interchangeably in this paper due to the nature of the real-life data set used for data generation. One class corresponds to one cluster in the example data set.

2.2. Different Approaches to Data Generation

The way synthetic data are generated varies depending on the downstream task they are generated for. In some cases, the characteristics of the data are less relevant than the number of data points, whereas in other cases reproducing the characteristics of real-life data is the main goal. Therefore, various algorithms and methods were reviewed to gain an overview of the field and further insight into the goals and challenges of data generation. The approaches were grouped according to apparent similarities. The following is a list of simplified data generation categories observed during research:

1. Methods that use a small sample or a small starting data set, the contents of which they replicate and add a certain amount of noise to [8,9];
2. Methods that input the starting data set into a simpler machine learning algorithm and train it, then extract elements of the trained solution to use for synthetic data generation [10–14];
3. Methods that leverage the capabilities of deep learning solutions to learn the characteristics of real-life data for tabular data synthesis, such as Generative Adversarial Networks (GANs) [15–23], Variational AutoEncoders (VAEs) [24–27], and Large Language Models (LLMs) [28–31];
4. Methods that rely on lists of values extracted from relevant (starting) data sets, where each attribute value of a synthetic data set is sampled from a corresponding value list [32–42];
5. Methods that use characteristics obtained from a starting data set to generate synthetic data [6,43–45].

In recent years, deep learning solutions have gained popularity due to the accessibility of generative models. Transfer learning and the possibility of fine-tuning existing models to specific tasks have also aided the adoption of such architectures. However, the variety of data and downstream tasks the data will be used for leads to great variety in the field [46]. Generalization is still an active problem due to the characteristics of the algorithms. In particular, deep learning solutions are prone to overfitting, mainly when their training sample is small [47], which is a possibility when using real-life data. In such cases, deep learning solutions are often outperformed by classic statistical methods [48]. Furthermore, they often require substantially more resources to achieve quality results, both when considering the size of the starting data set and the necessary processing power, especially when using an LLM [49] (such as in [28]). Another LLM specific issue is their susceptibility to hallucinations [50,51], which could greatly impact the quality of the synthetic data. GAN-based solutions can have issues reproducing the distributions of the original data set [25], as well as being particularly susceptible to model collapse [49]. Moreover, due to the adversarial nature of the training process, GAN models require careful hyperparameter tuning to achieve stable convergence [52]. Synthetic data generators based on VAEs often produce data that are too similar to the data from the original data set, often reducing overall synthetic data quality [25,49]. Therefore, machine learning solutions are frequently made

to solve a particular problem or generate a particular type of data set based on examples. In those cases when the examples are of real-life data, their tendency to overfit can lead to privacy concerns [53,54]. Also, explainability in deep learning is still lacking [55], especially in LLMs, a characteristic that could lead to further risks when generating synthetic data [56]. However, requiring real-life data is not unique to machine learning solutions; in the aforementioned list, only the last category can generate data without direct access to a starting (real-life) data set.

The aforementioned overview and working with personal data led the authors to explore methods from Category 5 in [7]. In this work, the authors used summary data (inter-attribute correlations and attribute distributions) to test generating quality synthetic inter-user relationship data based on real-life Facebook relationships. The authors showed that the method presented in [6] can be modified to generate data without using a real-life data set. This modification is necessary due to the original method resampling the real-life data set to form the basis of the synthetic data set. The modification of the method described in [7] uses the quantiles of each of the attributes of the starting data set, creating new data points by interpolating the quantiles. After analyzing the synthetic data sets generated by the modified approach, it was concluded the approach was a viable way of generating synthetic data without a direct need for a real-life data set.

The following subsection describes the original method presented in [6], and the steps needed to develop the modified method from [7]. The two approaches are the basis of the method proposed in this paper.

2.2.1. Ruscio–Kaczetow Method

The **Ruscio–Kaczetow method** (or the **original method**) is the method first proposed by Ruscio and Kaczetow in [6]. This method is a slight modification of the approach proposed by Ruscio et al. in [57]. The input of the Ruscio–Kaczetow method is a data set whose correlation matrix is calculated and whose data points are first sampled with repetition per each attribute. The sampled data points are the basis of the synthetic data set and there are as many of them as there are data points in the starting data set. After the sampling, the method instantiates three matrices: two are populated with normal values, and the third matrix temporarily stores the result of each iteration of the method. The matrices enable switching from calculation with normal values to calculation with non-normal values. The two random value matrices and the correlation matrix of the data set are used to calculate a weighted normal value for each sampled data point. The calculated values are saved into a placeholder matrix, then sorted and replaced with values from the sample data set. The correlation matrix of the resulting data set is calculated and compared to the correlation matrix of the starting data set. The comparison is based on the following formula:

$$RMSR = \sqrt{\frac{\sum r_{residual}^2}{\frac{k(k-1)}{2}}}, \tag{1}$$

where $r_{residual}$ denotes the elements under the main diagonal from the difference of the correlation matrix of the resulting data set and the correlation matrix of the starting data set, and k is the number of attributes present in the starting data set. If the stopping criterion of the method is met (the $RMSR$ value has not reduced for several iterations), the result of the calculation is the generated synthetic data set. The described steps have been written out in pseudocode form in Algorithm 1. Certain parts of the algorithm have been abstracted for easier viewing and understanding. A starting data set is the only requirement for generating synthetic data with this method.

Algorithm 1: Ruscio–Kaczetow data generation algorithm

Input: real-life data set D_{orig}
Output: synthetic data set D_{synth}

Function `GenData`(D_{orig})**:**
- $n \leftarrow$ number of rows(D_{orig}) `// Number of data points`
- $k \leftarrow$ number of columns(D_{orig}) `// Number of attributes`
- $rmsr_{best} \leftarrow 1$ `// Upper limit of RMSR`
- $t \leftarrow 5$ `// Number of trials`
- $C_{target}, C_{intermediate} \leftarrow$ correlation matrix(D_{orig}) `// Dimensions` $k * k$
- $D_{distribution} \leftarrow$ sample with repetition(D_{orig}, n) `// Dimensions` $n * k$
- sort($D_{distribution}$)
- $f \leftarrow$ parallel analysis($D_{distribution}, C_{target}$) `// No. of underlying factors`
- $S \leftarrow$ matrix(n, f) `// Matrix of shared components, dimensions` $n * f$
- $U \leftarrow$ matrix(n, k) `// Matrix of unique components, dimensions` $n * k$
- populate with random normal values(S, U)
- $counter \leftarrow 0$
- `/* Iterate until there is no improvement */`
- **while** $counter < t$ **do**
  - $W_{shared} \leftarrow$ calculate shared components($C_{intermediate}, f$) `// Shared component weight matrix, dimensions` $k * f$
  - $W_{unique} \leftarrow$ calculate unique components(W_{shared}) `// Unique component weight matrix, dimensions` $k * 1$
  - $D_{synth} \leftarrow$ calculate normal simulated values($S, W_{shared}, U, W_{unique}$)
  - sort(D_{synth})
  - replace values($D_{synth}, D_{distribution}$) `// (target, source)`
  - $C_{reproduced} \leftarrow$ correlation matrix(D_{synth})
  - $C_{residual} \leftarrow C_{target} - C_{reproduced}$
  - $rmsr_{current} \leftarrow \sqrt{\frac{\sum(\text{lower triangle}(C_{residual}))^2}{\frac{k(k-1)}{2}}}$ `// Use values below the diagonal in` $C_{residual}$
  - **if** $rmsr_{current} < rmsr_{best}$ **then**
    - $C_{best} \leftarrow C_{intermediate}$
    - $C_{residual\ best} \leftarrow C_{residual}$
    - $rmsr_{best} \leftarrow rmsr_{current}$
    - $C_{intermediate} \leftarrow C_{intermediate} + C_{residual}$
    - $counter \leftarrow 0$
  - **else**
    - $counter \leftarrow counter + 1$
    - $C_{intermediate} \leftarrow C_{best} + 0.5^{counter} * C_{residual\ best}$
  - **end**
- **end**
- `/* Construct the best solution found */`
- $W_{shared} \leftarrow$ calculate shared components(C_{best}, f)
- $W_{unique} \leftarrow$ calculate unique components(W_{shared})
- $D_{synth} \leftarrow$ calculate normal simulated values($S, W_{shared}, U, W_{unique}$)
- sort(D_{synth})
- replace values($D_{synth}, D_{distribution}$) `// (target, source)`
- **return** D_{synth}

For a clear and concise view of the steps necessary to generate synthetic data using the original Ruscio–Kaczetow method, a flowchart has been provided in Figure 1.

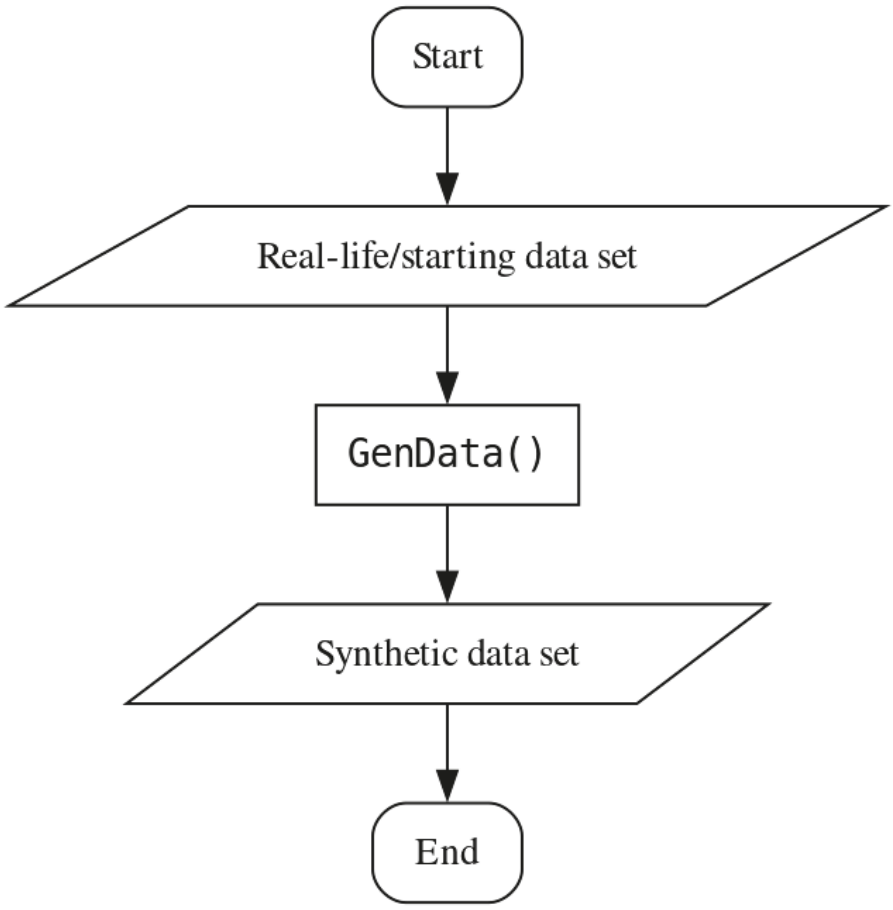


Figure 1. Flowchart for using the original synthetic data generation method.

2.2.2. Modified Ruscio–Kaczetow Method

The **modified method**, first proposed in [7], adds elements to and changes parts of the previously described method. Namely, instead of generating a synthetic data set from the starting data set directly, the modified method requires the correlation matrix and the quantiles of each of the attributes of the starting data set as input. The attribute quantiles are interpolated very finely and used as the basis of the synthetic data set. A dilation factor is set to create a large data set of interpolated values, from which the desired number of data points is sampled. This change of approach enables generating synthetic data sets of desired size. The sampling to form the attribute distributions from the original method is eliminated and the sampled interpolated data points are used for all the calculations directly. The correlation matrix of the starting data set is also calculated in advance and used as part of the input. These modifications offer several benefits when compared to the original method:

1. The method does not require a starting data set as direct input;
2. The data points used for data set generation are calculated and, therefore, less likely to exist in the starting data set;
3. There is no limit on the size of the data set the method can generate.

This method could allow researchers to work with data they cannot directly use due to privacy concerns since it does not require a starting data set as direct input. The synthetic data sets the modified method generates are more varied than those produced by the original method due to the use of calculated values instead of values sampled directly from

the starting data set. The theoretically limitless size of the synthetic data sets should be useful when an application requires more data points than are originally available.

This paper uses the described modified method as a benchmark to illustrate the effects of the proposed further changes to the method. It will be referred to as the **benchmark method** further in text.

Algorithm 2 describes the method used to interpolate the quantiles of a starting data set. It incorporates the possibility of choosing how numerous the resulting interpolated data should be (variable n), as well as how much larger the data set they are sampled from should be (variable d).

Algorithm 2: Data interpolation algorithm

Input: data set quantiles Q, resulting set size n, dilation factor d
Output: interpolated data set $D_{interpolated}$

Function `DataInterpolation(`Q, n, d`):`
  $D_{placeholder} \leftarrow Q$ `// Dimensions` $m * k$
  while number of rows$(D_{placeholder}) < (d * n)$ **do**
    `// The number of rows is the number of data points`
    $m \leftarrow$ number of rows$(D_{placeholder})$
    for $i \leftarrow 1, i < (2 * m - 1), i \leftarrow i + 2$ **do**
      $D_{tmp} \leftarrow \frac{D_{placeholder}[i] + D_{placeholder}[i+1]}{2}$
      `// Calculated for the whole row at index i`
      insert row$(D_{placeholder}, i)$ `// (target, index)`
      $D_{placeholder}[i] \leftarrow D_{tmp}$
    end
  end
  $D_{interpolated} \leftarrow$ sample without repetition$(D_{placeholder}, n)$ `// Dimensions` $n * k$
  return $D_{interpolated}$

Algorithm 3 details the changes made to Algorithm 1 when switching from generating synthetic data directly from a real-life data set to generating the data from an interpolated data set. Only the lines that need to be changed in Algorithm 1 are included.

Algorithm 3: Changes made to the Ruscio–Kaczetow algorithm

Input: interpolated data set $D_{interpolated}$, target correlation matrix C_{target}
...
Function `GenDataMod(`$D_{interpolated}$, C_{target}`):`
  $n \leftarrow$ number of rows$(D_{interpolated})$
  $k \leftarrow$ number of columns$(D_{interpolated})$
  ...
  $C_{intermediate} \leftarrow C_{target}$
  $D_{distribution} \leftarrow D_{interpolated}$
  ...

For a simple overview of the steps necessary for using the benchmark method, please refer to Figure 2. The steps below the dashed line do not require direct access to the starting data set, which enables external researchers to generate synthetic data based on attribute quantiles and inter-correlations.

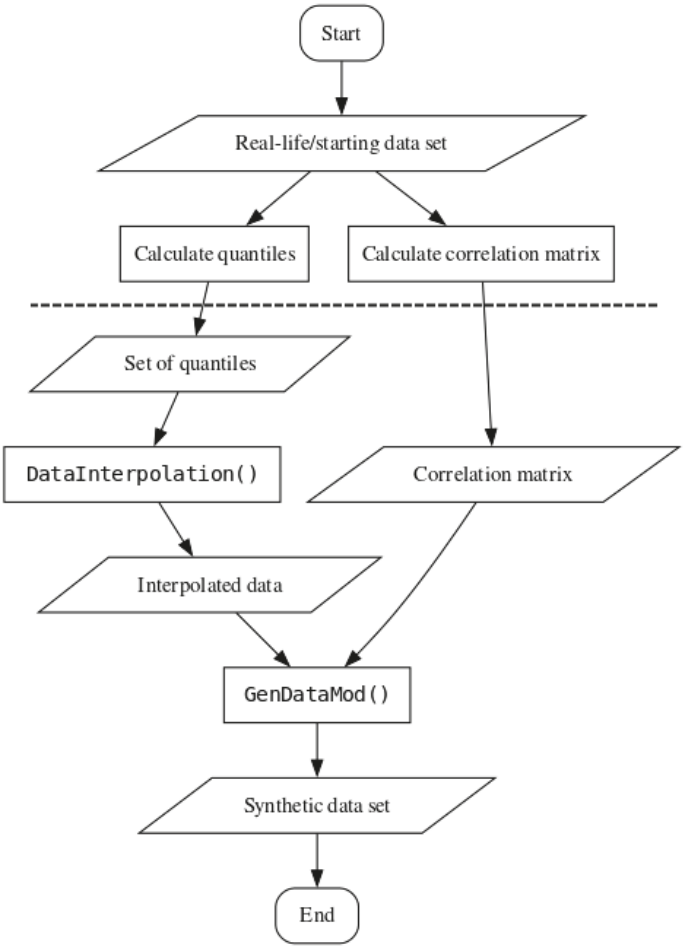


Figure 2. Flowchart for using the benchmark synthetic data generation method.

3. Methodology

After extensively testing the benchmark method with the correlations and distributions of a data set describing user interactions on Facebook (the analysis of the data set is available in [58]; benchmark method testing in [7]), it was not clear whether class or cluster information remained within the synthetic data. A suitable data set with clear clusters was required for determining whether the information was still extant after data generation. The clear clusters are necessary both for human readability and for clear clustering results when determining the quality of the resulting synthetic data set. The freely available "texture" data set was chosen for testing due to having suitably distinct clusters. The data set was filtered for greater result clarity.

Firstly, the benchmark method was used to generate a synthetic equivalent of the filtered "texture" data set. Analyses indicated that no clusters present in the real-life data set were preserved in the synthetic data set, even though the distributions and correlations were. Therefore, additional changes were made to the already modified method (following some of the recommendations from [57]).

When using the proposed approach, referred to as the **cluster preserving method** in the following text, the starting data set needs to be split into subsets according to class. The inter-attribute quantiles and correlation matrices of each subset need to be calculated separately, as well as the proportion of the subset in the starting data set. The quantiles and proportions are used to interpolate and sample a desired number of points that serve as the basis for data generation. The interpolated data points are then paired with the correlation matrix of the originating subset and run through the modified data generation algorithm (Algorithm 3). The resulting generated subsets are combined into a single synthetic data set

after data generation. This change in approach results in the following advantages when compared to the benchmark method:

1. The method preserves class or cluster information in the synthetic data;
2. The method maintains class or cluster proportions in the synthetic data.

The method even allows changing class proportions in the synthetic data set by providing different proportions during the interpolation step of data generation.

Algorithm 4 outlines the whole process of data generation with the cluster preserving method. The data generation starts with the real-life data set and the desired resulting synthetic data set size for clarity. However, the synthetic data set can be generated with a list of cluster quantiles, list of cluster correlation matrices, list of cluster proportions, a desired dilation factor, and the desired resulting synthetic data set size, entirely removing the need for directly using a real-life data set.

Algorithm 4: Generating a synthetic data set with preserved clusters

Input: real-life data set D_{orig}, dilation factor d, desired data set size n
Output: synthetic data set D_{synth}

Function `GenDataModCluster`(D_{orig}, d, n)**:**
  $L_{cluster} \leftarrow$ divide into clusters(D_{orig}) `// List of clusters`
  $D_{synth} \leftarrow$ empty data set()
  foreach *cluster* **in** $L_{cluster}$ **do**
    $C_{cluster} \leftarrow$ correlation matrix($cluster$)
    $q \leftarrow$ ceiling(0.1$*$ number of rows($cluster$))
      `// Number of quantiles to calculate`
    $Q_{cluster} \leftarrow$ quantiles($cluster, q$) `// (source, number of quantiles)`
    $proportion \leftarrow \dfrac{\text{number of rows}(cluster)}{\text{number of rows}(D_{orig})}$
    $D_{cluster\ interpolated} \leftarrow$ `DataInterpolation`($Q_{cluster}$, ($proportion * n$), d)
      `// (quantiles, desired cluster size, dilation factor)`
    $D_{intermediate} \leftarrow$ `GenDataMod`($D_{cluster\ interpolated}$, $C_{cluster}$)
    add($D_{synth}, D_{intermediate}$) `// (target, source)`
  end
  return D_{synth}

To enable quick comparison between the three described methods, a flowchart for the cluster preserving method can be seen in Figure 3. As with Figure 2, all the elements drawn underneath the dashed line do not require direct access to the starting data set. Also, if the starting data set does not contain clusters or classes, synthetic data generation should follow the steps outlined in Figure 2, as indicated by the right branch of the conditional block in Figure 3.

The cluster preserving method was used to generate a synthetic equivalent of the filtered "texture" data set as well. The synthetic data set was analyzed to determine whether the resulting distributions, correlations, and clusters were comparable to the correlations, distributions, and clusters of the real-life data set, as well as whether the resulting synthetic data set approximated the real-life data set more closely than the synthetic data set generated by the benchmark method.

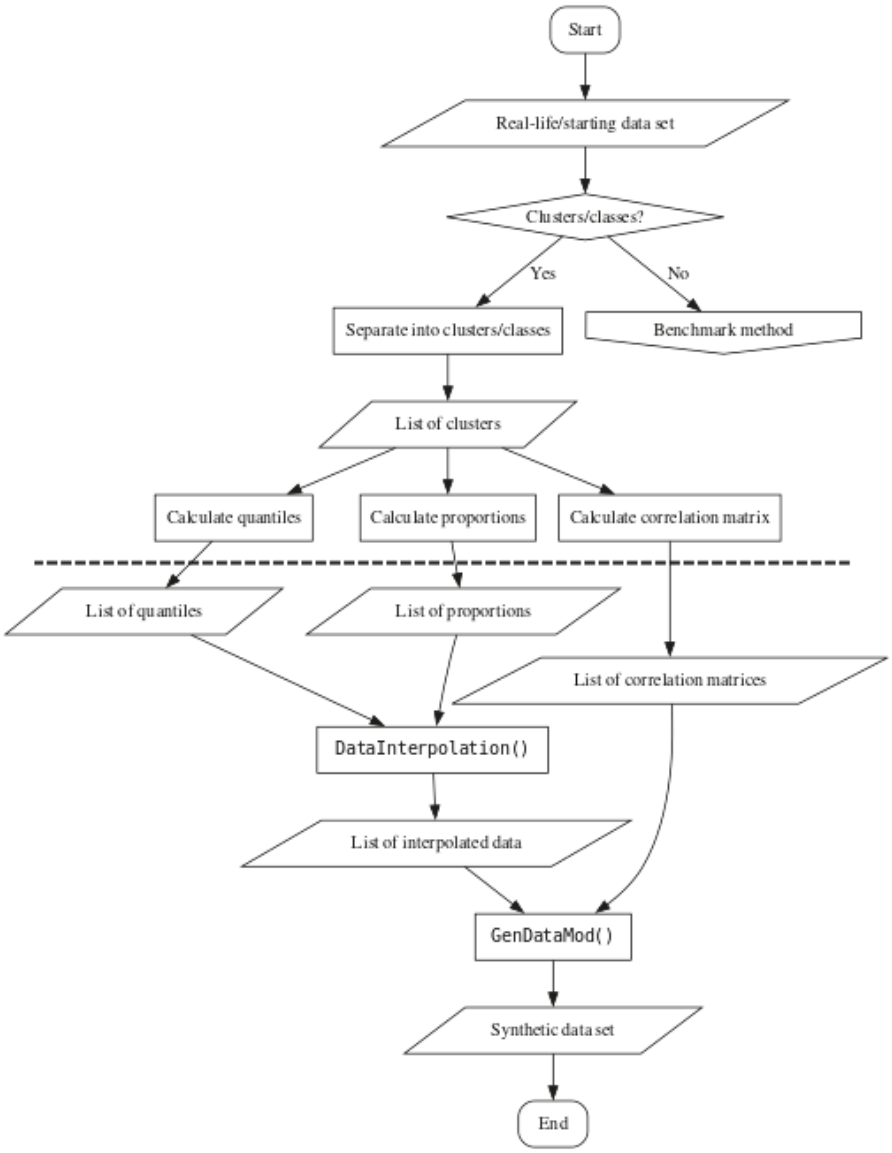


Figure 3. Flowchart for using the cluster preserving synthetic data generation method.

Implementation and Comparisons

Algorithm 4 was implemented in the programming language R ("a free software environment for statistical computing and graphics", https://www.r-project.org/, version 4.4.2), as were the previous versions. The flowcharts were created with the Graphviz software ("open source graph visualization software", https://graphviz.org/, version 12.1.0).

The quality of the generated synthetic data sets was determined through multiple analyses. The quantiles of the synthetic data sets were calculated and compared to the quantiles of the starting data set. The differences between the correlation matrices of the synthetic data sets and the correlation matrix of the starting data set were visualized. Statistical tests were used to determine the similarity of the synthetic data and the real-life data. The package "twosamples" ("Fast randomization based two sample tests.", https://cran.r-project.org/web/packages/twosamples/twosamples.pdf, accessed on 23 April 2025) was chosen due to enabling the comparison of two data samples with a different number of data points. The two statistical methods chosen from the package were "DTS" [59] and the Anderson–Darling statistical test [60]. The tests were used to test

the hypothesis that two samples come from the same distribution with an alpha value of $\alpha = 0.05$. Finally, the synthetic data sets were evaluated on a downstream task.

4. Data Set

The data set "texture" was used for data generation. The primary characteristics of the data set are: a sizable number of attributes (40), no missing values, numerous instances per class (500; overall 5500 instances), and clearly labeled classes (11; each corresponds to a real-life texture). It is also a real-life data set, which makes it suitable for data generation in the context of this paper. A spatial distribution of the data set can be seen in Figure 4 in three dimensions. The class labels can be seen on the right-hand side of the figure. Attributes 'V10', 'V23', and 'V30' (labeled as 'A10', 'A23', and 'A30' in the data set file) are used as the x-, y-, and z-axes, respectively.

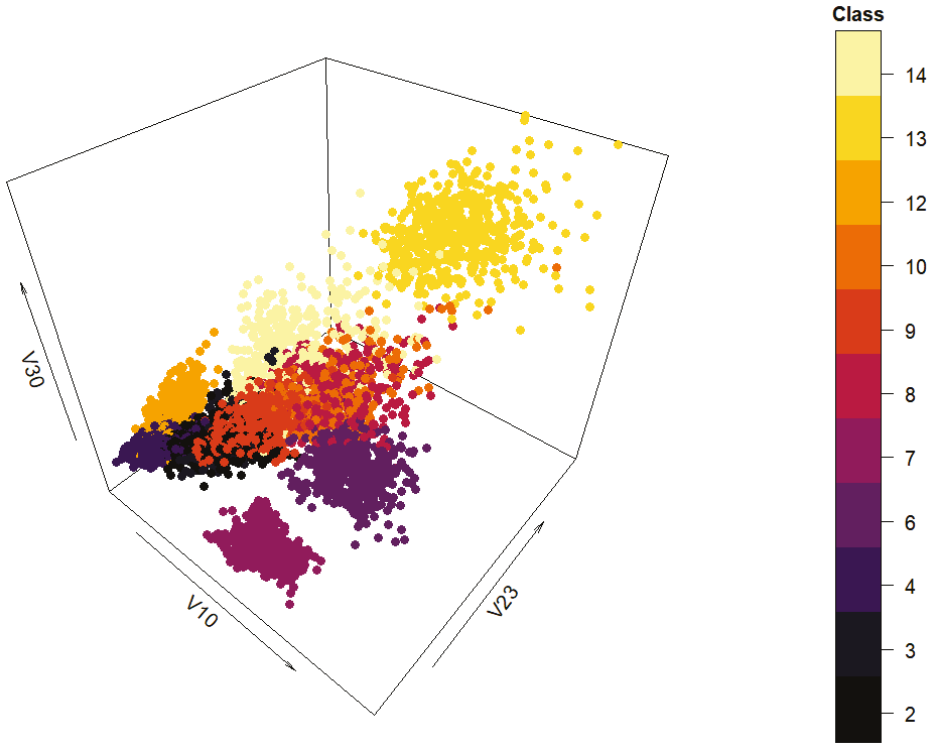


Figure 4. Three-dimensional view of the "texture" data set.

In order to make the preliminary results more human-readable and interpretable, 6 of the 40 attributes and 4 of the 11 classes were chosen for data generation—attributes 'A10', 'A19', 'A23', 'A25', 'A30', and 'A35' (renamed 'V10', 'V19', 'V23', 'V25', 'V30', and 'V35' in this paper) and classes '6', '7', '12', and '13'. The chosen attributes provide the most distance between the classes, and the chosen classes are spatially far apart. Both characteristics enable visual analysis by humans. The filtered data set has 2000 instances. The spatial relationship of the filtered data set can be seen in Figure 5 in three dimensions.

A second data set, the "yeast" data set, was also used to evaluate the proposed method. The description of the data set, the results of data generation, and their analysis can be found in Appendix A.

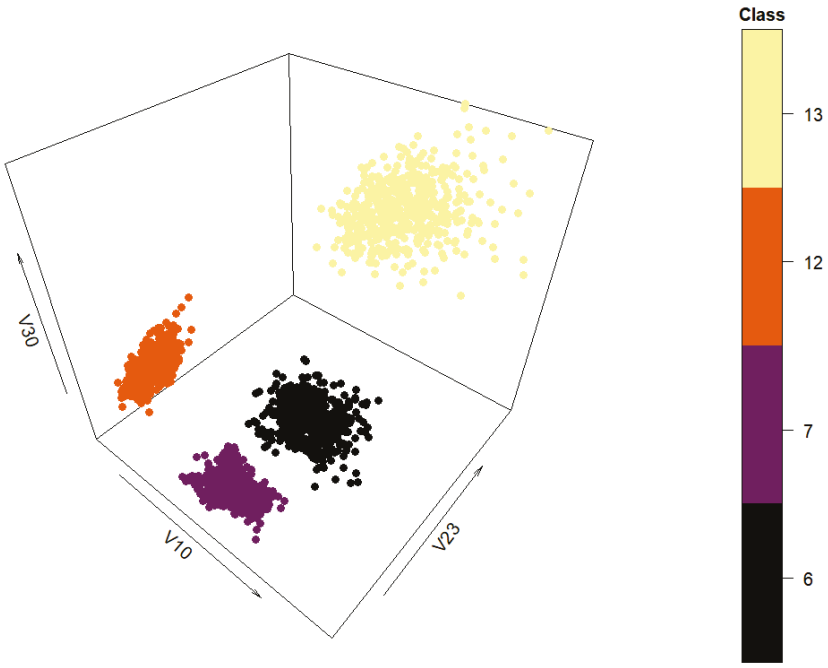


Figure 5. Three-dimensional view of the classes selected from the "texture" data set.

5. Results

Firstly, the entire filtered data set was used as the input for the benchmark method—201 quantiles were calculated for each of the attributes of the data set and the correlation matrix of the data set was calculated, after which both were used to generate a synthetic data set. The resulting data set can be seen in Figure 6.

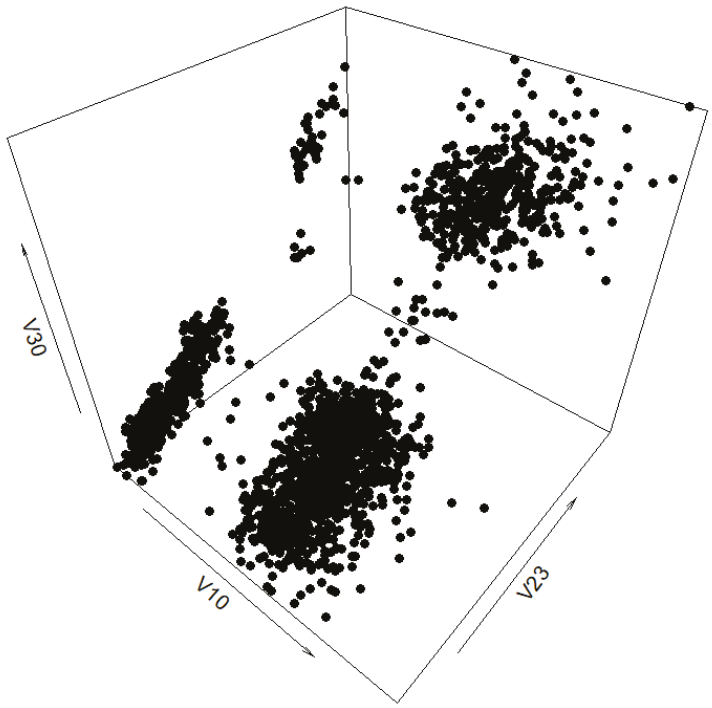


Figure 6. Three-dimensional view of the synthetic "texture" data set generated by the benchmark method **without** prior cluster separation for data interpolation.

However, after plotting the synthetic data set generated without any prior separation according to cluster, it was decided comparing synthetic data sets generated by the bench-

mark method and the cluster preserving method using the same interpolated data points would yield clearer and more mathematically sound results. Since the interpolation step can be separated from the generation step, the data points can be interpolated cluster by cluster, but then treated as a singular data set that can be used by the benchmark method (please note this paper compares the benchmark method and the cluster preserving method using the same data points obtained by interpolating cluster by cluster, unless otherwise stated). Therefore, 51 quantiles were calculated for each of the clusters chosen when filtering the data set, as were correlation matrices (both for the whole data set, to be used by the benchmark method, and one for each cluster, to be used by the cluster preserving method). The quantiles were then interpolated cluster by cluster. After this step, two synthetic data sets were generated from the same data points: one was generated by combining all the interpolated data points and using them and the calculated correlation matrix of the filtered real-life data set as the input for the benchmark method, whereas the other was generated by using the sets of data points interpolated from cluster quantiles and the corresponding correlation matrices as input for the cluster preserving method.

The synthetic data set generated by the benchmark method has 2000 data points and can be seen in Figure 7. Figure 8 is a comparison plot of the quantiles calculated from the real-life data set and the ones calculated from the synthetic data set. Figure 9 is the residual correlation matrix, which is the result of the following calculation:

$$C_{residual} = |C_{starting} - C_{synthetic}|, \tag{2}$$

where $C_{residual}$ denotes the matrix of correlation residuals, $C_{starting}$ the correlation matrix of the starting data set and $C_{synthetic}$ the correlation matrix of the synthetic data set. The Anderson–Darling and DTS statistical tests were also used to determine the similarity of the attributes of the real-life data set and the synthetic one. The *p*-values obtained from the tests can be observed in Table 1.

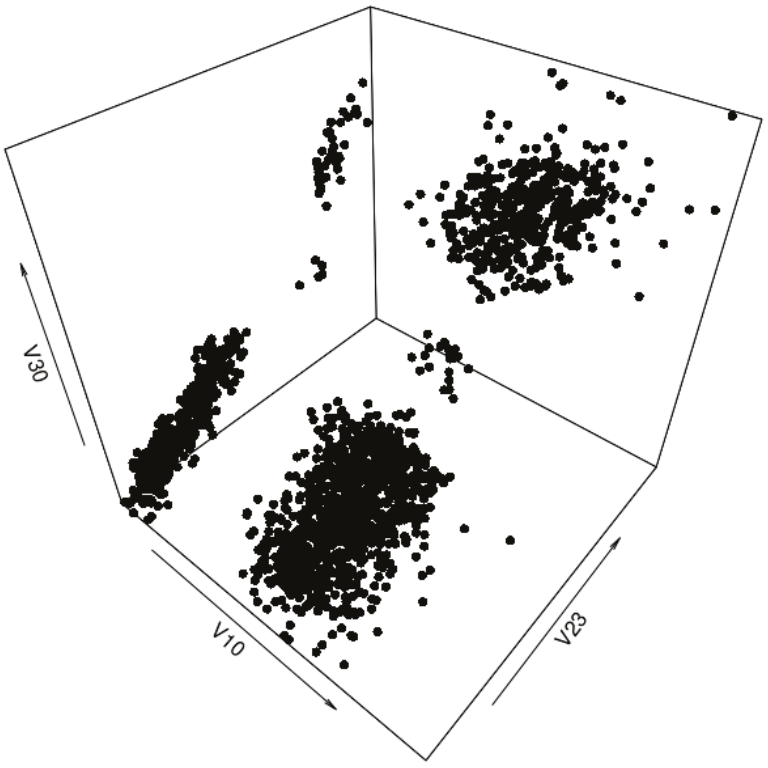


Figure 7. Three-dimensional view of the synthetic "texture" data set generated by the benchmark method **with** prior cluster separation for data interpolation.

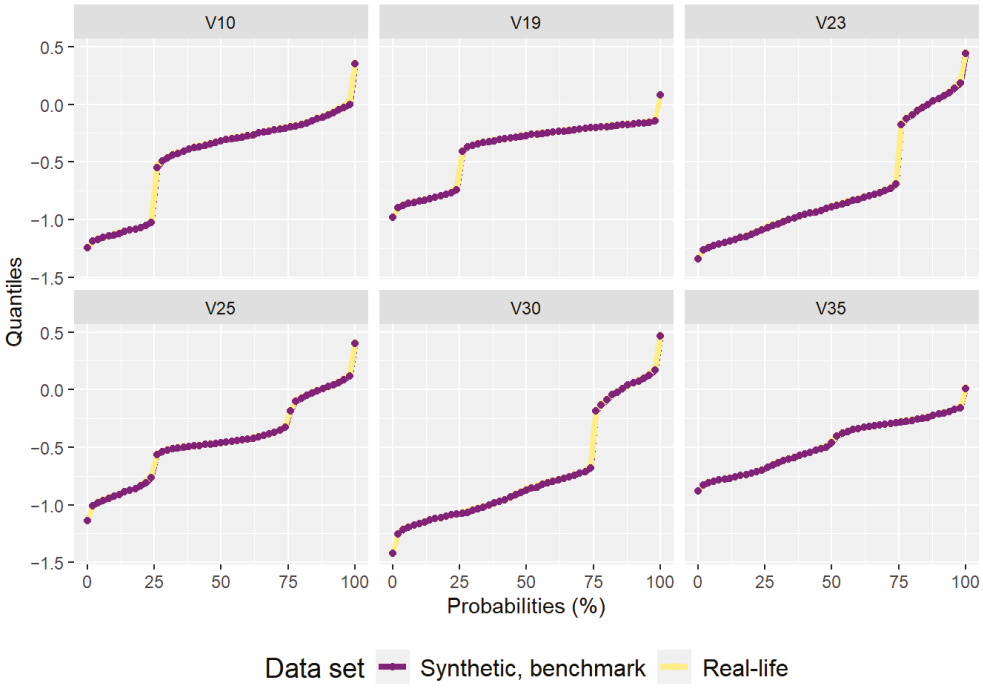


Figure 8. Plot of quantiles calculated for the real-life data set and the synthetic data set generated by the benchmark method.

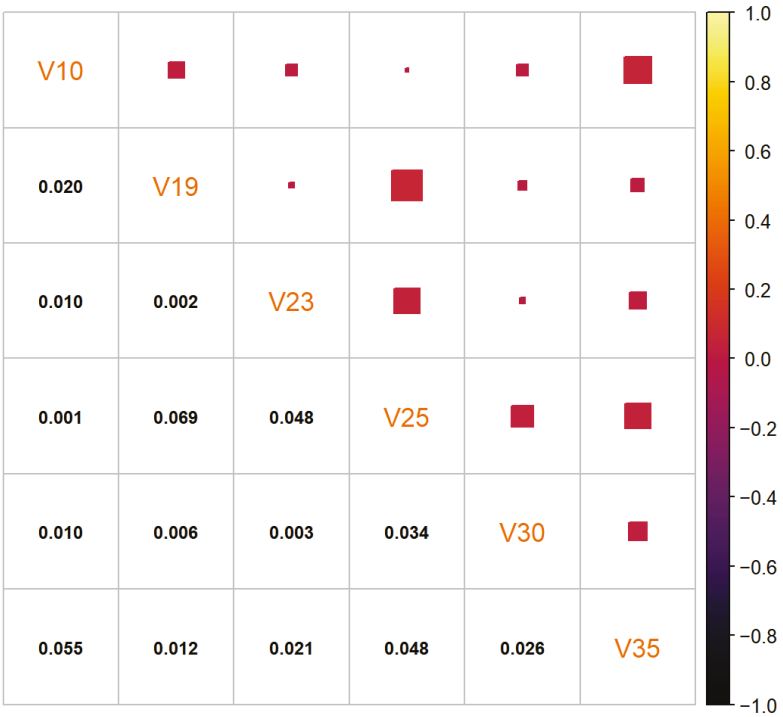


Figure 9. Residual correlation matrix of the synthetic "texture" data set generated by the benchmark method.

The synthetic data set generated by the cluster preserving method also has 2000 data points, with 500 data points being generated per each cluster (to correspond to the real-life data set). After generation, the individual cluster data sets were combined into a single synthetic data set that can be seen in Figure 10. A quantile comparison graph (Figure 11) and a residual correlation graph (Figure 12) were also plotted for this synthetic data set.

Please note that the residual correlation matrix was plotted for the combined data set. The matrix was calculated using (2). The results of the statistical tests can be seen in Table 2.

Table 1. Table of *p*-values calculated by the Anderson–Darling and DTS statistical tests comparing the distributions of the attributes of the filtered data set and the attributes of the synthetic data set generated by the benchmark method.

Attribute	Anderson–Darling	DTS
V10	0.4430	0.8700
V19	0.6650	0.9080
V23	0.8715	0.9650
V25	0.8955	0.9415
V30	0.8845	0.9815
V35	0.8820	0.8810

Table 2. Table of *p*-values calculated by the Anderson–Darling and DTS statistical tests comparing the distributions of the attributes of the filtered data set and the attributes of the synthetic data set generated by the cluster preserving method.

Attribute	Anderson–Darling	DTS
V10	0.4465	0.8525
V19	0.6925	0.9055
V23	0.8790	0.9560
V25	0.9025	0.9565
V30	0.8905	0.9755
V35	0.8690	0.8920

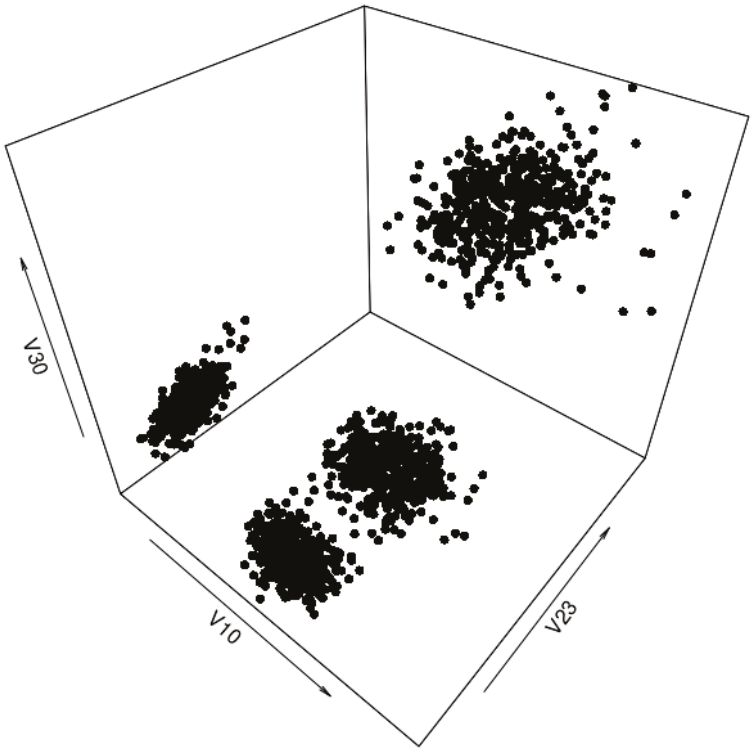


Figure 10. Three-dimensional view of the synthetic "texture" data set generated by the cluster preserving method.

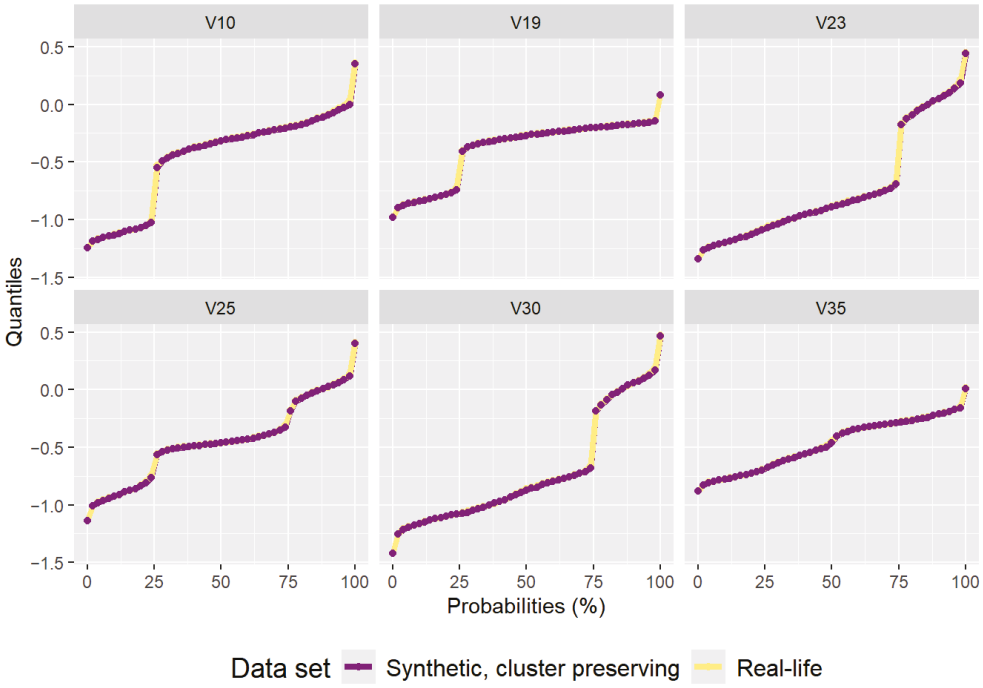


Figure 11. Plot of quantiles calculated for the real-life data set and the synthetic data set generated by the cluster preserving method.

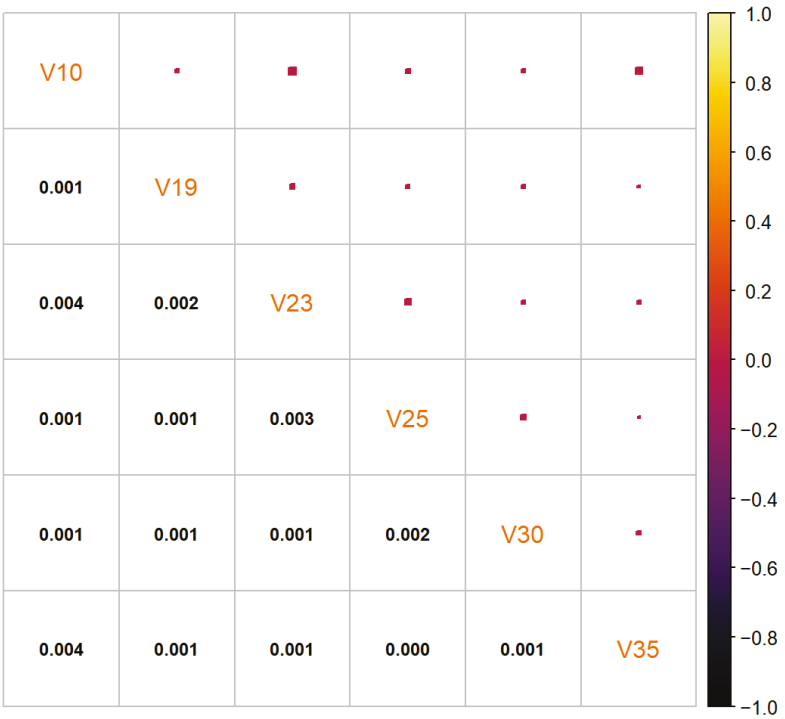


Figure 12. Residual correlation matrix of the synthetic "texture" data set generated by the cluster preserving method.

For a final comparison between the real-life data set, data set generated by the benchmark method, and data set generated by the cluster preserving method, the densities of each of the attributes were calculated and plotted on the same graph, as can be seen in Figure 13. The three densities overlap almost completely, an expected outcome that highlights the importance of choosing the right synthetic data set quality measures, an issue further discussed in detail in Section 6.

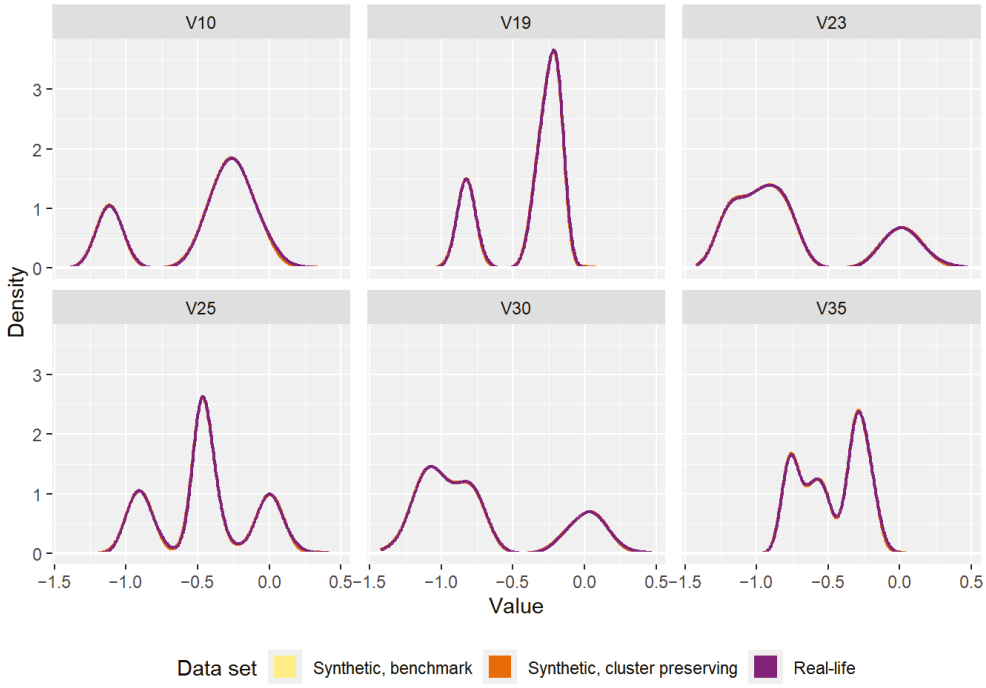


Figure 13. Comparison of the densities of the attributes of the real-life data set and the attributes of the synthetic data sets (one generated by the benchmark method, the other generated by the cluster preserving method).

Since the clusters in the synthetic data set generated by the benchmark method are not comparable with the ones present in the real-life data set, statistical tests calculated for each set of values an attribute takes for each of the distinct classes (and therefore, clusters) were calculated for the real-life data set and the synthetic data set with the preserved clusters. The results can be seen in Table 3.

Table 3. Table of *p*-values calculated by the Anderson–Darling and DTS statistical tests comparing the distributions of each of the attributes ("Attr." in table) of the filtered data set and the synthetic data set (cluster preserving method) depending on the class ("Cls." in table).

	Anderson–Darling				**DTS**			
Attr. \ **Cls.**	6	7	12	13	6	7	12	13
V10	0.4995	0.4920	0.4075	0.4585	0.7290	0.6615	0.5395	0.4890
V19	0.5975	0.5240	0.3090	0.3890	0.8010	0.5110	0.5860	0.4615
V23	0.4640	0.4095	0.3875	0.3855	0.5445	0.6055	0.5055	0.4620
V25	0.4630	0.4685	0.4350	0.4560	0.5185	0.5740	0.5620	0.6145
V30	0.4910	0.5930	0.4380	0.4005	0.5625	0.8075	0.6145	0.5195
V35	0.4080	0.5385	0.4880	0.3935	0.6030	0.6405	0.5715	0.4450

Multiple views of the filtered real-life data set, synthetic data set produced by the benchmark method, and the synthetic data set produced by the cluster preserving method can be found in Appendix B.

Downstream Task—Clustering

To test the hypothesis that the synthetic data generated by the cluster preserving method can be used for tasks that real-life data would otherwise be necessary for, both the real-life data set and the generated synthetic data set were subjected to clustering. Both were clustered using the k-means clustering method (the built-in `kmeans` function of the R programming language was used). Since both data sets retain the information which

cluster the data points belong to, one random data point from each cluster was used as the initial center for each cluster. The results of the clustering can be seen in Figure 14. Each of the clusters in both cases has 500 data points.

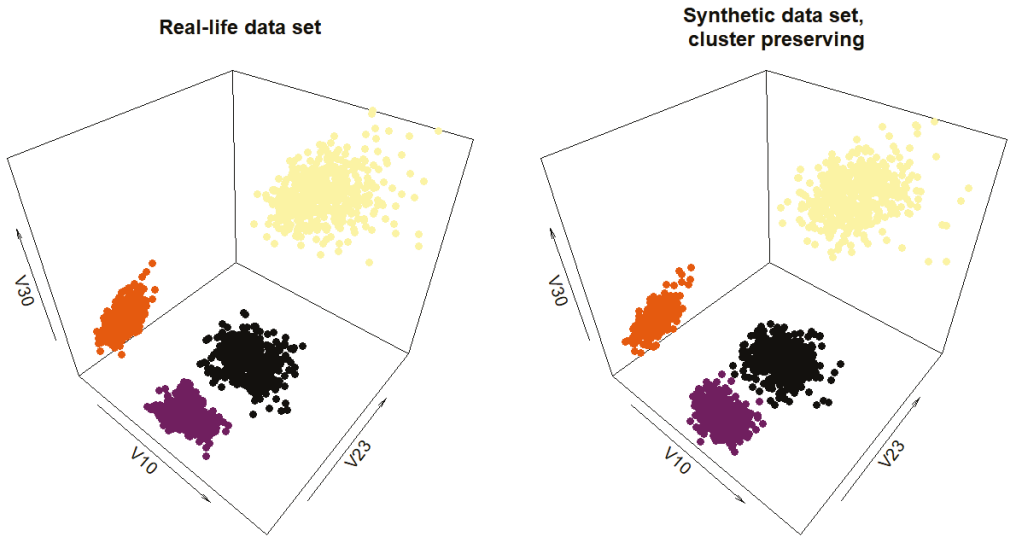


Figure 14. Plot of the clustering results for the real-life data set and the synthetic data set generated by the cluster preserving method.

The centroids of each of the clusters identified in the data sets were calculated as well and can be seen in Table 4.

Table 4. Calculated coordinates per attribute ("Attr." in table) for ("Cnt." in table) for both the real-life data set and the synthetic data set generated by the cluster preserving method, in tabular form.

	Real-Life				**Synthetic, Cluster Preserving**			
Cnt. / Attr.	1	2	3	4	1	2	3	4
V10	−0.1664	−0.3343	−1.1138	−0.2498	−0.1715	−0.3394	−1.1170	−0.2591
V19	−0.3040	−0.1919	−0.8237	−0.2290	−0.3064	−0.1937	−0.8271	−0.2319
V23	−0.8079	−1.1748	−0.9693	0.0277	−0.8123	−1.1788	−0.9735	0.0195
V25	−0.4270	−0.4762	−0.8987	0.0045	−0.4309	−0.4783	−0.9031	−0.0008
V30	−0.7797	−1.1489	−0.9835	0.0259	−0.7834	−1.1522	−0.9880	0.0184
V35	−0.3252	−0.5749	−0.7627	−0.2271	−0.3283	−0.5781	−0.7652	−0.2301

The within-cluster sum of squares can be seen in Table 5.

Table 5. Within-cluster sums of squares for both the real-life data set and the synthetic data set generated by the cluster preserving method, in tabular form.

	Real-Life				**Synthetic, Cluster Preserving**			
Cluster	1	2	3	4	1	2	3	4
Sum of squares	15.8481	13.6648	13.8702	36.2055	15.3899	13.4322	13.5477	34.8098

6. Discussion

Comparing Figures 5–7 leads to a clear conclusion that generating a synthetic data set using the benchmark method does not retain the spatial distribution of the data set used for generation. However, the following images, namely Figures 8 and 9, indicate that the quantiles and correlation matrix of the synthetic data set are still similar to those of the real-life data set. Table 1 also corroborates the possibility of the synthetic data set being a good reproduction of the real-life data set. The results of the various tests, both visual

and statistical, show the importance of choosing a humanly legible data set for initial tests. Without the clear visual comparison, the results could have been considered good enough, and an important aspect of the real-life data set would have been lost.

The synthetic data set generated by the cluster preserving method is visually similar to the real-life data set, unlike the synthetic data set generated by the benchmark method. A visual comparison of Figures 5, 7, and 10 leads to the conclusion that the starting data set should be separated into clusters (when it is a data set with clusters). It is important to note that the quantiles visible in Figure 11 are the quantiles for the full synthetic data set; therefore, even though the quantiles used to generate each of the clusters were exclusively the quantiles of that cluster, the full synthetic data set retains attribute distributions overall with minimal variation. Furthermore, since both of the synthetic data sets were generated from the same data points, the quantiles for each of the attributes are essentially the same. Therefore, even though the quantiles of the synthetic data sets are very similar to the quantiles of the real-life data set, they cannot help identify whether there is a difference in synthetic data set quality between the two data generation methods. However, the residual correlation matrices indicate a difference. A visual comparison of Figures 9 and 12 determines that splitting the starting data set by cluster when generating a synthetic data set results in the correlation matrix of the synthetic data set more closely matching the correlation matrix of the starting data set. When comparing the two numerically, the residual correlation matrix of the synthetic data set generated using the benchmark method has a sum of residuals above the main diagonal of 0.365 with an average residual of 0.024, whereas the residual correlation matrix of the synthetic data set generated using the cluster preserving method has a sum of 0.024 above the main diagonal with an average residual of 0.0016. Therefore, generating the synthetic data set cluster by cluster yields a reduction of correlation matrix residuals by at least an order of magnitude. Once again, even though the synthetic data set was generated with the correlation matrices of each of the clusters, the values of the correlation matrix of the full synthetic data set are almost equal to the values of the correlation matrix of the real-life data set.

The results of the statistical tests available in Tables 1 and 2 once again do not conclusively show whether one of the synthetic data sets simulates the real-life data set more closely. Neither the Anderson–Darling test nor the DTS test give a clear indication which of the two synthetic data sets is of better quality. However, since both tests randomly resample the data sets they are calculated with, the differences between the observed p-values are likely due to the seed the two test functions are seeded within the R programming language. The hypothesis that the synthetic attributes of either of the synthetic data sets belong to the distribution of the attributes of the real-life data set cannot be rejected. It is important to note that, even though the Anderson–Darling statistical test is one of the most frequently used statistical tests in scientific literature, DTS likely has more stable properties when presented with various distributions [59]. Since the distributions of the real-life data set are distributions of real-life properties, they cannot be expected to conform to standard mathematical distributions, which could explain the lower p-values observed in the Anderson–Darling test results.

Additionally, Figure 13 confirms both the synthetic data generation and subsequent quality analyses of synthetic data sets need to be executed carefully. Using the same data points for both synthetic data sets ensures their densities overlap completely. Similarly, due to the interpolation step of the synthetic data generation methods, the densities of the synthetic attributes correspond to the densities of the real-life attributes.

Finally, the p-values comparing the real-life data set and the synthetic data set generated using the cluster preserving method, calculated for each of the attributes depending on the cluster the data belong to (Table 3), show that the hypothesis the two samples are

part of the same distribution cannot be rejected. The difference between the two statistical tests is consistent with the trend of the results of the previous tests (Tables 1 and 2), where the results of the Anderson–Darling test are poorer than the results of the DTS test. Even though each of the clusters of the synthetic data set was generated with the corresponding real-life cluster attributes, the overall distributions of the attributes are mathematically more similar than the distributions of the attributes within the clusters. It is possible this is due to the number of data points in each of the clusters.

The downstream task (Section 5) further corroborates the assessment that the synthetic data retain a significant number of the qualities of the real-life data set. The clusters were perfectly identified, the calculated centroid coordinates for both the real-life data set and the synthetic data set are very close in value, as are the within-cluster sums of squares.

The analyses indicate that the chosen number of quantiles calculated from the real-life data set (51 per cluster with four clusters, therefore 204—around 10% of the number of data points in the real-life data set) is enough to create a synthetic data set whose attribute distributions, correlations, and classes correspond to those of the real-life data set. Even though the attributes were separated by cluster, resulting in the interpolated data points being sampled from a segment of the real-life attribute distributions and the final synthetic clusters being generated with just the cluster correlation matrices, the overall measures conform to the measures of the real-life data set.

All the tests run on the three data sets also emphasize the importance of thoroughly analyzing the data set being used as the starting data set for synthetic data generation. The method described in this paper is not only suitable but necessary when generating a data set with clustered data. On the other hand, if the starting data set does not have clusters, it should be generated outright, without splitting into clusters. A synthetic data set that could be substituted for a real-life data set can only be generated from the characteristics of a well-studied data set. Furthermore, once the synthetic data set is generated, multiple characteristics of this data set should be compared to those of the real-life data set, especially if the final goal is to have a synthetic data set that could in fact be used for analyses instead of the real-life data set. This can only be performed by researchers that have direct access to the real-life data set.

7. Conclusions

This paper presents an adaptation of the synthetic data set generation method presented in [7] that ensures the preservation of clusters in the synthetic data it generates. The previously proposed method generates a synthetic data set based on the distributions of individual attributes as well as the inter-attribute correlations of a starting data set, and achieves significant overlap between those measures in the synthetic and starting data sets. However, it demonstrates limited effectiveness when applied to starting data sets that contain clusters. In such cases, although correlations and attribute distributions are retained, the clusters characteristic of the original data set are typically lost in the synthetic counterpart.

The method proposed in this paper addresses this deficiency by introducing an adaptation that preserves not only attribute distributions and inter-attribute correlations, but also the inherent clustering structure of the original data set.

An important advantage of the proposed cluster preserving method is that it enables the generation of data sets larger than the original, while maintaining the relative proportions of all classes present in the source data.

Additionally, an important feature of the proposed approach is also its ability to alleviate data scarcity, which is useful in contexts where machine learning algorithms require large quantities of data to effectively learn relevant patterns. Since the method

produces previously unseen data points, it could also help mitigate overfitting in artificial neural networks and similar models.

Another notable benefit is the potential to facilitate greater data sharing. The method could allow organizations to generate synthetic data sets that obscure sensitive real-life data while retaining the analytical value of said data. This could enable external researchers to develop solutions based on these data sets without compromising privacy—an especially valuable feature in sensitive domains such as healthcare and finance.

It is also worth emphasizing that, unlike many deep-learning-based synthetic data generators, the method presented here is explainable.

Future study should investigate the assumptions related to machine learning and further assess how the proposed method could address common issues associated with such algorithms. Moreover, extending the method to support data sets with multiple nominal attributes beyond the class label represents a promising direction for future research. Such an enhancement would significantly broaden the range of data sets which this method could be applied to.

Author Contributions: Conceptualization, L.H., M.V. and L.P.; methodology, M.V., L.H. and L.P.; software, L.P.; validation, L.P., L.H. and M.V.; formal analysis, L.P., L.H. and M.V.; investigation, L.P., L.H. and M.V.; resources, M.V., L.H. and L.P.; writing—original draft preparation, L.P.; writing—review and editing, L.H., M.V. and L.P.; visualization, L.P. and L.H.; supervision, M.V. and L.H.; funding acquisition, M.V. and L.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been supported by the European Regional Development Fund under the grant PK.1.1.10.0007 (DATACROSS).

Informed Consent Statement: Not applicable.

Data Availability Statement: The "texture" data set, used to test the hypotheses presented in this work, is freely available as part of the repository of the Knowledge Extraction based on Evolutionary Learning (KEEL) software tool at: https://sci2s.ugr.es/keel/dataset.php?cod=72 (accessed on 13 March 2025). Furthermore, the "yeast" data set, used to further test the hypotheses, is also freely available as part of the same repository at: https://sci2s.ugr.es/keel/dataset.php?cod=1063 (accessed on 21 April 2025).

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

Abbreviations

The following abbreviations are used in this manuscript:

GAN	Generative Adversarial Network
LLM	Large Language Model
RMSR	Root Mean Square Residual
SVM	Support Vector Machine
VAE	Variational AutoEncoder

Appendix A. Analysis and Synthetic Data Generation for the "Yeast" Data Set

The "yeast" data set is an additional data set used to test the method described in this paper. The data set consists of 8 attributes, 1484 instances, and 10 classes the instances can belong to. There are no missing values in the data set. All the attributes are numerical in value. The number of instances per class is highly imbalanced, as can be seen in Table A1. As with the "texture" data set, the "yeast" data set is a real-life data set.

Table A1. Number of instances per class in the "yeast" data set.

Class	CYT	ERL	EXC	ME1	ME2	ME3	MIT	NUC	POX	VAC
No. of instances	463	5	35	44	51	163	244	429	20	30

The spatial distribution of the data set can be seen in Figure A1. Since the graph clearly indicates clustering would not be an appropriate downstream task for this data set, classification was chosen instead.

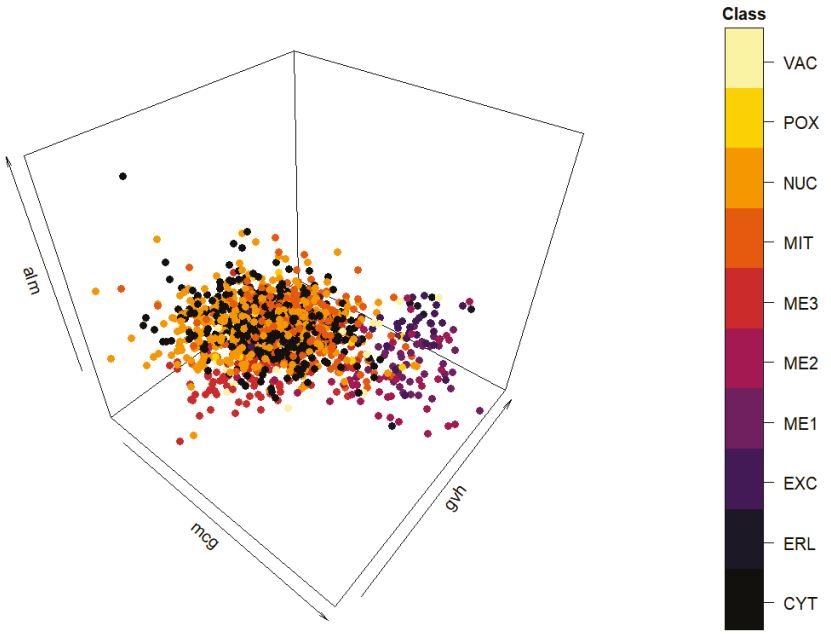


Figure A1. Three-dimensional view of the "yeast" data set.

This data set posed two challenges when generating synthetic data. The first challenge was that the values of the attributes 'erl' and 'pox' have very little variation. When splitting the data into classes before data generation, some of the subsets had only one value of these attributes, which led to the occurrence of undefined values (denoted by the logical constant `NA` in the R programming language) when calculating the class correlation matrices. There is a simple solution that can be built into the synthetic data generation methods (either the benchmark method or the cluster preserving one) without losing any previous functionality or generality—the attributes with no variance are excluded from the data interpolation and data generation step. Once data generation has been carried out, the attributes are added back into the synthetic data set. The second challenge stems from the smallest cluster having only five instances. Namely, Algorithm 3 uses parallel analysis to determine the number of underlying factors in a starting data set (if the value is not explicitly stated when calling the function in a program). This entails creating an $n * k$ (number of instances * number of attributes) sized matrix and populating it with attribute instances sampled with replacement from the starting data set. This matrix is used to calculate a correlation matrix whose eigenvalues are then computed. The small size of the 'ERL' class makes the probability of sampling the same value for all the instances of an attribute fairly likely. This leads to a higher likelihood of `NA` values occurring in the correlation matrix. The solution to this challenge can also be built into the approaches without changing their prior behavior and generality—the matrix populated with randomly sampled values should be checked for value uniqueness. If the values of the instances of an attribute are all the same, the value sampling should be redone. The sampling should be repeated until there are no columns

(attributes) with just one value. Once the two challenges have been solved, a synthetic "yeast" data set can be generated.

Synthetic data set generation closely followed the approach described in Section 3. The real-life data set was split by class. Quantiles, class proportions, and correlation matrices were calculated for each of the classes (the correlation matrix of the whole real-life data set was calculated for the benchmark method). The quantiles and the class proportions were used to create an interpolated data set for each of the classes. Those data and the corresponding correlation matrices were used to generate synthetic classes that were combined into a single data set (the interpolated data were also combined into one starting data set and used alongside the whole data set correlation matrix to generate a synthetic data set using the benchmark method).

The synthetic data set generated by the benchmark method can be seen in Figure A2. It has 1483 data points. Even though the data points are not of different colors, it is visible that the data do not follow the distribution of the real-life data set. The quantiles are almost perfectly replicated, as can be seen in Figure A3, as with the "texture" data set. The very low variance of the 'erl' and 'pox' attributes can clearly be seen in the quantiles. The residual correlation matrix can be seen in Figure A4. The sum of the residuals in the matrix above the main diagonal is 0.586 and the average residual value is 0.021. The results of the statistical tests can be seen in Table A2. Due to the low variance of some of the attributes the statistical tests reject (Anderson–Darling) or almost reject (DTS) the hypothesis the synthetic data are from the same distribution as the real-life data set.

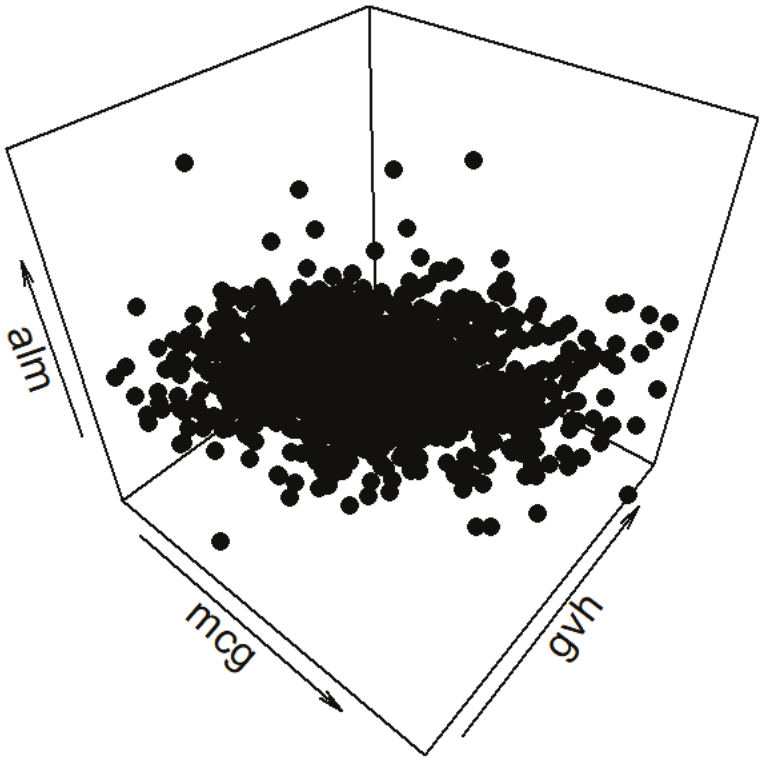


Figure A2. Three-dimensional view of the synthetic "yeast" data set generated by the benchmark method.

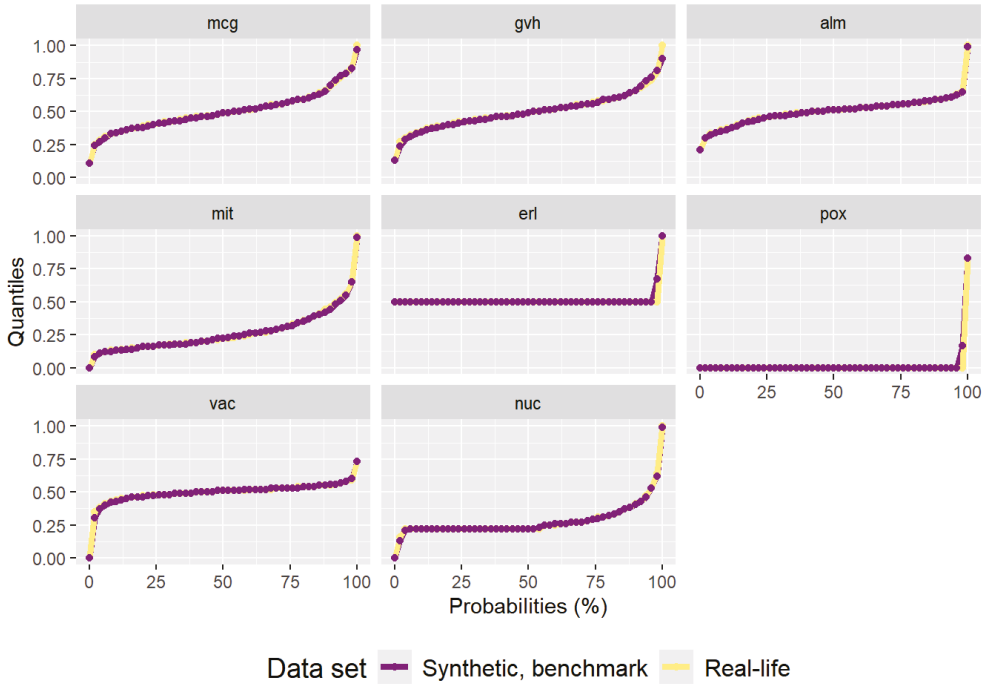


Figure A3. Plot of quantiles calculated for the real-life "yeast" data set and the synthetic data set generated by the benchmark method.

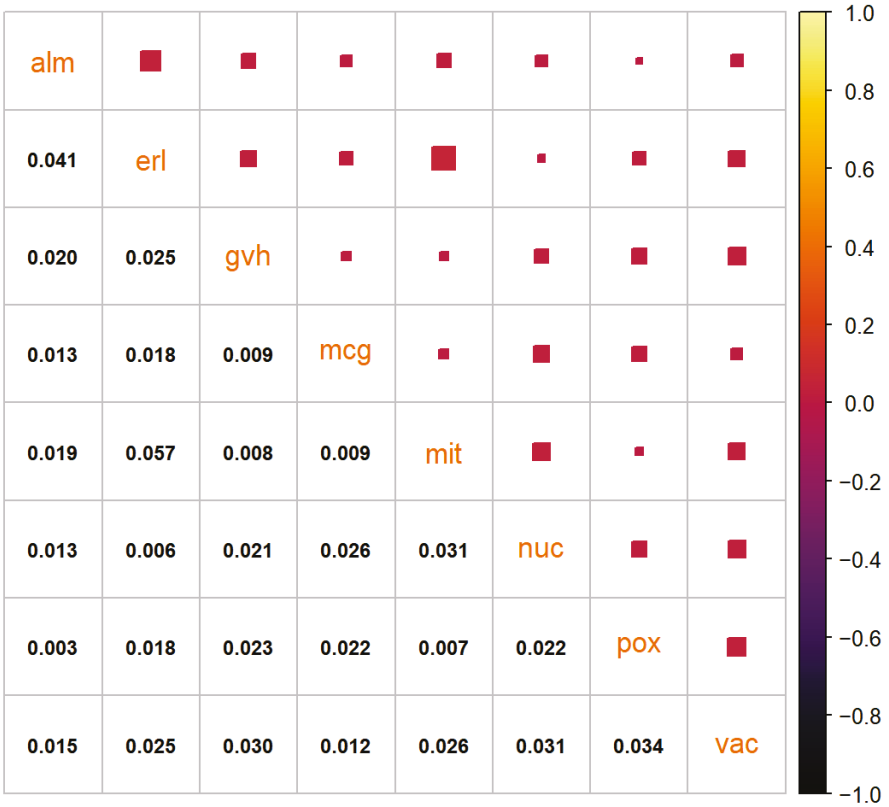


Figure A4. Residual correlation matrix of the synthetic "yeast" data set generated by the benchmark method.

Table A2. Table of *p*-values calculated by the Anderson–Darling and DTS statistical tests comparing the distributions of the attributes of the real-life "yeast" data set and the attributes of the synthetic data set generated by the benchmark method.

Attribute	Anderson–Darling	DTS
mcg	0.7715	0.7390
gvh	0.8530	0.5705
alm	0.9485	0.8020
mit	0.6165	0.7790
erl	0.0025	0.0615
pox	0.0020	0.0530
vac	0.6205	0.2050
nuc	0.2915	0.8525

The synthetic data set generated by the cluster preserving method can be seen in Figure A5. Even though it is not immediately evident, the spatial distribution of the synthetic data is closer to that of the real-life data. The synthetic data set has 1483 data points. The quantiles of the synthetic data set correspond to the quantiles of the real-life data set, as can be seen in Figure A6. As with the "texture" data set, using the cluster preserving method to generate the synthetic "yeast" data set results in smaller correlation residuals, as evidenced by Figure A7. The sum of residuals above the main diagonal is 0.356 and the average residual value is 0.013—both a significant reduction from the residual values of the synthetic data set generated by the benchmark method. However, the apparent improvement in correlation matrix residuals is absent from the *p*-values in Table A3.

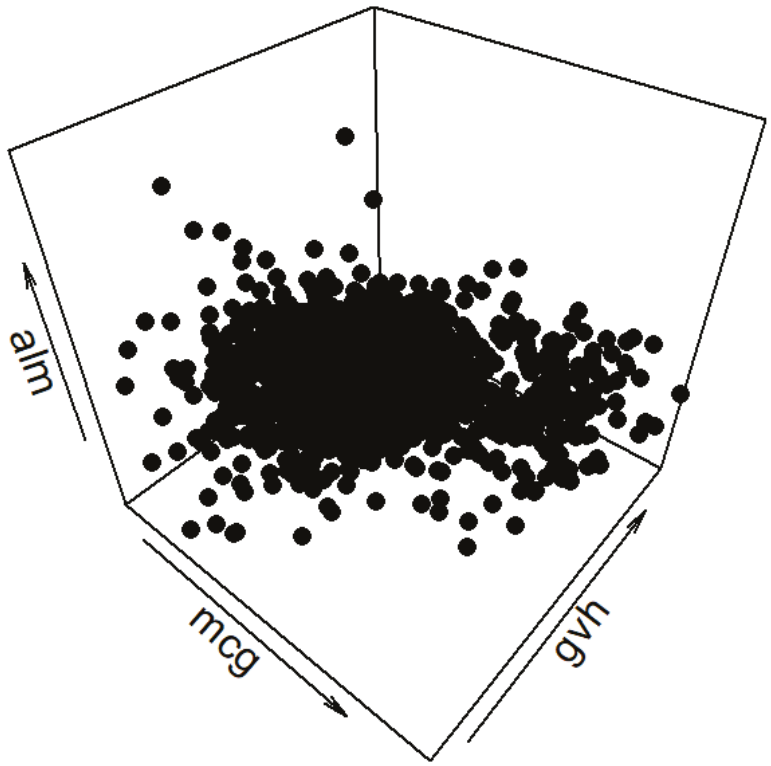


Figure A5. Three-dimensional view of the synthetic "yeast" data set generated by the cluster preserving method.

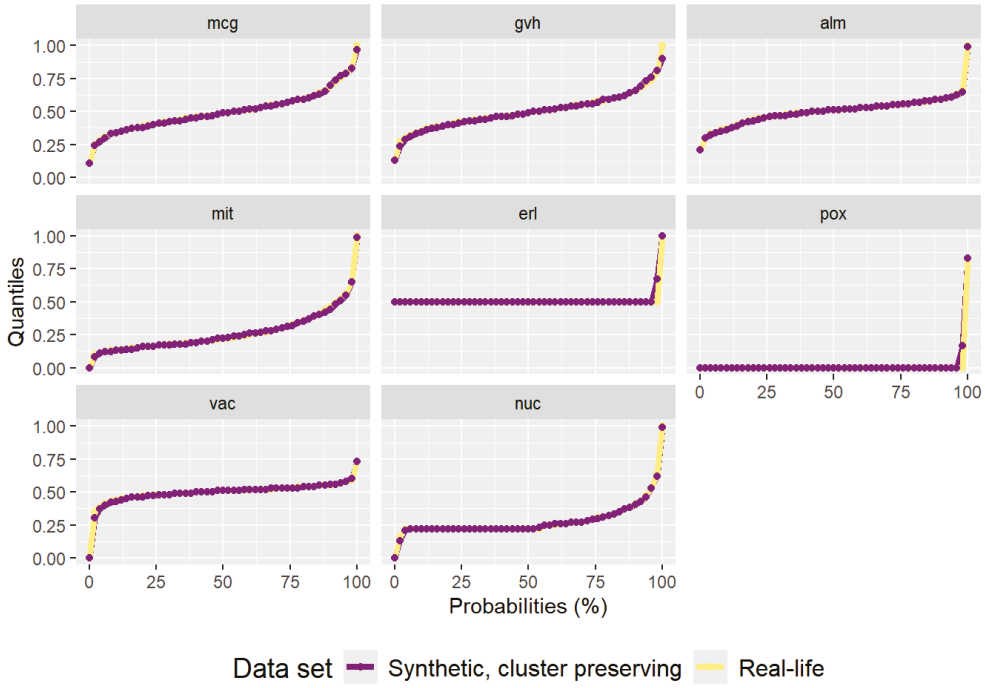


Figure A6. Plot of quantiles calculated for the real-life "yeast" data set and the synthetic data set generated by the cluster preserving method.

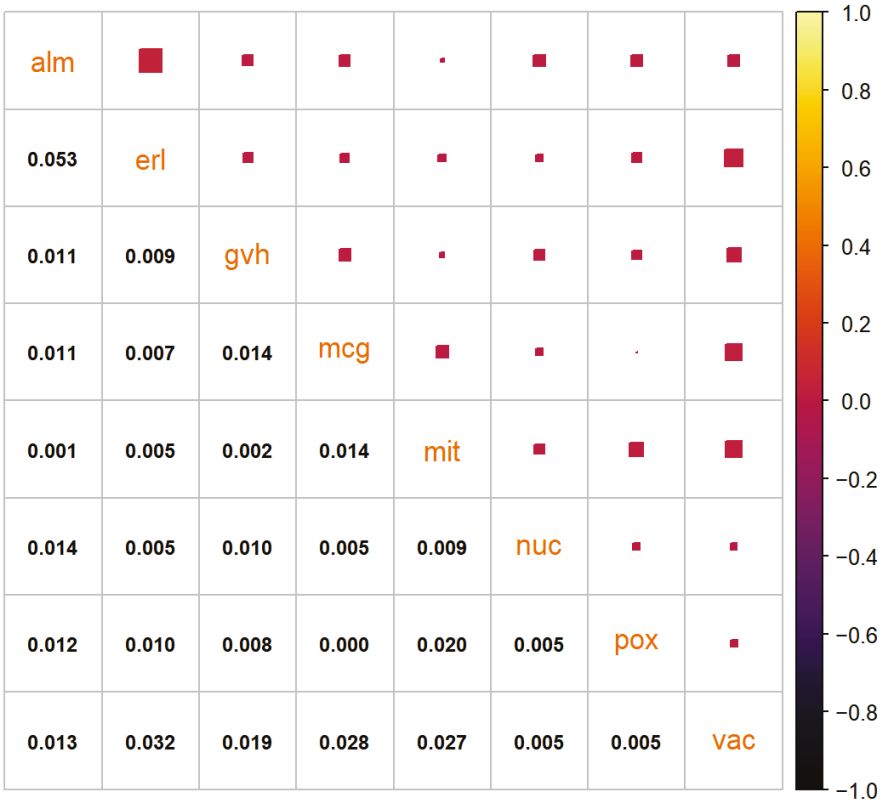


Figure A7. Residual correlation matrix of the synthetic "yeast" data set generated by the cluster preserving method.

Table A3. Table of *p*-values calculated by the Anderson–Darling and DTS statistical tests comparing the distributions of the attributes of the real-life "yeast" data set and the attributes of the synthetic data set generated by the cluster preserving method.

Attribute	Anderson–Darling	DTS
mcg	0.7805	0.7450
gvh	0.8515	0.5760
alm	0.9555	0.7910
mit	0.6370	0.7785
erl	0.0005	0.0620
pox	0.0045	0.0465
vac	0.6410	0.1965
nuc	0.3130	0.8315

The densities of each attribute in the three data sets can be seen in Figure A8. There are clear similarities between the "yeast" data set and the "texture" data set when comparing the effects of data generation. All the densities of the three data sets are nearly impossible to separate from one another. Furthermore, the results of the statistical tests in Tables A4 and A5 show the heterogeneous nature of the data set.

Table A4. Table of *p*-values calculated by the Anderson–Darling statistical test comparing the distributions of each of the attributes ("Attr." in table) of the real-life data set and the synthetic data set (cluster preserving method) depending on the class ("Cls." in table).

Anderson–Darling										
Cls. / Attr.	CYT	ERL	EXC	ME1	ME2	ME3	MIT	NUC	POX	VAC
mcg	0.972	0.424	0.622	0.975	0.764	0.538	0.707	0.922	0.769	0.316
gvh	0.958	0.318	0.886	0.626	0.500	0.517	0.743	0.986	0.692	0.364
alm	0.947	0.311	0.990	0.747	0.560	0.497	0.846	0.942	0.897	0.541
mit	0.987	0.290	0.844	0.845	0.521	0.562	0.828	0.970	0.827	0.146
erl	0.264	1.000	1.000	1.000	0.020	0.067	1.000	0.240	1.000	1.000
pox	0.039	1.000	1.000	1.000	1.000	1.000	0.304	1.000	0.006	1.000
vac	0.936	1.000	0.838	0.464	0.810	0.316	0.714	0.928	0.350	0.122
nuc	0.658	0.408	0.330	0.806	0.238	0.616	0.807	0.983	0.006	0.010

Table A5. Table of *p*-values calculated by the DTS statistical test comparing the distributions of each of the attributes ("Attr." in table) of the real-life data set and the synthetic data set (cluster preserving method) depending on the class ("Cls." in table).

DTS										
Cls. / Attr.	CYT	ERL	EXC	ME1	ME2	ME3	MIT	NUC	POX	VAC
mcg	0.970	0.386	0.929	0.909	0.903	0.576	0.848	0.760	0.706	0.383
gvh	0.885	0.278	0.954	0.651	0.493	0.427	0.428	0.983	0.912	0.337
alm	0.722	0.277	0.989	0.579	0.768	0.785	0.800	0.901	0.853	0.299
mit	0.914	0.388	0.961	0.426	0.606	0.786	0.836	0.773	0.705	0.178
erl	0.355	1.000	1.000	1.000	0.083	0.141	1.000	0.337	1.000	1.000
pox	0.136	1.000	1.000	1.000	1.000	1.000	0.336	1.000	0.034	1.000
vac	0.728	1.000	0.888	0.418	0.728	0.566	0.262	0.528	0.402	0.272
nuc	0.938	0.600	0.749	0.419	0.364	0.587	0.513	0.875	0.267	0.187

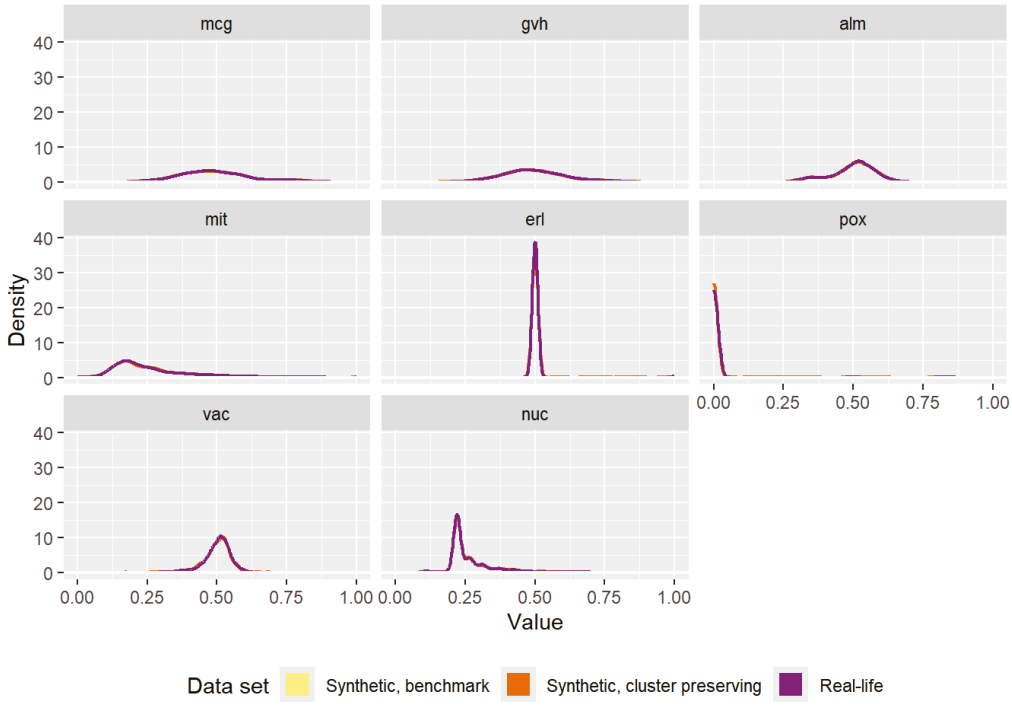


Figure A8. Comparison of the densities of the attributes of the real-life data set and the attributes of the synthetic data sets (one generated by the benchmark method, the other generated by the cluster preserving method).

Appendix A.1. Downstream Task—Classification

To test the hypothesis a model could be trained on the synthetic data generated by the cluster preserving method and then evaluated on the real-life data set, a simple classification was performed using Support Vector Machines (SVM) from the R package "e1071" ("Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien.", https://cran.r-project.org/web/packages/e1071/e1071.pdf, accessed on 25 April 2025). Both the real-life data set and the synthetic data set generated by the cluster preserving method were split into training and testing subsets, with 80% of the data set being used for training and the remainder for testing. All the attributes were used to predict the class and no further optimizations were implemented.

Evaluating the model trained on the real-life training set on the real-life test set yielded an accuracy of 0.623. Evaluating the model trained on the synthetic training set on the synthetic test set yielded an accuracy of 0.579. Finally, evaluating the model trained on the synthetic training set on the real-life test set yielded an accuracy of 0.66. These results indicate the hypothesis that the synthetic data generated by the cluster preserving method can be used to train a model that can then function well with real-life data holds.

Appendix B. Additional Graphs for the "Texture" Data Set

As was stated in Section 4, only a selection of the variables and classes of the full real-life data set was used to make the results visually interpretable. This section provides more detailed three-dimensional graphs to further illustrate the differences between the benchmark method (Section 2.2.2) and the cluster preserving method (Section 3).

Figure A9 shows four different views of the filtered real-life data set. Each of the four different classes is presented in a different color to make it more easily distinguishable from the others. Variables 'V10' and 'V23' were chosen as the x- and y-axis because they provide the most open and easily interpretable plane. The additional four variables are used as

the z-axis to provide more comprehensive views. Graphs (a,b) show that even the filtered classes have a certain amount of overlap, depending on how the data points are plotted.

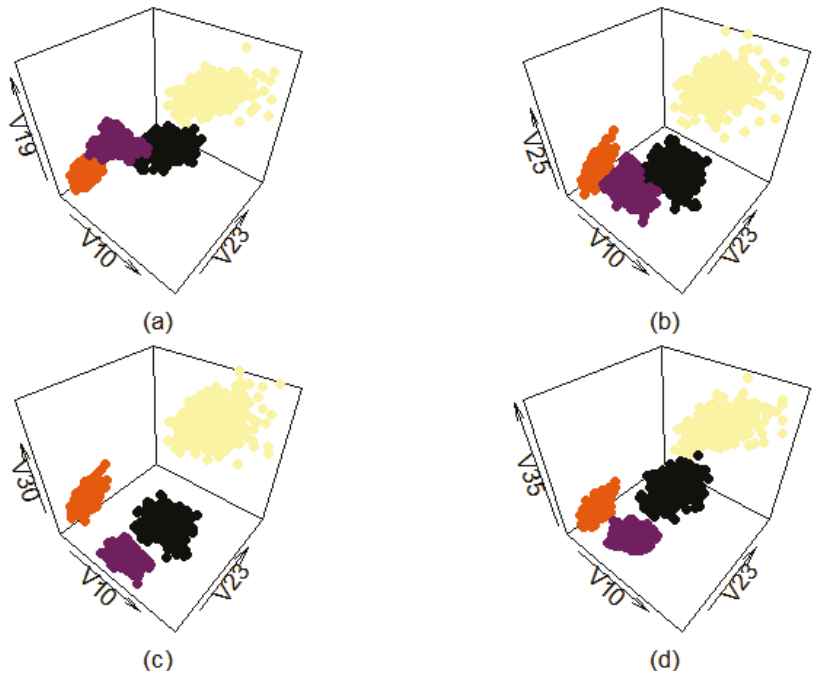


Figure A9. Three-dimensional view of classes selected from "texture" data set. Plotted with different z-axes.

Figure A10 shows four different views of the synthetic data set generated by the benchmark method. As can be seen, the synthetic data roughly follow the orientation of the real-life data, and, in some cases (graph (c)), even mostly replicate the layout of the real-life data set. However, this method produces multiple artifacts as well, which result in data points being present in places the filtered real-life data set does not have any.

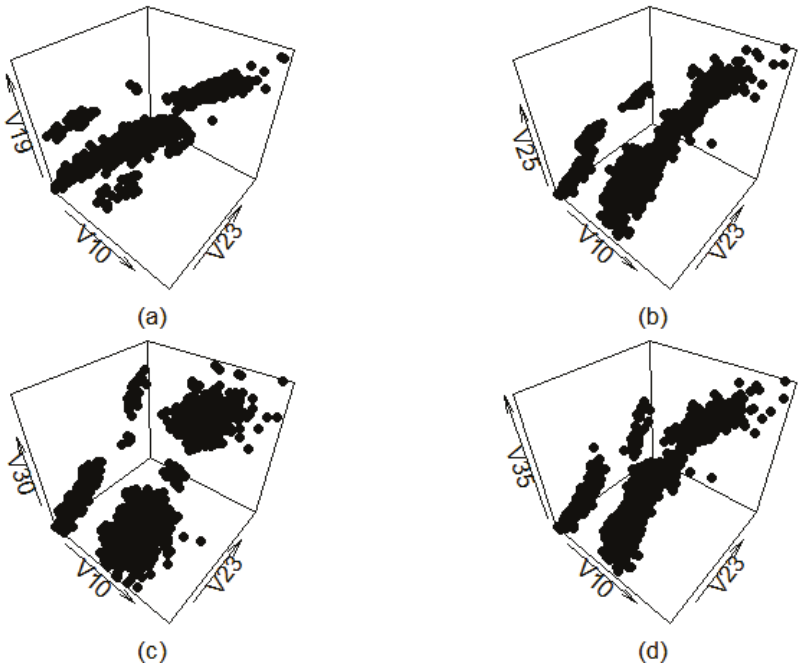


Figure A10. Three-dimensional view of classes generated by the benchmark method. Plotted with different z-axes.

Figure A11 shows four different views of the synthetic data set generated by the cluster preserving method. There is very little visual difference between this data set and the filtered real-life data set (apart from the classes being of different color in Figure A9). The cluster shapes do not correspond to the cluster shapes of the real-life data set completely, but that is expected and even desired. It is likely the generated data points are not present in the real-life data set, indicating that using data generated by the cluster preserving method to test the abilities of machine learning or clustering methods could be worthwhile.

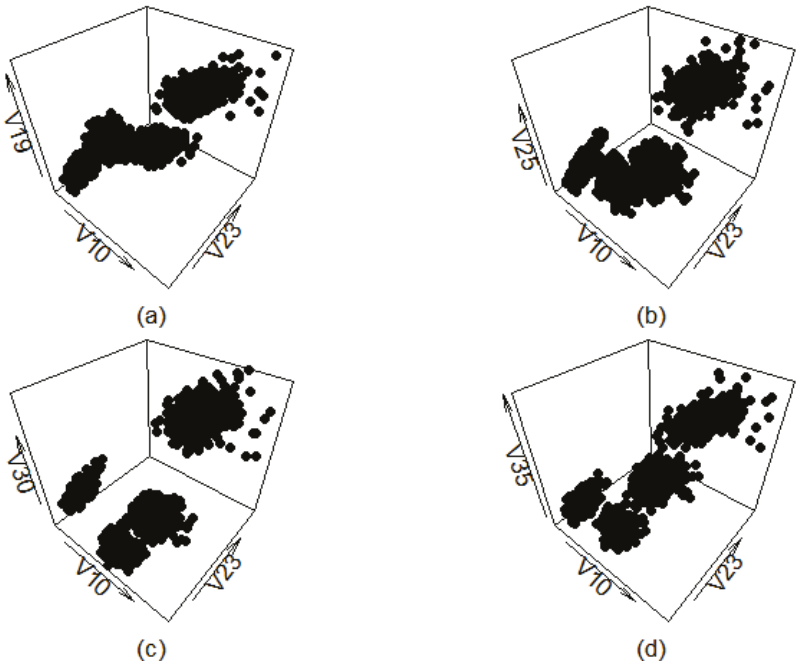


Figure A11. Three-dimensional view of classes generated by the cluster preserving method. Plotted with different z-axes.

References

1. Chen, R.J.; Lu, M.Y.; Chen, T.Y.; Williamson, D.F.K.; Mahmood, F. Synthetic data in machine learning for medicine and healthcare. *Nat. Biomed. Eng.* **2021**, *5*, 493–497. [CrossRef] [PubMed]
2. Liu, Q.; Khalil, M.; Jovanovic, J.; Shakya, R. Scaling While Privacy Preserving: A Comprehensive Synthetic Tabular Data Generation and Evaluation in Learning Analytics. In Proceedings of the 14th Learning Analytics and Knowledge Conference, Kyoto, Japan, 18–24 March 2024; LAK '24, pp. 620–631. [CrossRef]
3. Morley-Fletcher, E. New Solutions to Biomedical Data Sharing: Secure Computation and Synthetic Data. In *Personalized Medicine in the Making: Philosophical Perspectives from Biology to Healthcare*; Beneduce, C., Bertolaso, M., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 173–189. [CrossRef]
4. Giuffrè, M.; Shung, D.L. Harnessing the power of synthetic data in healthcare: Innovation, application, and privacy. *NPJ Digit. Med.* **2023**, *6*, 186. [CrossRef] [PubMed]
5. Potluru, V.K.; Borrajo, D.; Coletta, A.; Dalmasso, N.; El-Laham, Y.; Fons, E.; Ghassemi, M.; Gopalakrishnan, S.; Gosai, V.; Kreačić, E.; et al. Synthetic Data Applications in Finance. *arXiv* **2024**, arXiv:2401.00081. [CrossRef]
6. Ruscio, J.; Kaczetow, W. Simulating multivariate nonnormal data using an iterative algorithm. *Multivar. Behav. Res.* **2008**, *43*, 355–381. [CrossRef]
7. Petricioli, L.; Humski, L.; Vranić, M.; Pintar, D. Data Set Synthesis Based on Known Correlations and Distributions for Expanded Social Graph Generation. *IEEE Access* **2020**, *8*, 33013–33022. [CrossRef]
8. Iftikhar, N.; Liu, X.; Nordbjerg, F.E.; Danalachi, S. A Prediction-Based Smart Meter Data Generator. In Proceedings of the 19th International Conference on Network-Based Information Systems, NBiS 2016, Ostrava, Czech Republic, 7–9 September 2016; pp. 173–180. [CrossRef]

9. Syahaneim; Hazwani, R.A.; Wahida, N.; Shafikah, S.I.; Zuraini; Ellyza, P.N. Automatic Artificial Data Generator: Framework and implementation. In Proceedings of the 1st International Conference on Information and Communication Technology, ICICTM 2016, Kuala Lumpur, Malaysia, 16–17 May 2016; IEEE: Piscataway, NJ, USA, 2017; pp. 56–60. [CrossRef]
10. Liu, R.; Fang, B.; Tang, Y.Y.; Chan, P.P. Synthetic data generator for classification rules learning. In Proceedings of the 7th International Conference on Cloud Computing and Big Data, CCBD 2016, Macau, China, 16–18 November 2016; IEEE: Piscataway, NJ, USA, 2017; pp. 357–361. [CrossRef]
11. Nowok, B.; Raab, G.M.; Dibben, C. Synthpop: Bespoke creation of synthetic data in R. *J. Stat. Softw.* **2016**, *74*, 1–26. [CrossRef]
12. Robnik-Sikonja, M. Data Generators for Learning Systems Based on RBF Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 926–938. [CrossRef]
13. Sagduyu, Y.E.; Grushin, A.; Shi, Y. Synthetic social media data generation. *IEEE Trans. Comput. Soc. Syst.* **2018**, *5*, 605–620. [CrossRef]
14. Yang, S.; Zhou, Y.; Guo, Y.; Farneth, R.A.; Marsic, I.; Randall, B.S. Semi-Synthetic Trauma Resuscitation Process Data Generator. In Proceedings of the IEEE International Conference on Healthcare Informatics, ICHI 2017, Park City, UT, USA, 23–26 August 2017; IEEE: Piscataway, NJ, USA, 2017, p. 573. [CrossRef]
15. Xu, L.; Veeramachaneni, K. Synthesizing Tabular Data using Generative Adversarial Networks. *arXiv* **2018**, arXiv:1811.11264.
16. Xu, L.; Skoularidou, M.; Cuesta-Infante, A.; Veeramachaneni, K. Modeling tabular data using conditional GAN. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Curran Associates Inc.: Red Hook, NY, USA, 2019; Number 659, pp. 7335–7345.
17. Zhao, Z.; Kunar, A.; Birke, R.; Chen, L.Y. CTAB-GAN: Effective Table Data Synthesizing. In Proceedings of the 13th Asian Conference on Machine Learning, Virtual, 17–19 November 2021; PMLR: New York, NY, USA, 2021; pp. 97–112.
18. Rajabi, A.; Garibay, O.O. TabFairGAN: Fair Tabular Data Generation with Generative Adversarial Networks. *Mach. Learn. Knowl. Extr.* **2022**, *4*, 488–501. [CrossRef]
19. Rai, R.; Sural, S. Tool/Dataset Paper: Realistic ABAC Data Generation using Conditional Tabular GAN. In Proceedings of the 13th ACM Conference on Data and Application Security and Privacy, Charlotte, NC, USA, 24–26 April 2023; pp. 273–278. [CrossRef]
20. He, G.; Zhao, Y.; Yan, C. Application of tabular data synthesis using generative adversarial networks on machine learning-based multiaxial fatigue life prediction. *Int. J. Press. Vessel. Pip.* **2022**, *199*, 104779. [CrossRef]
21. Gao, N.; Xue, H.; Shao, W.; Zhao, S.; Qin, K.K.; Prabowo, A.; Rahaman, M.S.; Salim, F.D. Generative Adversarial Networks for Spatio-temporal Data: A Survey. *ACM Trans. Intell. Syst. Technol.* **2022**, *13*, 22:1–22:25. [CrossRef]
22. Liu, T.; Fan, J.; Li, G.; Tang, N.; Du, X. Tabular data synthesis with generative adversarial networks: Design space and optimizations. *VLDB J.* **2023**, *33*, 255–280. [CrossRef]
23. Zhou, Y.; Qi, J. Synthesizing Tabular Data Using Selectivity Enhanced Generative Adversarial Networks. *arXiv* **2025**, arXiv:2502.21034. [CrossRef]
24. Suh, N.; Lin, X.; Hsieh, D.Y.; Honarkhah, M.; Cheng, G. AutoDiff: Combining Auto-encoder and Diffusion model for tabular data synthesizing. *arXiv* **2023**, arXiv:2310.15479. [CrossRef]
25. Anshelevich, D.; Katz, G. Synthetic Tabular Data Generation Using a Vae-Gan Architecture, 2024. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4902016 (accessed on 13 March 2025).
26. Tazwar, S.; Knobbout, M.; Quesada, E.; Popa, M. Tab-VAE: A Novel VAE for Generating Synthetic Tabular Data:. In Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods, Rome, Italy, 24–26 February 2024; pp. 17–26. [CrossRef]
27. Wang, A.X.; Nguyen, B.P. Deterministic Autoencoder using Wasserstein loss for tabular data generation. *Neural Netw.* **2025**, *185*, 107208. [CrossRef] [PubMed]
28. Borisov, V.; Sessler, K.; Leemann, T.; Pawelczyk, M.; Kasneci, G. Language Models are Realistic Tabular Data Generators. In Proceedings of the 11th International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
29. Wang, Y.; Feng, D.; Dai, Y.; Chen, Z.; Huang, J.; Ananiadou, S.; Xie, Q.; Wang, H. HARMONIC: Harnessing LLMs for Tabular Data Synthesis and Privacy Protection. In Proceedings of the 38th Conference on Neural Information Processing Systems, Datasets and Benchmarks Track, Vancouver, BC, Canada, 10–15 December 2024.
30. Cromp, S.; GNVV, S.S.S.N.; Alkhudhayri, M.; Cao, C.; Guo, S.; Roberts, N.; Sala, F. Tabby: Tabular Data Synthesis with Language Models. *arXiv* **2025**. arXiv:2503.02152. [CrossRef]
31. Zhao, Z.; Birke, R.; Chen, L. TabuLa: Harnessing Language Models for Tabular Data Synthesis. *arXiv* **2025**, arXiv:2310.12746. [CrossRef]
32. Ayala-Rivera, V.; McDonagh, P.; Cerqueus, T.; Murphy, L. Synthetic Data Generation using Benerator Tool. *arXiv* **2013**. [CrossRef]
33. Eno, J.; Thompson, C.W. Generating synthetic data to match data mining patterns. *IEEE Internet Comput.* **2008**, *12*, 78–82. [CrossRef]

34. Erling, O.; Averbuch, A.; Larriba-Pey, J.; Chafi, H.; Gubichev, A.; Prat, A.; Pham, M.D.; Boncz, P. The LDBC social network benchmark: Interactive workload. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Melbourne, VIC, Australia, 31 May–4 June 2015; pp. 619–630. [CrossRef]
35. Hoag, J.E.; Thompson, C.W. A Parallel General-Purpose Synthetic Data Generator. *SIGMOD Rec.* **2007**, *36*, 19–24. [CrossRef]
36. Jeske, D.R.; Lin, P.J.; Rendón, C.; Xiao, R.; Samadi, B. Synthetic data generation capabilities for testing data mining tools. In Proceedings of the IEEE Military Communications Conference MILCOM, Washington, DC, USA, 23–25 October 2006; IEEE: Piscataway, NJ, USA, 2007; pp. 1–6. [CrossRef]
37. Khan, J.; Lee, S. Online Social Networks (OSN) evolution model based on homophily and preferential attachment. *Symmetry* **2018**, *10*, 654. [CrossRef]
38. Maciejewski, R.; Hafen, R.; Rudolph, S.; Tebbetts, G.; Cleveland, W.S.; Grannis, S.J.; Ebert, D.S. Generating synthetic syndromic-surveillance data for evaluating visual-analytics techniques. *IEEE Comput. Graph. Appl.* **2009**, *29*, 18–28. [CrossRef] [PubMed]
39. Nettleton, D.F. A synthetic data generator for online social network graphs. *Soc. Netw. Anal. Min.* **2016**, *6*, 44. [CrossRef]
40. Petermann, A.; Junghanns, M.; Müller, R.; Rahm, E. FoodBroker-Generating synthetic datasets for Graph-Based business analytics. In *Big Data Benchmarking* ; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Cham, Switzerland, 2015; Volume 8991, pp. 145–155. [CrossRef]
41. Pham, M.D.; Boncz, P.; Erling, O. S3G2: A scalable structure-correlated social graph generator. In *Selected Topics in Performance Evaluation and Benchmarking*; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin/Heidelberg, Germany, 2013; Volume 7755, pp. 156–172. [CrossRef]
42. Spasić, M.; Jovanovik, M.; Prat-Pérez, A. An RDF dataset generator for the social network benchmark with real-world coherence. *CEUR Workshop Proc.* **2016**, *1700*, 1–8.
43. Lee, J.; Hong, J.; Hong, B.; Ahn, J. A generator of test data set for tactical moving objects based on velocity. In Proceedings of the IEEE International Conference on Big Data, Big Data 2016, Washington, DC, USA, 5–8 December 2016; pp. 4011–4013. [CrossRef]
44. Pérez-Rosés, H.; Sebé, F. Synthetic generation of social network data with endorsements. *J. Simul.* **2015**, *9*, 279–286. [CrossRef]
45. Vale, C.D.; Maurelli, V.A. Simulating multivariate nonnormal distributions. *Psychometrika* **1983**, *48*, 465–471. [CrossRef]
46. Eigenschink, P.; Reutterer, T.; Vamosi, S.; Vamosi, R.; Sun, C.; Kalcher, K. Deep Generative Models for Synthetic Data: A Survey. *IEEE Access* **2023**, *11*, 47304–47320. [CrossRef]
47. Rohlfs, C. Generalization in neural networks: A broad survey. *Neurocomputing* **2025**, *611*, 128701. [CrossRef]
48. Wang, A.X.; Chukova, S.S.; Simpson, C.R.; Nguyen, B.P. Challenges and opportunities of generative models on tabular data. *Appl. Soft Comput.* **2024**, *166*, 112223. [CrossRef]
49. Goyal, M.; Mahmoud, Q.H. A Systematic Review of Synthetic Data Generation Techniques Using Generative AI. *Electronics* **2024**, *13*, 3509.
50. Perković, G.; Drobnjak, A.; Botički, I. Hallucinations in LLMs: Understanding and Addressing Challenges. In Proceedings of the 47th MIPRO ICT and Electronics Convention (MIPRO), Opatija, Croatia, 20–24 May 2024; pp. 2084–2088.
51. Xu, Z.; Jain, S.; Kankanhalli, M. Hallucination is Inevitable: An Innate Limitation of Large Language Models. *arXiv* **2025**, arXiv:2401.11817. [CrossRef]
52. Miyato, T.; Koyama, M. cGANs with Projection Discriminator. *arXiv* **2018**, arXiv:1802.05637. [CrossRef]
53. Raghunathan, T.E. Synthetic Data. *Annu. Rev. Stat. Its Appl.* **2021**, *8*, 129–140. [CrossRef]
54. Wang, Z.; Myles, P.; Tucker, A. Generating and evaluating cross-sectional synthetic electronic healthcare data: Preserving data utility and patient privacy. *Comput. Intell.* **2021**, *37*, 819–851. [CrossRef]
55. Hosain, M.T.; Jim, J.R.; Mridha, M.F.; Kabir, M.M. Explainable AI approaches in deep learning: Advancements, applications and challenges. *Comput. Electr. Eng.* **2024**, *117*, 109246. [CrossRef]
56. Zhao, H.; Chen, H.; Yang, F.; Liu, N.; Deng, H.; Cai, H.; Wang, S.; Yin, D.; Du, M. Explainability for Large Language Models: A Survey. *ACM Trans. Intell. Syst. Technol.* **2024**, *15*, 20:1–20:38. [CrossRef]
57. Ruscio, J.; Ruscio, A.M.; Meron, M. Applying the bootstrap to taxometric analysis: Generating empirical sampling distributions to help interpret results. *Multivar. Behav. Res.* **2007**, *42*, 349–386. [CrossRef] [PubMed]
58. Humski, L.; Pintar, D.; Vranić, M. Analysis of Facebook Interaction as Basis for Synthetic Expanded Social Graph Generation. *IEEE Access* **2019**, *7*, 6622–6636. [CrossRef]
59. Dowd, C. A new ECDF Two-Sample test statistic. *arXiv* **2020**, arXiv:2007.01360 . [CrossRef]
60. Anderson, T.W.; Darling, D.A. Asymptotic Theory of Certain "Goodness of Fit" Criteria Based on Stochastic Processes. *Ann. Math. Stat.* **1952**, *23*, 193–212. [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

Data Driven Analytics for Distribution Network Power Supply Reliability Assessment Method Considering Frequency Regulating Scenario

Yu Zhang [1,†], Jinyue Shi [1], Shicheng Huang [1], Liang Geng [1], Zexiong Wang [1], Hao Sun [1], Qingguang Yu [2,*], Xin Yao [2,†], Ding Liu [2], Weihua Zuo [2], Min Guo [2] and Xiaoyu Che [2]

[1] Shijiazhuang Power Supply Company, State Grid Hebei Electric Power Co., Ltd., Shijiazhuang 050000, China; sjz_zhangy16@he.sgcc.com.cn (Y.Z.); sjz_shijy2@he.sgcc.com.cn (J.S.); sjz_huangsc@he.sgcc.com.cn (S.H.); sjz_gengl@he.sgcc.com.cn (L.G.); sjz_wangzx6@he.sgcc.com.cn (Z.W.); sjz_sunh2@he.sgcc.com.cn (H.S.)

[2] Department of Electrical Engineering, Tsinghua University, Beijing 100084, China; yaox23@mails.tsinghua.edu.cn (X.Y.); d-liu24@mails.tsinghua.edu.cn (D.L.); zuowh25@mails.tsinghua.edu.cn (W.Z.); guomin1994@mail.tsinghua.edu.cn (M.G.); chexy22@mails.tsinghua.edu.cn (X.C.)

* Correspondence: qinggy@tsinghua.edu.cn

† These authors contributed equally to this work.

Abstract

Islanded microgrids face significant frequency stability challenges due to limited system capacity, low inertia levels, and the strong variability in renewable energy sources. Traditional reliability assessment methods, often based on static power balance, struggle to comprehensively reflect frequency dynamic characteristics and their impact on power supply reliability. To address this issue, this paper proposes a sequential Monte Carlo reliability assessment method integrated with a system frequency response model. First, an SFR model for the isolated microgrid, incorporating diesel generators, gas turbines, energy storage, and wind turbines, is established. For synchronous units, a frequency deviation-based failure rate correction mechanism is introduced to characterize the impact of frequency fluctuations on equipment reliability. State transitions are achieved by integrating failure and repair rates to reach threshold values. Second, sequential Monte Carlo simulation is employed to conduct time-series simulations of annual operation. Random sampling of unit failure and repair times is used to calculate reliability metrics. MATLAB/Simulink simulation results demonstrate that system frequency fluctuations caused by power imbalance worsen unit failure rates, leading to microgrid reliability values lower than static calculations. This provides reference for planning, design, and operational scheduling of isolated microgrids.

Keywords: islanded microgrids; sequential monte carlo; system frequency regulate model; reliability assessment

1. Introduction

With the large-scale integration of renewable energy, the operating conditions and control characteristics of power systems are undergoing profound changes. Traditional power systems rely on large synchronous generators to provide rotational inertia and primary frequency control, thereby maintaining frequency stability in the presence of disturbances [1]. However, power-electronics–interfaced resources such as wind power, photovoltaic generation, and energy storage are gradually replacing a portion of synchronous units, markedly reducing system inertia and introducing new challenges to frequency stability [2,3]. In

islanded microgrids in particular, limited system capacity, insufficient reserves, and the strong variability in renewable outputs [4] make frequency security issues especially acute, highlighting an urgent need for comprehensive assessments that jointly consider reliability and frequency characteristics [5].

Current microgrid reliability studies predominantly employ simulation-based and analytical approaches. Analytical methods, which typically rely on computing failure (state) transition rates, cannot provide reliability indices that explicitly depend on system frequency and time. In fact, the post-disturbance frequency recovery of an islanded microgrid is highly contingent on the frequency-regulation performance and control characteristics of its constituent units; if the frequency deviation exceeds preset thresholds, it may trigger under-frequency load shedding or even system tripping. Traditional static reliability indices thus struggle to fully represent the security level of a microgrid during dynamic operation. In addition, the stochastic nature introduced by renewable energy resources must be addressed—accounting for forced outages and random power outputs—while the state of charge of energy storage systems exhibits pronounced temporal dependence. Consequently, the sequential Monte Carlo method is better suited to contemporary microgrid reliability assessment, albeit at the cost of substantial computational time.

In recent years, extensive research has been conducted on microgrid reliability, frequency security, and renewable energy variability. Reference [6] employs a Markov process to model the output uncertainty of wind power and energy storage, analyzing its effect on supply reliability; however, it does not account for post-disturbance frequency excursions. Regarding frequency stability, ref. [7] introduces a reliability assessment framework based on a system frequency response model, which correlates unit failure and repair rates with frequency deviations to enable integrated frequency-reliability evaluation. Nonetheless, the case studies in [7] adopt piecewise correction methods that are unsuitable for complex time-sequential simulations. References [8,9] propose frequency-security metrics such as the number of under-frequency events, their duration, and the probability of low-frequency recovery. These approaches, however, rely on analytical approximations or static simulations and fail to adequately represent the dynamic operational constraints of microgrids. Reference [10] applies sequential Monte Carlo simulation to model the temporal processes of unit failure and repair, incorporating real-world wind speed and load time series. Despite this, it only considers power fluctuations from renewable resources and overlooks the feedback effect of frequency on unit reliability. Reference [11] proposes a stochastic method for risk aversion based on weather routing, which provides a new idea for multi-energy co-optimization. However, the method mainly focuses on the operational optimization level, lacks an in-depth analysis of the system dynamic characteristics, especially the frequency stability, and does not consider the feedback effect of frequency fluctuation on equipment reliability. Reference [12] proposed an N-1 assessment framework for the integrated electric-gas system considering the information physical dependence, which reveals the coupling relationship between the energy system and the information system. However, the study focused on static safety assessment and failed to react to the dynamic response process of the system under perturbation.

The main contribution of this paper is the development of an equipment reliability model considering frequency variations and the proposal of a sequential Monte Carlo simulation method that introduces a frequency response model of the system, which overcomes the limitations of traditional static assessment methods. On the basis of this, this paper presents a comprehensive reliability assessment approach for islanded microgrids that incorporates frequency-dependent effects. An integrated SFR model is developed, encompassing diesel generators, gas turbines, energy storage systems, and wind turbines. A frequency-adaptive adjustment mechanism for unit failure and repair rates is introduced,

along with an under-frequency load-shedding strategy. On this basis, the sequential Monte Carlo method is employed to simulate long-term operational scenarios, simultaneously capturing stochastic unit outages, repairs, and fluctuations in renewable generation and load. The proposed framework evaluates both supply reliability and frequency security under various operating conditions, generating a set of corresponding reliability and frequency-security indices. Results demonstrate that the operational reliability of islanded microgrids is closely coupled with frequency regulation measures. The impact of system frequency on unit failure rates proves non-negligible in reliability assessments, offering theoretical insights and methodological support for subsequent system planning and operational optimization.

2. SFR Model of Islanded Microgrid

The frequency behavior of a microgrid depends on the types of sources and loads it comprises. Synchronous generator units exhibit pronounced frequency-droop characteristics and load regulation effects, whereas the frequency behavior of converter-interfaced sources is determined by their control models. Therefore, clarifying the characteristics of each type of source within the microgrid is a prerequisite for reliability studies. In this section, we first establish an islanded microgrid system frequency response model that includes both synchronous generator units and converter-interfaced sources. The synchronous units consist of diesel generators and gas turbines, while the converter-interfaced sources include an energy storage system and wind turbine generators. A typical microgrid system architecture is shown in Figure 1.

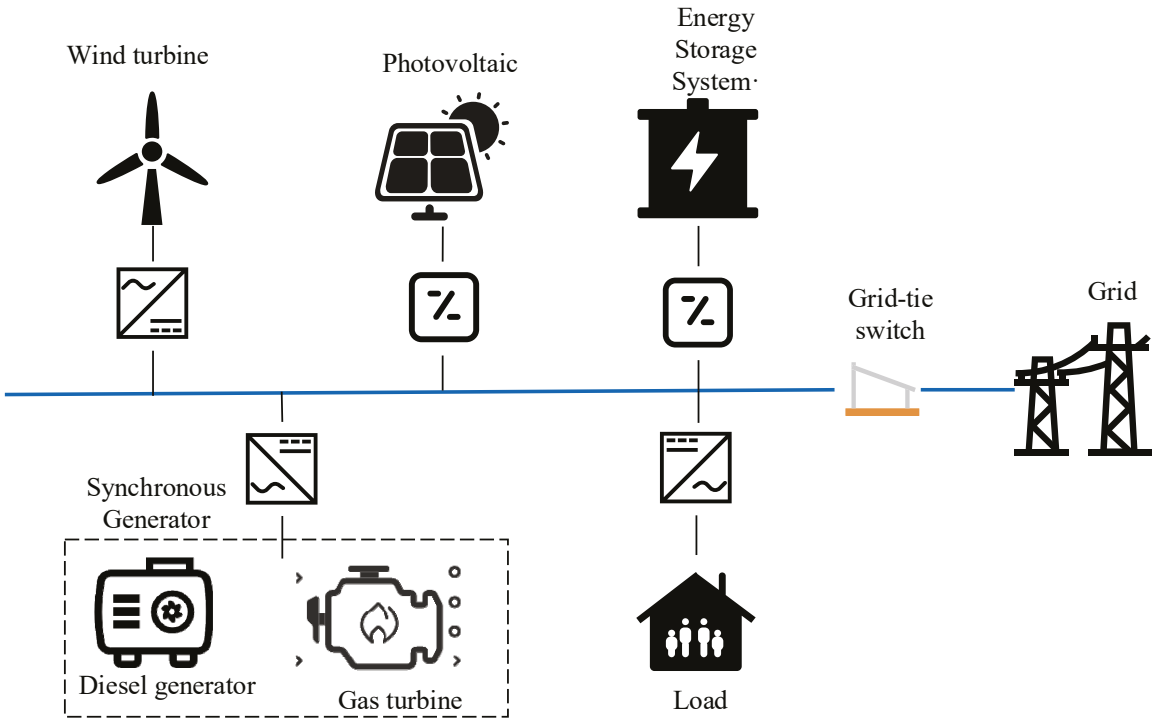


Figure 1. Microgrid system architecture.

2.1. Diesel Generator/Gas Turbine

Diesel generators and gas turbines are typical rotating synchronous generation sources in microgrids. Their operation relies on fossil fuels, and the fuel supply is generally assumed to be sufficiently ample. The rotor speed of these units operates at synchronous speed; when the external active power demand increases, the unit's active power output rises. In the unit's control model, this is reflected as an increase in electromagnetic torque, which

causes a drop in rotor speed. The governor system then increases the mechanical torque in response to the sensed speed decrease to raise the unit speed; once the mechanical and electromagnetic torques are rebalanced, the speed—and hence the frequency—stabilizes.

The frequency dynamics of diesel generators and gas turbines are jointly determined by rotor inertia and primary frequency control. The diesel generator is modeled using a first-order prime-mover model, as in Equation (1) [13].

$$\Delta P = (-\frac{S}{R}) \cdot \frac{1}{1+sT_G} \cdot \frac{1}{1+sT_E} \cdot \Delta f \tag{1}$$

where S denotes the unit (rated) capacity; R is the primary frequency-droop coefficient; T_G is the governor time constant, characterizing the dynamic response speed of the diesel governor; and T_E is the prime-mover time constant, representing the dynamic lag due to diesel-engine combustion and the inertia of mechanical output.

For typical small- to medium-sized diesel generators, the dynamic response is relatively fast but constrained by the mechanical subsystem and combustion dynamics. In accordance with IEEE interconnection performance requirements for distributed resources, their ramp rate does not exceed 20% p.u./min.

The gas turbine is modeled using the GGOV1 model, as in Equation (2) [14].

$$\Delta P = (-\frac{S}{R}) \cdot \frac{1}{1+sT_G} \cdot \frac{(1+sT_2)}{(1+sT_1)(1+sT_3)} \cdot \Delta f \tag{2}$$

where T_1 is the combustor time constant, describing the lag of the working gas along the combustion path; T_2 is the inertia-coupling time constant, reflecting the lead behavior induced by the air/fuel-gas dynamics within the turbine; and T_3 is the turbine mechanical time constant, representing the inertial lag associated with the gas turbine's mechanical energy conversion process.

For gas-turbine generator units, power ramp-up is constrained by combustion stability and thermal stress, and the ramp rate does not exceed 10% p.u./min.

2.2. Wind-Driven Generator

This study employs a doubly fed induction generator wind turbine for simulation. The turbine's output power is strongly correlated with wind speed. Depending on the ambient wind speed, the DFIG operates in four regions, as shown in Figure 2: when the wind speed is below the cut-in threshold, the turbine remains shut down; once the wind speed reaches the cut-in speed, the turbine starts and operates under maximum power point tracking, maintaining the power coefficient Cp at its maximum; in the constant-power region, the generator torque is limited to its rated value and the blade pitch angle is adjusted to reduce Cp so as to deliver the rated power; when the wind speed exceeds the cut-out threshold, the turbine is shut down to avoid faults.

The wind turbine generator mathematical model comprises the aerodynamic model, drivetrain model, generator power control model, and pitch-angle control model. The aerodynamic power captured by the rotor from the wind is calculated as follows:

$$P_m = \frac{1}{2}\rho\pi R^2 C_p(\lambda,\beta)v_w^3 \tag{3}$$

where ρ, R, C_P, λ, β, and v_w denote the air density, rotor radius, power coefficient, tip-speed ratio, pitch angle, and wind speed, respectively.

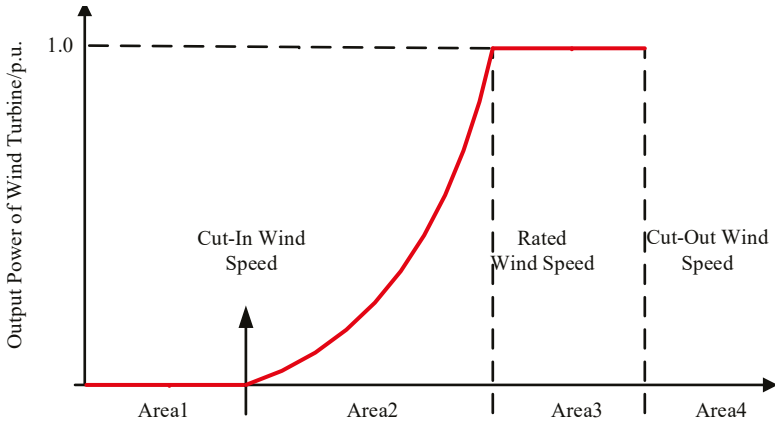


Figure 2. Wind turbine power output vs. wind speed chart.

The power coefficient C_p is a function of λ and β; a commonly used empirical fitting formula is given by [15]:

$$C_p = 0.5176(\frac{116}{\lambda_i} - 0.4\beta - 5)e^{-\frac{21}{\lambda_i}} + 0.0068\lambda \tag{4}$$

$$\frac{1}{\lambda_i} = \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1} \tag{5}$$

$$\lambda = \frac{\omega R}{v_w} \tag{6}$$

In the equation, λ_i is the intermediate variable, and ω is the blade rotational speed. Equations (3)–(6) constitutes the aerodynamic model for fan blades.

Current wind turbine drivetrain models include single-mass, two-mass, three-mass, and six-mass representations. In this study, a two-mass drivetrain model is adopted, and the governing equations are given by:

$$J_m \frac{d\omega}{dt} = T_m - T_s \tag{7}$$

$$T_s = K_s\psi + K_{vi}\frac{d\psi}{dt} \tag{8}$$

$$J_e \frac{d\omega_e}{dt} = \frac{T_s}{N} - T_e \tag{9}$$

$$\frac{d\psi}{dt} = \omega - \frac{\omega_e}{N} \tag{10}$$

$$J_{total} = J_m + N^2 J_e \tag{11}$$

In the equation, J_m represents the rotor inertia, J_e represents the generator inertia, J_{total} represents the equivalent inertia of the wind turbine, ω_e represents the generator rotational speed, T_m represents the mechanical torque, T_s represents the main shaft torque, T_e represents the electromagnetic torque, ψ represents the transmission chain torque, K_s represents the main shaft elasticity coefficient, K_{vi} represents the main shaft viscous friction coefficient, and N represents the gearbox transmission ratio. Equations (5)–(9) constitute the motion model of the wind turbine rotor.

When wind turbines participate in frequency regulation, control strategies include droop control, virtual inertia control, and combined inertia control. This study adopts the combined inertia control, and the frequency-regulation power of each turbine is given by:

$$\Delta P = -(K_p \Delta f + K_d \cdot \frac{d\Delta f}{dt})S \tag{12}$$

In the equation, Δf represents the system frequency deviation, K_p and K_d denote the primary frequency regulation coefficients of the wind turbine generator.

The output of wind turbines is controlled by converters, enabling rapid response. However, due to potential mechanical shocks causing fatigue loads and mechanical impacts, their ramping speed typically does not exceed 20% p.u./min.

2.3. Energy Storage System

Given that the storage capacity in microgrids is constrained by power electronic hardware, the energy storage system in this study is assumed to provide inertial response only. Virtual inertia control enables the ESS to emulate the inertial behavior of synchronous generators, thereby increasing the system inertia time constant, reducing the rate of change in frequency, and slowing the deterioration of frequency following disturbances. A derivative-based control law is employed to operate the ESS and deliver virtual inertia support, as shown in Figure 3.

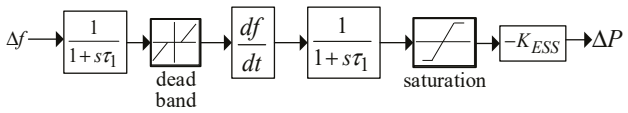


Figure 3. Energy storage system frequency response control block diagram.

A deadband prevents the energy storage system from responding to small frequency deviations. A low-pass filter is also applied to remove noise from the frequency-derivative signal, as derivative control is highly sensitive to noise. The per-unit output PIR is limited to ± 1 p.u. The power injected into the grid by the ESS under virtual inertia control can be calculated as:

$$\Delta P = -\frac{-K_{ESS}}{(1+s\tau_1)(1+s\tau_2)}\frac{df}{dt}S \tag{13}$$

where K_{ESS} is the virtual inertia response coefficient of the energy storage system, τ_1 and τ_2 are the filter time constants.

The ESS output is likewise controlled by power converters and therefore responds rapidly; its ramp rate can reach up to 100% p.u./min.

2.4. Microgrid SFR Model

By integrating the above component models of the diesel generator, steam-turbine generator, wind turbine generator, and energy storage system, this paper constructs a frequency-control model for islanded operation, as illustrated in Figure 4.

When a power imbalance causes a frequency deviation in the microgrid, each component adjusts its output to prevent excessive deviations. The system's equivalent inertia constant is computed as a capacity-weighted combination of the inertia time constants of the synchronous generators in service, as given by the corresponding formula.

$$H = (\sum_{i \in N} H_i \cdot S_i)/(\sum_{i \in N} S_i) \tag{14}$$

where N denotes the set of synchronous generating units operating normally at the current time, H_i represents the inertia time constant of the normally operating synchronous

generating unit i, and S_i denotes the rated capacity of the normally operating synchronous generating unit i.

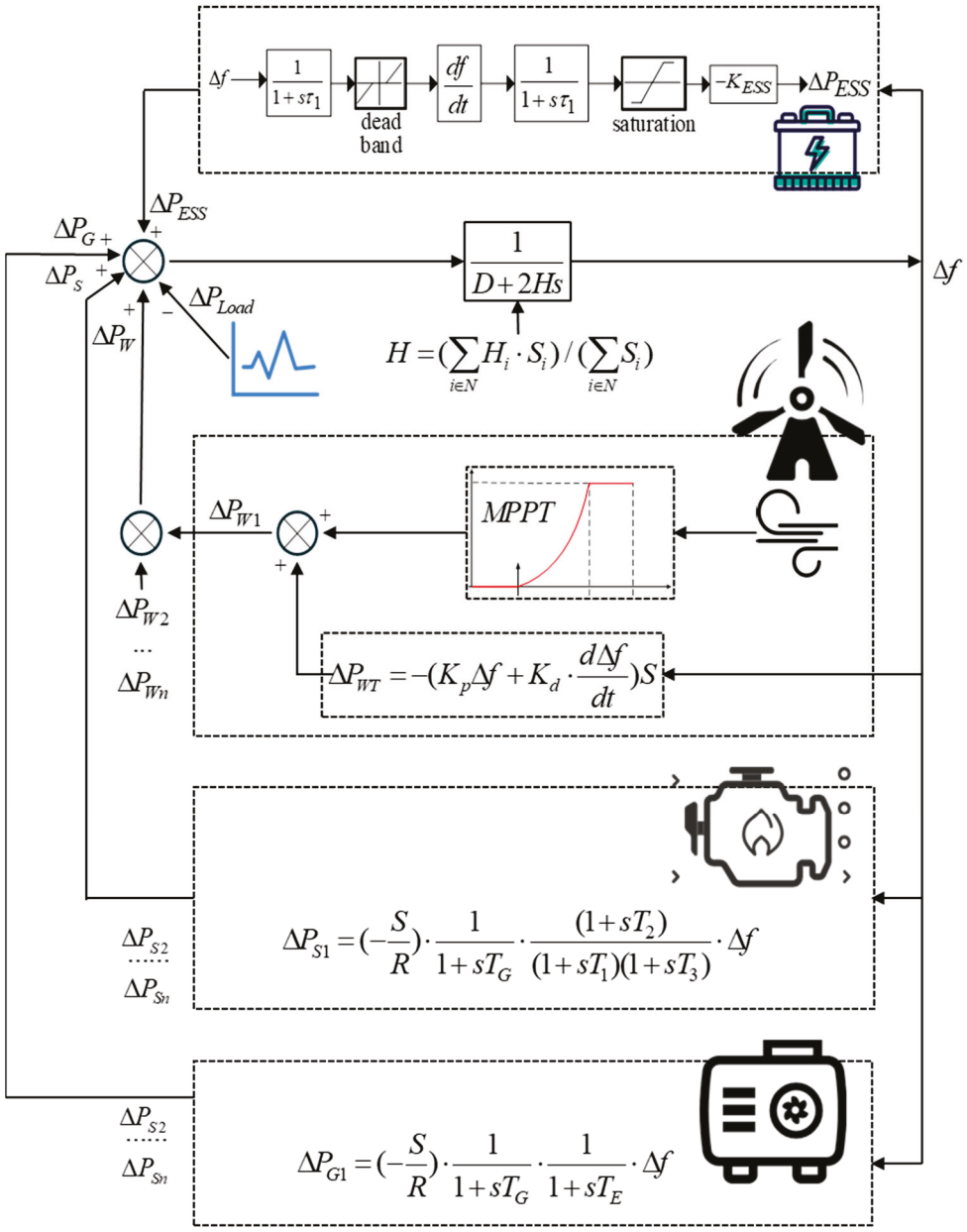


Figure 4. Block Diagram of the Microgrid System SFR Model.

3. Assessment Method of Reliability

3.1. Correction of Unit Maintenance Failure Rate Considering Frequency Deviation

The ESS tracks the grid frequency via the inverter's phase-locked loop (PLL). When frequency deviations become large, the control effort and switching stress increase, which may trigger overcurrent/overtemperature protection but seldom causes immediate failures. The impact of frequency deviations on the ESS primarily manifests as lifetime degradation; therefore, in this study the ESS failure rate is assumed to be independent of frequency. Likewise, wind turbine generators (WTGs) are decoupled from the microgrid through power converters and typically only invoke power-limiting or shutdown protections under large frequency excursions; hence, the WTG failure rate is also assumed frequency-independent. In other words, for converter-interfaced units such as WTGs and ESSs, we assume frequency-independent failure rates in view of their decoupling characteristics.

By contrast, the rotors of synchronous machines such as diesel generators and gas turbines are directly coupled to the microgrid. Frequency deviations translate into speed deviations, which in turn elevate the risk of mechanical stress, torque oscillations, and inadequate cooling, among other failure mechanisms. Following the frequency-excursion classes and tiered multipliers for failure rates reported by [16], we construct a continuous frequency-dependent function for the failure rate of each unit type to enable failure-rate correction, with the correction given by the corresponding formula.

$$m_\lambda(f) = 1 + (M-1)(1 - e^{-k|f-\delta|}) \tag{15}$$

$$\lambda(f) = m_\lambda(f)\lambda_0 \tag{16}$$

where m_λ is the failure rate multiplier, M is the frequency deviation limit multiplier, k is the shape parameter, δ is the frequency health dead zone, and λ_0 is the base failure rate. The setting of M is based on the frequency tolerance criterion for synchronous generating sets, which indicates that the risk of equipment failure may increase to M times the base value in case of frequency deviation. The shape parameter k is the rate at which the failure rate grows with frequency deviation from the deadband δ, indicating the increase in risk as the frequency crosses the limit. The deadband δ is typically set to 0.5 Hz, indicating that frequency fluctuations within this range are considered normal operating conditions and do not increase the risk of equipment failure. The failure rate frequency curve is shown in Figure 5 and characteristic data used in this paper are shown in Table 1.

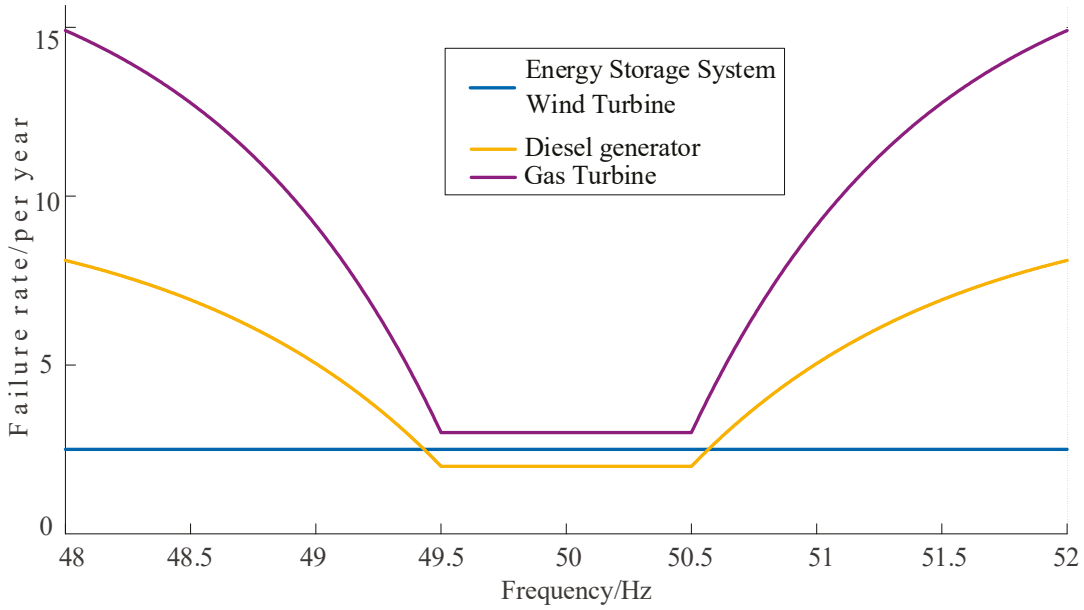


Figure 5. Frequency characteristics curve of unit failure rate.

Table 1. Failure Rate Frequency Characteristics Curve Parameters for Each Unit.

Equipment	M	k	δ
10 MW diesel generator	5	0.9575	0.5
5 MW diesel generator	4	0.9232	0.5
5 MW gas turbine	6	1.0522	0.5

Under standard assumptions, unit failures are considered memoryless; consequently, the operating time and outage time can be sampled accordingly, with the sampling formulas given as follows:

$$\begin{aligned} TTF &= -\tfrac{1}{\lambda} \ln R \\ TTR &= -\tfrac{1}{\mu} \ln R \end{aligned} \tag{17}$$

TTF denotes the time-to-failure, TTR represents the time-to-repair, and R is a random number between 0 and 1 obtained through sampling. Considering the temporal characteristics of failure rates, repairs do not occur precisely after TTR elapses following a failure. This paper introduces safety thresholds and repair thresholds, which involve performing temporal integration on failure rates and repair rates, triggering a switch when the threshold is reached.

3.2. Load Shedding Strategy

In islanded microgrids, limited inertia and reserve margins mean that relying solely on generator-provided frequency regulation to suppress the maximum rate of change in frequency (RoCoF) and frequency deviations may be insufficient to maintain stability and security. To confine contingencies within a controllable range, prevent frequency collapse, and preserve system stability, load shedding offers a rapid and effective measure. By shedding load, the probability of frequency limit violations can be substantially reduced, converting unavoidable supply interruptions into controllable, staged load curtailments.

This study adopts a staged under-frequency load shedding scheme. Multiple frequency trigger thresholds are set so that load shedding proceeds in several stages, acting sequentially according to the severity of the frequency decline. In addition, intentional delays are configured to effectively avoid nuisance operations caused by motor starting, cold-load pickup, and similar effects. The parameters of the staged UFLS scheme are given in Table 2.

Table 2. Parameters set for the underfrequency load shedding strategy.

Level	Trigger Frequency (Hz)	Single-Stage Load-Shedding Ratio	Delay (s)
UFLS-1	49.0	10%	0.3 s
UFLS-2	48.7	10%	0.3 s
UFLS-3	48.5	10%	0.5 s
UFLS-4	48.2	5%	0.5 s
UFLS-5	48.0	5%	1.0 s

In addition to configuring the staged under-frequency load shedding scheme, a load restoration strategy must be defined. Hysteresis-like, frequency-based staged actions are required to prevent oscillations caused by repeated shedding and restoration near certain frequency thresholds. The parameter settings for the restoration of shed load are listed in Table 3.

Table 3. Parameters set for the graded load restoration strategy.

Level	Trigger Frequency (Hz)	Single-Stage Load Recovery Ratio	Delay (s)
UFLS-1	49.5	10%	0.3 s
UFLS-2	49.0	10%	0.3 s
UFLS-3	48.7	10%	0.5 s
UFLS-4	48.5	5%	0.5 s
UFLS-5	48.2	5%	1.0 s

3.3. System Reliability Parameter

(1) Expected Energy Not Supplied(EENS)

Expected Energy Not Supplied (EENS) is an index used to characterize the risk of supply loss. When the microgrid frequency falls below the preset UFLS trigger thresholds and the corresponding intentional delays have elapsed, load curtailment is initiated. The EENS is calculated as:

$$EENS = \sum \alpha_{shed} L t_{shed} \tag{18}$$

Here, α_{shed} represents the load reduction ratio, L denotes the load value, and t_{shed} indicates the load reduction duration.

(2) Expected Indirect Energy Not Supplied (EIENS)

EIENS is used to characterize the power imbalance that arises during the frequency control process. Notably, this imbalance is not directly manifested as load curtailment; rather, it is reflected in the system's frequency dynamics, where such imbalances directly induce frequency fluctuations. The EIENS is computed as follows:

$$EIENS = \int \min[0, (P_G + P_{WT} + P_S + P_{ESS} - P_{Load})]dt \tag{19}$$

(3) Expected Number of Under-Frequency events (ENUF)

ENUF quantifies the expected annual number of under-frequency events. In this study, we tally the occurrences of the five under-frequency event categories defined earlier, each satisfying the UFLS staging thresholds and duration requirements.

(4) Expected Under Frequency Duration, EUFD

The microgrid frequency may deviate from the normal range and remain within a low-frequency band for a period of time until generator output is ramped up or load is reduced to restore the frequency to the normal range. The frequency will then gradually return to its nominal value through frequency-regulation actions on both the supply and demand sides. The Expected Under-Frequency Duration denotes the expected duration for which the frequency remains below its normal value, and is calculated as:

$$EIENS = \int bool[if(f < f_{UFLS})]dt \tag{20}$$

3.4. Sequential Monte Carlo Method

Given the pronounced temporal characteristics of wind turbines and energy storage systems, this study employs the sequential Monte Carlo (SMC) method to approximate the long-term reliability statistics of an islanded microgrid. Advancing continuously along the time axis, SMC yields time-sequential information that better captures the microgrid's frequency dynamics. It faithfully represents the coupling among unit state evolution, load fluctuations, and frequency dynamics, and it naturally accommodates frequency-dependent failure-rate correction, health-window constraints, and staged under-frequency load shedding (UFLS). Compared with non-sequential methods, SMC provides the event order and the system's dynamic response, aligning more closely with the actual operating characteristics of islanded microgrids.

The SMC procedure adopted in this paper is as follows:

1. Initialization: set up unit reliability data, unit control parameters, wind-speed data, and load data.
2. Enumerate microgrid state i, with i = 1 at the start: for each unit, randomly generate its initial status. If the unit is up, additionally sample its residual time to failure; if it is down, sample its residual time to repair.

3. Update the microgrid's equivalent inertia constant based on the set of units currently in outage.
4. Run the SFR model: compute each unit's output in real time from the frequency information and correct each unit's failure rate accordingly. If the frequency falls below the UFLS thresholds and the intentional delay is exceeded, automatically shed the corresponding proportion of load.
5. Unit state transitions: for each unit in normal operation, perform time integration of its failure rate and determine whether it enters the failed state. When the accumulated hazard reaches the pre-sampled cumulative hazard threshold, the unit fails; sample the repair time and reduce its output to zero subject to the ramp/power limits applicable at the outage instant. For units in outage, when the sampled repair time elapses, return the unit to service and pre-sample the next cumulative hazard threshold for the subsequent failure event.
6. Index tallying: output the reliability and frequency-security indices proposed in Section 2.3.

The overall workflow is illustrated in Figure 6.

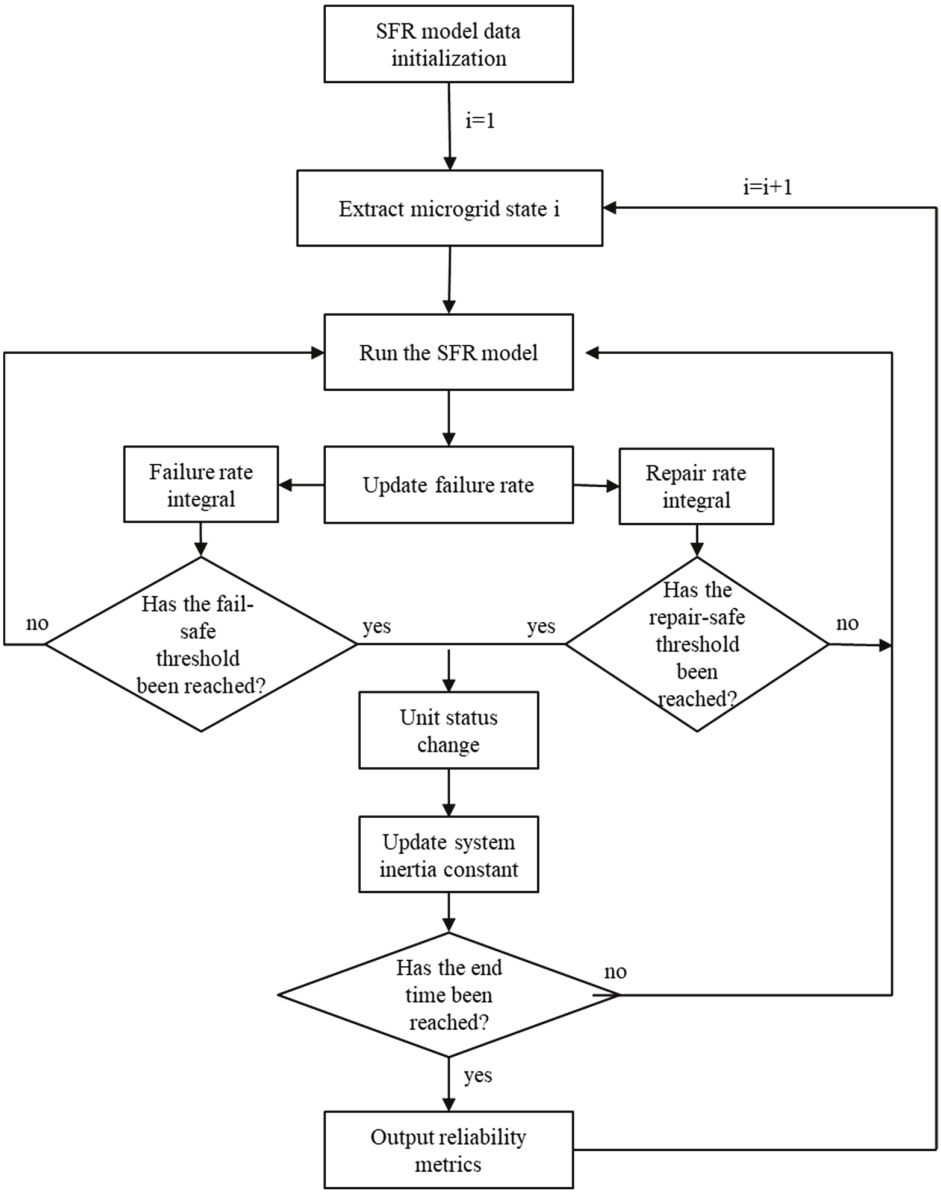


Figure 6. Sequential monte carlo algorithm process.

4. Case Study

Based on the SFR model established above, we conduct simulation studies. The model includes three diesel generators and one gas turbine with capacities of 10 MW, 10 MW, 5 MW, and 5 MW, respectively. Their failure rates (failures/year) are set to 2, 2, 2.5, and 3, and their repair rates (repairs/year) are 20, 20, 20, and 30, respectively. In addition, four identical wind turbine generators (WTGs) and one energy storage system (ESS) are included; each WTG has a capacity of 2.5 MW and the ESS is 5 MW. The WTG and ESS failure rates (failures/year) are set to 4 and 2.5, and the corresponding repair rates (repairs/year) are 35 and 40. Load data are taken from EnergyPlus annual load profiles for the New Jersey climate zone (USA), and wind-speed data are from the NREL Flatirons Campus.

4.1. Validation of the SFR Model

To verify the correctness and effectiveness of the proposed SFR model, a 90,000 s time-domain simulation is performed for the islanded microgrid configured above. In this subsection, both failure and repair rates are increased by a factor of 100 to accentuate the impact of unit state transitions. Figures 7–9, respectively, present the simulated microgrid frequency, the fraction of load connected, and the operating states of the units.

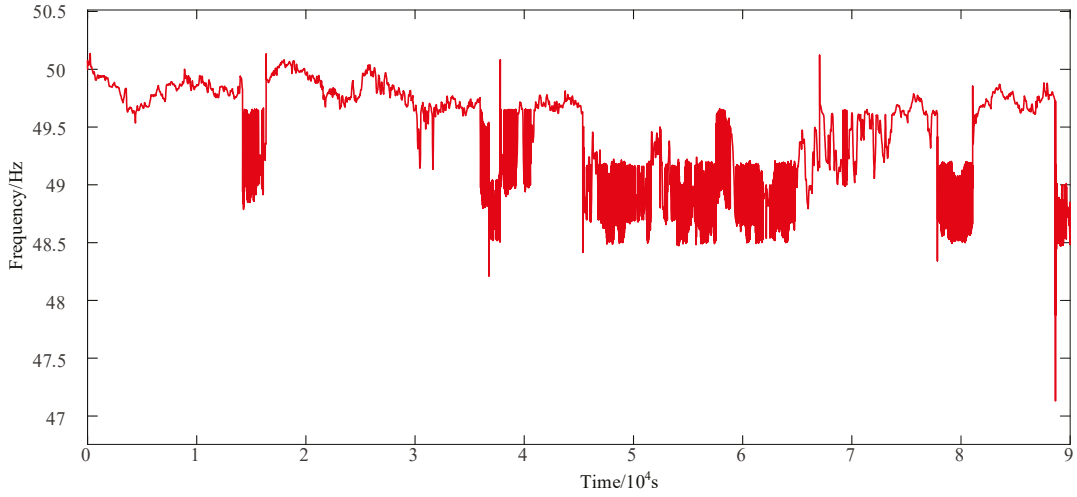


Figure 7. Dynamic frequency status of microgrids.

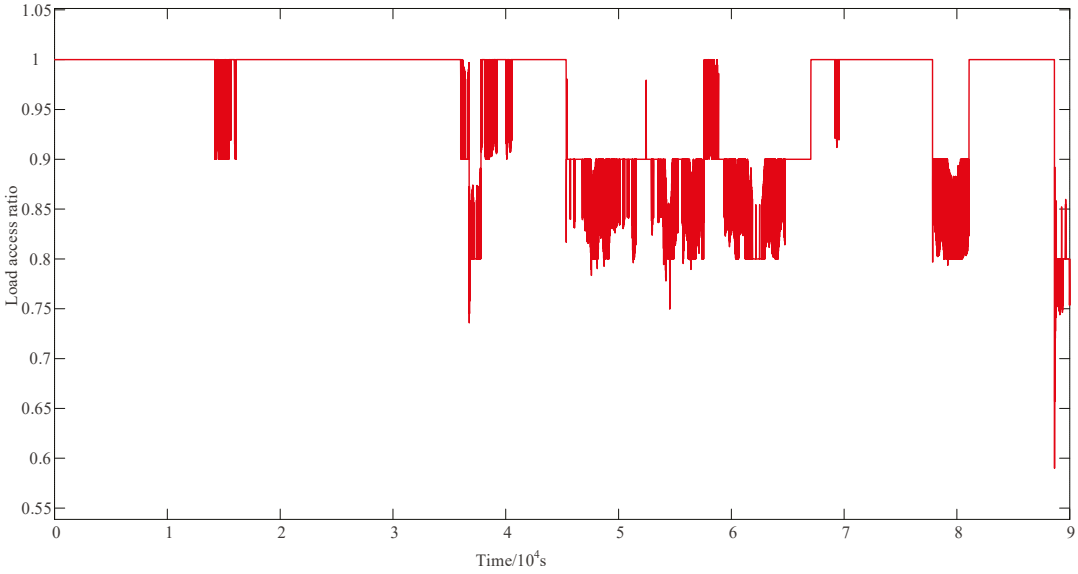


Figure 8. Dynamic status of load access ratio.

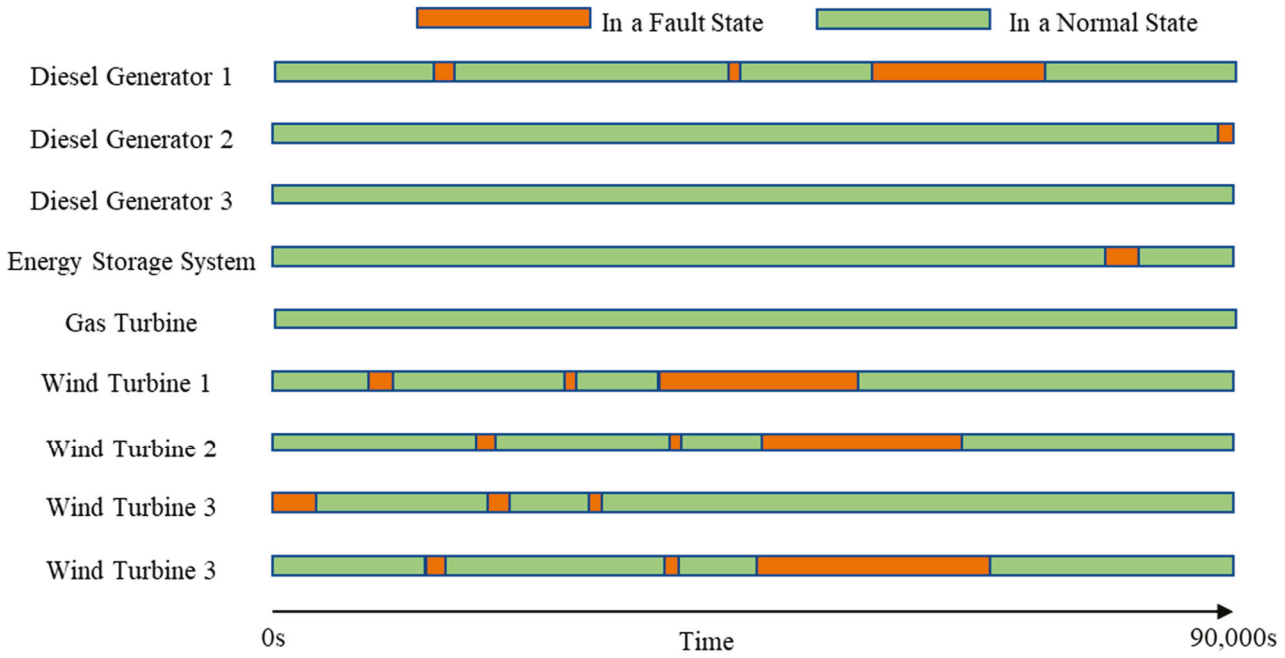


Figure 9. Operational status of the unit.

4.1.1. Impact of a Single Synchronous Unit Outage

Around 15,000 s, Diesel Generator 1 trips. As it is the primary source in the microgrid, its outage causes a steep frequency drop and imposes supply pressure, leading to frequency oscillations within 49–49.5 Hz. The served-load fraction is controlled to remain above 90%. After a brief repair, Diesel Generator 1 returns to service and the frequency stabilizes. Around 58,000 s, although the load has decreased and the frequency begins to rise to about 49.5 Hz, the microgrid attempts to restore 100% load; however, Diesel Generator 1 remains in a prolonged outage. Not until after 65,000 s, when the load declines to a level supportable by the remaining units, does the microgrid frequency return to the normal range. In addition, Diesel Generator 2 fails around 88,000 s, driving the frequency down to nearly 47 Hz for a period—an extremely hazardous range for an islanded microgrid—which forces the load to be curtailed to its minimum connection level.

4.1.2. Impact of WTG/ESS Outage

During 45,000–58,000 s, 10–20% of the load is shed for an extended period, with the frequency confined to 48.5–49 Hz. In this interval, three WTGs are out of service, so renewable output is limited, while the nominally required load is 27–33 MW—exceeding the maximum output of the diesel generators and gas turbine. The lack of supplemental renewable power prevents the microgrid from actively restoring frequency to the normal range. For most of the time, WTG output can supplement that of synchronous units; however, during load peaks, when the synchronous units are at their limits and WTGs cannot supply the excess demand, the microgrid must shed load to ensure security. After 65,000 s, an ESS failure leads to a low-frequency event around 80,000 s, similar to the case of WTG outages.

The long-term simulation above demonstrates that the SFR model can accurately capture the dynamic impact of different types of unit faults and recovery processes on system frequency. Single-unit failure causes short-term frequency fluctuations and triggers UFLS; multiple-unit failures lead to prolonged low-frequency operation; and energy storage withdrawal results in rapid loss of system support capability, triggering more severe low-frequency events. These results align with theoretical expectations, demonstrating

the validity and effectiveness of the established SFR model. This provides a reliable modeling foundation for subsequent reliability assessments based on sequential Monte Carlo methods.

4.2. Sequential Monte Carlo Results

In this section, building on the SFR model and Monte Carlo framework established above, we perform sequential Monte Carlo simulations of the system scenarios to analyze the impact of frequency deviations on generator (unit) reliability parameters, thereby demonstrating the necessity of incorporating frequency dynamics into reliability assessment.

All simulations in this study were performed on a personal workstation equipped with an Intel Ultra7-265K CPU and an NVIDIA Ultra W OC 5060Ti 16 GB GPU, running MATLAB/Simulink R2024a. Owing to the time-sequential nature of the method, parallelization is limited and the runtime mainly depends on CPU performance. Under this configuration, a single annual simulation (one full Monte Carlo sample path) required about 7–8 min. To achieve statistically convergent reliability indices, 1000 Monte Carlo trials were executed, leading to a total runtime of approximately 120 h.

The simulations compare two cases—one that accounts for the influence of frequency on unit failure rates and one that does not—with the results summarized in Table 4.

Table 4. Statistical results of reliability indicators.

Indicator	Considering the Effect of Frequency on Failure Rate	Without Considering the Effect of Frequency on Failure Rate	Deviation
EENS	18.63	15.24	22.2%
EIENS	7.41	6.12	21.1%
ENUF1	8.72	7.58	15.0%
ENUF2	5.31	4.63	14.7%
ENUF3	3.18	2.71	17.3%
ENUF4	2.07	1.62	27.8%
ENUF5	0.78	0.49	59.2%
EUFD1	422.5	367.9	14.8%
EUFD2	311.8	274.5	13.6%
EUFD3	208.6	180.3	15.7%
EUFD4	149.3	128.7	16.0%
EUFD5	92.7	76.2	21.7%

The comparison results show that the traditional static assessment method shows optimistic bias in all reliability indicators, which is mainly due to the fact that the static method fails to reflect the frequency dynamic process triggered by the power imbalance and ignores the resulting triggered ENUF and its duration EUFD; and the static method does not take into account the aggravating effect of frequency fluctuations on the synchronous generator failure rate, thus underestimating the real failure of the system under perturbation risk under disturbance.

With respect to EENS and EIENS, the values are significantly higher when frequency effects are considered than when they are ignored. This indicates that frequency deviations not only lead directly to load shedding but also indirectly increase the expected energy not supplied by exacerbating the failure rates of synchronous machines, thereby causing the supply risk to be underestimated if frequency is neglected.

For the ENUF and EUFD indices, both the expected number and the duration of under-frequency events at each stage increase when frequency effects are included. For example, the first-stage metrics ENUF1 and EUFD1 rise markedly, showing that low-frequency load-shedding actions are triggered more frequently and persist longer under frequency

disturbances. This underscores the critical role of energy storage and conventional units in frequency regulation and further indicates that omitting frequency effects in reliability assessment will fail to capture the true severity of under-frequency events.

Examining the distribution of staged load shedding, the lower levels (UFLS-1 and UFLS-2) have the highest trigger probabilities, while higher levels (UFLS-4 and UFLS-5) are relatively rare. This aligns with operational practice: most disturbances can be mitigated by low-level shedding, whereas only severe contingencies trigger high-level shedding.

Therefore, neglecting the influence of frequency on unit failure rates directly leads to an overestimation of microgrid reliability and introduces substantial bias into judgments of long-term operational reliability.

5. Conclusions

Targeting the low inertia and strong renewable output variability of islanded microgrids, this paper proposes a sequential Monte Carlo reliability assessment method integrated with a system frequency response model. An integrated SFR model comprising diesel generators, a gas turbine, an energy storage system, and wind turbine generators is established, and frequency-dependent corrections to unit failure and repair rates are introduced together with an under-frequency load-shedding strategy. In this way, reliability indices and frequency-security indices are jointly evaluated.

Simulation results show that if the influence of frequency on unit failure rates is ignored, indices such as EENS, the number of under-frequency events, and their durations will all be underestimated, leading to an overly optimistic reliability assessment. Incorporating frequency-dependent corrections more accurately reflects the potential threats posed by frequency fluctuations to microgrid operation. Compared with traditional static methods, the proposed approach can produce not only conventional reliability indices such as EENS and LOLP, but also dynamic frequency indices including ENUF, EUFD, and the low-frequency recovery probability, thus offering higher practical value.

In summary, the proposed method provides a more comprehensive reliability analysis tool for the planning, design, and operational scheduling of islanded microgrids. Future work may further integrate multi-energy complementary operation strategies, demand-side response, and comparative analyses of different control schemes to enhance the reliability and frequency-security assessment framework under high renewable penetration.

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

Funding: This project is supported by the State Grid R&D Program of China (kjgw2024-008).

Data Availability Statement: The datasets used and analyzed during this current study are available from the corresponding author on reasonable request.

Conflicts of Interest: Author Yu Zhang, Jinyue Shi, Shicheng Huang, Liang Geng, Zexiong Wang and Hao Sun were employed by the company Shijiazhuang Power Supply Company, State Grid Hebei Electric Power Co., Ltd. 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.

References

1. Kundur, P.; Balu, N.J.; Lauby, M.G. *Power System Stability and Control*; McGraw-Hill: New York, NY, USA, 1994.
2. Panteli, M.; Mancarella, P. Influence of extreme weather and climate change on the resilience of power systems: Impacts and possible mitigation strategies. *Electr. Power Syst. Res.* **2015**, *127*, 259–270. [CrossRef]
3. Viral, R.; Khatod, D.K. Optimal planning of distributed generation systems in distribution system: A review. *Renew. Sustain. Energy Rev.* **2012**, *16*, 5146–5165. [CrossRef]
4. Olivares, D.E.; Mehrizi-Sani, A.; Etemadi, A.H.; Canizares, C.A.; Iravani, R.; Kazerani, M.; Hajimiragha, A.H.; Gomis-Bellmunt, O.; Saeedifard, M.; Palma-Behnke, R.; et al. Trends in Microgrid Control. *IEEE Trans. Smart Grid* **2014**, *5*, 1905–1919. [CrossRef]
5. Billinton, R.; Allan, R.N. *Reliability Evaluation of Power Systems*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 1996.
6. Adefarati, T.; Bansal, R. Reliability and economic assessment of a microgrid power system with the integration of renewable energy resources. *Appl. Energy* **2017**, *206*, 911–933. [CrossRef]
7. Zhao, Y.; Xie, K.; Shao, C.; Lin, C.; Shahidehpour, M.; Tai, H.-M.; Hu, B.; Du, X. Integrated Assessment of the Reliability and Frequency Deviation Risks in Power Systems Considering the Frequency Regulation of DFIG-Based Wind Turbines. *IEEE Trans. Sustain. Energy* **2023**, *14*, 2308–2326. [CrossRef]
8. Chu, Z.; Zhang, N.; Teng, F. Frequency-Constrained Resilient Scheduling of Microgrid: A Distributionally Robust Approach. *IEEE Trans. Smart Grid* **2021**, *12*, 4914–4925. [CrossRef]
9. Guo, J.; Zhao, T.; Liu, W.; Zhang, J. Reliability Modeling and Assessment of Isolated Microgrid Considering Influences of Frequency Control. *IEEE Access* **2019**, *7*, 50362–50371. [CrossRef]
10. Fei, Z.; Zou, Y.; Hua, W.; Banda, O.V.; Guerrero, J.M.; Li, Z. Weather Routing Based Multi-Energy Ship Microgrid Operation Under Diverse Uncertainties: A Risk-Averse Stochastic Approach. *IEEE Trans. Smart Grid* **2025**. *Early Access*. [CrossRef]
11. Zhang, S.; Zhang, X.; Zhang, R.; Gu, W.; Cao, G. N-1 Evaluation of Integrated Electricity and Gas System Considering Cyber-Physical Interdependence. *IEEE Trans. Smart Grid* **2025**, *16*, 3728–3742. [CrossRef]
12. Nargeszar, A.; Ghaedi, A.; Nafar, M.; Simab, M. Reliability evaluation of the renewable energy-based microgrids considering resource variation. *IET Renew. Power Gener.* **2023**, *17*, 507–527. [CrossRef]
13. Rocabert, J.; Luna, A.; Blaabjerg, F.; Rodríguez, P. Control of Power Converters in AC Microgrids. *IEEE Trans. Power Electron.* **2012**, *27*, 4734–4749. [CrossRef]
14. Zamani, M.A.; Yazdani, A.; Sidhu, T.S. A Control Strategy for Enhanced Operation of Inverter-Based Microgrids Under Transient Disturbances and Network Faults. *IEEE Trans. Power Deliv.* **2012**, *27*, 1737–1747. [CrossRef]
15. Lin, J.; Yang, X.; Niu, S.; Yu, H.; Zhong, J.; Jian, L. Inflatable Savonius wind turbine with rapid deployment and retrieval capability: Structure design and performance investigation. *Energy Convers. Manag.* **2024**, *310*, 118480. [CrossRef]
16. Xue, S.; Yang, Z.; Yin, Z.; Yu, F.; Wang, Z. The Frequency Regulation Scheme of a Wind Turbine Generator Based on Variable Tracking Operation. *Electronics* **2024**, *13*, 2575. [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

LACX: Locality-Aware Shared Data Migration in NUMA + CXL Tiered Memory

Hayong Jeong, Binwon Song, Minwoo Jo and Heeseung Jo *

Department of Computer Science, Chungbuk National University, Cheongju 28644, Republic of Korea; gkdyd7167@chungbuk.ac.kr (H.J.); qlsdnjs236@cbnu.ac.kr (B.S.); chminoo@g.cbnu.ac.kr (M.J.)
* Correspondence: heesn@cbnu.ac.kr

Abstract

In modern high-performance computing (HPC) and large-scale data processing environments, the efficient utilization and scalability of memory resources are critical determinants of overall system performance. Architectures such as non-uniform memory access (NUMA) and tiered memory systems frequently suffer performance degradation due to remote accesses stemming from shared data among multiple tasks. This paper proposes LACX, a shared data migration technique leveraging Compute Express Link (CXL), to address these challenges. LACX preserves the migration cycle of automatic NUMA balancing (AutoNUMA) while identifying shared data characteristics and migrating such data to CXL memory instead of DRAM, thereby maximizing DRAM locality. The proposed method utilizes existing kernel structures and data to efficiently identify and manage shared data without incurring additional overhead, and it effectively avoids conflicts with AutoNUMA policies. Evaluation results demonstrate that, although remote accesses to shared data can degrade performance in low-tier memory scenarios, LACX significantly improves overall memory bandwidth utilization and system performance in high-tier memory and memory-intensive workload environments by distributing DRAM bandwidth. This work presents a practical, lightweight approach to shared data management in tiered memory environments and highlights new directions for next-generation memory management policies.

Keywords: Compute Express Link (CXL); tiered memory; optimization; page hotness

1. Introduction

In modern data centers and high-performance computing (HPC) environments, memory demand continues to escalate due to memory-intensive workloads such as artificial intelligence (AI), machine learning, and large-scale data analytics. This trend not only pushes system memory capacity to its limits but also exposes critical bottlenecks in memory bandwidth utilization [1,2]. Although multi-core CPUs provide high computational throughput through parallelism, bursts of concurrent memory accesses can saturate DRAM bandwidth, leading to severe system performance degradation. These bandwidth bottlenecks are particularly detrimental in real-time analytics, transactional systems, and massively parallel computations, where predictable latency and throughput are essential.

To overcome the limitations of memory capacity and bandwidth, various techniques have been explored over the past decade. Prior research has proposed tiered memory structures such as NUMA-aware placement strategies [3–6], high-bandwidth memories (e.g.,

HBM [7]), and memory compression schemes [8]. These methods combine fast yet capacity-limited DRAM as high-tier memory with slower but larger-capacity devices as low-tier memory, aiming for cost-effective scalability. While these techniques have achieved partial success, they face limitations in universality and scalability due to hardware complexity, cost, and non-portable optimizations. In particular, relying solely on DRAM expansion cannot simultaneously satisfy the capacity and bandwidth demands of modern heterogeneous workloads [1].

Recently, Compute Express Link (CXL) has emerged as a promising interconnect technology to bridge this gap [9–11]. CXL enables high-speed, low-latency communication between CPUs and memory devices, allowing a dynamic expansion of system memory beyond the constraints of traditional architectures. Through the flexible attachment of diverse memory types, CXL facilitates the construction of software-managed tiered memory systems that can adaptively allocate resources according to workload characteristics. Importantly, CXL offers a pathway for efficiently managing shared data placement and access across nodes, a challenge that conventional NUMA and AutoNUMA approaches do not explicitly address [12,13].

Furthermore, the proposed LACX framework can also provide potential benefits in emerging computing paradigms such as in-memory computing (IMC) based on non-volatile memory (NVM) technologies. In IMC environments, computation and storage are tightly coupled, and efficient data locality management becomes essential for maintaining both performance and energy efficiency. Since shared data management remains a key challenge in IMC systems where multiple processing units may concurrently access the same NVM regions, the shared page-aware design of LACX could be extended to mitigate data contention and optimize bandwidth utilization. This alignment with NVM-based IMC directions highlights the adaptability of LACX to next-generation memory-centric architectures [14,15].

Despite advances in tiered memory and automatic page migration, a fundamental limitation remains: existing systems primarily base migration decisions on access frequency (data "hotness") without distinguishing whether the accessed pages are shared among multiple tasks [16–18]. As a result, shared data—often a major contributor to remote access contention and bandwidth pressure—tends to remain fixed on a particular node or is conservatively excluded from migration. This oversight leads to suboptimal bandwidth utilization and unpredictable performance in NUMA and heterogeneous memory environments.

To address this problem, this work proposes LACX, a Linux kernel mechanism designed to identify and manage shared data pages explicitly in CXL-extended memory systems. LACX introduces a shared page-aware migration policy that relocates shared data to CXL nodes, while keeping node-local hot data in DRAM. This design mitigates bandwidth contention and improves effective memory utilization across heterogeneous tiers. By operating at the kernel level, LACX leverages intrinsic memory tracking structures to minimize overhead and maintain compatibility with the existing AutoNUMA framework.

LACX makes the following key contributions:

1. Low-overhead shared page-aware migration: It implements an efficient shared-page recognition and migration mechanism that reuses existing NUMA structures, adding minimal overhead and preserving AutoNUMA's scanning cycle.
2. Seamless integration and stability: LACX cooperates with AutoNUMA's placement policies to prevent migration conflicts, ensuring stability across diverse and dynamic workloads.

3. Balanced bandwidth utilization: By distributing local and shared data between DRAM and CXL memory, LACX achieves more balanced bandwidth usage, alleviating DRAM saturation and improving system throughput.

Through these contributions, LACX addresses a major gap in current memory management research—the lack of shared data-oriented optimization in heterogeneous memory systems. The proposed approach demonstrates that operating systems can efficiently coordinate DRAM and CXL tiers to achieve scalable bandwidth utilization and predictable performance in next-generation data centers and HPC platforms.

2. Background and Related Work

This section describes the theoretical background of our research, covering the principles and latest research trends in NUMA architecture, tiered memory, and next-generation memory interconnect technologies such as CXL [3,6,9,10]. Based on this discussion, we analyze how the LACX technique proposed in this paper differs from existing approaches.

2.1. NUMA and AutoNUMA

The NUMA architecture [19] divides a system into multiple nodes, each consisting of physically coupled CPUs and memory. When a CPU accesses memory within its own node (local memory), it achieves the highest speed, whereas accessing memory in other nodes (remote memory) incurs longer latency. As a result, it is crucial for the operating system to ensure that CPUs access memory as locally as possible, minimizing remote accesses and maximizing local accesses to reduce bottlenecks and latency [3,4].

Linux has officially supported the AutoNUMA feature since kernel version 3.8 for automatic optimization in NUMA systems. AutoNUMA automatically migrates pages and tasks to minimize the distance between them. Specifically, it triggers NUMA, hinting faults to intentionally induce page faults, thereby identifying the locations of CPUs that recently accessed the page and the node where the page is allocated [16,19].

Existing NUMA optimization research has primarily focused on improving page access locality. As a representative example, AutoNUMA, which was introduced earlier, is not without its limitations [5,16]. Excessive page migration can degrade performance, and the migration process itself can introduce additional overhead due to conflicts with the CPU load balancer, increased translation lookaside buffer (TLB) misses, and page table walks. In particular, frequent TLB misses can cause significant delays if the page table is located on a remote node. Furthermore, when all CPUs attempt to optimize local accesses, memory controllers on certain nodes may become overloaded, resulting in local access latencies that exceed those of remote accesses.

Notably, AutoNUMA primarily focuses on optimizing memory access locality for individual tasks and is less effective in handling shared data accessed by multiple tasks [5,6]. Since AutoNUMA performs migration based only on page-level locality, it is difficult to achieve clear locality benefits by migrating pages containing shared data to any particular node. Consequently, AutoNUMA tends to handle shared data conservatively or even avoid migrating it altogether. These limitations remain as important challenges for future NUMA optimization research.

To overcome these limitations, this paper proposes LACX, a novel mechanism that actively identifies and manages shared data across tasks. While AutoNUMA avoids migrating shared pages to prevent performance degradation, LACX leverages the same hinting-fault mechanism to detect frequently shared pages and strategically isolate them in dedicated memory nodes, thereby improving locality for private data and reducing contention around shared memory.

2.2. Tiered Memory

In tiered memory systems, DRAM serves as the high-speed, high-bandwidth high-tier memory, while non-volatile memory (NVM) or other extension memory technologies, which offer larger capacity and cost efficiency, act as low-tier memory. Low-tier memory is connected to the system via high-speed interfaces such as PCIe; NVM technologies like Optane are representative examples [20,21]. These low tiers are critical for transaction-based services requiring large capacity and data integrity.

The primary strategy in tiered memory involves analyzing workloads' memory access patterns to place data in appropriate tiers based on access frequency (hotness). Ideally, frequently accessed (hot) data should reside in high-tier memory, while less frequently accessed (cold) data should be allocated to low-tier memory. Automated techniques that migrate data from low-tier to high-tier when frequently accessed are widely used, and ongoing research aims to improve their efficiency [12,17].

Prior work such as Transparent Page Placement (TPP) [17] focuses on identifying hot and cold pages at the OS level and proactively demotes cold pages from high to low tiers via lightweight mechanisms. These approaches improve throughput and reduce overhead in data center workloads by ensuring that high-tier memory is mainly occupied by frequently accessed pages. However, hotness-based methods can be sensitive to rapid changes in access patterns and may incur a high migration overhead.

More recent schemes like Colloid [22] aim to balance effective access latency across tiers rather than solely relying on access frequency. Despite reducing some shortcomings of hotness-based methods, they still struggle to manage pages accessed concurrently by multiple tasks, limiting their effectiveness for shared data.

Recent work [23] shows that access frequency alone does not always indicate performance-critical data. They introduce Amortized Offcore Latency (AOL), a metric capturing the true impact of memory accesses by considering both latency and memory-level parallelism. Using AOL, they propose allocation (SOAR) and migration (ALTO) policies that significantly outperform traditional hotness-based mechanisms across diverse workloads.

Building upon these insights, LACX explicitly targets shared data in tiered memory environments with CXL. By offloading shared pages to CXL nodes, LACX reduces contention on DRAM and preserves DRAM capacity for latency-sensitive private data. Unlike device-side techniques such as HeteroMem [11], which rely on hardware-managed page promotions, LACX uses a host-side NUMA fault-based mechanism to efficiently manage data placement with minimal overhead, enhancing memory bandwidth scalability in multi-threaded workloads.

2.3. CXL

CXL has emerged as an effective solution for overcoming the capacity limitations of traditional memory in server and data center environments with high memory demands [9,10]. CXL provides high bandwidth by leveraging the PCIe 5.0 interface and enables the construction of heterogeneous memory pools through the cxl.mem protocol. From the perspective of the host CPU, CXL devices are recognized as independent NUMA nodes, ensuring compatibility with existing NUMA-based memory management techniques [12,13].

However, CXL exhibits inherent performance differences compared to high-tier memory (DRAM). CXL is classified as a lower-priority, low-tier memory and shows marked differences from high-tier memory in terms of access latency and bandwidth. This performance heterogeneity makes the limitations of traditional NUMA policies evident even when CXL devices are recognized as NUMA nodes [22]. While existing NUMA policies

determine memory placement based on a single distance metric between memory domains, heterogeneous memory environments such as those using CXL require consideration of multiple performance metrics, including latency, bandwidth, and capacity [21,24].

Accordingly, when conventional NUMA policies—which focus primarily on access distance—are applied directly to CXL-based systems, they fail to adequately account for performance heterogeneity, potentially leading to inefficient tiered memory. This can prevent the system from achieving the expected performance improvements or even result in performance degradation. Moreover, improper migrations may increase overhead and reduce overall system performance.

In response, LACX incorporates CXL as a strategic target for shared data placement. Rather than attempting to maintain DRAM locality for all pages, LACX isolates highly contended shared data into CXL nodes. This allows DRAM to be reserved for private pages that are latency-sensitive, while shared data benefits from increased aggregate bandwidth via off-node access. This architectural distinction enables LACX to both respect the constraints of performance heterogeneity in CXL and actively utilize it to mitigate NUMA contention.

3. Design

This section details the design of LACX, focusing on its core functionalities for identifying shared data and migrating it to CXL nodes. LACX operates at the page level, which is the fundamental unit of memory migration in the Linux kernel. Therefore, in the following discussion, shared data is referred to as a "shared page" from an implementation perspective.

3.1. Approach

In environments where system memory resources are intensively utilized, data sharing among different processes and threads becomes increasingly common. As a result, remote memory accesses between nodes frequently occur in NUMA systems, which is a major cause of overall system performance degradation [3,16]. The Linux kernel's AutoNUMA identifies and distinguishes pages that experience shared access, known as shared fault pages. However, it employs a conservative policy that keeps these pages on the node where they were first accessed, rather than distributing or relocating them [16]. This approach induces repeated remote accesses to shared pages, further exacerbating bottlenecks in NUMA systems.

In tiered memory environments, if the system has sufficient free memory, some shared data remaining in high-tier memory does not pose a significant problem [12,17]. However, when multiple workloads run concurrently or a single workload consumes large amounts of memory, high-tier memory quickly becomes saturated. In such cases, if shared data still occupies certain portions of high-tier memory, it leads to inefficient utilization and a decline in overall memory performance [22].

To address these issues, this paper focuses on populating high-tier memory with pages that are highly likely to be accessed locally. Instead of analyzing the locality of each individual page access, our approach determines the sharing status of pages and leverages this information for placement decisions [4,5]. By focusing on whether a page is shared among multiple tasks or primarily accessed by a single local task, we ensure that high-tier memory is predominantly occupied by pages with strong locality to the local node.

Therefore, our approach designates shared data as the primary target for migration to CXL nodes, leveraging the advantages that CXL provides [9,10]. By migrating shared pages, LACX maximizes the utilization of high-tier memory for local accesses, thereby

amplifying the performance benefits and enabling more efficient maintenance of overall system performance.

3.2. Shared Fault

AutoNUMA aims to reduce latency by migrating pages to appropriate NUMA nodes based on memory access patterns [16]. However, incorrect migrations can introduce overhead and degrade overall system performance. To prevent such issues, AutoNUMA interprets hinting faults in two categories.

The first category is local fault and remote fault. These are determined by whether the page resides on the same physical node as the CPU core where the current task is executing. If the page is located on the current task's node, a local fault occurs; otherwise, a remote fault occurs [16].

The second category is private fault and shared fault. These are determined by whether the current accessing task is the same as the previous task that accessed the page. If the same task accesses the page, it is classified as a private fault; if a different task accesses the page, it is classified as a shared fault [16]. When a shared fault occurs, AutoNUMA recognizes the page as a shared page.

AutoNUMA combines these two categories to determine whether to migrate a page. For example, if a remote fault and a private fault occur together on a page primarily used by a specific task, it indicates that the task is accessing its main pages remotely. In this case, AutoNUMA migrates the task to the node where the page is located [16].

However, the situation is different when a shared fault occurs. A shared fault arises when different tasks access the same page, and in this case, migration cannot ultimately avoid remote access [16]. Figure 1 shows how page access varies depending on the page location when a shared fault occurs. Assuming there are three NUMA nodes, if two tasks are located on separate nodes and the page is located on a third node, both tasks will access the page remotely. If the page is located on one of the tasks' nodes, the other task will still access the page remotely. In other words, if a shared fault occurs among tasks on different nodes, at least one task will always incur a remote access. As a result, AutoNUMA generally avoids migrating pages on which shared faults have occurred.

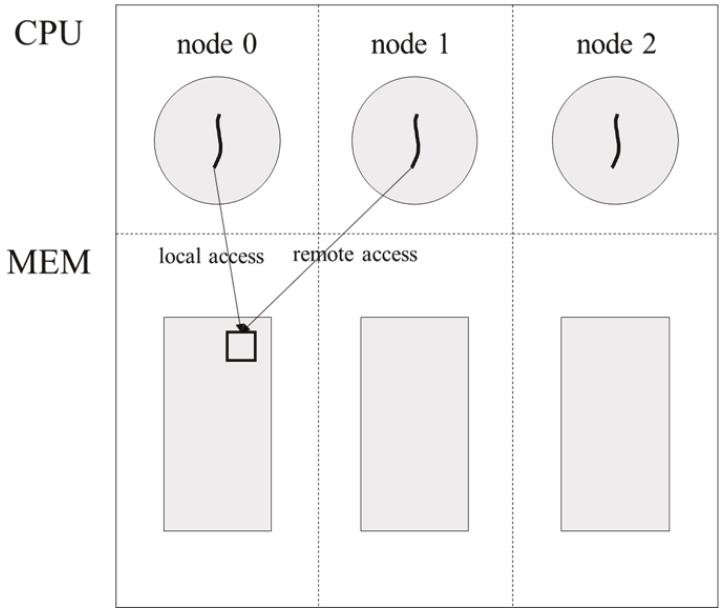


Figure 1. Shared fault based on page location.

While AutoNUMA's conservative policy for shared data—retaining pages that incur at least one remote access in high-tier memory—is rational given the inevitability of remote accesses, it does not constitute an efficient use of high-tier memory resources [6,16]. By keeping shared pages in high-tier memory, AutoNUMA allows these pages to occupy valuable space that could otherwise be allocated to data with high local access frequency, thereby limiting the overall efficiency of memory utilization. This can lead to performance degradation as memory usage increases, especially in workloads where multiple threads share memory.

In contrast, LACX adopts a proactive strategy that explicitly migrates shared data to low-tier memory (CXL), reserving high-tier memory for data with strong local access patterns [17,22]. This approach not only prevents the wasteful allocation of high-tier memory to shared pages but also maximizes the benefits of tiered memory improved locality for private data and expanded bandwidth for the system as a whole. As a result, LACX enables more effective utilization of high-tier memory and better accommodates the demands of memory-intensive, multi-threaded workloads.

3.3. Shared Data and Migration

The existing AutoNUMA shared fault classification mechanism is designed to minimize performance loss caused by incorrect migrations [16]. To achieve this, AutoNUMA checks for conditions that make migration unsuitable and only migrates pages that do not meet any of these exclusion criteria. As a result, AutoNUMA marks pages shared among different tasks as unsuitable for migration. For example, pages such as memory-mapped (mmap) file pages and anonymous shared memory pages are excluded from migration. However, since these pages are often accessed by tasks on multiple nodes, LACX actively identifies and considers such shared pages as migration candidates.

LACX identifies shared pages based on information already present in existing Linux kernel data structures. Specifically, it utilizes three types of information: (1) the tasks that have accessed the page, (2) the NUMA nodes where the page and tasks are located, and (3) the identifier of the most recent task to access the page. Using these, LACX determines whether a page is shared among tasks on multiple nodes. The specific pseudo code for shared page identification can be found in Algorithm 1.

Algorithm 1: Pseudo code for IS_SHARED_PAGE

```
is_shared_page page, current_task
    current_cpu ← get_current_cpu();
    current_node ← get_numa_node(current_cpu);
    current_cpupid ← encode_cpupid(current_cpu, current_task.pid);
    last_cpupid ← exchange_page_last_cpupid(page, current_cpupid);
    if page is mapped by more than one task then
        if current_task.pid ≠ last_cpupid then
            return true;
    return false;
```

Shared pages identified through Algorithm 1 are subsequently considered as migration candidates. The actual migration is performed within the `do_numa_page()` function, which is responsible for making migration decisions in AutoNUMA [16]. During this function, LACX adopts the existing page migration mechanism of AutoNUMA but specifically identifies shared pages and permits their migration, explicitly fixing the target location to the CXL node [17]. Migration is performed only after precondition checks and resource

verification to minimize bottlenecks. If migration fails, the page remains on its original node instead of being moved to the CXL node. This approach precisely identifies shared pages and promotes their migration to CXL nodes, thereby mitigating conflicts and imbalances in high-tier memory and improving the overall memory utilization efficiency of the system.

Figure 2 shows the process by which LACX identifies shared pages and migrates them to CXL. Migrating shared pages to CXL nodes may result in remote accesses, raising potential concerns about performance degradation. However, this migration increases the available space in high-tier memory so that it can be more effectively filled with pages exhibiting local access frequency [22]. This improves overall memory access efficiency and ultimately contributes to enhanced system performance.

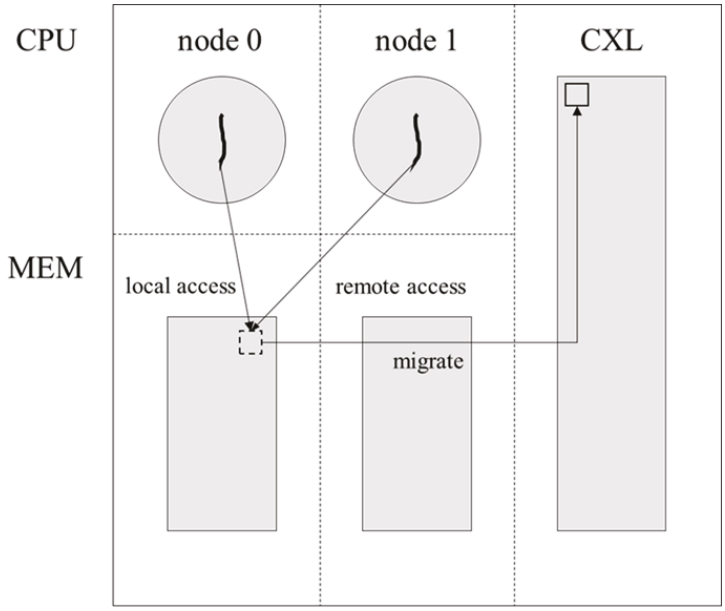


Figure 2. Identify shared data and migrate to CXL.

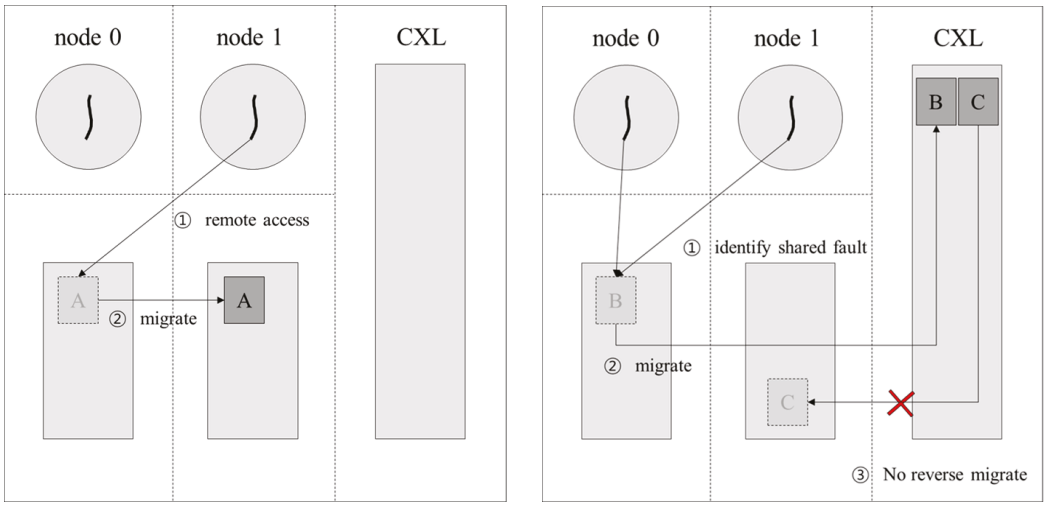
3.4. Avoiding Overhead

Although more effective identification of shared pages improves accuracy, determining the sharing status of all pages can introduce significant overhead [5,21]. Most existing tiered memory techniques employ monitoring and decision mechanisms for the precise measurement of page access frequency (hotness), which have been noted as significant sources of performance overhead. To address this, our approach identifies the sharing status of pages within the existing AutoNUMA scan cycle, rather than relying on additional sampling-based mechanisms or hardware-based performance monitoring tools [16]. This enables the detection of shared data without introducing extra overhead. Furthermore, to minimize the overhead associated with migrating shared pages to CXL nodes, our implementation maximizes the use of existing kernel functions and data structures [16]. This strategy ensures that high-tier memory remains populated primarily by pages with a high probability of local access, while avoiding additional memory consumption, thereby maintaining memory access efficiency and overall system performance.

3.5. Avoiding Collision with AutoNUMA

When shared pages are migrated to CXL nodes, AutoNUMA may misinterpret these pages as being placed in incorrect locations. Figure 3a shows the default behavior of AutoNUMA [16]. By default, AutoNUMA follows a policy of preferentially placing data in high-tier memory. In contrast, Figure 3b demonstrates how LACX identifies shared pages and migrates them to CXL nodes. In this figure, page B is a shared page accessed

by CPUs on both node 0 and node 1 and is identified by LACX as such and migrated to a CXL node. Page C is also migrated to a CXL node by LACX, but AutoNUMA mistakenly considers it to be in an incorrect location and attempts to migrate it back to its original node. When a shared page resides in high-tier memory, AutoNUMA generally does not migrate it, since it is considered to belong to the high-tier regardless of its specific node. However, if a shared page is located in a lower-priority node such as CXL, AutoNUMA regards it as being in an "abnormal location" and tries to migrate it back to a high-tier node. To prevent such unnecessary reverse migrations, LACX marks pages that are identified as shared and migrated to CXL nodes. This ensures that these pages are excluded from AutoNUMA's migration priority list, effectively avoiding conflicts between the two policies [16].



(**a**) Page migration in AutoNUMA (**b**) AutoNUMA with LACX

Figure 3. Overview of AutoNUMA and LACX.

4. Evaluation

This section assesses the performance of LACX by examining its approach of identifying shared data and migrating it to CXL nodes, using experimental results from both single-workload and multi-workload scenarios to analyze the impact of memory usage.

4.1. Environment

The experimental machine is equipped with eight sockets, a total of 192 cores (24 cores per socket), and 512 GB of memory (64 GB per node). To simulate a CXL environment, virtual machines were employed. Table 1 shows the system configuration. Among the eight NUMA nodes, three were utilized for experiments. Specifically, two NUMA nodes (each with 24 CPU cores and 18 GB DDR4-3200 memory) and one CPU-less 18 GB DDR4-3200 memory node were configured. To emulate the memory latency characteristics of CXL, the CPU-less memory node was assigned a higher NUMA distance [10,12]. Throughout this paper, the experimental nodes are referred to as follows: the two nodes with both CPU and memory (nodes 0 and 1) are designated as high-tier nodes, and the CPU-less memory-only node (node 2) is referred to as the CXL node.

Table 2 summarizes the per-node memory access latency according to NUMA distance. The rows represent the requesting nodes, and the columns denote the target memory nodes. The memory access latency for high-tier nodes (node 0 → node 0, node 1 → node 1) is the lowest, at 151.5 ns and 152.0 ns, respectively. In contrast, remote accesses between different high-tier nodes (node 0 → node 1, node 1 → node 0) range from 394.1 ns to

400.4 ns, approximately 2.6 times higher. Notably, accesses from high-tier nodes to the CXL node exhibit the highest latency, ranging from 454.8 ns to 456.6 ns, which exceeds even the inter-node latency between high-tier nodes. These latency characteristics closely mirror the performance differences observed in real CXL-based tiered memory environments, where CXL access latency typically falls in the range of 400–500 ns [10,12]. In this study, we configured the evaluation environment by assigning a high NUMA distance to a CPU-less node to emulate the latency behavior of a CXL tier. This approach follows the same methodology validated by previous studies [17,22] and aims not at reproducing absolute physical performance but at analyzing the relative performance trends of data migration between DRAM and CXL memory. Therefore, the results obtained under this setup effectively illustrate how the proposed policy alleviates bandwidth pressure on high-tier memory and enables balanced utilization of the entire system bandwidth, which remains valid even in real CXL-based environments.

Table 1. System configuration.

OS	Ubuntu 24.04 LTS
Kernel	Linux Kernel 6.8.0
CPU	Intel Xeon E7-8890 v4 @ 2.20 GHz 8 sockets, 192 cores total
V.CPU	48 cores total (24 cores per virtual node 0, 1)
Memory	DDR4-3200, 512 GB
V.Memory	54 GB total (18 GB per virtual node 0, 1, 2)

Table 2. Per-node memory access latency of experimental system (unit: ns). (Requesting : node in rows, target memory node in columns).

Requesting Node	**Target Memory Node**		
	Node 0	Node 1	Node 2 (CXL)
Node 0	151.5	400.4	454.8
Node 1	394.1	152.0	456.6

Experiments were conducted by comparing the performance of the Linux kernel 6.8.0 with LACX implemented and the standard kernel of the same version. To effectively identify shared data and validate CXL migration, we used SHM-bench [25], a microbenchmark designed for this purpose. SHM-bench accepts the shared memory size and the number of threads as arguments, and the generated threads are distributed across all nodes to perform sequential read and write operations on the shared memory, thereby inducing intentional shared faults. For real-world workloads, we employed yahoo! cloud serving benchmark (YCSB) [26], which is widely used for key-value store performance evaluation, and the mg.D workload from NASA parallel benchmarks (NPB) [27].

The evaluation aims to answer the following key questions:

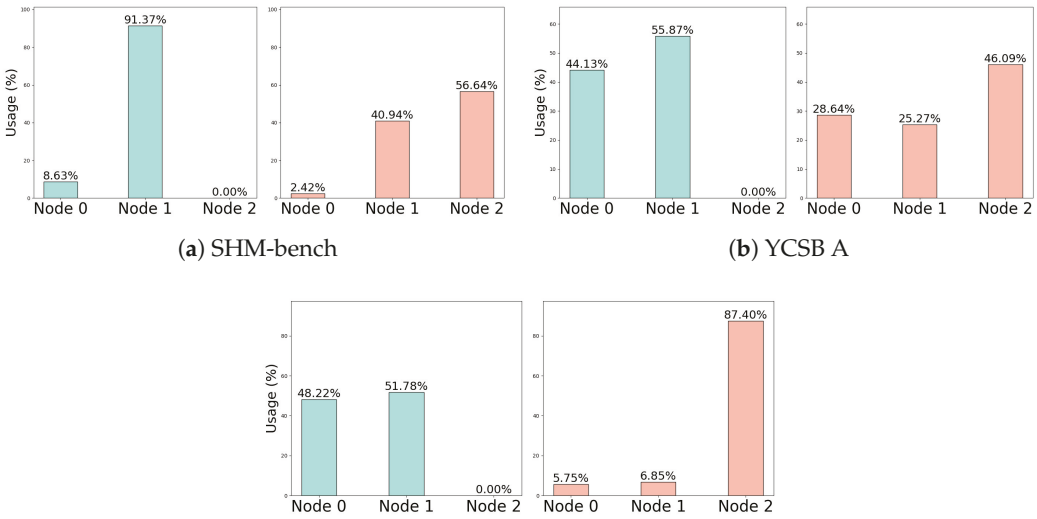
1. Can shared data be accurately identified and migrated?
   This evaluation quantitatively assesses how accurately LACX can detect shared data and how effectively it can migrate such data to the CXL node. In particular, the precision of shared data detection and the appropriateness of migration are analyzed in detail according to workload characteristics and access patterns.
2. How does the migration policy impact system performance?

To evaluate the impact of shared data migration on system performance, we measured overall performance changes under both single- and multi-workload scenarios. Specifically, we closely analyzed how migration overhead and the latency of accessing low-tier nodes affect workload execution time and resource efficiency. This allows us to verify whether LACX's policy contributes to real-world performance improvements.

3. How does LACX affect memory tier utilization and bandwidth usage?
   By analyzing changes in data distribution between high-tier nodes and CXL nodes as well as bandwidth utilization patterns, we assess how LACX influences the balance and efficiency of the entire memory hierarchy. In particular, we focus on the bandwidth usage of high-tier nodes and CXL nodes according to overall system memory utilization.

4.2. Distinction and Migration of Shared Data

To quantitatively evaluate whether LACX can effectively identify shared data and accurately migrate it to CXL nodes, we used SHM-bench, YCSB A, and NPB workloads. Figure 4 compares the per-node memory usage ratios between the standard kernel and LACX environments for each workload. The x-axis represents the NUMA node number, and the y-axis indicates the proportion of total memory allocated to each node.



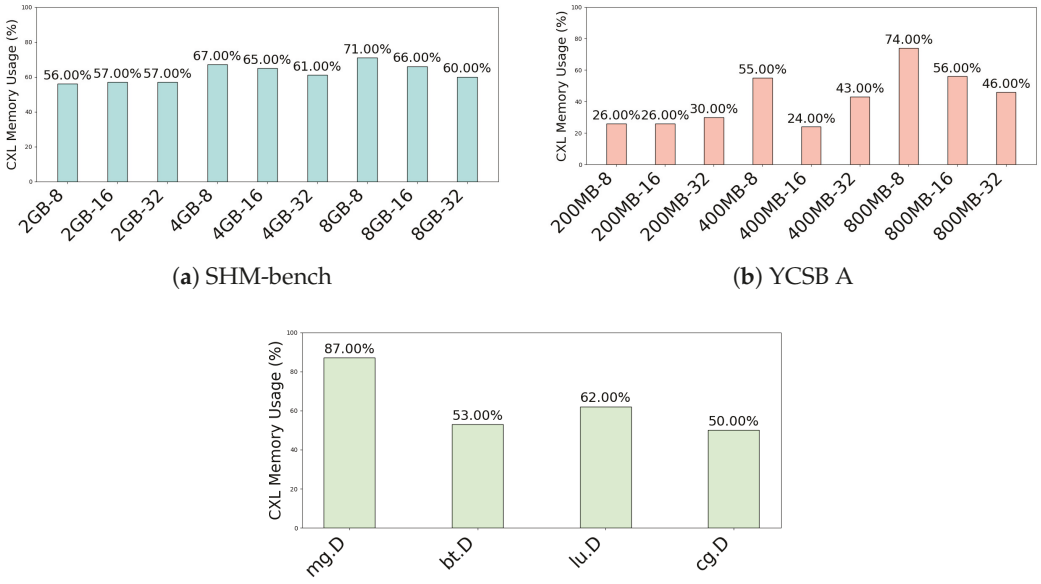
(**a**) SHM-bench (**b**) YCSB A

(**c**) NPB mg.D

Figure 4. Comparison of memory usage for each node in a general kernel and LACX.

Figure 4a shows the results for SHM-bench, where 32 threads operate on a 2 GB shared memory segment. Under the standard kernel, most memory is allocated to the high-tier nodes with CPUs, and no memory is allocated to the CXL node. This is because the standard kernel deprioritizes allocation to the CXL node to avoid remote accesses that could degrade performance. In contrast, LACX migrates approximately 56.64% of the total memory used by SHM-bench to the CXL node, demonstrating its ability to effectively identify and migrate pages shared by many threads. Figure 4b shows the results for the YCSB A workload, where 32 threads operate on a 800 MB record. Although YCSB A is a real-world, unstructured workload, it exhibits patterns where multiple threads reference common data during execution. Accordingly, LACX migrates about 46.09% of the total memory to the CXL node. Figure 4c shows the results for NPB mg.D, a numerically intensive parallel workload in which many processes repeatedly access a global matrix structure. As a result, LACX migrates 87% of the total memory to the CXL node, the highest migration ratio among the three workloads.

Figure 5 shows how the migration ratio to the CXL node varies with different workload configurations in the LACX environment. The x-axis represents the workload settings, combining memory usage and thread count, and the y-axis indicates the proportion of memory migrated to the CXL node for each configuration.



(**a**) SHM-bench (**b**) YCSB A

(**c**) NPB

Figure 5. Migration rate from LACX to CXL node by workload.

Figure 5a shows that SHM-bench, which intentionally generates shared data, maintains a high migration ratio across all settings. The migration ratio ranges from a minimum of 56% (2 GB, 8 threads) to a maximum of 71.00% (8 GB, 8 threads), with larger memory usage leading to more shared pages. Figure 5b for YCSB A shows that the migration ratio varies from 24.00% (400 MB, 16 threads) to 74% (800 MB, 8 threads), with a sharp increase as memory usage increases, even with fewer threads. Figure 5c for NPB shows different results depending on the workload type, with cg.D exhibiting about a 50% migration ratio and mg.D reaching up to 87.40%. This variation is attributed to differences in the amount of shared data within the internal parallel processing structures of each NPB workload.

These results quantitatively demonstrate that LACX can precisely identify shared data based on workload characteristics and access patterns and migrate it appropriately to CXL nodes. In particular, workloads with a larger number of threads, higher memory usage, or parallel computation structures with shared data exhibit higher migration ratios. This clearly indicates that LACX successfully implements a sharing-aware migration policy across diverse workload environments.

4.3. Performance Impact

Migrating shared data to CXL nodes can lead to performance degradation in single-workload environments. This is primarily due to the overhead associated with remote memory access and the costs incurred during data migration. However, as the number of concurrent workloads increases and overall system memory utilization grows, actively leveraging the CXL node can result in system-wide performance improvements.

Figure 6 presents a comparison of execution times between the standard kernel and LACX across various memory and thread count configurations for SHM-bench, YCSB A, and NPB workloads. In SHM-bench, when 32 threads access 2 GB of shared memory, LACX exhibits approximately a 14% increase in execution time. For YCSB A, processing

200 MB of records with 32 threads results in about a 12% execution time increase under LACX. Although only YCSB A results are shown here, additional experiments with other YCSB workloads revealed similar trends, justifying the use of YCSB A as a representative example. Single-instance NPB workloads display comparable patterns, where execution times increase under LACX across mg, bt, lu, and cg workloads, with cg.D showing the highest increase of nearly 46.6%. This degradation is linked to high shared data access intensity and computational dependencies in these workloads, where remote access latency and migration overhead become critical bottlenecks.

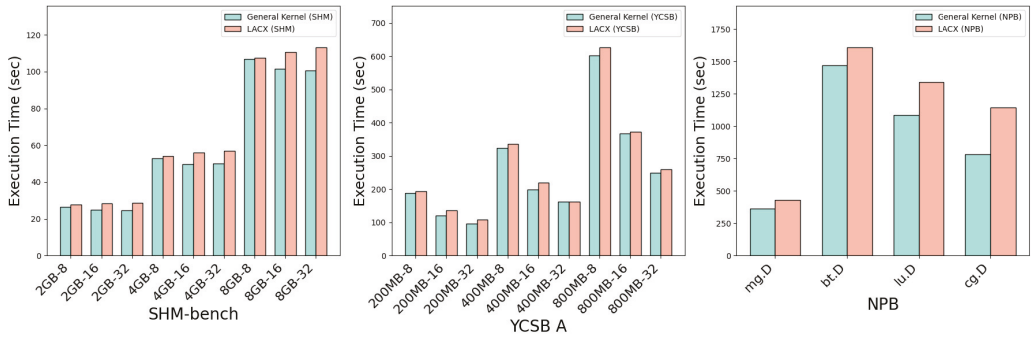


Figure 6. Comparison of single-instance workload execution time differences between the general kernel and LACX.

Workloads with greater thread-level parallelism and denser shared data tend to experience heightened impact from remote access latency and migration overhead, which appears to be the principal cause of performance degradation in single-workload scenarios. Conversely, in multi-workload or memory-constrained situations, LACX's strategy of migrating shared data to the CXL node reduces saturation on high-tier nodes, thereby improving memory resource utilization and bandwidth efficiency, as supported by bandwidth and memory distribution measurements. The impact of LACX varies by workload characteristics. SHM-bench, featuring a straightforward shared memory access pattern, benefits less from increased parallelism, while migration overhead scales with shared access intensity. YCSB A leverages thread-level parallelism, leading to decreased execution times with more threads; however, this improvement is partially offset by LACX's migration of shared data to the CXL node. For NPB workloads, the effect is strongly influenced by how shared data is structured and accessed, resulting in varying performance outcomes.

Extending this analysis, Figure 7 illustrates execution time changes when running ten sequential YCSB A workloads. In low memory pressure environments with a single workload, LACX causes up to a 14% increase in execution time; however, under multitasking conditions with higher aggregated memory demands, LACX reduces execution time by up to 3%. This gain is attributed to the retention of high-locality pages in high-tier nodes and the proactive shared data distribution to the CXL node, which alleviates pressure on high-tier resources. Consequently, memory utilization and effective bandwidth increase, leading to shorter execution times. This behavior is closely tied to NUMA memory allocation strategies. When multiple workloads run sequentially, early workloads primarily consume high-tier node memory, forcing later workloads to allocate from the CXL node as high-tier nodes saturate. LACX's proactive migration of shared data preserves high-tier memory for private data, maximizing local access and enhancing overall system performance.

Similar trends emerge in NPB workloads. Running five repeated executions of mg, bt, lu, and cg (size D) with nine threads each, mg and lu demonstrate execution time reductions of approximately 5% and 11%, respectively, evidencing LACX's effectiveness in highly parallel workloads with well-distributed shared data. Conversely, bt and cg

exhibit execution time increases (about 25% and 15%), likely due to concentrated shared access patterns or strong computational dependencies, where migration overhead surpasses the benefits.

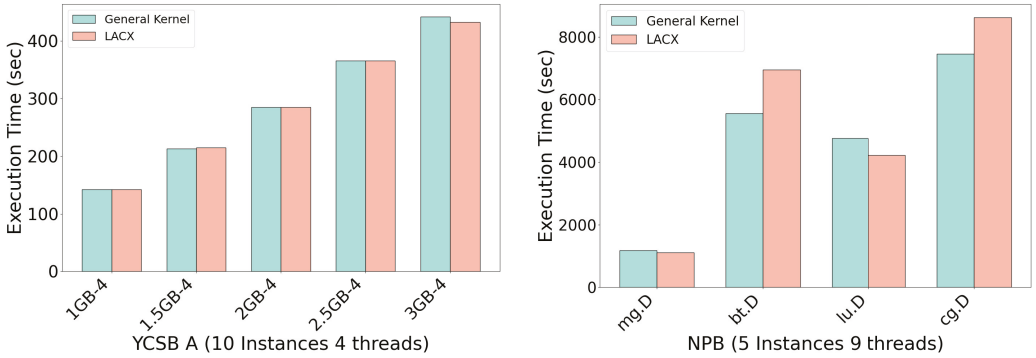


Figure 7. Comparison of multi-instance workload execution time between general kernel and LACX.

To investigate LACX's impact under severe memory constraints, we evaluated the mg.D workload with 40 threads and a memory demand of nearly 90% of the total VM capacity. In this scenario, high-tier node memory alone cannot accommodate the workload, necessitating data allocation to the CXL node. Figure 8 compares execution times between the standard kernel and LACX across different memory capacity ratios of high-tier to CXL nodes. When high-tier nodes have 2.5 times the memory of the CXL node, performance differences are negligible. At a 1.5:1 ratio, LACX achieves a 2.1% reduction in execution time; when the CXL node's capacity exceeds that of high-tier nodes (1:1.5), LACX attains a 6.2% performance improvement. These results highlight that in memory-constrained environments, proactive shared data migration to the CXL node is especially beneficial. Unlike the standard kernel's arbitrary data distribution, which neglects locality, LACX selectively migrates shared data to the CXL node and reserves high-tier memory predominantly for private data. This strategy maximizes local memory access and yields increasingly significant performance advantages as reliance on the CXL node grows.

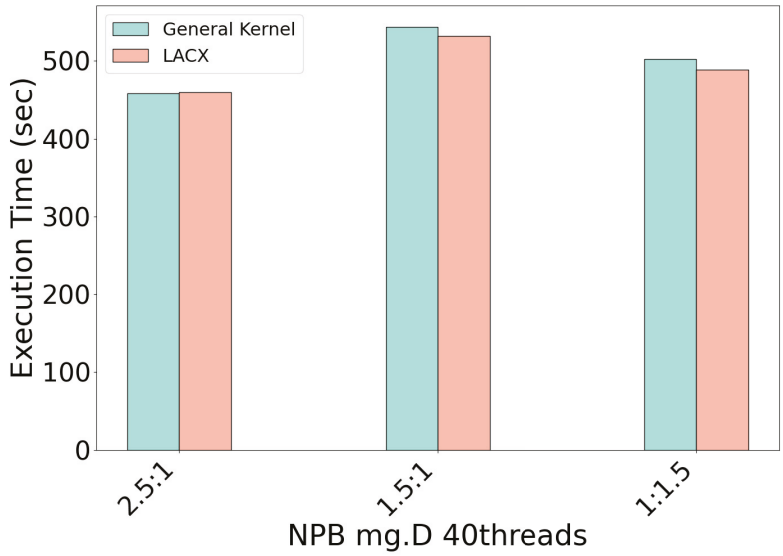
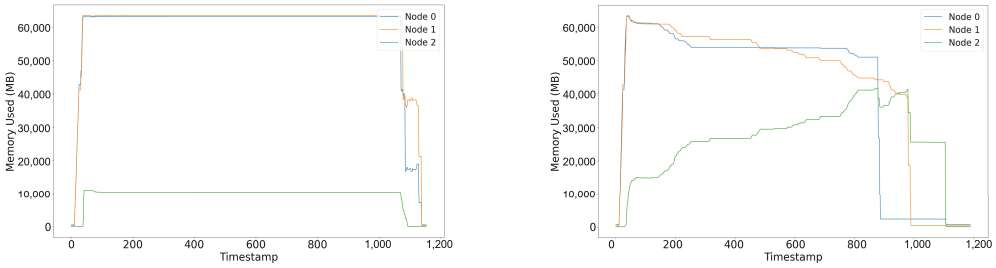


Figure 8. Execution time of NPB mg.D under memory-constrained conditions with varying CXL usage.

4.4. Memory Usage and Bandwidth Analysis

Figure 9a and 9b show the changes in memory usage per node over time while running five instances of the NPB mg.D workload under the general kernel and LACX environments,

respectively. Each instance uses nine threads in a similar configuration. In both figures, the x-axis represents the elapsed time from the start to the completion of the workloads, while the y-axis indicates the amount of memory allocated to each node.

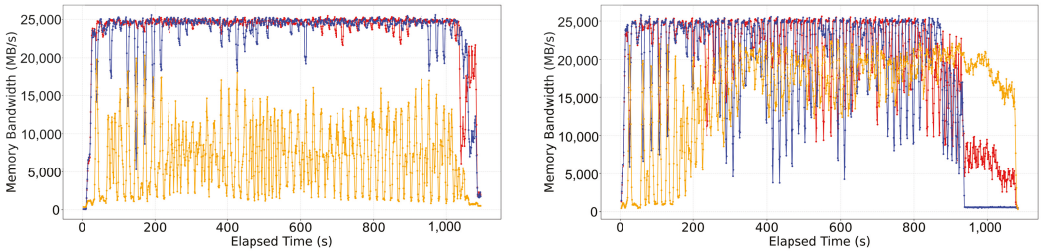


(**a**) General kernel (**b**) LACX

Figure 9. Memory usage over time for general and LACX under NPB mg.D workload with 5 concurrent operations.

In the general kernel environment, the memory usage of the CXL node begins to increase only after the memory of the high-tier nodes is fully exhausted. In contrast, under LACX, memory migration to the CXL node occurs in parallel with the usage of the high-tier nodes, resulting in a rapid and earlier increase in CXL memory utilization. Each workload is executed sequentially, and the total memory demand exceeds the capacity of the high-tier nodes, necessitating the use of the CXL node. In this scenario, LACX delays the point at which the DRAM limit is reached compared to the general kernel. By migrating shared data between tasks to the CXL node, LACX ensures that only non-shared data with high local access likelihood remain in the high-tier nodes, thereby improving the overall proportion of local accesses.

Figure 10 shows the results of measuring per-node memory bandwidth usage over time using PCM-memory during the sequential execution of five NPB mg.D workloads. The five workloads were executed sequentially, utilizing all high-tier nodes and allocating some memory pages to CXL nodes as well. In Figure 10a, which shows the general kernel environment, the bandwidth usage of the high-tier nodes (red and blue) remains consistently high, reaching nearly 25,000 MB/s for most of the execution period. In contrast, the CXL node (orange) exhibits low and unstable bandwidth throughout, indicating minimal actual utilization.



(**a**) General kernel (**b**) LACX

Figure 10. Memory bandwidth usage over time for general and LACX under NPB mg.D workload with 5 concurrent operations.

In Figure 10b, which shows the LACX environment, the bandwidth usage of the high-tier nodes also starts high, similar to the general kernel. However, starting around 250–300 s into execution, the bandwidth of the CXL node increases sharply and remains mostly between 15,000 MB/s and 20,000 MB/s, occasionally surpassing 20,000 MB/s. This demonstrates that LACX effectively redistributes memory traffic by migrating shared

data to CXL nodes, thereby improving overall bandwidth utilization efficiency across the memory hierarchy.

This trend is also quantitatively confirmed in Table 3, which summarizes the maximum, minimum, and average bandwidth observed for each node in Figure 9 under both the general kernel and LACX environments. Notably, the CXL node shows a +149.2% increase in average bandwidth usage, representing the largest improvement. Meanwhile, the high-tier nodes experience a slight decrease on average, but the total average bandwidth across the system increases by approximately 6.7%, indicating improved resource utilization of the entire memory tier.

Table 3. Bandwidth comparison between two experiments.

Node	Max Bandwidth Change (%)	Min Bandwidth Change (%)	Avg Bandwidth Change (%)
0	−3.1%	+15.3%	−9.7%
1	+6.0%	+2.8%	−18.8%
2 (CXL)	+0.6%	+10.4%	+149.2%
Total (Avg Sum)	–	–	+6.7%

5. Discussion and Future Research

Overall, our evaluations demonstrate that LACX effectively optimizes NUMA memory utilization in multi-workload environments with high-tier memory usage. In particular, migrating shared data to CXL nodes alleviates memory pressure on high-tier memory and improves overall memory bandwidth efficiency across the system.

In contrast, in environments that are not memory-intensive or are dominated by a single workload, migration costs become relatively higher, and performance degradation may occur in certain scenarios. This can be attributed to the increased frequency of migrations for shared data and the resulting rise in remote accesses. While CXL offers the advantages of expanded memory capacity and flexible tiering, its placement in low-tier memory means that it cannot match the performance of high-tier memory. Therefore, if there is available capacity in high-tier memory, it is preferable to use high-tier memory to maximize the system performance.

This study focuses on scenarios where memory usage exceeds the capacity of high-tier memory, and in such cases, CXL acts not merely as auxiliary storage but as an essential resource. This addresses the growing demand for large memory pools and expanded bandwidth in modern workloads that surpass the limits of DRAM. In such environments, the shared data migration mechanism proposed by LACX has been shown to alleviate bottlenecks in high-tier memory and improve bandwidth utilization across all memory tiers.

Building on the current implementation of LACX, future research will focus on dynamically adjusting the migration policy based on real-time system conditions to further optimize performance and reduce overhead. Two key metrics will guide this adaptation: page reference frequency and memory tier saturation. By analyzing page reference frequency, the system can identify pages migrated to CXL memory that are likely to be reused frequently in DRAM and promote them back to high-tier memory, minimizing unnecessary remote accesses and maintaining data locality, especially in workloads with low sharing or dominance by a single process. Additionally, monitoring memory saturation will enable LACX to activate migration policies only when the DRAM nodes surpass specified utilization thresholds, avoiding migration overhead during periods of low memory pressure. Conversely, when sufficient capacity is available in high-tier memory, the migration mechanisms can be automatically disabled to prevent performance degradation. This state-based activation and fallback framework

aims to ensure consistent performance improvements across diverse workload profiles and system states, making LACX adaptable to practical deployment challenges in heterogeneous memory environments. Through these adaptive strategies, LACX aims to maintain consistent performance benefits and efficient bandwidth utilization across diverse workload patterns and dynamic operating environments.

6. Conclusions

This study proposes LACX, a novel technique for efficiently identifying and managing shared data in CXL-based resource consolidation environments. By migrating shared data to CXL nodes, LACX alleviates bandwidth pressure on high-tier memory and effectively addresses the problem of remote access to shared data in NUMA environments. We developed and implemented this low-cost policy to optimize memory bandwidth utilization without introducing significant overhead. Our evaluation results show that, in multi-workload environments with high memory usage, LACX achieves up to a 14% performance improvement over the original kernel by distributing bandwidth pressure on high-tier memory and optimizing local data access. These results demonstrate that LACX is particularly effective in large-scale workloads that exceed the capacity of high-tier memory, providing meaningful bandwidth bottleneck mitigation and improving overall system efficiency.

Author Contributions: Conceptualization, H.J. (Hayong Jeong) and M.J.; methodology, H.J. (Hayong Jeong); software, H.J. (Hayong Jeong); validation, B.S. and H.J. (Heeseung Jo); formal analysis, H.J. (Hayong Jeong); investigation, H.J. (Hayong Jeong); writing—original draft preparation, H.J. (Hayong Jeong); writing—review and editing, M.J. and H.J. (Heeseung Jo); supervision, H.J. (Heeseung Jo). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Electronics and Telecommunications Research Institute(ETRI) grant funded by the Korean government [25ZS1100, Research on High-Performance Computing to overcome Limitations of AI] and by Chungbuk National University BK21 program(2025).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to institutional restrictions.

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.

References

1. Farmahini-Farahani, A.; Gurumurthi, S.; Loh, G.; Ignatowski, M. Challenges of High-Capacity DRAM Stacks and Potential Directions. In Proceedings of the MCHPC'18: Workshop on Memory Centric High Performance Computing, Dallas, TX, USA, 11 November 2018; pp. 4–13. [CrossRef]
2. Luo, H.; Shahroodi, T.; Hassan, H.; Patel, M.; Yağlıkçı, A.G.; Orosa, L. CLR-DRAM: A Low-Cost DRAM Architecture Enabling Dynamic Capacity-Latency Trade-off. In Proceedings of the 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA), Valencia, Spain, 30 May–3 June 2020; pp. 666–679. [CrossRef]
3. Lepers, B.; Quéma, V.; Fedorova, A. Thread and Memory Placement on NUMA Systems: Asymmetry Matters. In Proceedings of the USENIX Annual Technical Conference (USENIX ATC), Santa Clara, CA, USA, 8–10 July 2015; pp. 277–289. Available online: https://www.usenix.org/conference/atc15/technical-session/presentation/lepers (accessed on 12 February 2025).
4. Dashti, M.; Fedorova, A.; Funston, J.; Gaud, F.; Lachaize, R.; Lepers, B.; Quema, V.; Roth, M. Traffic Management: A Holistic Approach to Memory Placement on NUMA Systems. In Proceedings of the ASPLOS 13: Architectural Support for Programming Languages and Operating Systems, Houston, TX, USA, 16–20 March 2013; pp. 381–394.
5. Gaud, F.; Lepers, B.; Funston, J.; Dashti, M.; Fedorova, A.; Quéma, V.; Lachaize, R.; Roth, M. Challenges of Memory Management on Modern NUMA System: Optimizing NUMA Systems Applications with Carrefour. *ACM Queue* **2015**, *13*, 70. Available online: https://queue.acm.org/detail.cfm?id=2852078 (accessed on 28 February 2025). [CrossRef]

6. Gaud, F.; Lepers, B.; Decouchant, J.; Funston, J.; Fedorova, A.; Quéma, V. Large Pages May Be Harmful on NUMA Systems. In Proceedings of the USENIX ATC 14: 2014 USENIX conference on USENIX Annual Technical Conferen, Philadelphia, PA, USA, 19–20 June 2014; pp. 231–242. Available online: https://www.usenix.org/conference/atc14/technical-sessions/presentation/gaud (accessed on 5 March 2025).
7. Author, A.; Author, B.; Author, C. Shuhai: A Tool for Benchmarking High Bandwidth Memory on FPGAs. *IEEE Trans. Comput.* **2021**, *70*, 922–935.
8. Choukse, E.; Erez, M.; Alameldeen, A.R. Compresso: Pragmatic Main Memory Compression. In Proceedings of the 2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), Fukuoka, Japan, 20–24 October 2018; pp. 546–558. [CrossRef]
9. Compute Express Link (CXL). Available online: https://www.computeexpresslink.org/ (accessed on 20 March 2025).
10. Micron Technology, Inc. CXL Memory Expansion: A Closer Look on Actual Platform. Available online: https://www.micron.com/content/dam/micron/global/public/products/white-paper/cxl-memory-expansion-a-close-look-on-actual-platform.pdf (accessed on 15 April 2025).
11. Chen, Y.; Dong, X.; Zhou, Z.; Wang, Z.; Zhang, J.; Sun, G. FPGA-based Emulation and Device-Side Management for CXL-based Memory Tiering Systems. *arXiv* **2025**, arXiv:2502.19233. [CrossRef]
12. Zhong, Y.; Berger, D.S.; Waldspurger, C.; Wee, R.; Agarwal, I.; Agarwal, R.; Hady, F.; Kumar, K.; Hill, M.D.; Chowdhury, M.; et al. Managing Memory Tiers with CXL in Virtualized Environments. In Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI), Santa Clara, CA, USA, 10–12 July 2024; pp. 37–56. Available online: https://www.usenix.org/conference/osdi24/presentation/zhong-yuhong (accessed on 30 May 2025).
13. Li, H.; Berger, D.S.; Hsu, L.; Ernst, D.; Zardoshti, P.; Novakovic, S.; Shah, M.; Rajadnya, S.; Lee, S.; Agarwal, I.; et al. Pond: CXL-Based Memory Pooling Systems for Cloud Platforms. In Proceedings of the ASPLOS 23: 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, Vancouver, BC, Canada, 25–29 March 2023; pp. 574–587. [CrossRef]
14. Wang, Z.; Nalla, P.S.; Krishnan, G.; Joshi, R.V.; Cady, N.C.; Fan, D. Digital-Assisted Analog In-Memory Computing with RRAM Devices. In Proceedings of the 2023 International VLSI Symposium on Technology, Systems and Applications (VLSI-TSA/VLSI-DAT), HsinChu, Taiwan, 17–20 April 2023; pp. 1–4. [CrossRef]
15. Vignali, R.; Zurla, R.; Pasotti, M.; Rolandi, P.L.; Singh, A.; Gallo, M.L. Designing Circuits for AiMC Based on Non-Volatile Memories: A Tutorial Brief on Trade-Off and Strategies for ADCs and DACs Co-Design. *IEEE Trans. Circuits Syst. II Express Briefs* **2024**, *71*, 1650–1655. [CrossRef]
16. NUMA Balancing (AutoNUMA). Available online: https://mirrors.edge.kernel.org/pub/linux/kernel/people/andrea/autonuma/autonuma_bench-20120530.pdf (accessed on 30 June 2025).
17. Maruf, H.A.; Wang, H.; Dhanotia, A.; Weiner, J.; Agarwal, N.; Bhattacharya, P.; Petersen, C.; Chowdhury, M.; Kanaujia, S.; Chauhan, P. TPP: Transparent Page Placement for CXL-Enabled Tiered-Memory. In Proceedings of the ASPLOS 23: 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3, Vancouver, BC, Canada, 25–29 March 2023; pp. 742–755. [CrossRef]
18. Achermann, R.; Panwar, A.; Bhattacharjee, A.; Roscoe, T.; Gandhi, J. Mitosis: Transparently Self-Replicating Page-Tables for Large-Memory Machines. In Proceedings of the ASPLOS 20: Architectural Support for Programming Languages and Operating Systems, Lausanne, Switzerland, 16–20 March 2020; pp. 283–300. [CrossRef]
19. Entezari-Maleki, R.; Cho, Y.; Egger, B. Evaluation of memory performance in NUMA architectures using Stochastic Reward Nets. *J. Parallel Distrib. Comput.* **2020**, *144*, 172–188. [CrossRef]
20. Intel. What Is Optane Technology? Available online: https://www.intel.com/content/dam/www/public/us/en/documents/technology-briefs/what-is-optane-technology-brief.pdf (accessed on 30 May 2025).
21. Duraisamy, P.; Xu, W.; Hare, S.; Rajwar, R.; Culler, D.; Xu, Z.; Fan, J.; Kennelly, C.; McCloskey, B.; Mijailovic, D.; et al. Towards an Adaptable Systems Architecture for Memory Tiering at Warehouse-Scale. In Proceedings of the ASPLO 2023: 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3, Vancouver, BC, Canada, 25–29 March 2023; pp. 727–741. [CrossRef]
22. Vuppalapati, M.; Agarwal, R. Tiered Memory Management: Access Latency Is the Key! In Proceedings of the SOSP 24: ACM SIGOPS 30th Symposium on Operating Systems Principles, Austin, TX, USA, 4–6 November 2024; pp. 79–94. [CrossRef]
23. Liu, J.; Hadian, H.; Xu, H.; Li, H. Tiered Memory Management Beyond Hotness. In Proceedings of the 19th USENIX Symposium on Operating Systems Design and Implementation (OSDI), Boston, MA, USA, 7–9 July 2025; pp. 731–747.
24. Ni, Y.; Mehra, P.; Miller, E.; Litz, H. TMC: Near-Optimal Resource Allocation for Tiered-Memory Systems. In Proceedings of the SoCC 23: ACM Symposium on Cloud Computing, Santa Cruz, CA, USA, 30 October 2023–1 November 2023; pp. 376–393. [CrossRef]
25. cslabcbnu. SHM-Bench. Available online: https://github.com/cslabcbnu/SHM-bench (accessed on 12 June 2025).

26. Cooper, B.F.; Silberstein, A.; Tam, E.; Ramakrishnan, R.; Sears, R. Benchmarking Cloud Serving Systems with YCSB. In Proceedings of the SoCC 10: 1st ACM Symposium on Cloud Computing in Conjunction with SIGMOD 2010, Indianapolis, IN, USA, 10–11 June 2010; pp. 143–154. [CrossRef]
27. NASA Advanced Supercomputing Division. NAS Parallel Benchmarks. Available online: https://www.nas.nasa.gov/software/npb.html (accessed on 12 June 2025).

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 Editor. 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 Editor 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.





Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-8032-4