electronics

Special Issue Reprint

# Computational Intelligence and Machine Learning

## Models and Applications: 2nd Edition

Edited by
Grzegorz Dudek and Arkadiusz Tomczyk

mdpi.com/journal/electronics

MDPI

# Computational Intelligence and Machine Learning: Models and Applications: 2nd Edition

# Computational Intelligence and Machine Learning: Models and Applications: 2nd Edition

Guest Editors

**Grzegorz Dudek**
**Arkadiusz Tomczyk**

MDPI

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

*Guest Editors*

Grzegorz Dudek
Department of Automatic Control, Electrical Engineering and Optoelectronics
Częstochowa University of Technology
Częstochowa
Poland

Arkadiusz Tomczyk
Institute of Information Technology
Lodz University of Technology
Lodz
Poland

*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/GF6RK2352Q.

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 978-3-7258-6902-2 (Hbk)**
**ISBN 978-3-7258-6903-9 (PDF)**
**https://doi.org/10.3390/books978-3-7258-6903-9**

# Contents

# About the Editors

**Grzegorz Dudek**

Grzegorz Dudek received his PhD in Electrical Engineering from Czestochowa University of Technology, Poland, in 2003 and completed his habilitation in Computer Science at Lodz University of Technology, Poland, in 2013. In 2023, he was appointed as a full professor. His research primarily focuses on machine learning and artificial intelligence, with a strong emphasis on their applications in classification, regression, forecasting, and optimization. He has authored and co-authored 4 books and over 140 scientific papers in these areas.

**Arkadiusz Tomczyk**

Arkadiusz Tomczyk is employed as an Assistant Professor at the Institute of Information Technology of the Lodz University of Technology in Poland. His initial research expertise encompassed image processing and analysis, with particular emphasis on active contour methods, as well as pattern recognition and machine learning techniques. Between 2013 and 2017, he served as the Principal Investigator of a research project focused on integrating active contour approaches with structural representations of image content, supported by the National Science Centre. He was and currently is involved in several projects funded by the National Centre for Research and Development, which explicitly combine the outcomes of fundamental research with practical applications and industrial implementation. His present research interests focus on advanced machine learning methods, including convolutional neural networks, transformer-based architectures, and graph neural networks, applied to the analysis of images, time series, and graph-structured data. He is the author or co-author of approximately 50 scientific publications, including journal articles, book chapters, and conference contributions.

# Preface

This Reprint presents a curated collection of research articles originally published in the Special Issue "Computational Intelligence and Machine Learning: Models and Applications". It brings together recent advances in computational intelligence and machine learning that address both methodological challenges and practical applications across diverse domains.

The scope of this Reprint spans modern learning paradigms, including distributed, semi-supervised, and weakly supervised approaches, as well as application-driven studies in areas such as agriculture, cybersecurity, document analysis, sports analytics, and human–AI interaction. The primary aim of this Reprint is to highlight how contemporary machine learning models are being adapted to operate effectively under real-world constraints, such as data uncertainty, decentralization, and the growing need for interpretability and trust.

The motivation for compiling this Reprint arises from the rapid expansion of machine learning beyond controlled experimental settings into complex socio-technical environments. By presenting a coherent set of contributions that combine theoretical insights with domain-specific applications, this Reprint seeks to provide readers with a structured overview of current trends, challenges, and opportunities in computational intelligence.

This Reprint is addressed to researchers, practitioners, and graduate students in machine learning, data science, and artificial intelligence, as well as to professionals interested in the deployment of intelligent systems in applied contexts. It is intended to serve both as a reference to recent developments and as a source of inspiration for future research and innovation in the field.

**Grzegorz Dudek and Arkadiusz Tomczyk**

*Guest Editors*

*electronics*

MDPI

*Editorial*

# Advances in Computational Intelligence and Machine Learning Models and Applications

**Grzegorz Dudek [1,2,3,*] and Arkadiusz Tomczyk [4]**

[1] Faculty of Electrical Engineering, Czestochowa University of Technology, Al. AK 17, 42-201 Czestochowa, Poland
[2] Faculty of Mathematics and Computer Science, University of Łódź, ul. Banacha 22, 90-238 Łódź, Poland
[3] CAMINO—Centre for Data Analysis, Modelling and Computational Sciences, University of Łódź, ul. Narutowicza 68, 90-136 Łódź, Poland
[4] Institute of Information Technology, Faculty of Technical Physics, Information Technology and Applied Mathematics, Łódź University of Technology, al. Politechniki 8, 93-590 Łódź, Poland; arkadiusz.tomczyk@p.lodz.pl
* Correspondence: grzegorz.dudek@pcz.pl

## 1. Introduction

Machine learning (ML) and artificial intelligence (AI) have entered a phase of accelerated evolution, reshaping the computational landscape and influencing an ever-growing spectrum of scientific and industrial activities. What once were specialized tools designed for narrow analytical tasks have now become integral components of complex pipelines that support decision-making, automate perception, enhance prediction, and facilitate interaction between humans and digital systems. This expansion has been driven not only by advances in algorithmic design but also by the increasing availability of heterogeneous data, the maturation of distributed computing ecosystems, and a heightened societal expectation for systems that are adaptive, transparent, and contextually aware.

As ML systems are deployed in environments that diverge markedly from controlled laboratory conditions, researchers face new methodological tensions. Data encountered in real-world settings may be incomplete, noisy, weakly labeled, fragmented across devices or institutions, or evolving in ways that challenge static modeling assumptions. These constraints have motivated the exploration of learning paradigms capable of functioning under limited supervision, handling uncertainty, and respecting privacy requirements. At the same time, domain specialists in areas such as agriculture, sports analytics, cyberdefense, and document intelligence are pushing for models that incorporate domain structure rather than relying on generic architectures. Complementing these technical imperatives is a parallel line of inquiry that addresses how people perceive, adopt, and interact with AI-driven tools, raising questions about interpretability, trust, accountability, and user behavior.

These developments converge around three major trajectories that increasingly define contemporary ML research: (1) distributed and semi-supervised learning frameworks, which aim to enable robustness and scalability when labels are scarce, data are decentralized, or uncertainty is inherent to the measurement process; (2) specialized, domain-responsive ML solutions that integrate contextual knowledge into model parameterization and evaluation, allowing AI to operate effectively in complex applied settings; and (3) human-centered behavioral modeling and technology acceptance studies, which investigate how individuals and organizations integrate intelligent systems into their workflows, and how model transparency, perceived utility, and personal traits influence adoption.

The ten articles included in this Special Issue exemplify these intertwined directions. They collectively expand the theoretical and practical boundaries of modern ML by introducing novel algorithmic strategies, validating them on demanding real-world datasets, and examining their implications for human-technology interaction. Their contributions resonate with several major currents in global research, including federated and distributed learning [1], multimodal transformer architectures [2], predictive models of user acceptance and behavior [3], and the development of trustworthy, explainable, and ethically aligned AI systems [4].

To guide readers through this multidisciplinary landscape, the remainder of this editorial is organized as follows. Section 2 provides a structured synthesis of the ten contributions, grouping them into three thematic categories that reflect the aforementioned research trajectories. Section 3 discusses conceptual themes that cut across these categories, highlighting methodological synergies and shared challenges. Section 4 concludes with reflections on emerging research opportunities and the broader significance of the advances presented in this Special Issue.

## 2. Summary of the Contributions

### *2.1. Distributed, Semi-Supervised and Weakly-Supervised Learning*

The first group of contributions addresses a core challenge in contemporary ML: how to learn reliable models when data are distributed across networks, labels are incomplete or ambiguous, and feature vectors may be both uncertain and partially missing. Rather than assuming centralized data access and fully supervised training, these works embrace the reality of fragmented, weakly supervised, and noisy environments. They do so by combining ideas from distributed optimization, semi-supervised learning, partial-label modeling, and multi-dimensional classification, thereby contributing to a growing body of research on federated and decentralized learning under imperfect supervision [1,5–7].

The paper "Distributed Semi-Supervised Multi-Dimensional Uncertain Data Classification over Networks" by Xu and Chen focuses on multi-dimensional classification in distributed networks where each node observes uncertain data and only a subset of instances have reliable labels. The authors consider a setting in which local nodes construct multi-dimensional classifiers based on their own data while exchanging limited information with neighboring nodes in a communication graph. The proposed method explicitly models the uncertainty of input data and couples this with a semi-supervised learning strategy that can exploit unlabeled observations to regularize the decision boundaries. By embedding this into a consensus-based framework over the network, the algorithm allows each node to benefit from global structure without centralizing the raw data, which is crucial in privacy-sensitive or bandwidth-limited scenarios [8]. Beyond incremental improvements in accuracy, the main innovation lies in the joint treatment of uncertainty, multi-dimensional outputs, and distributed semi-supervised optimization, providing a template for future work on robust learning in sensor networks and edge-intelligence applications.

The second contribution in this group, authored by Lee and Han, takes the perspective of cybersecurity and mobile platforms. Here, the data distribution is not merely decentralized but also characterized by rapidly evolving threats and limited labeled examples. SMAD (Semi-Supervised Android Malware Detection) tackles Android malware detection by converting application packages into image-like representations and using a segmentation-oriented backbone to extract pixel-level, multi-scale features from these Android Package Kit (APK) images. On top of this representation, the authors introduce a dual-branch semi-supervised objective, in which two parallel prediction branches are encouraged to remain consistent on unlabeled samples. This consistency regularization enables the method to leverage large volumes of unlabeled telemetry data and to remain


https://doi.org/10.3390/electronics15010196

effective when the distribution of malware families drifts over time. In contrast to traditional signature-based or purely supervised ML systems [9], SMAD illustrates how modern semi-supervised techniques—rooted in consistency and perturbation-based learning—can be transplanted into the malware domain, bringing ideas from image-based SSL and self-training into security analytics. The fine-grained spatial modeling of APK imagery is particularly innovative, as it allows the detector to capture subtle structural cues that are difficult to encode with handcrafted features.

The remaining two papers in this section focus on partial labels and missing data, further relaxing the assumption of clean supervision. In "Distributed Partial Label Learning for Missing Data Classification", Xu and Chen study scenarios where each training instance is associated with a set of candidate labels (only one of which is correct), and where feature vectors are themselves incomplete. They propose a distributed partial-label missing-data classification (dPMDC) algorithm that combines generative and discriminative ideas into a unified framework. On the generative side, they design a probabilistic, information-theoretic imputation scheme that exploits the weak supervisory signal embedded in ambiguous labels to infer the missing features. On the discriminative side, once the features are imputed, a classifier is trained using random feature mappings of the $\chi^2$ kernel, enabling nonlinear decision boundaries at modest computational cost. The entire procedure is implemented in a distributed manner so that nodes collaboratively refine imputations and classifiers without pooling data centrally. This work extends classical partial-label learning [6] by coupling it explicitly with missing-feature imputation and by pushing the computation to the network edge, where data are produced.

In "Distributed Partial Label Multi-Dimensional Classification via Label Space Decomposition", the same authors further generalize partial-label learning to the multi-dimensional case. Here, each instance is associated with multiple heterogeneous label variables, and for each dimension only a subset of candidate labels is known. The proposed dPL-MDC algorithm performs a one-vs.-one decomposition of the original multi-dimensional output space, effectively transforming the problem into a collection of distributed partial multi-label learning tasks. This label-space decomposition serves two purposes: it reduces the complexity of directly modeling interactions in a high-dimensional label space, and it enables parallelized computation across decomposed subproblems, which is well suited to distributed environments. The approach connects naturally with broader research on multi-label and multi-dimensional classification [7] and complements recent advances in partial-label learning that address issues such as out-of-distribution candidate labels and long-tailed label distributions [10]. By extending these ideas into a decentralized framework, dPL-MDC helps bridge the gap between sophisticated label modeling and the constraints of networked data acquisition.

Taken together, the four articles form a coherent line of research on learning in distributed, weakly supervised, and imperfect environments. All of them leverage local computation and limited communication instead of assuming fully centralized access, aligning with contemporary developments in federated and edge learning [1,5]. At the same time, they explore complementary dimensions of supervision: semi-supervised consistency in SMAD for security telemetry; exploitation of unlabeled and uncertain instances in multi-dimensional classification; and partial labels combined with missing-feature imputation or label-space decomposition. Their novelty is not only algorithmic but also conceptual: they illustrate how ideas from semi-supervised learning, probabilistic modeling, and structured prediction can be systematically adapted to realistic deployment settings where uncertainty, ambiguity, and decentralization are the norm rather than the exception. As such, this cluster of works provides a strong methodological foundation for the subsequent, more application-oriented contributions in this Special Issue.


https://doi.org/10.3390/electronics15010196

### *2.2. Specialized Domain Applications with Machine Learning*

The second thematic group highlights how ML methods can be adapted, extended, and refined to meet the unique structural and operational challenges of real-world application domains. These three papers illustrate the broader movement toward domain-aware AI, in which models are not simply transferred from generic benchmarks but are redesigned to exploit domain structure—be it multimodal agricultural data, complex performance dynamics in sports, or geometric distortions in document images. This aligns with a growing body of work emphasizing task-specific inductive biases and multimodal fusion as central to achieving state-of-the-art results in applied ML [11–15].

The first contribution in this group, by Jácome Galarza et al., presents an innovative application of transformer architectures to agricultural forecasting. While crop yield prediction has traditionally relied on statistical agronomic models or convolutional networks applied to remote sensing data, AgriTransformer integrates multiple heterogeneous data streams—including satellite imagery and tabular data—within a unified transformer-based framework. By leveraging attention mechanisms, the model learns dynamic interdependencies across modalities. This design reflects the broader trend toward multimodal learning in Earth observation [16] and is particularly significant given the scarcity, noise, and spatial variability characteristic of agricultural datasets. The authors demonstrate that attention-driven fusion not only improves predictive accuracy but also, through the use of explainability techniques, enables the identification of which components of different modalities contribute most to yield estimates. In doing so, the work provides a path forward for decision support in sustainable agriculture, food security, and climate-resilient farming.

The paper by Chandru et al. turns to the rapidly evolving domain of sports analytics, where strategic decisions increasingly rely on quantitative assessment of player performance. Building on the expanding literature in sports data science [17], the authors design a ML pipeline that integrates performance metrics, game statistics, and contextual features to estimate players' contributions and forecast team strategy outcomes. Their approach explores, among other techniques, ensemble learning and the optimization of model selection to improve accuracy while accommodating nonlinear interactions among variables describing players' statistics. What distinguishes this contribution is the explicit linkage between predictive modeling and actionable strategy: the authors go beyond prediction to provide tactical insights for coaches and analysts. This demonstrates a broader pattern in domain-focused ML—shifting from merely descriptive analytics to prescriptive models that guide decision-making. In an era where professional sports are increasingly data-driven, the article offers a well-structured example of how ML can enhance competitive advantage through evidence-based strategic planning.

In the paper by Cai et al., the authors address a fundamental challenge in document intelligence: reliably detecting and geometrically correcting tables within scanned or photographed documents. Traditional OCR systems and naive detection architectures struggle with distortions, rotations, and variable table layouts, motivating recent research in document AI that leverages deep learning and geometric reasoning [18,19]. LocRecNet introduces a two-stage synergistic pipeline that first employs localization modules to detect characteristic points and then uses rectification module to compensate for skew, perspective distortion, and irregular cell geometry. The contribution is notable for its integration of detection and rectification rather than treating them as isolated tasks. This design mirrors trends in computer vision—especially in structured document understanding—where joint optimization leads to more stable and generalizable performance. By demonstrating strong results across diverse document types, the work presents a practical and scalable solution for applications such as automated data extraction, digital archiving, and financial or legal document processing.


https://doi.org/10.3390/electronics15010196

Viewed collectively, the three articles exemplify how ML methods can be specialized to leverage domain structure, multimodality, and task-specific constraints. Each contribution demonstrates a distinct innovation: attention-based multimodal fusion in agriculture, predictive strategy analytics in sports, and synergistic geometric reasoning for document understanding. At the same time, they share methodological themes with broader ML research, such as the importance of representation learning, the integration of learning with domain knowledge, and the increasing relevance of task-aware architectures. These works underscore the maturation of applied ML: moving from generic models to tailored, interpretable, and operationally meaningful intelligent systems.

### *2.3. Human Factors and Behavioral Prediction*

The third group of papers underscores a central insight of contemporary AI research: the success of intelligent systems ultimately depends not only on algorithmic performance but also on how humans perceive, interact with, and integrate these systems into their workflows. As AI becomes embedded in identity verification, customer service, and managerial decision-making, understanding user behavior, trust, and acceptance becomes essential. This evolution aligns with long-standing research in information systems and human-computer interaction [3,20,21], as well as with the growing demand for transparent and trustworthy AI [4,22].

The first article in this category, by Halász et al., tackles the crucial problem of biometric authentication under open-set conditions. Unlike closed-set classification, open-set recognition must identify whether a user is known or entirely novel, requiring models that can generalize beyond the classes observed during training. The authors adapt an existing open-set framework to time-series biometrics, demonstrating strong performance in identifying impostors, even when variability in user behavior is substantial. Their work ties into broader concerns about secure and explainable biometric systems [23,24], illustrating how open-set approaches can mitigate risks associated with spoofing and distributional shift. Importantly, the article shows that incorporating time-series dynamics and uncertainty modeling can help bridge the gap between theoretical open-set constructs and the operational realities of biometric authentication.

In the work by Gené-Albesa and de Andrés-Sánchez, the focus shifts toward human-AI interaction in service-oriented contexts. The authors examine how customers perceive AI-driven chatbots in insurance policyholder support—a domain where user trust, perceived usefulness, and clarity of communication directly influence adoption. By combining explainable AI (XAI) with Importance-Performance Map Analysis (IPMA), the study identifies which features most strongly affect acceptance and offers interpretable insights for system designers. This methodology reflects the ongoing shift toward AI systems whose outputs and recommendations must be understandable to end-users [4,25]. The integration of XAI with behavioral modeling is a notable innovation, illustrating how explainability can serve not only regulatory or ethical purposes but also practical design objectives.

Finally, the paper by Cuc et al. explores how individual characteristics influence the intention to adopt AI tools in professional decision-making. Managerial accounting, increasingly shaped by automation, predictive analytics, and decision-support systems, provides a compelling context for studying technology adoption. The authors use decision-tree regression to uncover nonlinear relationships between AI knowledge, personality factors, and usage intention—patterns that traditional linear acceptance models often fail to capture. Their findings resonate with the broader literature on technology adoption, particularly the emphasis on cognitive, affective, and contextual determinants of behavior [3]. By leveraging ML not only as a subject of study but also as an analytical tool, the paper demonstrates how predictive modeling can augment theory-building in behavioral research.


https://doi.org/10.3390/electronics15010196

Taken together, the three articles highlight a crucial dimension of modern AI systems: their embedding in socio-technical ecosystems. From biometric authentication to service chatbots to managerial decision support, human behavior emerges as both a constraint and a driver of system effectiveness. Across these studies, themes such as trust, transparency, and user diversity recur, aligning with ongoing global discussions about responsible AI. The emphasis on interpretability, open-set robustness, and behavior-aware modeling illustrates an important shift in AI research—from a technology-centered paradigm to a human-centered one – mirroring contemporary calls for systems that are trustworthy, explainable, and psychologically attuned to their users.

## 3. Cross-Domain Themes and Conceptual Integration

Although the ten contributions span diverse application areas and methodological perspectives, several overarching themes reveal a deeper conceptual alignment across the Special Issue. A central unifying motif is the movement toward learning under real-world constraints, where data are incomplete, weakly supervised, uncertain, or distributed across heterogeneous computational environments. This shift reflects a growing recognition that classical assumptions of IID data, centralized processing, and fully supervised labels rarely hold in operational contexts. As illustrated by recent surveys on federated and distributed learning [1,5], the increasing prominence of edge devices, privacy regulations, and decentralized infrastructures creates strong demand for algorithms capable of collaboration without centralized data aggregation. The contributions in distributed semi-supervised and partial-label learning directly speak to this global trend, offering models that balance local autonomy with global consistency.

Another cross-domain theme is the progressive specialization of ML architectures. Instead of relying on universal modeling paradigms, researchers increasingly adapt architectures to the structural properties of specific domains, whether through multimodal attention mechanisms in agriculture, geometric rectification in document analysis, or fine-grained spatial representations in cybersecurity. This reflects the broader shift in ML toward domain-aware inductive biases and task-specific architectures, a movement emphasized in contemporary discussions of transformer-based models [2]. The domain studies included in this Special Issue illustrate how such specialization enhances interpretability, robustness, and operational significance.

A third integrative thread emerges from the human-centered dimension of ML. As AI systems become embedded in professional workflows, consumer-facing services, and identity verification processes, understanding user perception, behavioral variability, and trust becomes as important as optimizing algorithmic accuracy. Research on technology acceptance, explainability, and decision support [3,4,26] highlights that the success of AI initiatives depends on an interplay between model capabilities, user characteristics, and organizational practices. The studies on chatbot acceptance, biometric identification under open-set conditions, and AI adoption in managerial accounting underscore that responsible AI development must integrate psychological, sociotechnical, and ethical considerations alongside computational ones.

Finally, a broader methodological synthesis can be observed across all contributions: the convergence toward data-centric and context-sensitive ML. This orientation prioritizes understanding the structure, quality, and limitations of data, whether through robust strategies for missing labels, multimodal fusion techniques, or behavioral modeling frameworks. Recent calls within the ML community [27] emphasize that improvements in data quality, representation, and alignment often yield greater gains than increasing model complexity alone. The collective work presented in this Special Issue exemplifies this philosophy


https://doi.org/10.3390/electronics15010196

by pairing methodological advances with a realistic appraisal of domain constraints and user-centered requirements.

Taken together, these cross-domain themes reflect an evolving ML landscape—one in which scalability, domain integration, and human-centered design are no longer optional enhancements but essential elements of modern intelligent systems. The convergence of these perspectives suggests a future in which ML research becomes increasingly interdisciplinary, context-aware, and attuned to both technical performance and societal impact.

## 4. Conclusions

The papers in this Special Issue collectively advance theoretical frameworks, methodological innovations, and practical applications across distributed learning, domain-specific AI, and human-AI interaction. Together, they demonstrate the richness and interdisciplinarity of modern machine intelligence research. We hope that the presented contributions inspire further exploration in distributed weakly supervised learning, transformer-driven multimodal modeling, human-centered predictive analytics, and trustworthy AI deployment.

**Funding:** This research received no external funding.

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

## List of Contributions

1. Xu, Z.; Chen, S. Distributed Semi-Supervised Multi-Dimensional Uncertain Data Classification over Networks. *Electronics* **2025**, *14*, 4634. https://doi.org/10.3390/electronics14234634.
2. Lee, S.; Han, S. SMAD: Semi-Supervised Android Malware Detection via Consistency on Fine-Grained Spatial Representations. *Electronics* **2025**, *14*, 4246. https://doi.org/10.3390/electronics14214246.
3. Xu, Z.; Chen, Z. Distributed Partial Label Learning for Missing Data Classification. *Electronics* **2025**, *14*, 1770. https://doi.org/10.3390/electronics14091770.
4. Xu, Z.; Chen, S. Distributed Partial Label Multi-Dimensional Classification via Label Space Decomposition. *Electronics* **2025**, *14*, 2623. https://doi.org/10.3390/electronics14132623.
5. Jácome Galarza, L.; Realpe, M.; Viñán-Ludeña, M.; Calderón, M.; Jaramillo, S. AgriTransformer: A Transformer-Based Model with Attention Mechanisms for Enhanced Multimodal Crop Yield Prediction. *Electronics* **2025**, *14*, 2466. https://doi.org/10.3390/electronics14122466.
6. Chandru, R.; Kaushik, A.; Jaiswal, P. Enhancing Basketball Team Strategies Through Predictive Analytics of Player Performance. *Electronics* **2025**, *14*, 2177. https://doi.org/10.3390/electronics14112177.
7. Cai, Z.; Feng, J.; Hou, Z.; Zhang, H.; Ma, H. LocRecNet: A Synergistic Framework for Table Localization and Rectification. *Electronics* **2025**, *14*, 1920. https://doi.org/10.3390/electronics14101920.
8. Halász, A.; Al Hemeary, N.; Daubner, L.; Juhász, J.; Zsedrovits, T.; Tornai, K. Adapting a Previously Proposed Open-Set Recognition Method for Time-Series Data: A Biometric User Identification Case Study. *Electronics* **2025**, *14*, 3983. https://doi.org/10.3390/electronics14203983.
9. Gené-Albesa, J.; de Andrés-Sánchez, J. Assessing Chatbot Acceptance in Policyholder's Assistance Through the Integration of Explainable Machine Learning and Importance–Performance Map Analysis. *Electronics* **2025**, *14*, 3266. https://doi.org/10.3390/electronics14163266.
10. Cuc, L.; Rad, D.; Cilan, T.; Gomoi, B.; Nicolaescu, C.; Almași, R.; Dzitac, S.; Isac, F.; Pandelica, I. From AI Knowledge to AI Usage Intention in the Managerial Accounting Profession and the Role of Personality Traits—A Decision Tree Regression Approach. *Electronics* **2025**, *14*, 1107. https://doi.org/10.3390/electronics14061107.

## References


1. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]
2. Xu, P.; Zhu, X.; Clifton, D.A. Multimodal Learning with Transformers: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 12113–12132. [CrossRef] [PubMed]
3. Venkatesh, V.; Morris, M.G.; Davis, G.B.; Davis, F.D. User Acceptance of Information Technology: Toward a Unified View. *MIS Q.* **2003**, *27*, 425–478. [CrossRef]
4. Doshi-Velez, F.; Kim, B. Towards a Rigorous Science of Interpretable Machine Learning. *arXiv* **2017**, arXiv:1702.08608. [CrossRef]
5. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and Open Problems in Federated Learning. *Found. Trends® Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
6. Tian, Y.; Yu, X.; Fu, S. Partial label learning: Taxonomy, analysis and outlook. *Neural Netw.* **2023**, *161*, 708–734. [CrossRef]
7. Tsoumakas, G.; Katakis, I.M. Multi-Label Classification: An Overview. *Int. J. Data Warehous. Min.* **2007**, *3*, 1–13. [CrossRef]
8. Mendez, J.; Bierzynski, K.; Cuéllar, M.P.; Morales, D.P. Edge Intelligence: Concepts, Architectures, Applications, and Future Directions. *ACM Trans. Embed. Comput. Syst.* **2022**, *21*, 1–41. [CrossRef]
9. Zadeh Nojoo Kambar, M.E.; Esmaeilzadeh, A.; Kim, Y.; Taghva, K. A Survey on Mobile Malware Detection Methods using Machine Learning. In Proceedings of the 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 26–29 January 2022; pp. 215–221. [CrossRef]
10. Huang, J.; Cheung, Y.M. PLOOD: Partial Label Learning with Out-of-distribution Objects. *arXiv* **2024**, arXiv:2403.06681v4.
11. Aijaz, N.; Lan, H.; Raza, T.; Yaqub, M.; Iqbal, R.; Pathan, M.S. Artificial intelligence in agriculture: Advancing crop productivity and sustainability. *J. Agric. Food Res.* **2025**, *20*, 101762. [CrossRef]
12. Mehdipour, S.; Mirroshandel, S.A.; Tabatabaei, S.A. Vision transformers in precision agriculture: A comprehensive survey. *Intell. Syst. Appl.* **2026**, *29*, 200617. [CrossRef]
13. Xu, T.; Baghaei, S. Reshaping the future of sports with artificial intelligence: Challenges and opportunities in performance enhancement, fan engagement, and strategic decision-making. *Eng. Appl. Artif. Intell.* **2025**, *142*, 109912. [CrossRef]
14. Pietraszewski, P.; Terbalyan, A.; Roczniok, R.; Maszczyk, A.; Ornowski, K.; Manilewska, D.; Kuliś, S.; Zając, A.; Gołaś, A. The Role of Artificial Intelligence in Sports Analytics: A Systematic Review and Meta-Analysis of Performance Trends. *Appl. Sci.* **2025**, *15*, 7254. [CrossRef]
15. Li, X.; Dong, J.; Wong, R. From Surface to Semantics: Semantic Structure Parsing for Table-Centric Document Analysis. *arXiv* **2025**, arXiv:2508.10311. [CrossRef]
16. Mena, F.; Ienco, D.; Dantas, C.F.; Interdonato, R.; Dengel, A. Multi-modal co-learning for Earth observation: Enhancing single-modality models via modality collaboration. *Mach. Learn.* **2025**, *114*, 279. [CrossRef]
17. Sarlis, V.; Gerakas, D.; Tjortjis, C. A Data Science and Sports Analytics Approach to Decode Clutch Dynamics in the Last Minutes of NBA Games. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 2074–2095. [CrossRef]
18. Li, D.L.; Lee, S.K.; Liu, Y.T. Printed document layout analysis and optical character recognition system based on deep learning. *Sci. Rep.* **2025**, *15*, 23761. [CrossRef]
19. Rajan, R.; Devasena, M.G. Deep learning based optimization model for document layout and text recognition. *Ain Shams Eng. J.* **2025**, *16*, 103587. [CrossRef]
20. Zhang, P.; Li, N. An assessment of human–computer interaction research in management information systems: Topics and methods. *Comput. Hum. Behav.* **2004**, *20*, 125–147. [CrossRef]
21. Gupta, R.D.; Rahman, A.; Showmick, M.I.H.; Rahat, M.Y.; Hossen, M.J. Exploring the Convergence of HCI and Evolving Technologies in Information Systems. *arXiv* **2025**, arXiv:2506.08549. [CrossRef]
22. Rudin, C. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nat. Mach. Intell.* **2019**, *1*, 206–215. [CrossRef] [PubMed]
23. Ayeswarya, S.; Singh, K.J. A Comprehensive Review on Secure Biometric-Based Continuous Authentication and User Profiling. *IEEE Access* **2024**, *12*, 82996–83021. [CrossRef]
24. Tucci, C.; Della Greca, A.; Tortora, G.; Francese, R. Explainable biometrics: A systematic literature review. *J. Ambient. Intell. Humaniz. Comput.* **2024** . [CrossRef]
25. Díaz-Rodríguez, N.; Del Ser, J.; Coeckelbergh, M.; López de Prado, M.; Herrera-Viedma, E.; Herrera, F. Connecting the dots in trustworthy Artificial Intelligence: From AI principles, ethics, and key requirements to responsible AI systems and regulation. *Inf. Fusion* **2023**, *99*, 101896. [CrossRef]



https://doi.org/10.3390/electronics15010196

26. Haque, A.B.; Islam, A.N.; Mikalef, P. Explainable Artificial Intelligence (XAI) from a user perspective: A synthesis of prior literature and problematizing avenues for future research. *Technol. Forecast. Soc. Change* **2023**, *186*, 122120. [CrossRef]
27. Sambasivan, N.; Kapania, S.; Highfill, H.; Akrong, D.; Paritosh, P.K.; Aroyo, L. "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, Yokohama, Japan, 8–13 May 2021.




https://doi.org/10.3390/electronics15010196

*electronics*

MDPI

*Article*

# Distributed Semi-Supervised Multi-Dimensional Uncertain Data Classification over Networks

**Zhen Xu [1,*,†] and Sicong Chen [2,†]**

[1] College of Computer Science and Artificial Intelligence Engineering, Wenzhou University, Wenzhou 325006, China
[2] Kasco Signal Co., Ltd., Shanghai 200072, China
* Correspondence: 20200588@wzu.edu.cn
† These authors contributed equally to this work.

**Abstract:** Distributed multi-dimensional classification, where multiple nodes over a network induce a multi-dimensional classifier based on their own local data and a little information exchanged from neighbors, has received extensive attention in the academic community recently. Nevertheless, we observe that the classical distributed multi-dimensional classification formulation requires all training data to have definite feature attributes and complete labels. However, in real-world scenarios, due to measurement errors in distributed networks, the collected data samples consist of attributes with uncertainty. Additionally, a substantial proportion of multi-dimensional data faces challenges in label acquisition. Therefore, the key to achieving satisfactory performance in such a case is designing an effective method to model the input uncertainty and exploit weakly supervised information from the training data. Considering this, in this paper, we design a novel misclassification loss function that extracts effective information from uncertain data by treating it as the integral of misclassification loss over the potential data distribution. Additionally, we propose a new explicit feature mapping for constructing a nonlinear discriminant function. Based on this, we further put forward a novel manifold regularization term to recover multi-dimensional labels and simplify the original objective function to enable it to be optimized. By leveraging the gradient descent method, we optimize the simplified decentralized cost function and obtain the global optimal solution. We evaluate the performance of the proposed distributed semi-supervised multi-dimensional uncertain data classification algorithm, namely the dSMUDC algorithm, on several real datasets. The results of our experiments indicate that, in terms of all metrics, our proposed algorithm outperforms existing approaches to a significant extent.

**Keywords:** semi-supervised learning; multi-dimensional classification; uncertain data classification; explicit feature map

## 1. Introduction

Currently, distributed learning (DL) stands as a widely used learning framework. In this framework, a specific global task can be executed across a group of individual nodes, with only a small amount of key information shared among neighbors [1–9]. Owing to its outstanding learning performance and the robustness of distributed networks to node/link failures, numerous DL approaches have been devised and widely applied across diverse fields in recent years. These applications span critical domains like intelligent computation [3,6], the Internet of Things (IoT) [7,8], and 6G-enabled intelligent transportation systems [5,9]. Recent studies have explored the performance enhancement of distributed

computing in real-world scenarios. For instance, in [6], parallel computing is leveraged to improve the computational efficiency of convolutional neural networks. In [5], device-side data processing and resource allocation are optimized via DL frameworks to boost IoT system performance. Additional research has focused on DL frameworks tailored for 6G intelligent transportation systems [9], as well as the validation of real-world implementations in 6G IoT scenarios [7]. Collectively, these works lay a solid foundation for the practical deployment of DL across diverse critical domains.

Nevertheless, conventional DL algorithms typically target binary or multi-class classification tasks and make the assumption that data labels are of a single dimension [1,5]. In practical application scenarios, however, a substantial volume of training data can often be categorized across multiple dimensions [10–15]. An illustrative example is the census, where survey agencies may classify individuals according to multiple dimensions, including gender, age, occupation, education level, etc.

To address such kinds of data, a series of multi-dimensional classification methods have been proposed [10–17]. Generally speaking, the majority of existing algorithms focus on exploiting the correlations among multi-dimensional labels in an explicit manner [10–17]. Currently, typical methods for modeling label correlations include merging multi-dimensional labels into new labels [10], exploiting explicit label chain order to explore heterogeneous class spaces [13], and learning label-pair dependencies in decomposed multi-dimensional spaces [14]. Recently, a few distributed multi-dimensional classification (MDC) algorithms have been successively proposed [16,17]. For instance, in [16], the authors proposed to decompose multi-dimensional heterogeneous spaces into multiple homogeneous class spaces via one-vs.-one decomposition and employ various techniques to learn the high-order label correlations among decomposed labels. In [17], a distributed MDC is proposed to map the multi-dimensional label space into several low-dimensional homogeneous subspaces based on subspace learning and learn the second-order correlations among embedded labels. Existing centralized/distributed MDC methods have made considerable progress, but they still face certain potential limitations.

First, as multi-dimensional data, training data is characterized by labels across multiple distinct dimensions. However, in many practical applications, due to the difficulty and high cost of label acquisition, a large volume of unlabeled data is easily accessible, while accurately annotating such data remains challenging. In most cases, we can only obtain a small amount of labeled data alongside a large amount of unlabeled data. In such scenarios, most existing algorithms belong to supervised learning algorithms [10–15], making it difficult for them to achieve satisfactory performance. Semi-supervised learning algorithms, which are capable of handling both labeled and unlabeled data simultaneously, may be a more suitable choice.

Second, these conventional distributed MDC algorithms, along with the majority of distributed classification methods, often necessitate that the attribute values of each training data instance be definitive. However, in a wide range of real application scenarios, due to the measurement errors of the equipment, the attributes of the collected data exhibit a certain level of uncertainty [18–21]. Traditional MDC/DL approaches overlook the uncertainty information present in training data, potentially resulting in inadequate classification results.

In the past two decades, some studies have considered the problem of data uncertainty caused by observation inaccuracies [22–26], which usually intend to adjusting the weights of data samples with high uncertainty so as to make the induced model more effective. In recent years, a small number of studies have also modeled uncertain data by characterizing data distributions. Nevertheless, these methods are constrained by the requirement for data with specific attributes, preventing them from achieving their intended effectiveness. For

example, the method proposed in [27] requires to employ a tuple-level model to characterize the data distribution, which is unavailable in many real-world scenarios. Besides, in the literature [28], the authors proposed to utilize a multi-dimensional Gaussian distribution to model the data uncertainty. However, this method can only be used for linearly separable binary classification problems, which limits its scope of applications.

Therefore, semi-supervised classification of multi-dimensional uncertain data constitutes a significant research challenge that needs investigation. This study presents a distributed semi-supervised classification algorithm for multi-dimensional uncertain data (dSMUDC) to address the challenges posed by absent labels and data uncertainty. The main contributions of this work are listed as follows:

1. To fully utilize the uncertainty information of uncertain data, we characterize uncertain data instances using a multi-dimensional Gaussian distribution. We then design the loss function for training data instances as the integral of misclassification losses over the Gaussian distribution.

2. To achieve nonlinear classification while simplifying integral computation, we reconstruct the probability density function of uncertain data using a new explicit feature mapping.

3. A one-vs.-one decomposition strategy is employed to transform the multi-dimensional heterogeneous class space into multiple homogeneous multi-label class spaces. Subsequently, the class labels of individual training data points are recovered by leveraging the manifold regularization term. Finally, the classifier is trained using the recovered class labels.

4. The theoretical analysis of the proposed algorithm is conducted.

The remainder of this work is organized as below. In Section 2, some preliminaries are briefly reviewed. In Section 3, a dSMUDC algorithm is developed, and its performance is analyzed. In Section 4, we show some numerical simulations on a series of datasets, and in Section 5, we draw some conclusions based on those simulation results.

## 2. Preliminaries

To ensure this paper is self-contained, some fundamental preliminaries should be briefly introduced.

### *2.1. Distributed Vector Quantization*

Vector quantization (VQ) is a method to achieve data compression while retaining as much relevant information as possible [29,30]. Its main idea is to partition the input data and generate a collection of reproduction vectors to represent the original data in each partition. VQ has several advantages, including a high compression rate, simple decoding, and good detail preservation, which makes it suitable for various applications.

The technological details of the distributed VQ (dVQ) method are presented as follows [29,30]: Given the training data set $\{\boldsymbol{x}_{j,n}\}$ at each node $j$, we define the reproduction vector and the associated set by $\{\boldsymbol{v}_{j,l}\}_{l=1}^{V}$ and $\{\mathcal{M}_{j,l}\}_{l=1}^{V}$, where $V$ represents the number of reproduction vectors.

**Initialization:** We set a small loop indexed by $\tau$, and initialize the values of $V$ reproduction vectors $\{\boldsymbol{v}_{j,l,0}\}_{l=1}^{V}$ with random values when $\tau = 0$.

**Data Partition:** At the following iterations $\tau > 0$, we can calculate the Euclidean distance between the reproduction vectors $\{\boldsymbol{v}_{j,l,\tau}\}_{l=1}^{V}$ and the dispersedly stored training data $\boldsymbol{x}$ as $\text{dist}_{j,n,l} = ||\boldsymbol{x}_{j,n} - \boldsymbol{v}_{j,l,\tau}||_2$. According to the calculation results, we can classify the training data into the nearest partition $\mathcal{M}_{j,l,\tau}$, i.e., $\boldsymbol{x}_{j,n} \in \mathcal{M}_{j,l,\tau}$, if $\text{dist}_{j,n,l} < \text{dist}_{j,n,r}, \forall r$.

**Local Update of Reproduction Vector:** We update the reproduction vectors based on the locally partitioned results

$$\boldsymbol{v}_{j,l,\tau}^{'} = \frac{\sum_{n \in \mathcal{M}_{j,l,\tau}} \boldsymbol{x}_{j,n}}{N_{j,l,\tau}^{'}}, \tag{1}$$

where the count of partition $\mathcal{M}_{j,l,\tau}$ is denoted by $N_{j,l,\tau}^{'}$.

**Information Fusion of Local Estimation:** We can obtain the global-like estimation of the reproduction vector by exchanging and fusing the immediate estimates among neighboring nodes, which is given by

$$\boldsymbol{v}_{j,l,\tau} = \frac{\boldsymbol{v}_{j,l,\tau}^{'} N_{j,l,\tau}^{'} + \exp(-\tau/J) \sum_{i \in \mathcal{B}_j} \boldsymbol{v}_{i,l,\tau}^{'} N_{i,l,\tau}^{'}}{N_{j,l,\tau}^{'} + \exp(-\tau/J) \sum_{i \in \mathcal{B}_j} N_{i,l,\tau}^{'}}, \tag{2}$$

where $\exp(-\tau/J)$ represents a coefficient that varies with time. This parameter can progressively reduce the impact of estimates from one-hop neighbors as the partitioned results are regularly updated until convergence.

**Loop Termination Criterion:** As long as the difference between the updated reproduction vectors at two successive iterations is smaller than the threshold $\varepsilon$, the small loop indexed by $\tau$ terminates. Otherwise, the small loop continues, i.e., $\tau = \tau + 1$.

### *2.2. Explicit Feature Map*

To achieve non-linear classification, a non-linear discriminant function should be employed for classifier induction.

As we all know, the learning performance of kernel-based classifiers is largely dependent on the settings of the kernel function. However, in many real-world situations, it is difficult to select optimal parameters due to a lack of prior knowledge. In some circumstances with complicated data distributions, the densities of distinct areas might differ greatly, making it impossible for a discriminant function that employs a single global kernel to develop an efficient decision boundary across all regions. To solve these problems, referring to [29], an explicit feature mapping method has been proposed based on VQ to replace the original kernel map. By using this explicit mapping method, the complex kernel parameter selection can be avoided, and the ability to characterize the boundary of complex data distribution can be improved.

The process of constructing an explicit feature mapping function based on VQ can be summarized as follows:

**Data partition:** Given the training data set at each node $j$, we employ the dVQ algorithm to obtain the global consensus reproduction vectors $\{\boldsymbol{v}_{j,l}\}_{l=1}^{V}$ and the partitions $\{\mathcal{M}_{j,l}\}_{l=1}^{V}$, where the reproduction vector $\boldsymbol{v}_{j,l}$ represents the mean vector of data instances in the partition $\mathcal{M}_{j,l}$, which is located at the center of the partition.

**Data Distribution Characterization:** We characterize the data distribution in a small region using the Gaussian distribution. Supposing that the training data is regularly distributed in a limited region, the probability density function (pdf) of the data distribution can be articulated as

$$p(\boldsymbol{x}|\boldsymbol{v}_{j,l}, \boldsymbol{\Pi}_{j,l}) = \frac{\exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{v}_{j,l})^T \Pi_{j,l}^{-1} (\boldsymbol{x} - \boldsymbol{v}_{j,l}))}{(2\pi)^{\frac{D}{2}} (\det(\Pi_{j,l}))^{\frac{1}{2}}}, \tag{3}$$

where the covariance matrix $\Pi_{j,l} \in \mathbb{R}^{D\times D}$ represents the degree of data dispersion in the partition $\mathcal{M}_{j,l}$, which can be calculated by

$$\mathbf{\Pi}_{j,l} = \frac{\sum_{n\in\mathcal{M}_{j,l}}(\boldsymbol{x}_{j,n} - \boldsymbol{v}_{j,l})(\boldsymbol{x}_{j,n} - \boldsymbol{v}_{j,l})^T}{N_{j,l}}. \tag{4}$$

**Explicit Feature Map using Gaussian pdf:** Utilizing the Gaussian pdfs, we may obtain the explicit feature map by consolidating the pdfs of all regions into a vector, represented as

$$\psi_j(\boldsymbol{x}) = (p(\boldsymbol{x}|\boldsymbol{v}_{j,1}, \Pi_{j,1}), \ldots, p(\boldsymbol{x}|\boldsymbol{v}_{j,V}, \Pi_{j,V}))^T \tag{5}$$

We use the modified mapping function with a multi-scale variant to boost its ability to cover data regions, making it applicable for regions with sparse training data instances, as referenced in [29]. To be specific, some positive coefficients scale the covariance matrix of the Gaussian model, so that the explicit feature map can be reformulated as

$$\psi_j(\boldsymbol{x}) = (\chi_j(\boldsymbol{x}|\boldsymbol{v}_{j,1}, \Pi_{j,1}), \ldots, \chi_j(\boldsymbol{x}|\boldsymbol{v}_{j,V}, \Pi_{j,V}))^T \tag{6}$$

where $\chi(\boldsymbol{x}|\boldsymbol{v}_{j,l}, \Pi_{j,l})$ denotes the Gaussian pdfs with the same mean vector and different covariance matrices

$$\chi_j(\boldsymbol{x}|\boldsymbol{v}_{j,l}, \Pi_{j,l}) = (p(\boldsymbol{x}|\boldsymbol{v}_{j,l}, s_1\Pi_{j,l}), \ldots, p(\boldsymbol{x}|\boldsymbol{v}_{j,l}, s_q\Pi_{j,l}))^T \tag{7}$$

with $s_1, \ldots, s_q$ being the scaling coefficients.

## 3. dSMUDC Algorithm

To achieve dSMUDC, we first formulate the issue of distributed semi-supervised classification of multi-dimensional uncertain data in Section 3.1. Then, we present the procedure of one-vs.-one decomposition encoding in Section 3.2 and design the global objective function in Section 3.3. Subsequently, it is followed by decentralized implementation in Section 3.4 and optimization in Section 3.5. Section 3.6 outlines the main steps of the decomposition decoding process. Ultimately, in Section 3.7, the performance of the proposed dSMUDC algorithm is analyzed.

### *3.1. Problem Formulation*

In this work, a distributed network consisting of $J$ nodes is considered. A total of $L + U$ data are distributed across these $J$ nodes. At each node $j$, there are $L_j$ labeled uncertain data $\{\boldsymbol{x}_{j,n}, \mathbf{\Theta}_{j,n}, \boldsymbol{y}_{j,n}\}_{n=1}^{L_j}$ and $U_j$ unlabeled uncertain data $\{\boldsymbol{x}_{j,n}, \mathbf{\Theta}_{j,n}, \boldsymbol{y}_{j,n}\}_{n=L_j+1}^{L_j+U_j}$, where $\{\boldsymbol{x}_{j,n}, \mathbf{\Theta}_{j,n}\}$ denotes the input features of uncertain data, and $\boldsymbol{y}_{j,n} \in \{+1, 0\}^{\sum_{m=1}^{M} K_m \times 1}$ denotes the corresponding labels.

The input features are represented as a multivariate Gaussian distribution with a mean vector $\boldsymbol{x}_{j,n} \in \mathbb{R}^{D\times 1}$ and a covariance matrix $\mathbf{\Theta}_{j,n} \in \mathbb{R}^{D\times D}$, as shown in Figure 1. More specifically, the Gaussian uncertainty pdf of the $n$-th data sample may be defined as [28]

$$f_{\boldsymbol{x}_{j,n}}(\boldsymbol{x}) = \frac{\exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_{j,n})^T(\mathbf{\Theta}_{j,n})^{-1}(\boldsymbol{x} - \boldsymbol{x}_{j,n}))}{(2\pi)^{\frac{D}{2}}|\mathbf{\Theta}_{j,n}|^{\frac{1}{2}}}. \tag{8}$$

Besides, the available class label vector of labeled data is composed of a total of $M$ heterogeneous class spaces $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_M$, i.e., $\boldsymbol{y}_{j,n} = [\boldsymbol{y}_{j,n,1}, \ldots, \boldsymbol{y}_{j,n,M}]^T$. In each $m$-th dimensional class space, the corresponding label vector $\boldsymbol{y}_{j,n,m} \in \{+1, 0\}^{K_m\times 1}$ is a $K_m$-dimensional binary vector, where the $k$-th element $y_{j,n,m,k} = 1$ if the $k$-th label is valid, and $y_{j,n,m,k} = 0$ otherwise.

In order to execute distributed information fusion, each node $j$ aims to identify the global optimal classifier utilizing its local data and the information shared with its neighboring nodes $\mathcal{B}_j$, which includes all one-hop neighbors of node $j$ as well as node $j$ itself.

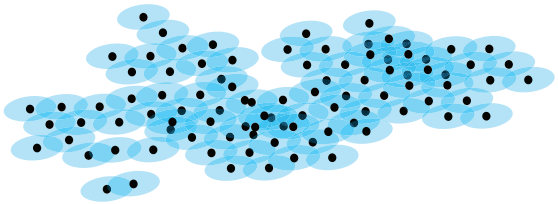


**Figure 1.** This diagram illustrates the uncertain data. Here, each data sample follows a Gaussian distribution. The dot represents the mean of the Gaussian distribution, while the translucent area surrounding the dot denotes the main data region covered by this Gaussian distribution.

### *3.2. One-vs.-One Decomposition Encoding*

The heterogeneity of multidimensional spaces is a substantial barrier in multidimensional classification challenges, as the output values of various spaces are incomparable. To address this issue, we use a one-vs.-one decomposition strategy to divide the heterogeneous multidimensional space into numerous homogeneous class spaces [16].

To be specific, taking the $m$-th dimensional class vector as an example, if the $n$-th training data sample is labeled, we utilize a one-vs.-one decomposition strategy to translate the $K_m$-dimensional vectors $\boldsymbol{y}_{j,n,m}$ in the original class space into $K_m^{'} = \binom{K_m}{2}$-dimensional ternary label vectors $\boldsymbol{y}_{j,n,m}^{'} \in \{-1, 0, +1\}^{K_m^{'} \times 1}$.

Likewise, if the $n$-th training data sample is unlabeled, we can obtain a $K_m^{'}$-dimensional transformed label vector with all elements being zero.

To explain the process of one-vs.-one decomposition, we present the corresponding transformation formula.

For the $m$-dimensional class space of the original label vector $\boldsymbol{y}_{j,n}$, if the element in the $r$-th dimension denotes the actual label, then the value of the $l$-th dimension in the newly converted vector $\boldsymbol{y}_{j,n}^{'}$ is [16]

$$\boldsymbol{y}_{j,n,m,l}^{'} = \begin{cases} +1, & \text{if} \quad l \in \mathcal{Q}_{j,n,m}, \\ -1, & \text{if} \quad l \in \mathcal{R}_{j,n,m}, \\ 0. & \text{otherwise.} \end{cases} \tag{9}$$

Here, when $y_{j,n,m,k} = 1$ and $1 \leq k < K_m$, the set $\mathcal{Q}_{j,n,m} = \{[(k-1)K_m - \sum_{i=1}^{k}(i-1)+1], [(k-1)K_m - \sum_{i=1}^{k}(i-1)+2] \ldots, [kK_m - \sum_{i=1}^{k} i]\}$ denotes the collection of values of sequence index $l$ for all elements where the transformed $\boldsymbol{y}_{j,n,m,l}^{'} = 1$. Correspondingly, the set $\mathcal{R}_{j,n,m} = \{k-1, [k-1+\sum_{i=0}^{1}(K_m-2-i)], [k+\sum_{i=0}^{2}(K_m-2-i)], \ldots, [k+\sum_{i=0}^{k-3}(K_m-2-i)]\}$ denotes the collection of values of sequence index $l$ for all elements where the transformed $\boldsymbol{y}_{j,n,m,l}^{'} = -1$.

For the sake of clarity, the process of label encoding is illustrated in Figure 2.

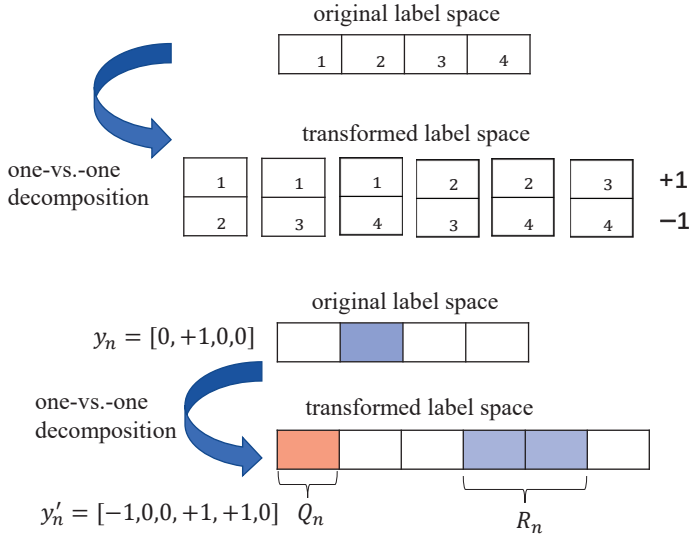



**Figure 2.** Schematic diagram of label encoding.

**Example 1.** *Given four data points, with their original labels in the m-th dimension being* $\boldsymbol{y}_{1,m} = [1,0,0,0]^T$, $\boldsymbol{y}_{2,m} = [0,1,0,0]^T$, $\boldsymbol{y}_{3,m} = [0,0,1,0]^T$, $\boldsymbol{y}_{4,m} = [0,0,0,1]^T$, *the transformed labels and the corresponding positive/negative index sets are in sequence via (9)*

$$\begin{aligned}
\boldsymbol{y}'_{1,m} &= [+1,+1,+1,0,0,0]^T, \quad \mathcal{Q}_{1,m} = \{1,2,3\}, \quad \mathcal{R}_{1,m} = \emptyset, \\
\boldsymbol{y}'_{2,m} &= [-1,0,0,+1,+1,0]^T, \quad \mathcal{Q}_{2,m} = \{4,5\}, \quad \mathcal{R}_{2,m} = \{1\}, \\
\boldsymbol{y}'_{3,m} &= [0,-1,0,-1,0,+1]^T, \quad \mathcal{Q}_{3,m} = \{6\}, \quad \mathcal{R}_{3,m} = \{2,4\} \\
\boldsymbol{y}'_{4,m} &= [0,0,-1,0,-1,-1]^T, \quad \mathcal{Q}_{4,m} = \emptyset, \quad \mathcal{R}_{4,m} = \{3,5,6\}.
\end{aligned}$$

Based on the decomposition results above, given that the discriminant function is non-linear, the output of the discriminant function $u_{j,n,m,k}$ corresponding to the $k$-th reconstructed class of the $m$-th class space can be expressed as

$$u_{j,n,m,k} = \boldsymbol{w}_{j,m,k}^T \cdot \boldsymbol{\psi}(\boldsymbol{x}_{j,n}) + b_{j,m,k}, \quad k = 1, \ldots, K'_m, \tag{10}$$

where $\boldsymbol{w}_{j,m,k}$ and $b_{j,m,k}$ denote the weight variable and bias variable, respectively. Besides, $\boldsymbol{\psi}(\boldsymbol{x}_{j,n})$ denotes the non-linear map.

### *3.3. Design of Global Objective Function*

This paper focuses on Gaussian uncertain data classification. For such data samples, not only their mean but also their Gaussian distribution area should be considered valid information. Therefore, this information should be taken into account for model induction.

To make full use of the abundant information offered by these Gaussian pdfs, we propose to formulate the loss function of the n-th uncertain data sample as the expected value of the misclassification loss over the Gaussian pdfs $f_{\boldsymbol{x}_{j,n}}(\boldsymbol{x})$ [28]. Guided by this idea, we develop the global objective function as follows:

$$\begin{aligned}
\mathcal{F}_{\boldsymbol{x}} = \sum_{m=1}^{M} \sum_{k=1}^{K'_m} \Bigg[ & \frac{\lambda_A}{2L} \sum_{n=1}^{L} (y'_{n,m,k} - z_{j,n,m,k})^2 + \frac{\lambda_B}{2(L+U)} \sum_{n=1}^{L+U} \left( z_{n,m,k} - \sum_{l=1}^{L+U} \omega_{n,l} z_{l,m,k} \right)^2 \\
& + \frac{\lambda_C}{L+U} \sum_{n=1}^{L+U} \int_{\mathcal{R}} h(z_{n,m,k} - u_{n,m,k}) f_{\boldsymbol{x}_n}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + \frac{\lambda_D}{2} ||\boldsymbol{w}_{m,k}||_2^2 + \frac{\lambda_E}{2} b_{m,k}^2 \Bigg].
\end{aligned} \tag{11}$$

The objective function comprises five separate terms.

The first term serves to obtain valid supervisory information from labeled data with parameter $\lambda_A$, providing support for the estimation of label confidence.

The second term captures the estimated error in labeling confidence over the complete training dataset, controlled by the parameter $\lambda_B$. Because the correct label contains no noise, it is natural to deduce that samples that are close in the feature space have comparable labels. By employing this concept, we may assess the labeling confidence of the candidate label by the minimization of the estimated error. The similarity measure between the $n$-th and $l$-th data samples is given by $\omega_{n,l}$, and its value satisfies $\sum_l \omega_{n,l} = 1$.

The third term is the loss function for the n-th uncertain data sample, associated with a weight parameter indicated as $\lambda_C$. In this paper, we implement an exponential hinge loss function $h(\cdot)$ to expedite convergence, which quantifies the misclassification loss of the sample, represented as

$$h(z_{n,m,k} - u_{n,m,k}) = \exp(\beta \max(0, z_{n,m,k} - u_{n,m,k})), \tag{12}$$

where $\beta$ denotes a positive coefficient.

The last two terms function as regularization components concerning the model parameters $\boldsymbol{w}_{m,k}$ and $b_{m,k}$, are employed to control the model's complexity. Here, $\lambda_D$ and $\lambda_E$ represent the respective weight parameters.

### *3.4. Reformulation and Decentralization of Global Objective Function*

In contrast to the conventional classification framework, the objective function outlined above incorporates all uncertainty distributions through integration calculations, which can help guide the classification model. Despite the simplicity of the global objective function's expression, there are also two challenges during the decentralized optimization process. First, conventional classification approaches implicitly realize nonlinear mapping through kernel methods. Since the explicit form of the nonlinear feature mapping $\psi(\cdot)$ is unknown, deriving a closed-form solution for the integral calculation becomes quite challenging. Second, an observation of the global objective function reveals that maximizing the first term requires the aggregation of all training data samples at a single fusion center, which is impractical in a distributed network.

Addressing these two challenges involves three steps: developing an explicit feature map, simplifying the integral computation, and redefining the estimated error function for label confidence.

#### 3.4.1. Construction of Explicit Feature Map

To tackle the first problem, this study designs an explicit mapping function grounded in dVQ. Through the use of this explicit mapping function to introduce reasonable approximations to the objective function, we are able to derive a closed-form solution for the integral calculation.

With reference to [29,30], we intend to use the dVQ method to obtain a series of reproduction vectors that characterize the global data distribution and employ the explicit mapping function to construct the discriminant function. To be specific, under the initial setting where each node $j$ has gathered $L_j + U_j$ local training data samples $\{\boldsymbol{x}_{j,n}, \boldsymbol{\Theta}_{j,n}\}_{n=1}^{L_j+U_j}$ in total, we initially implement the dVQ method to derive $V$ quantization partitions $\{\mathcal{M}_{j,l}\}_{l=1}^{V}$, where $\mathcal{M}_{j,l} = \{\boldsymbol{x}_{j,n}, \boldsymbol{\Theta}_{j,n}\}_{n=1}^{N_{j,l}}$. It should be noted that the uncertainty of training data must be included in the development of the explicit mapping function. The mean vectors and covariance matrices inside $\mathcal{M}_{j,l}$ may be calculated as follows:

$$\tilde{\boldsymbol{v}}_{j,l} = \frac{\sum_{n \in \mathcal{M}_{j,l}} \boldsymbol{x}_{j,n}}{N_{j,l}}, \tag{13}$$

$$\tilde{\mathbf{\Pi}}_{j,l} = \frac{1}{N_{j,l}} \sum_{n \in \mathcal{M}_{j,l}} (\mathbf{\Theta}_{j,n} + (\boldsymbol{x}_{j,n} - \tilde{\boldsymbol{v}}_{j,l})(\boldsymbol{x}_{j,n} - \tilde{\boldsymbol{v}}_{j,l})^T). \tag{14}$$

The newly formulated covariance matrix $\tilde{\mathbf{\Pi}}_{j,l}$ effectively captures the degree of dispersion of training data within partitions $\mathcal{M}_{j,l}$, thereby facilitating a more accurate decision boundary, particularly in scenarios where data distributions differ across various network regions. Referring to the method in Section 2.2, we derive the new explicit feature map as follows [29]:

$$\begin{aligned} \psi(\boldsymbol{x}_{j,n}) = [&p(\boldsymbol{x}_{j,n}|\tilde{\boldsymbol{v}}_{j,1}, s_1\tilde{\mathbf{\Pi}}_{j,1}), p(\boldsymbol{x}_{j,n}|\tilde{\boldsymbol{v}}_{j,1}, s_2\tilde{\mathbf{\Pi}}_{j,1}), \ldots, p(\boldsymbol{x}_{j,n}|\tilde{\boldsymbol{v}}_{j,1}, s_q\tilde{\mathbf{\Pi}}_{j,1}), \\ &p(\boldsymbol{x}_{j,n}|\tilde{\boldsymbol{v}}_{j,2}, s_1\tilde{\mathbf{\Pi}}_{j,2}), p(\boldsymbol{x}_{j,n}|\tilde{\boldsymbol{v}}_{j,2}, s_2\tilde{\mathbf{\Pi}}_{j,2}), \ldots, p(\boldsymbol{x}_{j,n}|\tilde{\boldsymbol{v}}_{j,2}, s_q\tilde{\mathbf{\Pi}}_{j,2}), \\ &\ldots \\ &p(\boldsymbol{x}_{j,n}|\tilde{\boldsymbol{v}}_{j,V}, s_1\tilde{\mathbf{\Pi}}_{j,V}), p(\boldsymbol{x}_{j,n}|\tilde{\boldsymbol{v}}_{j,V}, s_2\tilde{\mathbf{\Pi}}_{j,V}), \ldots, p(\boldsymbol{x}_{j,n}|\tilde{\boldsymbol{v}}_{j,V}, s_q\tilde{\mathbf{\Pi}}_{j,V}), ] \end{aligned} \tag{15}$$

where the expression of pdf $p(\boldsymbol{x}_{j,n}|\tilde{\boldsymbol{v}}_{j,l}, s_1\tilde{\mathbf{\Pi}}_{j,l})$

$$p(\boldsymbol{x}_{j,n}|\tilde{\boldsymbol{v}}_{j,l}, s_k\tilde{\mathbf{\Pi}}_{j,l}) = \frac{\exp(-\frac{1}{2}(\boldsymbol{x}_{j,n} - \tilde{\boldsymbol{v}}_{j,l})^T(s_k\tilde{\mathbf{\Pi}}_{j,l})^{-1}(\boldsymbol{x}_{j,n} - \tilde{\boldsymbol{v}}_{j,l}))}{(2\pi)^{\frac{D}{2}}|s_k\tilde{\mathbf{\Pi}}_{j,l}|^{\frac{1}{2}}}. \tag{16}$$

Note that for the sake of convenience, we use the notation $\boldsymbol{\psi}_{j,n}$ to represent $\psi(\boldsymbol{x}_{j,n})$.

Algorithm 1 delineates the essential steps involved in the construction of an explicit non-linear feature map for clarity.

**Algorithm 1** Explicit Non-linear Feature Map Construction

**Require:** Input vector $\{\boldsymbol{x}_{j,n}, \mathbf{\Theta}_{j,n}\}_{n=1}^{L_j+U_j}$, and set initial values of cluster centers $\{\boldsymbol{v}_{j,l,0}\}_{l=1}^{V}$.

**while** $(||\boldsymbol{v}_{j,l,\tau} - \boldsymbol{v}_{j,l,\tau-1}||_2 > \varepsilon$ for $l = 1, \ldots, V)$
  **for** $j \in \mathcal{J}$ **do**
    Group the training data into the closest partition.
    Calculate partition centers $\boldsymbol{v}'_{j,l,\tau}$ and the corresponding counts $N'_{j,l,\tau}$ via (1).
    Exchange $\{\boldsymbol{v}'_{n,l,\tau}, N'_{j,l,\tau}\}_{l=1}^{V}$ with its neighboring nodes.
  **end for**
  **for** $j \in \mathcal{J}$ **do**
    Update the reproduction vectors $\{\boldsymbol{v}'_{j,l,\tau}\}_{l=1}^{V}$ via (2).
  **end for**
  $\tau = \tau + 1$
Calculate the mean $\tilde{\boldsymbol{v}}_{j,l}$ and covariance matrix $\tilde{\mathbf{\Pi}}_{j,l}$ via (13) and (14).
Construct the non-linear explicit feature map via (15).

3.4.2. Simplification of Integration Calculation

While a finite-dimensional explicit feature map can be obtained, if a closed-form solution for the integral cannot be derived directly, classical optimization methods cannot be applied to minimize the objective function. Considering this, we propose to simplify the integral computation by converting the integral problem over the Gaussian uncertainty distribution of $\boldsymbol{x}_{j,n}$ to an integral problem with respect to the distribution of $\boldsymbol{\psi}_{j,n}$.

$$\begin{aligned} \mathcal{F}_{\boldsymbol{\psi}} = \sum_{m=1}^{M}\sum_{k=1}^{K'_m}\Bigg[&\frac{\lambda_A}{2L}\sum_{n=1}^{L}(y'_{n,m,k} - z_{n,m,k})^2 + \frac{\lambda_B}{2(L+U)}\sum_{n=1}^{L+U}\left(z_{n,m,k} - \sum_{l=1}^{L+U}\omega_{n,l}z_{l,m,k}\right)^2 \\ &+\frac{\lambda_C}{L+U}\sum_{n=1}^{L+U}\int_{\mathcal{R}} h(z_{n,m,k} - u_{n,m,k})f_{\boldsymbol{\psi}_n}(\boldsymbol{\psi})\mathrm{d}\boldsymbol{\psi} + \frac{\lambda_D}{2}||\boldsymbol{w}_{m,k}||_2^2 + \frac{\lambda_E}{2}b_{m,k}^2\Bigg], \end{aligned} \tag{17}$$

However, the nonlinear relationship between $\boldsymbol{\psi}_{j,n}$ and $\boldsymbol{x}_{j,n}$ makes directly transforming the integral problem challenging. To overcome the issue, we suggest to use simple distributions to approximate the Gaussian uncertainty distribution of the explicit feature map $f_{\boldsymbol{\psi}_{j,n}}(\boldsymbol{\psi})$.

Specifically, we first determine the entries of the first-order moment of $f_{\boldsymbol{\psi}_{j,n}}(\boldsymbol{\psi})$ using the formula provided below, which acts as a mean vector,

$$\begin{aligned}\delta_{j,n,lr} &= \int_{\mathcal{R}} p(\boldsymbol{x}|\tilde{\boldsymbol{v}}_{j,l}, s_r\tilde{\boldsymbol{\Pi}}_{j,l}) f_{\boldsymbol{x}_{j,n}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\ &= \exp[-\frac{1}{2}(2D\log 2\pi - \log|\Xi_1| - \log|\Xi_2| + \boldsymbol{\eta}_1^T\Xi_1^{-1}\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2^T\Xi_2^{-1}\boldsymbol{\eta}_2) \\ &\quad - \frac{1}{2}(D\log 2\pi - \log|\Xi_1+\Xi_2| + (\boldsymbol{\eta}_1+\boldsymbol{\eta}_2)^T(\Xi_1+\Xi_2)^{-1}(\boldsymbol{\eta}_1+\boldsymbol{\eta}_2))].\end{aligned} \tag{18}$$

Additionally, we use the formula provided below to calculate the diagonal entries of the second-order moment of $f_{\boldsymbol{\psi}_{j,n}}(\boldsymbol{\psi})$, and this matrix can act as an approximated covariance matrix

$$\begin{aligned}\Lambda_{j,n,lr} &= \int_{\mathcal{R}} (p(\boldsymbol{x}|\tilde{\boldsymbol{v}}_{j,l}, s_r\tilde{\boldsymbol{\Pi}}_{j,l}) - \delta_{j,n,lr})^2 f_{\boldsymbol{x}_{j,n}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\ &= \exp[-\frac{1}{2}(3D\log 2\pi - 2\log|\Xi_1| - \log|\Xi_2| + 2\boldsymbol{\eta}_1^T\Xi_1^{-1}\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2^T\Xi_2^{-1}\boldsymbol{\eta}_2) \\ &\quad - \frac{1}{2}(D\log 2\pi - \log|2\Xi_1+\Xi_2| + (2\boldsymbol{\eta}_1+\boldsymbol{\eta}_2)^T(2\Xi_1+\Xi_2)^{-1}(2\boldsymbol{\eta}_1+\boldsymbol{\eta}_2))] - (\delta_{j,n,kl})^2,\end{aligned} \tag{19}$$

where the variables

$$\Xi_1 = (s_r\tilde{\boldsymbol{\Pi}}_{j,l})^{-1}, \quad \Xi_2 = \boldsymbol{\Theta}_{j,n}^{-1}, \quad \boldsymbol{\eta}_1 = (s_r\tilde{\boldsymbol{\Pi}}_{j,l})^{-1}\tilde{\boldsymbol{v}}_{j,l}, \quad \boldsymbol{\eta}_2 = \boldsymbol{\Theta}_{j,n}^{-1}\boldsymbol{x}_{j,n}.$$

Using the approximated mean vector and covariance matrix, we can employ the following Gaussian probability density function to approximate the uncertainty distribution of data samples in the feature space

$$\tilde{f}_{\boldsymbol{\psi}_{j,n}}(\boldsymbol{\psi}) = \frac{\exp(-\frac{1}{2}(\boldsymbol{\psi}-\boldsymbol{\delta}_{j,n})^T\boldsymbol{\Lambda}_{j,n}^{-1}(\boldsymbol{\psi}-\boldsymbol{\delta}_{j,n}))}{(2\pi)^{\frac{Vq}{2}}|\boldsymbol{\Lambda}_{j,n}|^{\frac{1}{2}}} \tag{20}$$

where the mean vector $\boldsymbol{\delta}_{j,n} = [\delta_{j,n,11}, \delta_{j,n,12}, \ldots, \delta_{j,n,kl}, \ldots, \delta_{j,n,qV}]$ and the covariance matrix $\boldsymbol{\Lambda}_{j,n} = \mathrm{diag}([\Lambda_{j,n,11}, \Lambda_{j,n,12}, \ldots, \Lambda_{j,n,kl}, \ldots, \Lambda_{j,n,qV}])$.

Since there exists the linear correlation between $\boldsymbol{x}$ and $\boldsymbol{\psi}$ in a localized region, this novel Gaussian pdf is unlikely to result in significant information loss. Utilizing (20), we reformulate the integral problem with respect to the Gaussian uncertainty distribution of $\boldsymbol{x}$ as

$$\begin{aligned}\tilde{\mathcal{F}}_{\boldsymbol{\psi}} = \sum_{m=1}^{M}\sum_{k=1}^{K_m'}\Bigg[&\frac{\lambda_A}{2L}\sum_{n=1}^{L}(y_{n,m,k}' - z_{n,m,k})^2 + \frac{\lambda_B}{2(L+U)}\sum_{n=1}^{L+U}\left(z_{n,m,k} - \sum_{l=1}^{L+U}\omega_{n,l}z_{l,m,k}\right)^2 \\ &+ \frac{\lambda_C}{L+U}\sum_{n=1}^{L+U}\int_{\mathcal{R}} h(z_{n,m,k} - u_{n,m,k})\tilde{f}_{\boldsymbol{\psi}_n}(\boldsymbol{\psi})\mathrm{d}\boldsymbol{\psi} + \frac{\lambda_D}{2}||\boldsymbol{w}_{m,k}||_2^2 + \frac{\lambda_E}{2}b_{m,k}^2\Bigg],\end{aligned} \tag{21}$$

3.4.3. Reconstruction of Estimated Error Function on Labeling Confidence

To develop a decentralized objective function without aggregating all training data into a fusion center, we employ reproduction vectors to represent the complete dataset and reconstruct labeling confidence. This is justified as follows: Because the reproduction vectors are created using dVQ, their distribution is similar to the global data distribution.

Thus, it can be concluded that the labeling confidence generated from these vectors is close to that acquired from the total dataset. Based on this, we redesign the objective function as

$$\begin{aligned}\tilde{\tilde{\mathcal{F}}}_{\boldsymbol{\psi}} = \sum_{m=1}^{M}\sum_{k=1}^{K_m^{'}}\Bigg[&\frac{\lambda_A}{2L}\sum_{n=1}^{L}(y_{n,m,k}^{'} - z_{n,m,k})^2 + \frac{\lambda_B}{2(L+U)}\sum_{n=1}^{L+U}\left(z_{n,m,k} - \sum_{l=1}^{V}\omega_{n,l}^{v} z_{l,m,k}^{v}\right)^2 \\ &+\frac{\lambda_C}{L+U}\sum_{n=1}^{L+U}\int_{\mathcal{R}} h(z_{n,m,k} - u_{n,m,k})\tilde{f}_{\boldsymbol{\psi}_n}(\boldsymbol{\psi})\mathrm{d}\boldsymbol{\psi} + \frac{\lambda_D}{2}||\boldsymbol{w}_{m,k}||_2^2 + \frac{\lambda_E}{2}b_{m,k}^2\Bigg],\end{aligned} \tag{22}$$

where $\omega_{n,l}^{v}$ measures the degree of the similarity between $\boldsymbol{x}_n$ and $\tilde{\boldsymbol{v}}_l$, which is calculated by $p(\boldsymbol{x}|\tilde{\boldsymbol{v}}_l, \tilde{\boldsymbol{\Pi}}_l)$. Furthermore, $z_{l,m}^{v}$ stands for the confidence of the $m$-th label class for the $l$-th reproduction data.

3.4.4. Decentralization Implementation

Based on (22), we can achieve decentralization of global objective function by substituting global variables with local ones and enforcing consensus equality constraints [17], namely,

$$\begin{aligned}\tilde{\tilde{\mathcal{F}}}_{\boldsymbol{\psi}} = &\sum_{j=1}^{J}\sum_{m=1}^{M}\sum_{k=1}^{K_m^{'}}\Bigg[\frac{\lambda_A}{2L}\sum_{n=1}^{L_j}(y_{j,n,m,k}^{'} - z_{j,n,m,k})^2 + \frac{\lambda_B}{2(L+U)}\sum_{n=1}^{L_j+U_j}\left(z_{j,n,m,k} - \sum_{l=1}^{V}\omega_{n,l}^{v} z_{l,m,k}^{v}\right)^2 \\ &+\frac{\lambda_C}{L+U}\sum_{n=1}^{L_j+U_j}\int_{\mathcal{R}} h(z_{j,n,m,k} - u_{j,n,m,k})\tilde{f}_{\boldsymbol{\psi}_{j,n}}(\boldsymbol{\psi})\mathrm{d}\boldsymbol{\psi} + \frac{\lambda_D}{2}||\boldsymbol{w}_{j,m,k}||_2^2 + \frac{\lambda_E}{2}b_{j,m,k}^2\Bigg] \\ &= \sum_{j=1}^{J}\tilde{\tilde{\mathcal{F}}}_{\boldsymbol{\psi}}^{j} \\ \text{s.t.}\quad &\boldsymbol{w}_{j,m,k} = \boldsymbol{w}_{i,m,k}, \quad b_{j,m,k} = b_{i,m,k}, \quad j \in \mathcal{J}, \quad i \in \mathcal{B}_j.\end{aligned} \tag{23}$$

*3.5. Optimization*

In this subsection, we would like to use an alternating minimization strategy to simultaneously optimize the variables $\{\boldsymbol{w}_{j,m,k}, b_{j,m,k}\}$ and $\{z_{j,n,m,k}, z_{j,l,m,k}^{v}\}$. To obtain the update equations, referring to Lemma 1 [28], we simplify the decentralized objective function into

$$\begin{aligned}\tilde{\tilde{\mathcal{F}}}_{\boldsymbol{\psi}} = &\sum_{j=1}^{J}\sum_{m=1}^{M}\sum_{k=1}^{K_m^{'}}\Bigg[\frac{\lambda_A}{2L}\sum_{n=1}^{L_j}(y_{j,n,m,k}^{'} - z_{j,n,m,k})^2 + \frac{\lambda_B}{2(L+U)}\sum_{n=1}^{L_j+U_j}\left(z_{j,n,m,k} - \sum_{l=1}^{V}\omega_{n,l}^{v} z_{l,m,k}^{v}\right)^2 \\ &+\frac{\lambda_C}{2(L+U)}\sum_{n=1}^{L_j+U_j}\exp\left(\frac{\beta^2 d_{\boldsymbol{\Lambda}_{j,n,m,k}}^2}{4} + \beta d_{\delta_{j,n,m,k}}\right)\left(1 + \mathrm{erf}\left(\frac{d_{\delta_{j,n,m,k}}}{d_{\boldsymbol{\Lambda}_{j,n,m,k}}} + \frac{\beta d_{\boldsymbol{\Lambda}_{j,n,m,k}}}{2}\right)\right) \\ &+\frac{\lambda_D}{2J}||\boldsymbol{w}_{j,m,k}||_2^2 + \frac{\lambda_E}{2J}b_{j,m,k}^2\Bigg], \\ \text{s.t.}\quad &\boldsymbol{w}_{j,m,k} = \boldsymbol{w}_{i,m,k}, \quad b_{j,m,k} = b_{i,m,k}, \quad j \in \mathcal{J}, \quad i \in \mathcal{B}_j.\end{aligned} \tag{24}$$

where the variables $d_{\delta_{j,n,m,k}} = z_{j,n,m,k} - \boldsymbol{w}_{j,m,k}^T \cdot \boldsymbol{\delta}_{j,n} - b_{j,m,k}$, $d_{\boldsymbol{\Lambda}_{j,n,m,k}} = \sqrt{2\boldsymbol{w}_{j,m,k}^T \boldsymbol{\Lambda}_{j,n}\boldsymbol{w}_{j,m,k}}$, and the error function $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp(-t^2)\mathrm{d}t$.

The detailed derivation process is discussed in Lemma 1.

**Lemma 1.** *Assume that* $\boldsymbol{\psi} \in \mathcal{R}$ *is a random variable with a multivariate Gaussian distribution* $N(\boldsymbol{\delta}, \boldsymbol{\Lambda})$*, that is,*

$$p(\boldsymbol{\psi}) = \frac{\exp(-\frac{1}{2}(\boldsymbol{\psi}-\boldsymbol{\delta})^T\boldsymbol{\Lambda}^{-1}(\boldsymbol{\psi}-\boldsymbol{\delta}))}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Lambda}|^{\frac{1}{2}}}. \tag{25}$$

Given a hyperplane $z - \boldsymbol{w}^T \cdot \boldsymbol{\psi} - b = 0$, the expectation of the exponential hinge loss $h(z - \boldsymbol{w}^T \cdot \boldsymbol{\psi} - b)$ with respect to $\boldsymbol{\psi}$ can be calculated as follows:

$$\begin{aligned} E_{p(\boldsymbol{\psi})}(\boldsymbol{w}, b) &\triangleq \int_{\mathcal{R}} \exp\Big[\beta \max\Big(0, z - \boldsymbol{w}^T \cdot \boldsymbol{\psi} - b\Big)\Big] p(\boldsymbol{\psi}) \mathrm{d}\boldsymbol{\psi} \\ &= \frac{1}{2}\exp(\frac{\beta^2 d_{\boldsymbol{\Lambda}}^2}{4} + \beta d_{\delta})(1 + \mathrm{erf}(\frac{d_{\delta}}{d_{\boldsymbol{\Lambda}}} + \frac{\beta d_{\boldsymbol{\Lambda}}}{2})), \end{aligned} \tag{26}$$

where the variables $d_{\delta} = z - \boldsymbol{w}^T \cdot \boldsymbol{\delta} - b$ and $d_{\boldsymbol{\Lambda}} = \sqrt{2\boldsymbol{w}^T\boldsymbol{\Lambda}\boldsymbol{w}}$.

**Proof.** Referring to [28], we firstly perform eigenvalue decomposition on the matrix $\boldsymbol{\Lambda}$ and obtain $\boldsymbol{\Lambda} = \boldsymbol{U}^T D \boldsymbol{U}$. Then, we define the new integral variable as $\boldsymbol{\psi}^{'} = D^{-\frac{1}{2}}U(\boldsymbol{\psi} - \boldsymbol{\delta})$. By letting $\boldsymbol{w}^{'} = \boldsymbol{D}^{\frac{1}{2}}\boldsymbol{U}\boldsymbol{w}$, we have

$$\begin{aligned} E_{p(\boldsymbol{\psi})}(\boldsymbol{w}, b) &= \int_{\mathcal{R}} \exp\Big[\beta \max\Big(0, z - \boldsymbol{w}^T\boldsymbol{\delta} - b - (\boldsymbol{w}^{'})^T\boldsymbol{\psi}^{'}\Big)\Big] \frac{\exp(-\frac{1}{2}(\boldsymbol{\psi}^{'})^T\boldsymbol{\psi}^{'})}{(2\pi)^{\frac{D}{2}}} \mathrm{d}\boldsymbol{\psi}^{'} \\ &= \int_{\mathcal{R}_1} \exp\Big[\beta\Big(z - \boldsymbol{w}^T\boldsymbol{\delta} - b - (\boldsymbol{w}^{'})^T\boldsymbol{\psi}^{'}\Big)\Big] \frac{\exp(-\frac{1}{2}(\boldsymbol{\psi}^{'})^T\boldsymbol{\psi}^{'})}{(2\pi)^{\frac{D}{2}}} \mathrm{d}\boldsymbol{\psi}^{'}, \end{aligned} \tag{27}$$

where $\mathcal{R}_1 = \{\boldsymbol{\psi}^{'} \in \mathcal{R} : z - \boldsymbol{w}^T\boldsymbol{\delta} - b - (\boldsymbol{w}^{'})^T\boldsymbol{\psi}^{'} \geq 0\}$.

Then, we introduce a unit orthogonal matrix $\boldsymbol{B}$, which satisfies $\boldsymbol{B}\boldsymbol{w}^{'} = ||\boldsymbol{w}^{'}||_2\boldsymbol{e}_k$. Here, the $k$-th element of $\boldsymbol{e}_k$ is 1, and the rest are 0. We define another new integral variable $\boldsymbol{\psi}^{''} = \boldsymbol{B}\boldsymbol{\psi}^{'}$, and then obtain

$$E_{p(\boldsymbol{\psi})}(\boldsymbol{w}, b) = \int_{\mathcal{R}_2} \exp\Big[\beta\Big(z - \boldsymbol{w}^T\boldsymbol{\delta} - b - ||\boldsymbol{w}^{'}||_2\boldsymbol{e}_k^T\boldsymbol{\psi}^{''}\Big)\Big] \frac{\exp(-\frac{1}{2}(\boldsymbol{\psi}^{''})^T\boldsymbol{\psi}^{''})}{(2\pi)^{\frac{D}{2}}} \mathrm{d}\boldsymbol{\psi}^{''}, \tag{28}$$

where $\mathcal{R}_2 = \{\boldsymbol{\psi}^{''} \in \mathcal{R} : z - \boldsymbol{w}^T\boldsymbol{\delta} - b - ||\boldsymbol{w}^{'}||_2\boldsymbol{e}_k^T\boldsymbol{\psi}^{''} \geq 0\}$. In $\mathcal{R}_2$, only the integral domain of the $k$-th element of $\boldsymbol{\psi}^{''}$ is restricted.

Therefore, defining the $k$-th element of $\boldsymbol{\psi}^{''}$ as $r_k$, we can obtain

$$\begin{aligned} E_{p(\boldsymbol{\psi})}(\boldsymbol{w}, b) &= \int_{-\infty}^{\frac{z-\boldsymbol{w}^T\boldsymbol{\delta}-b}{||\boldsymbol{w}^{'}||_2}} \exp\Big[\beta\Big(z - \boldsymbol{w}^T\boldsymbol{\delta} - b - ||\boldsymbol{w}^{'}||_2 r_k\Big)\Big] \frac{\exp(-\frac{1}{2}(r_k)^2)}{(2\pi)^{\frac{1}{2}}} \mathrm{d}r_k \\ &\quad \times \prod_{j \neq k} \int_{-\infty}^{+\infty} \frac{\exp(-\frac{1}{2}(r_j)^2)}{(2\pi)^{\frac{1}{2}}} \mathrm{d}r_j \\ &= \frac{1}{2}\exp\left[\frac{\beta^2||\boldsymbol{w}^{'}||_2^2}{2} + \beta\Big(z - \boldsymbol{w}^T\boldsymbol{\delta} - b\Big)\right] \cdot \left[1 + \mathrm{erf}\left(\frac{z - \boldsymbol{w}^T\boldsymbol{\delta} - b}{\sqrt{2}||\boldsymbol{w}^{'}||_2} + \frac{\beta||\boldsymbol{w}^{'}||_2}{\sqrt{2}}\right)\right] \end{aligned} \tag{29}$$

For simplicity, we let $d_{\delta} \triangleq z - \boldsymbol{w}^T\boldsymbol{\delta} - b$ and $d_{\boldsymbol{\Lambda}} \triangleq \sqrt{2\boldsymbol{w}^T\boldsymbol{\Lambda}\boldsymbol{w}}$. Since $||\boldsymbol{w}^{'}||_2 = \sqrt{\boldsymbol{w}^T\boldsymbol{\Lambda}\boldsymbol{w}}$, we can obtain

$$E_{p(\boldsymbol{\psi})}(\boldsymbol{w}, b) = \frac{1}{2}\exp(\frac{\beta^2 d_{\boldsymbol{\Lambda}}^2}{4} + \beta d_{\delta})(1 + \mathrm{erf}(\frac{d_{\delta}}{d_{\boldsymbol{\Lambda}}} + \frac{\beta d_{\boldsymbol{\Lambda}}}{2})). \tag{30}$$

Lemma 1 is proven. □

3.5.1. Update of Labeling Confidence of Reproduction Data

Before the large loop indexed by $t$, we set another loop indexed by $\epsilon$ for the update of the labeling confidence of reproduction data. At the initial step $\epsilon = 0$, we initialize the labeling confidence $z^v_{j,l,m,k}$ at each node $j$

$$z^v_{j,l,m,k}(0) = \frac{1}{N_{j,l}} \sum_{n \in \mathcal{M}_{j,l}} y'_{j,n,m,k}. \tag{31}$$

At iteration $\epsilon > 0$, we can update the labeling confidence of the reproduction vector by fusing the intermediate estimates among one-hop neighbors, i.e.,

$$z^v_{j,l,m,k}(\epsilon + 1) = \sum_{i \in \mathcal{B}_j} c_{ji} z^v_{i,l,m,k}(\epsilon), \tag{32}$$

where $c_{ji}$ denotes the cooperative coefficient, which satisfies the Metropolis rule [17].

After a total of $T$ iterations, the global consensus estimation can be obtained, that is, $z^v_{l,m,k} = z^v_{1,l,m,k}(T) = \ldots = z^v_{J,l,m,k}(T)$.

3.5.2. Update of Labeling Confidence of Training Data Samples

Since the above equation seems to be complicated, we intend to employ the gradient descent method to optimize the objective function.

Using the gradient descent method, we have

$$z_{j,n,m,k}(t+1) = z_{j,n,m,k}(t) - \zeta_1(t+1) \nabla_{z_{j,n,m,k}} \tilde{\tilde{\mathcal{F}}}_{\boldsymbol{\psi}} |_{z_{j,n,m,k}}, \tag{33}$$

where $\zeta_1(t+1)$ denotes a time-varying step size. Besides, $\nabla_{z_{j,n,m,k}} \tilde{\tilde{\mathcal{F}}}_{\boldsymbol{\psi}} |_{z_{j,n,m,k}}$ denotes the partial derivation of $\tilde{\mathcal{F}}_{\boldsymbol{\psi}}$ with respect to $z_{j,n,m,k}$.

3.5.3. Update of Model Parameter

Similarly, we can utilize the gradient descent method and diffusion cooperative strategy [2] to seek the global optimal solution of objective function.

We can obtain the optimal values of $\boldsymbol{w}_{j,m,k}$ and $b_{j,m,k}$ by executing the following update equations until convergence:

$$\breve{\boldsymbol{w}}_{j,m,k}(t+1) = \boldsymbol{w}_{j,m,k}(t) - \zeta_2(t+1) \nabla_{\boldsymbol{w}_{j,m,k}} \tilde{\tilde{\mathcal{F}}}_{\boldsymbol{\psi}} |_{\boldsymbol{w}_{j,m,k}}, \tag{34}$$

$$\boldsymbol{w}_{j,m,k}(t+1) = \sum_{i \in \mathcal{B}_j} c_{ji} \breve{\boldsymbol{w}}_{i,m,k}(t+1), \tag{35}$$

and

$$\breve{b}_{j,m,k}(t+1) = b_{j,m,k}(t) - \zeta_2(t+1) \nabla_{b_{j,m,k}} \tilde{\tilde{\mathcal{F}}}_{\boldsymbol{\psi}} |_{b_{j,m,k}}, \tag{36}$$

$$b_{j,m,k}(t+1) = \sum_{i \in \mathcal{B}_j} c_{ji} \breve{b}_{i,m,k}(t+1), \tag{37}$$

where $\zeta_2(t+1)$ denotes the time-varying step size. Additionally, $\nabla_{\boldsymbol{w}_{j,m,k}} \tilde{\tilde{\mathcal{F}}}_{\boldsymbol{\psi}} |_{\boldsymbol{w}_{j,m,k}}$ and $\nabla_{b_{j,m,k}} \tilde{\tilde{\mathcal{F}}}_{\boldsymbol{\psi}} |_{b_{j,m,k}}$ denote the corresponding gradients.

### *3.6. One-vs.-One Decomposition Decoding*

Utilizing the induced MDC classifier, for an unseen data point $\boldsymbol{x}^*$, we can derive the predicted labels of $\boldsymbol{x}^*$ in the transformed class space $\boldsymbol{y}^{*'} \in \{+1, -1\}^{\sum_{m=1}^{M} K'_m}$. Subsequently, we employ a one-vs.-one decoding technique to derive the labels in the original class space. We enumerate the elements in vector $\boldsymbol{y}^{*'}$ whose sequence indexes are included in sets $\mathcal{Q}_{j,n,m}$ and $\mathcal{R}_{j,n,m}$, represented as $q_{m,k}$ and $r_{m,k}$, respectively. The $k$-th predicted labels of unseen data $\boldsymbol{x}^*$ in the $m$-dimensional original class space may be derived using the majority voting approach.

$$y^*_{m,k} = \begin{cases} +1, & \text{if } k = \arg\max_{1 \le l \le K_m} q_{m,k} + r_{m,k} \\ 0, & \text{otherwise,} \end{cases} \quad m = 1, \ldots, M. \tag{38}$$

To enhance clarity, the label decoding process is depicted in Figure 3.

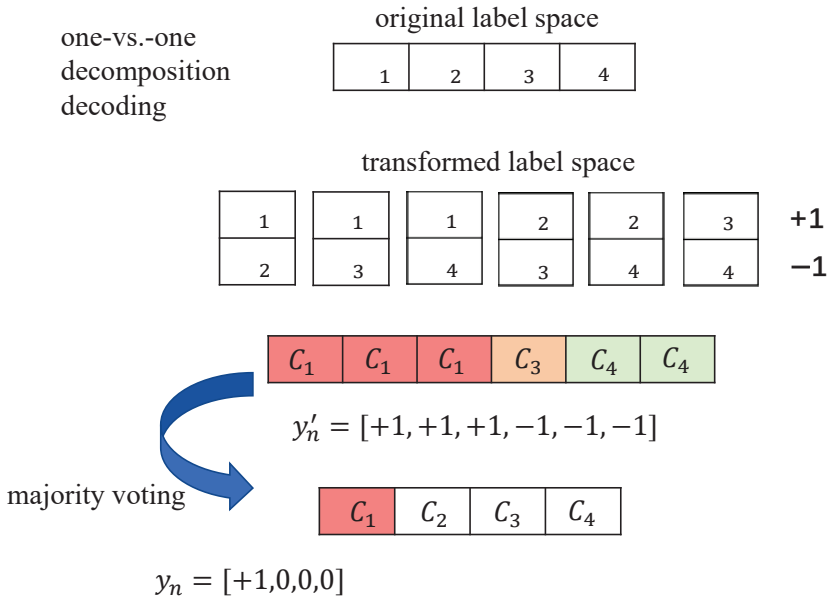



**Figure 3.** Schematic diagram of label decoding.

**Example 2.** *Given two unseen data points, with their predicted labels in the m-th dimension being* $\boldsymbol{y}^{*'}_{1,m} = [+1, +1, +1, -1, -1, -1]$, $\boldsymbol{y}^{*'}_{2,m} = [+1, +1, -1, +1, -1, -1]^T$, *the vote count w.r.t. four possible classes are in sequence*

$$\begin{aligned} &q_{1,m,1} + r_{1,m,1} = 3, \quad q_{1,m,2} + r_{1,m,2} = 0, \quad q_{1,m,3} + r_{1,m,3} = 1, \quad q_{1,m,4} + r_{1,m,4} = 2, \\ &q_{2,m,1} + r_{1,m,1} = 2, \quad q_{2,m,2} + r_{1,m,2} = 1, \quad q_{2,m,3} + r_{1,m,3} = 0, \quad q_{2,m,4} + r_{1,m,4} = 3. \end{aligned}$$

*Based on the voting results, the final decoded label can be obtained in accordance with (38) as follows:*

$$\boldsymbol{y}^*_{1,m} = [+1, 0, 0, 0]^T, \quad \boldsymbol{y}^*_{2,m} = [0, 0, 0, +1]^T.$$

For clarity, the flowchart of the proposed algorithm is presented in Figure 4, and the main steps of the dSMUDC algorithm are summarized in Algorithm 2.

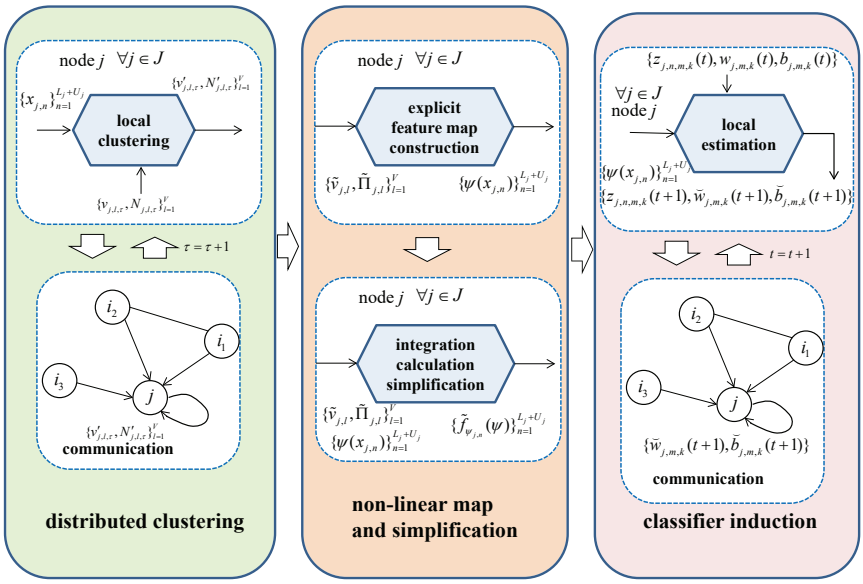



**Figure 4.** The flowchart of the dSMUDC algorithm.

**Algorithm 2** dSMUDC algorithm

**Require:** Collect uncertainty dataset $\{\boldsymbol{x}_{j,n}, \boldsymbol{\Theta}_{j,n}\}_{n=1}^{L_j+U_j}$, and initialize $\boldsymbol{w}_{j,m,k}(0) = \mathbf{0}_{Vq}$ and $b_{j,m,k}(0) = 0$ for each node $j$.

Transform the original class labels via (9).
Construct the explicit mapping function $\{\boldsymbol{\psi}_{j,n}\}$ via Algorithm 1.
Calculate the mean vector $\boldsymbol{\delta}_{j,n}$ and the covariance matrix $\boldsymbol{\Lambda}_{j,n}$ via (18) and (19).
**for** $\epsilon = 0, \ldots, T-1$ **do**
  **for** $j \in \mathcal{J}$ **do**
    Exchange $\{z_{j,l,m,k}^{v}(\epsilon)\}_{l=1}^{V}$ with its neighboring nodes.
  **end for**
  **for** $j \in \mathcal{J}$ **do**
    Update the labeling confidences $\{z_{j,l,m,k}^{v}(\epsilon+1)\}_{l=1}^{V}$ via (32).
  **end for**
**end for**
**for** $t = 0, 1, \ldots$ **do**
  **for** $j \in \mathcal{J}$ **do**
    Update the intermediate labeling confidences $\{z_{j,n,m,k}(t+1)\}_{n=1}^{L_j+U_j}$ via (33).
    Compute $\breve{\boldsymbol{w}}_{j,m,k}(t+1)$ and $\breve{b}_{j,m,k}(t+1)$ via (34) and (36).
    Exchange $\breve{\boldsymbol{w}}_{j,m,k}(t+1)$ and $\breve{b}_{j,m,k}(t+1)$ to neighboring nodes.
  **end for**
  **for** $j \in \mathcal{J}$ **do**
    Compute $\boldsymbol{w}_{j,m,k}(t+1)$ and $b_{j,m,k}(t+1)$ via (35) and (37).
  **end for**
**end for**
Obtain the predictive labels $\boldsymbol{y}^{*'}$ in the transformed class space.
Return $\boldsymbol{y}^{*}$ via decoding produce (38).

### *3.7. Performance Analysis*

In this subsection, we evaluate the convergence and complexity of the proposed algorithm.

To conduct the subsequent convergence investigation, several common assumptions regarding DL methods must be given at first.

**Assumption 1.** *Considering a connected network $\mathcal{G}$, the cooperative coefficient matrix $C$, whose elements $C_{ji} = c_{ji}$ are determined by the Metropolis rule [31], satisfies the following two conditions:*

*(1)* $C\mathbf{1}_J = \mathbf{1}_J$, $\mathbf{1}_J^T C = \mathbf{1}_J^T$, *(2) spectral norm* $\rho(C - \frac{1}{J}\mathbf{1}_J\mathbf{1}_J^T) \leq 1$.

**Theorem 1.** *If the above assumption holds, then we have* $\lim_{t\to\infty} |z_{j,n,m,k}(t) - z^*_{j,n,m,k}| = 0$, $\lim_{t\to\infty} ||\boldsymbol{w}_{j,m,k}(t) - \boldsymbol{w}^*_{j,m,k}|| = 0$ *and* $\lim_{t\to\infty} |b_{j,m,k}(t) - b^*_{j,m,k}| = 0$, *when t tends to* $\infty$, *where* $z^*_{j,n,m,k}$, $\boldsymbol{w}^*_{j,m,k}$, *and* $b^*_{j,m,k}$ *denote their corresponding optimal values.*

Theorem 1 can be demonstrated in the literature [17], hence we cannot provide the detailed proof here. Theorem 1 shows that with a connected network, all nodes can obtain the global optimal classifier via sufficient iterations, demonstrating the theoretical efficiency of DL.

Additionally, to evaluate the complexity of the dSMUDC method, we calculated two key metrics: the volume of computations required at each node in each iteration, and the number of variables that need to be exchanged between each node and its neighboring nodes. Table 1 summarizes the number of addition operations (AO) and multiplication operations (MO) required for the proposed method, with detailed breakdowns of these two computational metrics per iteration and per node across four key steps of the algorithm: the derivation of reproduction vectors, the construction of explicit feature maps, the simplification of integral calculations, and the induction of the classifier.

**Table 1.** The MOs and AOs of the dSMUDC algorithm per iteration per node $j$.

| | | |
|---|---|---|
| $\{\tilde{v}_{j,l}\}_l$ | MO | $VD(\lvert\mathcal{B}_j\rvert + 3\rvert)$ |
| | AO | $VD(\lvert\mathcal{M}_{j,l}\rvert + \lvert\mathcal{B}_j\rvert)$ |
| $\{\boldsymbol{\psi}_{j,n}\}_n$ | MO | $(L_j + U_j)V(D + 2D^2 + q(D^4 + 2D^3 + 2D^2))$ |
| | AO | $(L_j + U_j)V(D^3 + \lvert\mathcal{M}_{j,l}\rvert D^2 + \lvert\mathcal{M}_{j,l}\rvert D + q(D^4 + 2D^3 + 2D^2))$ |
| $\{f_{\boldsymbol{\psi}_{j,n}}(\boldsymbol{\psi})\}$ | MO | $(L_j + U_j)Vq(4D^4 + 10D^3 + 8D^2)$ |
| | AO | $(L_j + U_j)Vq(4D^4 + 10D^3 + 8D^2)$ |
| $\{z^v_{j,l,m,k}\}$ | MO | $\sum_{m=1}^{M} K'_m TV(1 + \lvert\mathcal{B}_j\rvert)$ |
| | AO | $\sum_{m=1}^{M} K'_m TV(\lvert\mathcal{M}_{j,l}\rvert + \lvert\mathcal{B}_j\rvert)$ |
| $\{z_{j,n,m,k}\}$ | MO | $\sum_{m=1}^{M} K'_m(L_j + U_j)(2V^2q^2 + 2Vq + V + 8) + \sum_{m=1}^{M} K'_m L_j$ |
| | AO | $\sum_{m=1}^{M} K'_m(L_j + U_j)(2V^2q^2 + 2Vq + V + 5) + \sum_{m=1}^{M} K'_m L_j$ |
| $\{\boldsymbol{w}_{j,m,k}\}$ | MO | $\sum_{m=1}^{M} K'_m Vq(11(L_j + U_j) + 2Vq + \lvert\mathcal{B}_j\rvert + 3)$ |
| | AO | $\sum_{m=1}^{M} K'_m Vq(7(L_j + U_j) + 2Vq + \lvert\mathcal{B}_j\rvert + 3)$ |
| $\{b_{j,m,k}\}$ | MO | $\sum_{m=1}^{M} K'_m(V^2q^2 + 2Vq + 7(L_j + U_j) + \lvert\mathcal{B}_j\rvert + 2)$ |
| | AO | $\sum_{m=1}^{M} K'_m(V^2q^2 + 2Vq + 4(L_j + U_j) + \lvert\mathcal{B}_j\rvert + 1)$ |

From Table 1, we can notice that the computation complexity of the proposed algorithm depends on the number of reproduction vectors $V$, the value of the scaling coefficient $q$, and the number of one-hop neighbors $|\mathcal{B}_j|$, except for the characteristics of the dataset. Given that the quantity of one-hop neighbors $|\mathcal{B}_j|$ is moderate in practical networks, and provided that the values of $V$ and $q$ are acceptable, the computational complexity of the proposed dSMUDC algorithm can be controlled within an appropriate range.

Besides, during the process of obtaining reproduction vectors, a total of $VD$ scalars need to be exchanged among neighboring nodes at each iteration $\tau$. In the classifier induction process, at each iteration $t$, each node $j$ must communicate $\sum_{m=1}^{M} K'_m(MVq + M)$ scalars to $|\mathcal{B}_j|$ neighboring nodes. In general, provided that the values of $V$ and $q$ are acceptable, the communication cost of the proposed dSMUDC algorithm is moderate.

## 4. Experiments

In this section, we conduct multi-faceted validation of the performance of the proposed algorithm based on several existing multi-dimensional datasets. Detailed information about the datasets is provided in Table 2. Note that in Table 2, the notation ♯ denotes the set cardinality. Besides, in the column of the number of class labels for each dimension, if the number of class labels is the same across all dimensions, only one number is retained. Otherwise, the number of class labels for each dimension is listed in sequence.

**Table 2.** Detailed profiles of used datasets.

| Dataset | ♯ Training Exam. | ♯ Testing Exam. | ♯ Feature | ♯ Lab./Dim. |
|---|---|---|---|---|
| Flare | 2580 | 650 | 3 | 3, 4, 2 |
| Cal500 | 450 | 52 | 10 | 2 |
| Jura | 2870 | 720 | 2 | 4, 5 |
| Music | 530 | 61 | 6 | 2 |
| Song | 3140 | 785 | 3 | 3 |
| WQ | 950 | 110 | 14 | 4 |
| Belae | 1740 | 190 | 5 | 5 |

To facilitate the reproduction of the following experiments, several necessary explanations are provided. This experiment involves a randomly generated distributed network with 10 nodes and 23 links. The training data samples are randomly assigned to the 10 nodes. In the subsequent trials, we perform 50 separate Monte Carlo cross-validation simulations. In each Monte Carlo simulation, the training data is randomly divided into 10 folds, with 9 folds designated as training data and the remaining fold utilized for testing data.

In order to simulate this kind of uncertain data uncertainty, this work add Gaussian white noise with a mean of zero and a standard deviation of $0.25eA_d$ to the data, which serves as the uncertainty distribution for this attribute. Here, $A_d$ denotes the value range of feature $x_d$ across the entire dataset.

Besides, to test the performance variation of the proposed algorithm under different proportions of labeled uncertain data, we define a measurement metric called Labeled uncertain Data to total Ratio (LADR), which describes the ratio of the labeled uncertain data in the whole dataset.

To evaluate the algorithm's performance from multiple perspectives, we utilize commonly used performance metrics in multi-dimensional classification algorithms, including Hamming loss, exact match, and semi-exact match. Due to page limitations, please refer to the literature [11,13,14,17] for the specific definitions of the aforementioned metrics.

Firstly, to examine the sensitivity of the proposed algorithm, we investigate the influence of parameters $\lambda_A$, $\lambda_B$, $\lambda_C$, $\lambda_D$ and $\lambda_E$ on the hamming loss of the dSMUDC algorithm on the "Flare" dataset [32] in Figure 5. In the simulation experiments, we adjusted the value of one parameter while keeping all other parameters unchanged. This design ensures that any observed changes in Hamming loss can be attributed solely to the variation of the target parameter, eliminating potential interference from cross-parameter interactions. The simulation results in Figure 5 reveal a consistent trend in Hamming loss across all five parameters. As the value of a parameter increases from a low initial level, the Hamming loss first decreases gradually. Once the parameter enters a suitable range, the Hamming loss stabilizes and remains unchanged even as the parameter value varies slightly within this interval, indicating that the algorithm's classification accuracy is not sensitive to minor adjustments of these parameters when they are properly configured. However, when the parameter value continues to rise beyond this suit-

able range, the Hamming loss begins to increase noticeably. The simulation results in Figure 5 indicate that when parameters are configured within suitable ranges, the proposed algorithm's classification accuracy is not affected by the values of the parameters. Therefore, we can determine the appropriate designation of the weighted parameters as $\lambda_A \in [0.5, 5], \lambda_B \in [0.5, 5], \lambda_C \in [0.5, 5], \lambda_D \in [5 \times 10^{-3}, 0.05], \lambda_E \in [5 \times 10^{-3}, 0.05]$.

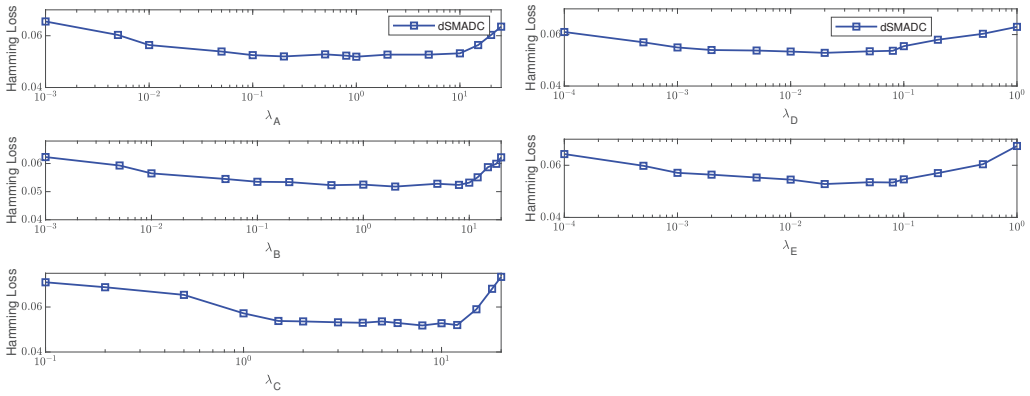



**Figure 5.** Hamming loss of dSMUDC algorithm under varying values of parameters $\lambda_A$, $\lambda_B$, $\lambda_C$, $\lambda_D$, and $\lambda_E$ on "Flare" dataset.

In addition, we also simulate the influence of the dimension of the explicit feature map (i.e., the product of the number of reproduction vectors $V$ and the number of scaling coefficients $q$) on the classification accuracy of the proposed algorithm in Figure 6. The simulation results indicate that provided that the values of $V$ and $q$ are larger than 30 and 4, respectively, good learning performance can be obtained. Considering that increasing the dimension may add to the burden on the computational system, $V$ and $q$ are appropriately adjusted to be no greater than 30 and 4, respectively.

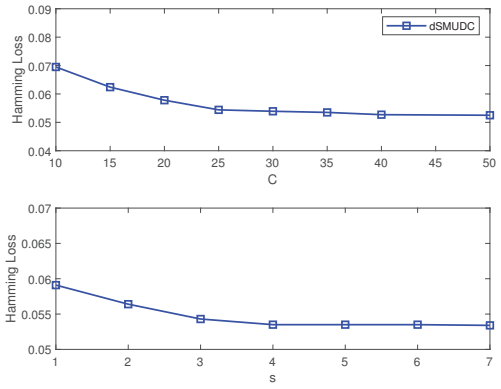



**Figure 6.** Hamming loss of dSMUDC algorithm under varying numbers of reproduction vectors $V$ and scaling coefficients $q$ on "Flare" dataset.

Furthermore, to evaluate the proposed algorithm's performance under different degrees of data uncertainty, we tested the variation of Hamming loss with different $e$ values on the Flare, Cal500, Jura, and Music datasets [11,32] in Figure 7. The experimental results show that the Hamming loss of our algorithm increases slightly as $e$ rises. Specifically, when $e$ increases from 0.03 to 0.15, the Hamming loss rises by less than 0.01. This indicates that increased data uncertainty has a certain negative impact on the algorithm's performance, but this impact is limited. When $e$ exceeds 0.15, the Hamming loss exhibits a more noticeable increase compared to the 0.03–0.15 range, yet the magnitude of this growth remains controllable and does not lead to sharp performance degradation. Collectively, these results

confirm that as long as data uncertainty is constrained within a reasonable range, the performance damage caused by uncertainty is limited, and the algorithm maintains stable and reliable prediction capabilities, verifying its robustness against moderate to moderately high levels of data uncertainty.

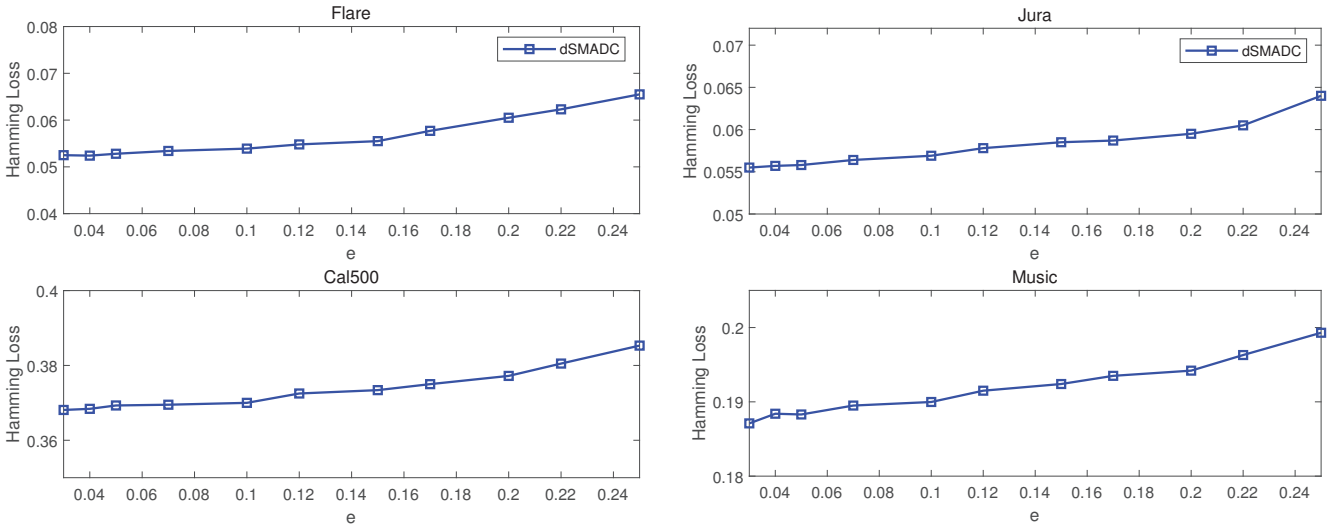



**Figure 7.** Hamming loss of dSMUDC algorithm under varying parameter $e$ on "Flare", "Cal500", "Jura" and "Music" datasets.

Additionally, to further highlight the advantage of our proposed algorithm, we testify to its classification accuracy on the 7 datasets, including the Flare, Cal500, Jura, Music, Song, WQ, and Belae datasets [11,12,32–34]. Besides, three evaluation metrics (hamming loss, exact match and semi-exact match) of the other comparison algorithms, including dS$^2$PMDL [17], dS$^2$MLL [2], PLEM [15], DLEM [14] and KARM [11], are also investigated. All the simulation results are presented in Tables 3–5. Furthermore, we also tested the performance of the centralized version of the SMADC algorithm (called cSMUDC), with its results serving as a baseline reference.

**Table 3.** Hamming loss of different algorithms versus LADR on MDC datasets (Bold entity here represents the lowest hamming loss in that case).

| Dataset | Hamming Loss | | | | | | |
|---|---|---|---|---|---|---|---|
| | **Flare** | **Cal500** | **Jura** | **Music** | **Song** | **WQ** | **Belae** |
| LADR | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| dSMUDC | **0.0505** | 0.3685 | 0.0564 | 0.1915 | 0.1265 | 0.3512 | 0.5761 |
| cSMUDC | 0.0507 | **0.3656** | **0.0551** | **0.1907** | **0.1236** | **0.3503** | **0.5742** |
| dS$^2$PMDL | 0.0517 | 0.3715 | 0.0627 | 0.1932 | 0.1287 | 0.3535 | 0.5811 |
| dS$^2$MLL | 0.0564 | 0.3807 | 0.0735 | 0.1973 | 0.1437 | 0.3668 | 0.5792 |
| PLEM | 0.0526 | 0.3747 | 0.0703 | 0.1942 | 0.1340 | 0.3557 | 0.5770 |
| DLEM | 0.0513 | 0.3705 | 0.0727 | 0.1948 | 0.1308 | 0.3573 | 0.5754 |
| KARM | 0.0557 | 0.3819 | 0.0792 | 0.1978 | 0.1447 | 0.3685 | 0.5805 |

The experimental results in Table 3 illustrate the Hamming loss of various algorithms under an LADR value of 0.5 across seven MDC datasets. Among all compared algorithms, cSMUDC achieves the lowest Hamming loss on most datasets, including Cal500, Music, Song, WQ, and Belae. Besides, the dSMUDC algorithm performs competitively, ranking second on most datasets and achieving the minimum Hamming loss on the Flare dataset. Furthermore, the dS$^2$MLL and KARM algorithms consistently exhibit higher Hamming loss values, particularly on the Jura and Song datasets.

**Table 4.** Exact match of different algorithms versus LADR on MDC datasets (Bold entity here represents the highest exact match in that case).

| Dataset | Exact Match | | | | | | |
|---|---|---|---|---|---|---|---|
| | **Flare** | **Cal500** | **Jura** | **Music** | **Song** | **WQ** | **Belae** |
| LADR | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| dSMUDC | 0.8248 | 0.0180 | 0.7726 | **0.3326** | 0.5380 | 0.0096 | 0.0287 |
| cSMUDC | **0.8267** | **0.0202** | **0.7761** | 0.3318 | **0.5412** | **0.0098** | **0.0298** |
| dS$^2$PMDL | 0.8218 | 0.0120 | 0.7552 | 0.3286 | 0.5302 | 0.0076 | 0.0265 |
| dS$^2$MLL | 0.8070 | 0.0105 | 0.7378 | 0.3184 | 0.5124 | 0.0072 | 0.0211 |
| PLEM | 0.8172 | 0.0135 | 0.7450 | 0.3224 | 0.5280 | 0.0092 | 0.0247 |
| DLEM | 0.8203 | 0.0146 | 0.7423 | 0.3240 | 0.5271 | 0.0085 | 0.0252 |
| KARM | 0.8184 | 0.0122 | 0.7148 | 0.3180 | 0.5144 | 0.0070 | 0.0202 |

The experimental results in Table 4 present the exact match scores of various algorithms under an LADR value of 0.5 across seven MDC datasets. By observing Table 4, we can notice that the cSMUDC and dSMUDC algorithms demonstrate superior performance on most datasets. The cSMUDC algorithm achieves the highest exact match scores on Flare, Cal500, Jura, Song, WQ, and Belae. Besides, dSMUDC exhibits comparable performance, securing the second-highest exact match scores across the majority of datasets. Algorithms including dS$^2$MLL and KARM consistently show the lowest exact match scores. This is particularly evident on Cal500, WQ, and Belae.

**Table 5.** Semi-exact match of different algorithms versus LADR on MDC datasets (Bold entity here represents the highest semi-exact match in that case).

| Dataset | Semi-Exact Match | | | | | | |
|---|---|---|---|---|---|---|---|
| | **Flare** | **Cal500** | **Jura** | **Music** | **Song** | **WQ** | **Belae** |
| LADR | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| dSMUDC | 0.9564 | 0.0727 | 0.9703 | 0.6760 | 0.9257 | **0.0468** | 0.1454 |
| cSMUDC | **0.9578** | **0.0748** | **0.9722** | **0.6796** | **0.9275** | 0.0466 | **0.1470** |
| dS$^2$PMDL | 0.9532 | 0.0716 | 0.9610 | 0.6693 | 0.9216 | 0.0446 | 0.1416 |
| dS$^2$MLL | 0.9412 | 0.0631 | 0.9506 | 0.6532 | 0.9130 | 0.0389 | 0.1265 |
| PLEM | 0.9501 | 0.0688 | 0.9549 | 0.6606 | 0.9178 | 0.0426 | 0.1355 |
| DLEM | 0.9516 | 0.0712 | 0.9627 | 0.6602 | 0.9163 | 0.0432 | 0.1381 |
| KARM | 0.9282 | 0.0653 | 0.9423 | 0.6505 | 0.9115 | 0.0392 | 0.1272 |

Table 5 presents the semi-exact match performance of various algorithms for MDC across seven datasets under a LADR value of 0.5. From the simulation results, we can find that cSMUDC outperforms others on five datasets: Flare, Cal500, Jura, Music, and Belae. Besides, dSMUDC remains highly competitive, achieving the optimal semi-exact match score on the Song dataset and ranking second across most other datasets. In contrast, dS$^2$MLL and KARM consistently demonstrate the lowest semi-exact match values. This is most pronounced on WQ, Belae, and Flare.

In summary, the experimental results presented in Tables 3–5 are similar, leading to the following key conclusions:

(1) The dS$^2$MLL algorithm and KARM algorithm perform relatively poorly among all methods. A possible reason is that the dS$^2$MLL algorithm, as a distributed multi-label learning algorithm, cannot effectively handle label correlations in heterogeneous multi-dimensional class spaces. The KARM algorithm, on the other hand, uses a classifier trained via the KNN method to perform initial data classification and incorporates the classification results into feature vectors as augmented information. Although the KARM

algorithm can address the training issue of multi-dimensional classifiers to a certain extent, its performance heavily relies on the selection of the number of neighboring data, resulting in weak generalizability.

(2) The DLEM and PLEM algorithms outperform the aforementioned KARM and dS$^2$MLL algorithms since they effectively explore label correlations in heterogeneous multi-dimensional class spaces. Despite this advantage, both the DLEM and PLEM algorithms are constrained by their inability to effectively handle unlabeled data and uncertain data. Consequently, their performance is inferior to that of our proposed method.

(3) The dS$^2$PMDL algorithm performs well because it can simultaneously leverage information from both labeled and unlabeled data and exploit the label correlations of multi-dimensional classes by learning subspace. However, in this experiment, the training data has a certain level of uncertainty, and the dS$^2$PMDL algorithm lacks the ability to characterize this uncertainty. Therefore, its performance is weaker than that of our proposed algorithm.

(4) Our proposed dSMUDC outperforms all comparative algorithms, signifying its efficacy in label recovery, data uncertainty exploitation, and classifier induction.

To further demonstrate the performance differences between these comparison algorithms, we use the Friedman test [35]. At a significance level of 0.05, the critical value is equivalent to 2.36. At this point, we can calculate the value of the Friedman statistic in terms of three metrics, please see Table 6. The Friedman test statistics are significantly greater than the critical value. Therefore, we reject the null hypothesis that there is no significant performance difference among these comparison algorithms.

**Table 6.** Summary of the Friedman statistics $F_F$ and the critical value in teams of Hamming loss, Exact match and Semi-exact match.

| **Metric** | $F_F$ | **Critical Value** ($\alpha = 0.05$) |
|---|---|---|
| Hamming loss | 36.50 | |
| Exact match | 21.92 | 2.36 |
| Semi-exact match | 46.87 | |

Furthermore, we use the Bonferroni-Dunn test [35] to testify whether there is a significant performance difference between a pair of comparison algorithms. Here, the cSMUDC is assumed to be the controlled algorithm. For the Bonferroni-Dunn test, at a significance level of 0.05, a schematic diagram is depicted in Figure 8. In this figure, for all algorithms whose average ranking difference with the controlled algorithm is less than one critical distance, we can justify that there is no significant performance difference between it and the controlled algorithm. Therefore, we connect it to the controlled algorithm with a red line. By observing Figure 8, we can see that the average performance ranking of the dSMUDC algorithm has significant performance improvements compared to the dS$^2$MLL and KARM comparison algorithms.

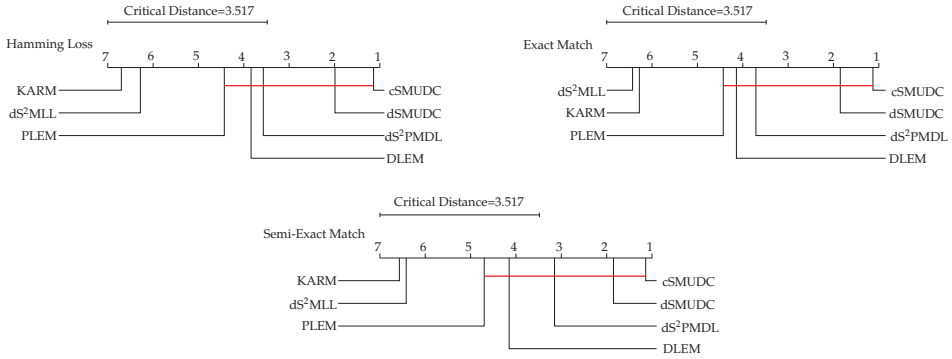



**Figure 8.** Comparison of cSMUDC (the control algorithms) in contrast to other comparing algorithms using the Bonferroni-Dunn test.

Finally, we test the performance of each comparative algorithm under different proportions of labeled uncertain data, and the experimental results are presented in the Figures 9–11. Figure 9 compares the Hamming loss of multiple algorithms across four datasets (Flare, Cal500, Jura and Music) as LADR varies, revealing clear performance gaps between algorithms. Across all four datasets, dSMUDC and cSMUDC achieve good learning performance with consistently lower Hamming loss than other methods. On the Jura dataset, dSMUDC/cSMUDC algorithms have 0.01–0.03 lower Hamming loss than dS$^2$PMDC/PLEM/DLEM algorithms, and 0.02–0.05 lower than dS$^2$MLL/KARM algorithms. On Flare, Cal500 and Music datasets, they lead dS$^2$PMDC/PLEM/DLEM algorithms by 0.005–0.01 and dS$^2$MLL/KARM algorithms by 0.01–0.02.

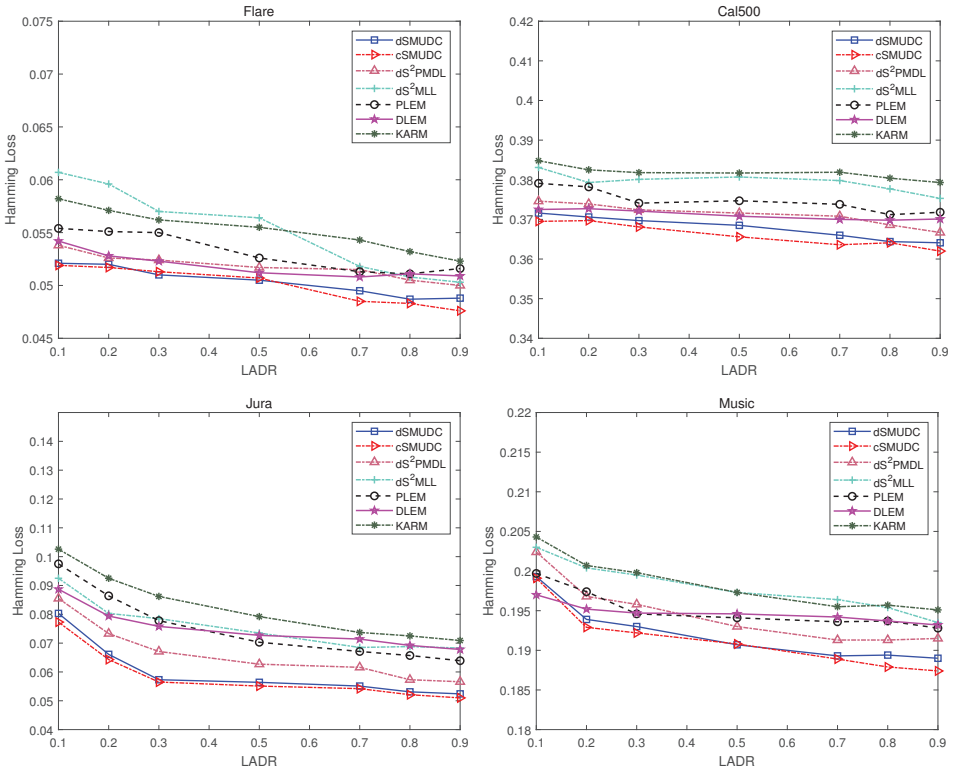



**Figure 9.** Hamming loss of different comparison algorithms versus the LADR on "Flare", "Cal500", "Jura" and "Music" datasets.

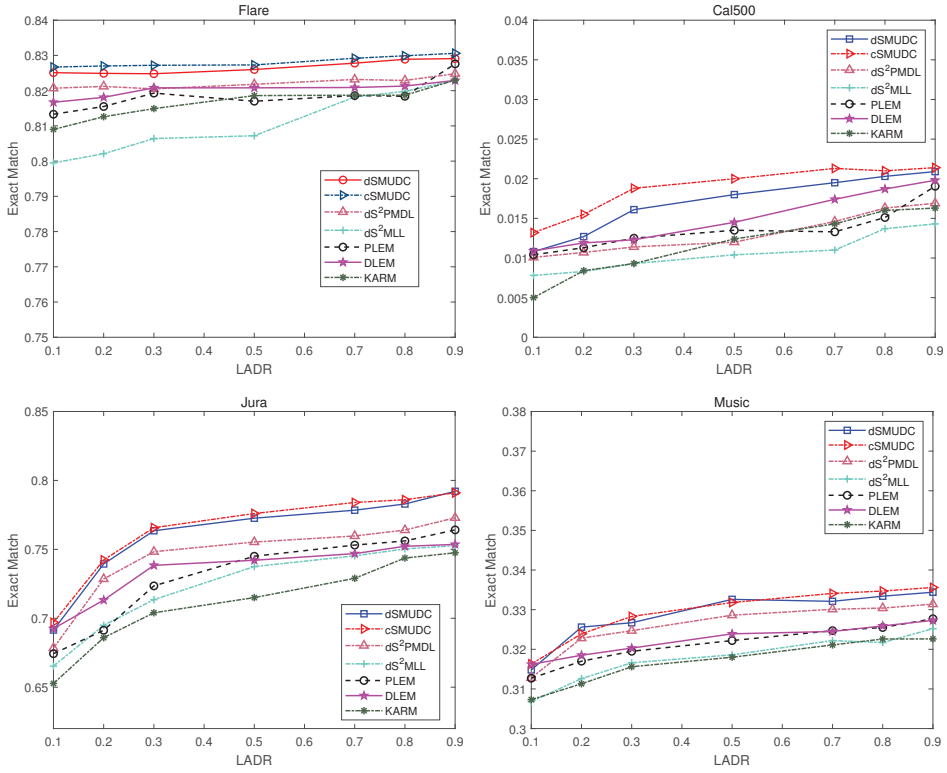



**Figure 10.** Exact match of different comparison algorithms versus the LADR on "Flare", "Cal500", "Jura" and "Music" datasets.

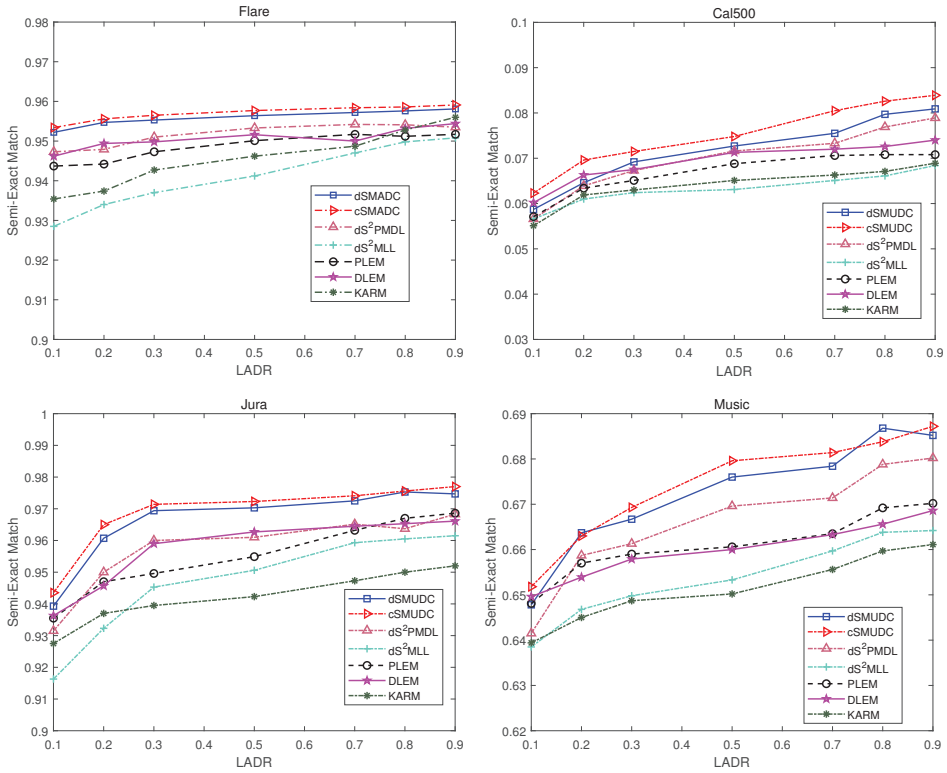



**Figure 11.** Semi-exact match of different comparison algorithms versus the LADR on "Flare", "Cal500", "Jura" and "Music" datasets.

Figure 10 evaluates the exact match performance of various algorithms (including dSMUDC, cSMUDC, dS²PMDL, etc.) across four datasets as LADR ranges from 0.1 to 0.9. It can be seen that the dSMUDC and cSMUDC perform the best across all scenarios,

and their exact match scores are consistently the highest, regardless of dataset or LADR value. Specifically, on Flare and Jura, they outperform dS$^2$PMDC/PLEM/DLEM methods by roughly 0.01–0.03, and dS$^2$MLL/KARM by 0.02–0.04. Even on Cal500 and Music, this pattern holds. The dSMUDC/cSMUDC algorithms maintain a 0.005–0.01 lead over dS$^2$PMDC/PLEM/DLEM algorithms, and a 0.01–0.03 advantage over dS$^2$MLL/KARM.

Figure 11 presents the semi-exact match performance of algorithms across four datasets as LADR varies (0.1–0.9), revealing distinct performance differences. We observe that cSMUDC and dSMUDC are the top performers overall: their scores remain consistently highest in most cases. On Jura and Music, they outperform mid-tier methods (e.g., PLEM, DLEM) by 0.01–0.02, and dS$^2$MLL/KARM by 0.02–0.05. This pattern persists on the Flare and Cal500 datasets as well.

Across Figures 9–11 (covering Hamming loss, exact match, and semi-exact match metrics), the following key conclusions emerge. The performance of all algorithms gradually improves as the LADR value increases. This indicates that the incorporation of more supervised information is beneficial to the training of classification models. Furthermore, we can also observe that the performance of our proposed algorithm is close to that of its corresponding centralized counterpart in most scenarios, and both are significantly superior to the other comparative algorithms. This superiority is particularly prominent when the amount of labeled data is relatively small.

## 5. Conclusions

In this paper, we have addressed the problem of distributed classification for partially labeled uncertain data and developed the dSMUDC algorithm. Within this proposed algorithm, we have employed the integral of the hinge loss of a sample over its uncertainty distribution as the misclassification loss for training samples, enabling effective utilization of the uncertainty of data distribution. Additionally, we have designed a mechanism for estimating labeling confidence to mitigate the negative impact on classification performance. Experimental results across multiple datasets have confirmed that the proposed algorithm can effectively tackle the challenge of classifying uncertain data in distributed scenarios.

The potential limitation of our algorithm mainly lies in two aspects. First, it assumes that the data across all nodes follows the same distribution, which may not align with real-world scenarios where data heterogeneity is common. Second, it presupposes that the uncertainty in the data originates from noise conforming to a Gaussian distribution. However, in practical applications, uncertainty can stem from diverse sources (e.g., measurement errors with non-Gaussian characteristics or incomplete data sampling) that do not follow Gaussian distribution. When facing such non-Gaussian uncertainty, the performance of the algorithm may be compromised.

Therefore, we aim to address these two issues in future research. On one hand, we intend to design a distributed personalized uncertain data classification algorithm to handle heterogeneously distributed uncertain data over a network, breaking the constraint of the identical data distribution assumption. On the other hand, we also aim to develop non-parametric methods that avoid strict distributional assumptions, which will help enhance the algorithm's adaptability to complex real-world uncertainty scenarios.

**Author Contributions:** Conceptualization, Z.X.; Methodology, Z.X.; Software, S.C.; Validation, S.C.; Formal analysis, Z.X.; Investigation, S.C.; Resources, S.C.; Data curation, S.C.; Writing—original draft, S.C.; Writing—review & editing, S.C.; Visualization, S.C.; Supervision, Z.X.; Project administration, S.C.; Funding acquisition, Z.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partly supported by the National Natural Science Foundation of China (Grant No. 62201398).

**Data Availability Statement:** All data generated or analyzed during this study are included in this published article.

**Conflicts of Interest:** Author Sicong Chen was employed by the company Kasco Signal 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. Shen, X.; Liu, Y. Privacy-preserving distributed estimation over multitask networks. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 1953–1965. [CrossRef]
2. Xu, Z.; Liu, Y.; Li, C. Distributed information theoretic semi-supervised learning for multi-label classification. *IEEE Trans. Cybern.* **2022**, *52*, 821–835. [CrossRef] [PubMed]
3. Lao, X.; Du, W.; Li, C. reproduction Distributed Estimation With Adaptive Combiner. *IEEE Trans. Signal Inf. Process. Netw.* **2022**, *8*, 187–200.
4. Verbraeken, J.; Wolting, M.; Katzy, J.; Kloppenburg, J.; Verbelen, T.; Rellermeyer, J.S. A survey on distributed machine learning. *ACM Comput. Surv.* **2020**, *53*, 1–33. [CrossRef]
5. Liu, M.; Yang, K.; Zhao, N.; Chen, Y.; Song, H.; Gong, F. Intelligent signal classification in industrial distributed wireless sensor networks based industrial internet of things. *IEEE Trans. Ind. Inf.* **2020**, *17*, 4946–4956. [CrossRef]
6. Aach, M.; Inanc, E.; Sarma, R.; Riedel, M.; Lintermann, A. Large scale performance analysis of distributed deep learning frameworks for convolutional neural networks. *J. Big Data* **2023**, *10*, 96. [CrossRef]
7. Naseh, D.; Abdollahpour, M.; Tarchi, D. Real-World Implementation and Performance Analysis of Distributed Learning Frameworks for 6G IoT Applications. *Information* **2024**, *15*, 190. [CrossRef]
8. Le, M.; Huynh-The, T.; Do-Duy, T.; Vu, H.; Hwang, W.-J.; Pham, Q.-V. Applications of Distributed Machine Learning for the Internet-of-Things: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2025**, *27*, 1053–1100. [CrossRef]
9. Naseh, D.; Bozorgchenani, A.; Shinde, S.S.; Tarchi, D. Unified Distributed Machine Learning for 6G Intelligent Transportation Systems: A Hierarchical Approach for Terrestrial and Non-Terrestrial Networks. *Network* **2025**, *5*, 41. [CrossRef]
10. Read, J.; Bielza, C.; Larranaga, P. Multi-dimensional classification with super-classes. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 1720–1733. [CrossRef]
11. Jia, B.-B.; Zhang, M.-L. Multi-dimensional classification via kNN feature augmentation. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 3975–3982.
12. Jia, B.-B.; Zhang, M.-L. Multi-dimensional classification via stacked dependency exploitation. *Sci. China Inf. Sci.* **2020**, *63*, 222102. [CrossRef]
13. Jia, B.-B.; Zhang, M.-L. Decomposition-based classifier chains for multi-dimensional classification. *IEEE Trans. Artif. Intell.* **2022**, *3*, 176–191. [CrossRef]
14. Jia, B.-B.; Zhang, M.-L. Multi-dimensional classification via decomposed label encoding. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 1844–1856. [CrossRef]
15. Tang, J.; Chen, W.; Wang, K.; Zhang, Y.; Liang, D. Probability-based label enhancement fro multi-dimensional classification. *Inf. Sci.* **2024**, *653*, 119790. [CrossRef]
16. Xu, Z.; Chen, S. Distributed partial label multi-dimensional classification via label space decomposition. *Electronics* **2025**, *14*, 2623. [CrossRef]
17. Xu, Z.; Chen, W. Distributed semi-supervised partial multi-dimensional learning via subspace learning. *Complex Intell. Syst.* **2025**, *11*, 318. [CrossRef]
18. Tsang, S.; Kao, B.; Yip, K.Y.; Ho, W.S.; Lee, S.D. Decision trees for uncertain data. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 64–78. [CrossRef]
19. Ozdemir, S.; Xiao, Y. Secure data aggregation in wireless sensor networks: A comprehensive overview. *Comput. Netw.* **2009**, *53*, 2022–2037. [CrossRef]
20. Srisooksai, T.; Keamarungsi, K.; Lamsrichan, P.; Araki, K. Practical data compression in wireless sensor networks: A survey. *J. Netw. Comput. Appl.* **2012**, *35*, 37–59. [CrossRef]
21. Aleroud, A.; Yang, F.; Pallaprolu, S.C.; Chen, Z.; Karabatis, G. Anonymization of network traces data through condensation based differential privacy. *Digit. Threat. Res. Pract.* **2021**, *2*, 1–23. [CrossRef]
22. Zhang, W.; Stella, X.Y.; Teng, S.H. Power SVM: Generalization with exemplar classification uncertainty. In Proceedings of the 2012 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 16–21 June 2012; pp. 2144–2151.
23. Liu, B.; Xiao, Y.; Cao, L.; Deng, F. Svdd-based outlier detection on uncertain data. *Knowl. Inf. Syst.* **2013**, *34*, 597–618. [CrossRef]

24. Dredze, M.; Crammer, K.; Pereira, F. Confidence-weighted linear classification. In Proceeding of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 June 2008; pp. 264–271.
25. Liu, B.; Xiao, Y.; Philip, S.Y.; Cao, L.; Zhang, Y.; Hao, Z. Uncertain one-class learning and concept summarization learning on uncertain data streams. *IEEE Trans. Knowl. Data Eng.* **2012**, *26*, 468–484. [CrossRef]
26. Yan, X.; Luo, Q.; Sun, J.; Luo, Z.; Chen, Y. Online dynamic working-state recognition through uncertain data classification. *Inf. Sci.* **2023**, *555*, 1–16. [CrossRef]
27. Jiang, B.; Pei, J. Outlier detection on uncertain data: Objects, instances, and inferences. In Proceedings of the 27th IEEE International Conference on Data Engineering, Hannover, Germany, 11–16 April 2011; pp. 422–433.
28. Tzelepis, C.; Mezaris, V.; Patras, I. Linear maximum margin classifier for learning from uncertain data. *IEEE Trans. Patt. Anal. Mach. Intell.* **2018**, *40*, 2948–2962. [CrossRef] [PubMed]
29. Pang, J.; Pu, X.; Li, C. A hybrid algorithm incorporating vector quantization and one-class support vector machine for industrial anomaly detection. *IEEE Trans. Ind. Inf.* **2022**, *18*, 8786–8796. [CrossRef]
30. Li, C.; Luo, Y. Distributed vector quantization over sensor network. *Int. J. Dist. Sensor Netw.* **2014**, *10*, 189619. [CrossRef]
31. Wang, S.; Li, C. Distributed stochastic algorithm for global optimization in networked system. *J. Optim. Theory Appl.* **2018**, *179*, 1001–1007. [CrossRef]
32. Jia, B.-B.; Zhang, M.-L. Maximum margin multi-dimensional classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 7185–7198. [CrossRef]
33. Aggarwal, C.C.; Yu, P.S. A condensation approach to privacy preserving data mining. In Proceedings of the 9th International Conference on Extending Database Technology, Heraklion, Crete, Greece, 14–18 March 2004; pp. 183–199.
34. Bielza, C.; Li, G.; Larranaga, P. Multi-dimensional classification with Bayesian networks. *Int. J. Approx. Reason.* **2011**, *52*, 705–727. [CrossRef]
35. Demsar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.

*electronics*

MDPI

*Article*

# SMAD: Semi-Supervised Android Malware Detection via Consistency on Fine-Grained Spatial Representations

**Suchul Lee and Seokmin Han ***

Department of Railroad Data Science, Korea National University of Transportation, Uiwang-si 16106, Gyeonggi-do, Republic of Korea; sclee@ut.ac.kr
* Correspondence: seokmin.han@ut.ac.kr

**Abstract:** Malware analytics suffer from scarce, delayed, and privacy-constrained labels, limiting fully supervised detection and hampering responsiveness to zero-day threats. We propose SMAD, a Semi-supervised Android Malicious App Detector that integrates a segmentation-oriented backbone—to extract pixel-level, multi-scale features from APK imagery—with a dual-branch consistency objective that enforces predictive agreement between two parallel branches on the same image. We evaluate SMAD on CICMalDroid2020 under label budgets of 0.5, 0.25, and 0.125 and show that it achieves higher accuracy, macro-precision, macro-recall, and macro-F1 with smoother learning curves than supervised training, a recursive pseudo-labeling baseline, a FixMatch baseline, and a confidence-thresholded consistency ablation. A backbone ablation (replacing the dense encoder with WideResNet) indicates that pixel-level, multi-scale features under agreement contribute substantially to these gains. We observe a coverage–precision trade-off: hard confidence gating filters noise but lowers early-training performance, whereas enforcing consistency on dense, pixel-level representations yields sustained label-efficiency gains for image-based malware detection. Consequently, SMAD offers a practical path to high-utility detection under tight labeling budgets—a setting common in real-world security applications.

**Keywords:** semi-supervised learning (SSL); Android malware detection; consistency regularization; segmentation-based pixel-level features; APK-to-image representation

## 1. Introduction

Advances in artificial intelligence (AI), propelled by information technology (IT), have profoundly reshaped numerous domains. AI has transformed information systems through heightened automation, predictive capability, decision accuracy, and efficiency, enabling widespread innovations in service delivery across applications such as large-scale language models (LLMs) [1], healthcare [2], edge computing [3], smart cities [4], and cybersecurity [5].

AI methods increasingly learn fluid decision boundaries by modeling underlying semantics rather than relying on surface terms in classification problems [5]. In cybersecurity, this boundary typically separates malicious from benign software and is learned using machine learning (ML) paradigms, including supervised, unsupervised, and reinforcement learning. Supervised learning fits labeled inputs to designated outputs [5], whereas unsupervised learning discovers latent structure in unlabeled data, grouping samples without prior categories [6].

Within cybersecurity, AI-based malware detection is increasingly employed to advance automated identification beyond traditional static and dynamic analyses. Yet these approaches often face accuracy limitations—most notably high false-positive and false-negative rates—that impede practical deployment. Consequently, supervised learning

remains a preferred strategy for known threats because it explicitly models input–label relationships and delivers more reliable performance [7,8]. Empirical studies indicate that supervised models trained on well-labeled datasets outperform unsupervised methods on structured detection tasks such as malware classification and intrusion detection [9].

However, the effectiveness of supervised learning is tightly coupled to access to large, high-quality labeled corpora—resources that are especially difficult to obtain in security contexts. Label creation demands expert-level effort, including reverse engineering, behavioral profiling, and the curation of threat intelligence [10]. The continual emergence of zero-day malware further complicates labeling and detection [11]; previously unseen threats often evade conventional defenses and cannot be accurately labeled at the time of appearance, undermining methods that rely on known classes. Privacy and confidentiality constraints also restrict data sharing, further hampering the construction of labeled datasets.

Within this landscape, semi-supervised learning (SSL) has emerged as a pragmatic response to label scarcity in security and privacy (S&P). Canonical methods—Mean Teacher [12], MixMatch [13], ReMixMatch [14], UDA [15], FixMatch [16], and large-scale self-training via Noisy Student [17]—operationalize consistency regularization and pseudo-labeling at scale, while highlighting evaluation pitfalls and confirmation-bias risks [18,19]. In cybersecurity, SSL has delivered gains across phishing, malware, intrusion, and encrypted-traffic detection [20]; representative instances include SSL-based Android malware detection leveraging labeled and unlabeled samples [21], multimodal frameworks that combine Gated Recurrent Units (GRUs) and Graph Convolutional Networks (GCNs) for encrypted traffic under limited labels [22], continual SSL that adapts to evolving malware without full retraining [23], and retrieval-augmented few-shot classification (MalMixer) [24]. These results motivate our focus on dense, pixel-level representations and dual-branch consistency for image-based Android malware detection. A concise review of these SSL foundations and their security applications appears in Section 2.

Building on these insights, we present SMAD (Semi-supervised Android Malicious App Detector), a framework designed to mitigate the dependence on costly annotations while improving resilience to previously unseen (zero-day) malware. SMAD integrates a segmentation-oriented backbone that extracts pixel-level, multi-scale features from APK imagery with a dual-branch consistency objective that enforces predictive agreement between two parallel branches on the same image. This combination leverages the structure present in abundant unlabeled telemetry to stabilize optimization under label scarcity and to enhance generalization when family distributions drift.

We evaluate SMAD on CICMalDroid2020 under label budgets $\texttt{rol} \in \{0.5, 0.25, 0.125\}$ and observe consistent gains in accuracy, macro-precision, macro-recall, and macro-F1 over supervised training and a recursive pseudo-labeling baseline, with smoother learning curves across epochs. We also include a controlled comparison to FixMatch [16] under the same schedule and a backbone ablation (replacing our backbone with WideResNet) that attributes a substantial portion of the gains to dense, pixel-level multi-scale features under agreement. An ablation with a fixed confidence gate clarifies the coverage–precision trade-off under a cold start.

This paper makes three contributions:

1. A semi-supervised detector with segmentation-derived pixel-level features. We introduce SMAD, which couples dual-branch consistency with a segmentation-oriented backbone that extracts dense, pixel-level, multi-scale representations from APK imagery via an Atrous Spatial Pyramid Pooling (ASPP) module. These features combined with agreement between parallel branches provide higher-SNR unsupervised targets—improving early calibration, training stability, and label efficiency under label scarcity.

2. Improved robustness to unknown behaviors. By enforcing agreement between two branches processing the same APK image, SMAD exhibits enhanced generalization to previously unseen malware families without additional expert labels.
3. Controlled and balanced evaluation. We conduct extensive experiments and report accuracy, macro-precision, macro-recall, and macro-F1 with mean $\pm$ std over three runs; compare against FixMatch under the same schedule; and perform a backbone ablation (dense encoder vs. WideResNet) to isolate encoder effects.

Section 2 reviews related work on semi-supervised security analytics and malware detection. Section 3 details the SMAD architecture, loss functions, and design rationales. Section 4 describes datasets, setup, and metrics. Section 5 reports results and discusses operational implications. Section 6 concludes.

## 2. Related Work

### *2.1. Motivation and Theoretical Foundations*

The increasingly complex and dynamic cyber-threat landscape renders purely supervised approaches impractical: collecting and annotating high-quality security data (e.g., malware corpora, encrypted traffic, intrusion logs) is costly, slow, and quickly outdated under zero-day threats and concept drift. Semi-supervised learning (SSL) addresses these constraints by leveraging abundant unlabeled data alongside minimal labels to sustain performance. A comprehensive survey [20] documents growing SSL adoption in phishing, network intrusion, web spam, and malware detection. Representative frameworks include Dapper [25], which attains near-supervised accuracy with $\sim$10% labels via pseudo-label propagation and automated hyperparameter selection, and SF-IDS [26], which couples pseudo-label filtering with hybrid losses to mitigate label scarcity and class imbalance.

Underpinning these successes are canonical SSL methods that operationalize consistency regularization and pseudo-labeling at scale: Mean Teacher stabilizes targets via weight-averaged teachers [12]; MixMatch unifies consistency and entropy minimization with mixup [13]; ReMixMatch augments this with distribution alignment and augmentation anchoring [14]; UDA enforces agreement under strong augmentations [15]; FixMatch combines confidence-thresholded pseudo-labels with weak/strong augmentation [16]; and Noisy Student demonstrates scalable self-training [17]. These developments rest on classical principles—entropy minimization and manifold regularization [27,28]—and are informed by cautions on evaluation protocol and confirmation-bias dynamics [18,19].

### *2.2. SSL for Malware and Encrypted-Traffic Detection*

SSL has been actively adapted to security subdomains with promising results. For Android malware, Memon et al. [21] implement a feature-based SSL model over permissions and API logs, achieving robust detection with limited labels. In encrypted traffic analysis, multimodal architectures that combine Gated Recurrent Units (GRUs) with Graph Convolutional Networks (GCNs) improve F1 under low-label regimes [22]. Continual SSL enables adaptation to evolving malware families without full retraining [23]. Retrieval-augmented SSL (MalMixer) supports few-shot malware family classification [24], while bidirectional normalizing-flow models reduce reliance on labeled anomalies [29]. Semi-supervised traffic clustering (e.g., SCOUT) can isolate malicious flows for downstream signature generation, and large-scale intrusion-detection pipelines report gains on CIC-DDoS2019 and UNSW-NB15 via self-training and co-training strategies [30].

### *2.3. Advanced SSL Architectures and Emerging Directions*

Recent directions underscore practicality and robustness in real deployments. Multi-stage designs like M3S-UPD deliver fine-grained encrypted-traffic classification and zero-

shot detection through continual learning [31]. Contrastive and multimodal pretraining further enhances generalization to unseen attack patterns in encrypted traffic [32]. Interpretable SSL systems for industrial cyber-attack detection improve transparency and trust [34**?** ]. In wireless-sensor-network (WSN) intrusion detection, pseudo-label-based SSL achieves high F1 in label-scarce settings [9]. Collectively, these advances chart a path toward robust, explainable, and deployable SSL for modern cybersecurity.

Within this landscape, image-based malware analytics are appealing because APK-to-image renderings expose narrow, local artifacts—padding bands, packing/obfuscation traces, and layout regularities—without hand-crafted features [35]. Patch-token pipelines (e.g., ViT) summarize content into coarse tokens and emphasize global relations [36], which can blur such artifacts; in contrast, segmentation-oriented encoders retain dense, multi-scale pixel-wise descriptors that preserve fine textures and local discontinuities [37], naturally aligning with consistency-style SSL objectives [12,16].

Following this rationale, we use a segmentation-oriented backbone to extract dense, multi-scale features from APK imagery and apply dual-branch consistency on the same image with a decoder-free, image-level classifier. Preserving pixel-wise detail and enforcing agreement at the artifact's spatial granularity stabilizes predictions under benign spatial/photometric shifts, providing a concise rationale for robustness to padding alignment changes and common obfuscation in APK-to-image renderings [35].

## 3. Semi-Supervised Android Malware Detection

In this study, we introduce SMAD (Semi-supervised Android Malicious App Detector), a semi-supervised learning (SSL) framework for detecting malicious Android applications. Rather than depending exclusively on large, fully labeled datasets, SMAD leverages unlabeled data to maintain high detection accuracy in environments where threat patterns evolve rapidly. By integrating a segmentation-oriented backbone for rich feature extraction with a two-branch consistency strategy for stable semi-supervised training, SMAD achieves strong generalization to both known and emerging attack types.

### *3.1. Architecture*

SMAD is built upon a modified DeepLabV3+ [37] backbone originally developed for semantic segmentation tasks. Conventional image classification networks employ a stack of convolutional layers followed by global pooling and fully connected layers. In contrast, DeepLabV3+ consists of a backbone feature extractor, an Atrous Spatial Pyramid Pooling (ASPP) module for multi-scale context aggregation, and a decoder module for pixel-wise segmentation. To use DeepLabV3+ for image-level classification rather than pixel-wise segmentation, we retain its multi-scale encoder unchanged and modify only the task-specific heads, as summarized below and illustrated in Figure 1.

Shared encoder (unchanged from DeepLabV3+). In the proposed scheme, we reuse the DeepLabV3+ encoder as is—a ResNet backbone followed by an ASPP module. ASPP comprises four branches (one $1 \times 1$ convolution and three $3 \times 3$ atrous convolutions with dilation rates 6, 12, and 18 at output stride 16), aggregating multi-scale context while preserving pixel-level detail. The encoder outputs a dense feature map that is shared by both SMAD's dual-branch heads; only the task-specific heads depart from the standard DeepLabV3+ pipeline.

Head change from DeepLabV3+ to the proposed classifier. Relative to the DeepLabV3+ segmentation head—which upsamples and refines features for pixel-wise prediction—the proposed classifier removes the decoder and adopts a lightweight path: global average pooling ($256 \rightarrow 1D$) $\rightarrow$ fully connected layer, producing image-level logits. This shifts from

spatial preservation (dense masks) to spatial collapse (a compact global representation) while retaining the encoder's multi-scale evidence.

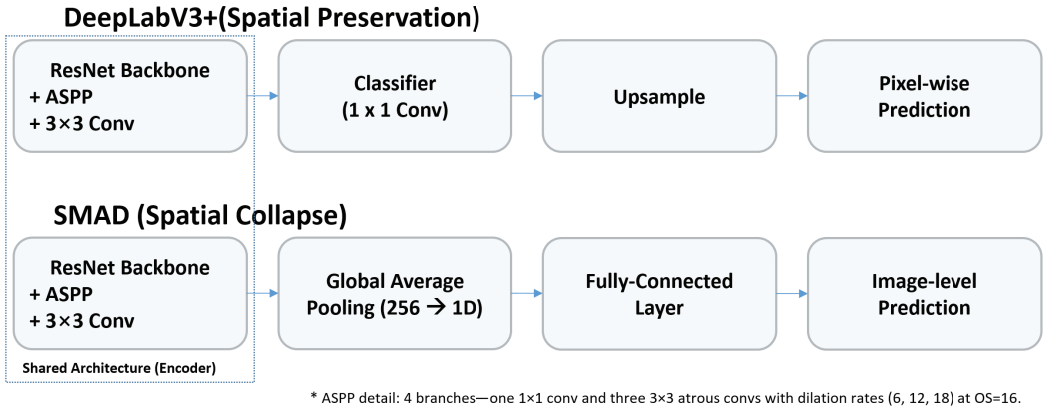



**Figure 1.** Architecture comparison between DeepLabV3+ and the proposed method: both share the same encoder (ResNet + ASPP), while the proposed classifier replaces the segmentation decoder with a global average pooling (256 → 1D) and a fully connected layer for image-level logits.

The motivation for adopting DeepLabV3+ is its ability to expand the receptive field and capture multi-scale context via atrous convolutions—capabilities uncommon in conventional classification models. This allows the network to fuse fine-grained local features with global contextual information, which is advantageous for classification tasks with substantial variation in object scale and spatial arrangement. In addition, reusing a segmentation-oriented backbone promotes dense feature representation learning, potentially yielding improved generalization over standard classification backbones.

Dual-branch layout and inference fusion. Figure 2 presents the SMAD network architecture. Given an image input $x$, a high-capacity backbone extracts multi-scale features that capture both global structure and fine-grained cues. The features are processed by two parallel semi-supervised branches on the same input; consistency between branches promotes better generalization.

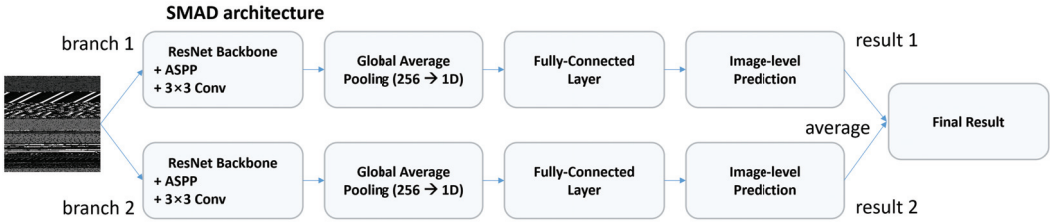



**Figure 2.** The network adopts a dual-branch semi-supervised architecture: two parallel branches process the same input, and inter-branch agreement is enforced between their predictions to promote robust generalization under limited labels.

Both branches are architecturally identical and route the shared encoder features directly to the same classifier head; we keep two instances solely to apply inter-branch consistency during training, and average their predictions at inference. The per-branch predictions are fused to produce the final decision. Training jointly minimizes a supervised loss $L_{sup}$ on labeled samples and a consistency loss $L_{con}$ on unlabeled samples, aligning the two branches' predictions. This design encourages representation learning via inter-branch agreement, mitigates overfitting under label scarcity, and yields robust performance against perturbations and evolving attack strategies, enabling reliable detection of both known threats and zero-days. The training objectives that enforce inter-branch agreement are mathematically defined in Section 3.2.

Although the two branches share the same architecture, we maintain the asymmetry needed for effective consistency regularization via two mechanisms: (i) distinct random initializations and (ii) explicitly decoupled parameters that are optimized independently by stochastic gradient descent (SGD). This induces immediate and sustained divergence in parameter space while preserving comparable capacity across branches, thereby ensuring that the inter-branch agreement term remains informative.

### *3.2. Training Objectives*

Using two parallel branches on the same input, the model aims to improve prediction stability and generalization. The training set is partitioned into labeled and unlabeled subsets.

Let $m = 1 \ldots M$ index images and $n = 1 \ldots N$ pixels. The model uses two branches, $i \in \{1, 2\}$. For labeled inputs, $y^l_{mn}$ denotes the one-hot ground truth at pixel $n$ of image $m$, and $\hat{y}^l_{mn,i}$ the class-probability prediction from branch $i$. For unlabeled inputs, $\hat{y}^u_{mn,i}$ denotes the prediction from branch $i$.

Let $C$ denote the number of classes with $c \in \{1, \ldots, C\}$ indexing classes. We use the standard pixel-wise categorical cross-entropy $\mathrm{L}_{ce}(\mathrm{p}, \mathrm{q}) = -\sum_{c=1}^{C} p_c \log(q_c)$; here, $\mathrm{p} = (p_c)$ and $\mathrm{q} = (q_c)$ are class-probability vectors of length $C$ (non-negative entries summing to 1). In implementation, $\mathrm{L}_{ce}$ is computed from logits for numerical stability (e.g., softmax-with-logits). For the labeled subset, ground-truth labels supervise the network via the supervised loss $L_{sup}$ defined as:

$$L_{sup,i} = \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N} \sum_{n=1}^{N} \mathrm{L}_{ce}(y^l_{mn}, \hat{y}^l_{mn,i}); \tag{1}$$

$$L_{sup} = \frac{1}{2} \cdot (\, L_{sup,1} + \, L_{sup,2}). \tag{2}$$

To improve training stability, we incorporate an additional consistency loss constraint [38], defined as follows:

$$L_{con} = \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N} \sum_{n=1}^{N} \mathrm{L}_{ce}(\hat{y}^u_{mn,1}, \hat{y}^u_{mn,2}). \tag{3}$$

Additionally, we can incorporate branch-specific confidence weighting into the consistency term (Equation (3)), using $c^{(1)}$ and $c^{(2)}$ for subnet 1 and subnet 2, respectively. For each unlabeled example $x$, the loss contributes only when both branches report confidence above a preset threshold $\tau$. Formally,

$$L_{\text{con\_thr}} = \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\big[c^{(1)}(x) > \tau \wedge c^{(2)}(x) > \tau\big] \; \mathrm{L}_{ce}(\hat{y}^u_{mn,1}, \hat{y}^u_{mn,2}), \tag{4}$$

where $\mathbf{1}[\cdot]$ is the indicator function. In what follows, we refer to this scheme as *SMAD-THR*. Unless stated otherwise, SMAD uses a DeepLabV3+ backbone. We also consider SMAD$^{\mathrm{W}}$, where "W" indicates a WideResNet backbone; all other components and training settings remain identical.

Combining the supervised loss constraint for labeled data and consistency loss for unlabeled data, the final loss constraint is defined as follows:

$$L_{final} = \lambda_{sup} \, L_{sup} + \lambda_{con} \, L_{con} \tag{5}$$

where $\lambda_{sup}$ and $\lambda_{con}$ are empirically tuned balancing coefficients. Unless otherwise stated, we set $\lambda_{sup} = 1$ and $\lambda_{con} = 1$ for all experiments. This constant-weight setting follows common SSL practice—e.g., FixMatch [16] uses a fixed unlabeled-loss weight $\lambda_u$ and reports that ramping this weight is unnecessary—so adopting 1:1 weighting avoids con-

founding from an additional hyperparameter while keeping comparisons fair across backbones and baselines. Exhaustive fine-tuning of the $\lambda$ coefficients is outside the scope of this work; our focus is on the method design and label-budget regime rather than hyperparameter optimization.

## 4. Experiments

### *4.1. Setup*

Datasets. We conduct experiments on the CICMalDroid 2020 dataset [39], a publicly available collection of Android Package Kit (APK) files—standard Android application bundles—compiled from sources such as VirusTotal and the Contagio blog (December 2017–December 2018).

The dataset comprises 17,341 applications labeled as Benign or one of four malware families (Adware, Banking, Mobile Riskware, SMS); in our image-rendered subset of 16,787 samples, the per-class counts are Benign 4039 (24.1%), Adware 1514 (9.0%), Banking 2505 (14.9%), SMS 4821 (28.7%), and Mobile Riskware 3908 (23.3%). We map APK bytes to a grayscale image using a simple row-major stream order (left-to-right, top-to-bottom); for the stream-order procedure and implementation details, see [40].

For evaluation, we perform a single class-stratified partition of the full corpus into an 8:1:1 training–validation–test split, preserving family proportions with a fixed random seed. All metrics are reported on the held-out test set; the validation split is used exclusively for model selection and early stopping. The semi-supervised label budget is parameterized by `rol` and applies only to the training split: for a given `rol`$\in \{0.5, 0.25, 0.125\}$, that fraction of the training samples is treated as labeled, while the remainder of the training data is available as unlabeled input to SSL methods (the validation and test sets are never used for unsupervised updates).

Input preprocessing. To standardize inputs and reduce overfitting, we apply a single, shared transform per image before it is fed to both branches: (i) random scaling with a factor $s \in [0.5, 2.0]$ while preserving aspect ratio (bilinear resampling), (ii) zero-padding if needed followed by a random square crop, (iii) random horizontal flip with probability 0.5, and (iv) uniform resize to $224 \times 224$.

Implementation Platform. PyTorch [41] is widely adopted in contemporary research owing to its dynamic computation graph, mature ecosystem and community support, and efficient, scalable Python implementation. Leveraging these properties, all experimental code for this study was implemented in PyTorch 2.9.0.

Performance metrics. We report accuracy together with macro-precision, macro-recall, and macro-F1 on the held-out test set. Macro-averaging computes each metric per class and then averages them with equal weight, mitigating skew from class imbalance.

Hardware & training details. All experiments were run on a single workstation with one NVIDIA RTX 4090 (24 GB VRAM), an Intel Core i7-13700KF CPU, and 64 GB RAM. Unless otherwise noted, we use stochastic gradient descent (SGD) as the optimizer, with a batch size of 4 and an initial learning rate of 0.005.

### *4.2. Compared Schemes*

We evaluate the following baselines and ablations alongside our semi-supervised consistency method.

Supervised only (*SUP*). Train only on the labeled subset with cross-entropy $L_{ce}$ (Section 3.2); unlabeled data are not used for training or model selection. This serves as a conventional lower-bound reference.

Recursive pseudo-labeling (*REC*). Train on labeled data, infer pseudo-labels on unlabeled data with the current model, accept all pseudo-labels (no confidence threshold),

and retrain on the union; the process may be repeated. This simple loop can accumulate label errors across iterations.

FixMatch. A popular semi-supervised baseline using confidence-thresholded pseudo-labels under weak/strong views. We adopt the same backbone (WideResNet), optimizer, and schedule as ours for a controlled comparison.

Confidence-threshold ablation (*SMAD-THR*). Our method with a fixed confidence gate inside the consistency term (Section 3.2) was used to study the effect of gating under limited labels.

Backbone ablation (SMAD$^{W}$). Replace the segmentation-oriented encoder with a WideResNet while keeping the SSL objective and schedule fixed. Motivated by the canonical FixMatch recipe, we use a WideResNet encoder here so that the ablation aligns with that backbone and supports apples-to-apples cross-method comparison. This isolates the encoder's contribution under identical training conditions.

## 5. Results

### *5.1. Overall Performance Across Label Ratios*

We evaluate all methods under varying label budgets on a fixed training pool. For a given `rol`, every method receives the identically labeled subset, while the remainder of the training set is treated as unlabeled data (consumable only by SSL methods). To provide a strong supervised reference point, we also report *SUP*($\texttt{rol} = 1$), trained on all available labels. Unless otherwise noted, all methods share the same backbone, optimizer, and training schedule for fairness. We report test accuracy (%) at every epoch over the entire training horizon. Results are visualized with per-`rol` breakdowns in Figures 3 and 4.

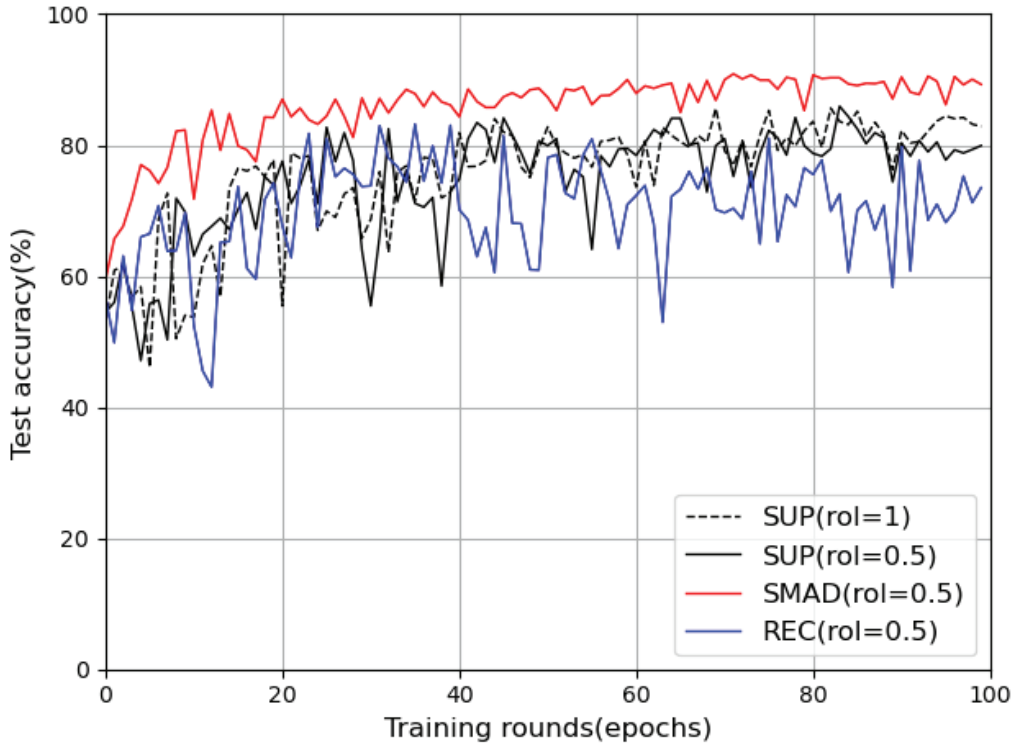
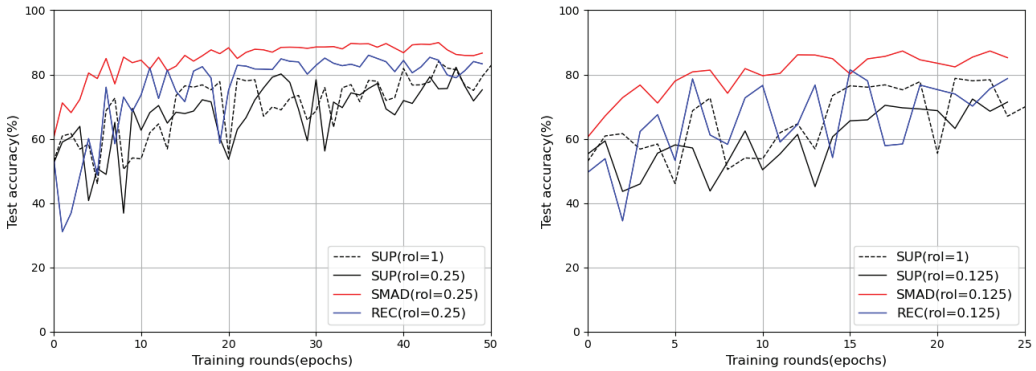



**Figure 3.** Test accuracy vs. epochs at `rol` = 0.5.




(**a**) `rol` = 0.25 (**b**) `rol` = 0.125

**Figure 4.** Side-by-side comparison under reduced label ratios.

Results at `rol` = 0.5 (Figure 3): With half of the labels available, SMAD rises rapidly to strong test performance and stabilizes above both supervised references and the pseudo-labeling variant; by the end of training, it matches or exceeds the full-label supervised reference while maintaining a smoother trajectory than *REC*. In contrast, *REC* attains intermediate accuracy but exhibits higher variance across epochs, consistent with noisier target signals. Beyond the generic benefits of consistency regularization, we attribute a substantive fraction of this margin to SMAD's segmentation-oriented backbone, which performs pixel-level feature extraction and aggregation with ASPP. The resulting dense, multi-scale representations fuse fine-grained local cues with global layout, yielding descriptors that are more invariant to APK packing/repackaging and obfuscation artifacts when binaries are rendered as images. Aggregating agreement over many spatial locations provides a higher signal-to-noise ratio for the unsupervised target, improves confidence calibration early in training, and dampens view-specific artifacts—mechanisms that plausibly account for the observed stability and the higher asymptote at `rol` = 0.5. Overall, Figure 3 indicates that, in this setting, unlabeled data—channeled through dual-branch consistency on dense features—yields measurable gains over purely supervised optimization on the same backbone and schedule.

Results at `rol` = 0.25 and `rol` = 0.125 (Figure 4). When the labeled budget is reduced to one quarter and one eighth, SMAD preserves a pronounced advantage over both the same-budget supervised baseline and the pseudo-labeling variant, while exhibiting accelerated convergence—it reaches a stable operating point in fewer epochs. Importantly, this stability is achieved at a slightly lower accuracy asymptote than in higher-label regimes, reflecting the diminished ceiling imposed by scarce labeled supervision. The method ordering is invariant across the training horizon: SMAD attains the highest terminal accuracy at both `rol` = 0.25 and `rol` = 0.125; purely supervised training plateaus earlier at a lower level, and the pseudo-labeling variant persistently trails.

The joint pattern of faster stabilization and modestly attenuated asymptotes is consistent with established semi-supervised learning principles. Cross-view consistency encourages decision boundaries to align with low-density regions of the data manifold, yielding strong regularization that accelerates optimization dynamics, whereas the limited volume of labeled evidence constrains the attainable peak performance [27]. Complementarily, entropy minimization on unlabeled instances sharpens posterior assignments without overfitting to the small labeled set—provided augmentations and targets are well calibrated—thereby sustaining margins over supervised training even when labels are sparse [28]. Empirically, contemporary consistency-based methods (e.g., MixMatch [13], ReMixMatch [14], and FixMatch [16]) report analogous behavior in low-label regimes—earlier convergence with a slightly reduced ceiling relative to richer supervi-

sion—while maintaining decisive gains over purely supervised and pseudo-labeling baselines. The curves in Figure 4 mirror these reports, evidencing that SMAD continues to leverage the unlabeled corpus effectively, converging rapidly yet retaining superior terminal accuracy under severe label scarcity.

Why consistency-based SSL outperforms supervised and pseudo-labeling baselines. Dual-view consistency imposes a prior agreement that encourages low-density separation and constrains function complexity, which is known to improve generalization under label scarcity; weight-averaged or consistency-target variants (e.g., Mean Teacher [12]) routinely outperform comparable supervised training with few labels. And modern confidence-aware formulations (e.g., FixMatch [16]) demonstrate that SSL can match or even surpass full-label supervised references in standard vision benchmarks [18]. By contrast, pure pseudo-labeling is susceptible to confirmation bias—early mistakes reinforce themselves—leading to oscillatory learning curves and inferior asymptotes unless additional regularization is introduced [19]. These mechanisms align with what we observe at `rol` = 0.5 in Figure 3.

Comparison with a popular SSL baseline (FixMatch [16]). To contextualize performance against a widely adopted SSL method, we include FixMatch [16] trained under the schedule and label splits. Table 1 summarizes accuracy, macro-precision, macro-recall, and macro-F1 (mean ± std over three runs) for SMAD and FixMatch under identical SSL setups. SMAD outperforms FixMatch consistently at all label ratios. For example, at `rol` = 0.5, SMAD improves accuracy by +3.2 pts (91.2 vs. 88.0) and F1 by +4.2 pts (90.3 vs. 86.1); at `rol` = 0.25, the gaps widen to +4.5 accuracy/+5.3 F1; and at `rol` = 0.125, to +8.3 accuracy/+9.2 F1. The precision/recall margins follow the same trend (e.g., recall +4.8, +6.1, and +10.5 pts for `rol` = 0.5, 0.25, and 0.125, respectively), indicating that SMAD maintains higher sensitivity without sacrificing precision as supervision decreases. These gains exceed the reported standard deviations, suggesting robust improvements rather than noise.

**Table 1.** Comparison with FixMatch across label ratios (values are mean ± std over three runs).

| **Method (`rol`)** | **Accuracy** | **Precision** | **Recall** | **F1** |
|---|---|---|---|---|
| SMAD (0.5) | $91.2\% \pm 0.5\%$ | $89.9\% \pm 0.6\%$ | $90.7\% \pm 0.6\%$ | $90.3\% \pm 0.5\%$ |
| SMAD (0.25) | $90.3\% \pm 0.4\%$ | $88.7\% \pm 0.7\%$ | $89.7\% \pm 0.6\%$ | $89.1\% \pm 0.5\%$ |
| SMAD (0.125) | $89.4\% \pm 0.6\%$ | $87.2\% \pm 0.8\%$ | $88.7\% \pm 0.5\%$ | $87.8\% \pm 0.7\%$ |
| FixMatch (0.5) | $88.0\% \pm 1.4\%$ | $86.5\% \pm 0.9\%$ | $85.9\% \pm 1.8\%$ | $86.1\% \pm 1.4\%$ |
| FixMatch (0.25) | $85.8\% \pm 1.1\%$ | $84.4\% \pm 0.9\%$ | $83.6\% \pm 0.9\%$ | $83.8\% \pm 0.9\%$ |
| FixMatch (0.125) | $81.1\% \pm 0.7\%$ | $79.6\% \pm 1.1\%$ | $78.2\% \pm 1.2\%$ | $78.6\% \pm 0.9\%$ |

Per-class accuracy at `rol` = 0.5. Per-class accuracy at `rol` = 0.5 shows that SMAD achieves 91.5/82.3/92.5/88.1/96.8(%), exceeding FixMatch (77.1/76.7/90.8/87.4/95.4) and SMAD$^{W}$ (87.7/78.0/93.6/86.1/95.5) across all classes.

### *5.2. Ablation Study*

Dense pixel-wise multi-scale vs. WideResNet. Table 2 isolates the effect of the encoder by swapping SMAD's segmentation-oriented backbone for a standard WideResNet while keeping the SSL objective and schedules fixed. The dense, pixel-wise multi-scale backbone yields higher accuracy/F1 at every budget, and the gap increases as labels become scarcer (e.g., F1 +1.3 @`rol` = 0.5, +3.5 @`rol` = 0.25, +6.8 @`rol` = 0.125; accuracy +1.4, +2.5, +5.4). This pattern supports our design claim: spatially dense multi-scale features provide stronger unsupervised targets under consistency, improving label efficiency when annotations are limited.

Effect of confidence-thresholded consistency (Figure 5). Introducing a confidence gate into the consistency term (*SMAD-THR*) leads to a systematic reduction in accuracy across all label budgets relative to SMAD, with the gap widening as the labeled ratio decreases. At `rol` = 0.5, *SMAD-THR* converges to a visibly lower plateau than SMAD and exhibits a noisier early trajectory; at `rol` = 0.25 and `rol` = 0.125, the attenuation is more pronounced and the early-phase volatility persists longer, indicating that the gate suppresses a substantial fraction of unlabeled updates when predictors are initially underconfident. This behavior is consistent with the precision–recall trade-off inherent to confidence filtering: while high thresholds are designed to exclude erroneous targets, they simultaneously curtail the recall of informative unlabeled instances and can starve the unsupervised signal during the period when it is most needed. Moreover, requiring both branches to exceed the threshold tightens the criterion further and magnifies the effect under class imbalance or calibration mismatch. Prior studies reported analogous sensitivities—confidence-thresholding yields gains when calibration is strong and augmentation is aggressive (e.g., FixMatch [16], Noisy Student [17]), but can underperform when thresholds are overly conservative or confidence is miscalibrated [16–18]. The curves in Figure 5 align with the latter regime: A harder gate reduces detrimental noise yet leaves performance below that of agreement-driven SMAD that exploits the unlabeled pool more continuously.

**Table 2.** SMAD vs. SMAD$^{W}$. Values are mean ± std over three runs.

| Method (with `rol`) | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| SMAD (0.5) | 91.0% ± 0.8% | 89.2% ± 1.1% | 90.2% ± 1.1% | 89.6% ± 1.1% |
| SMAD (0.25) | 90.3% ± 0.4% | 88.7% ± 0.7% | 89.7% ± 0.6% | 89.1% ± 0.5% |
| SMAD (0.125) | 89.4% ± 0.6% | 87.2% ± 0.8% | 88.7% ± 0.5% | 87.8% ± 0.7% |
| SMAD$^{W}$ (0.5) | 89.6% ± 0.3% | 88.0% ± 0.5% | 88.6% ± 0.4% | 88.3% ± 0.5% |
| SMAD$^{W}$ (0.25) | 87.8% ± 0.5% | 86.4% ± 0.9% | 85.1% ± 0.8% | 85.6% ± 0.8% |
| SMAD$^{W}$ (0.125) | 84.0% ± 1.2% | 82.9% ± 0.5% | 80.5% ± 2.3% | 81.0% ± 2.0% |

SMAD$^{W}$ uses WideResNet instead of SMAD's segmentation-oriented backbone.

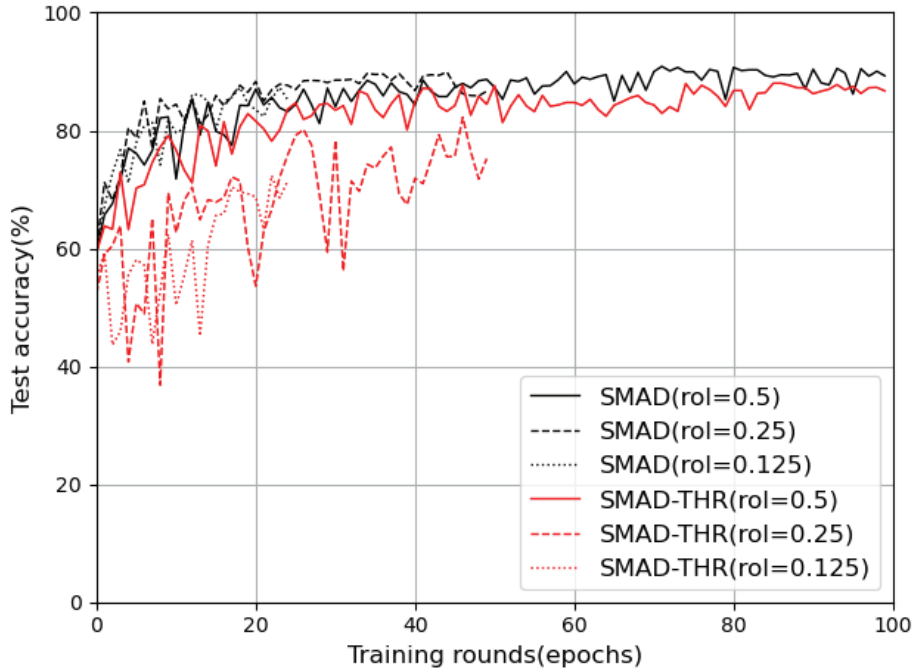



**Figure 5.** Effect of confidence-thresholded consistency (*SMAD-THR*) across label ratios.

Computational efficiency. On the same hardware (Section 4.1), the average training time per epoch over 100 epochs is 6.26 min (FixMatch), 6.50 min (SMAD$^{W}$), and 8.49 min (SMAD). Thus, SMAD incurs a modest overhead of +35.6% vs. FixMatch and +30.6% vs. SMAD$^{W}$, reflecting the heavier encoder. For inference on the fixed test set of 1680 images, total latency is 45 s (FixMatch), 105 s (SMAD$^{W}$), and 140 s (SMAD), corresponding to throughputs of $\sim$37.3 img/s ($\approx$26.8 ms/img), $\sim$16.0 img/s ($\approx$62.5 ms/img), and $\sim$12.0 img/s ($\approx$83.3 ms/img), respectively. In practice, these latencies remain compatible with offline scanning and scheduled batch analytics, and the observed gains in detection accuracy, macro-precision, macro-recall, and macro-F1 (Section 5.1) often outweigh the moderate increase in compute for SMAD in security operations where false decisions are costlier than additional milliseconds at inference.

### *5.3. Summary and Concluding Remarks*

Under realistic labeling constraints characteristic of security telemetry, models that leverage abundant unlabeled data deliver the greatest return on supervision. Across all label budgets, SMAD's dual-branch consistency achieves superior terminal accuracy and smoother optimization than purely supervised training on the same labeled subset and a pseudo-labeling variant, remaining competitive with—often surpassing—the full-label supervised reference. In the more constrained regimes ($\texttt{rol} = 0.25, 0.125$), SMAD reaches a stable operating point in fewer epochs with only a modest reduction in the final ceiling, evidencing strong label efficiency without sacrificing late-stage stability. A contributing factor is the segmentation-oriented backbone, whose pixel-level, multi-scale features yield descriptors more invariant to APK packing/repackaging and obfuscation; aggregating agreement over dense spatial cues provides a higher-SNR unsupervised signal and better early calibration, thereby strengthening the benefits of consistency training in this domain.

The confidence-gated ablation clarifies an operational trade-off. While hard thresholds effectively suppress low-confidence noise, they also curtail the recall of informative unlabeled instances during cold start, yielding lower asymptotes in our setting. Consequently, calibration-aware or curriculum mechanisms—such as temperature scaling, dynamic or branch-aware thresholds, or soft weighting—are preferable to fixed gates when calibration is uncertain or class priors drift.

Taken together, these results indicate that dense pixel-wise multi-scale encoders paired with dual-branch agreement realize the most value when labels are scarce. In malware-image scenarios where padding/obfuscation and layout idiosyncrasies are common, preserving fine spatial cues before classification appears particularly beneficial.

## 6. Conclusions

This paper introduced SMAD, a semi-supervised detector for Android malware that couples dual-branch agreement with a segmentation-oriented encoder to exploit unlabeled APK imagery under label scarcity. Across label budgets, SMAD surpassed supervised training on the same backbone and a recursive pseudo-labeling variant while exhibiting smoother optimization dynamics. In addition, a comparison against FixMatch [16] under the same training schedule showed higher accuracy/precision/recall/F1 across label ratios, and a backbone ablation confirmed that dense pixel-wise multi-scale features yield consistent gains over a standard WideResNet. The confidence-thresholded ablation clarified an operational trade-off: hard gates filter low-confidence noise but reduce unlabeled coverage during cold start, lowering the asymptote in our setting. Taken together, these results support SMAD as a label-efficient, deployment-oriented approach for malware analytics where annotations are limited.

Limitations and Future Work. We use only weak augmentations and a fixed confidence threshold (no tuning). Future work will focus on augmentation-policy design and confidence-threshold optimization (adaptive or schedule-based).

**Author Contributions:** Conceptualization, S.L.; Methodology, S.L.; Validation, S.H.; Writing—original draft, S.L.; Writing—review & editing, S.L.; Visualization, S.L. and S.H.; Supervision, S.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Regional Innovation System & Education (RISE) program through the (Chungbuk Regional Innovation System & Education Center), funded by the Ministry of Education (MOE) and the (Chungcheongbuk-do), Republic of Korea (2025-RISE-11-004).

**Data Availability Statement:** Publicly available datasets were analyzed in this study. The CICMalDroid 2020 dataset, released by the Canadian Institute for Cybersecurity (University of New Brunswick), is accessible at: https://www.unb.ca/cic/datasets/maldroid-2020.html (accessed on 26 October 2025).

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

## References

1. Zaim bin Ahmad, M.S.; Takemoto, K. Large-scale moral machine experiment on large language models. *PLoS ONE* **2025**, *20*, e0322776. [CrossRef]
2. Hasanzadeh, F.; Josephson, C.B.; Waters, G.; Adedinsewo, D.; Azizi, Z.; White, J.A.T. Bias recognition and mitigation strategies in artificial intelligence healthcare applications. *npj Digit. Med.* **2025**, *8*, 154. [CrossRef] [PubMed]
3. Ahmed, M.; Soofi, A.A.; Raza, S.; Khan, F.; Ahmad, S.; Khan, W.U.; Asif, M.; Xu, F.; Han, Z. Advancements in RIS-Assisted UAV for Empowering Multiaccess Edge Computing: A Survey. *IEEE Internet Things J.* **2025**, *12*, 6325–6346. [CrossRef]
4. Wolniak, R.; Stecuła, K. Artificial Intelligence in Smart Cities—Applications, Barriers, and Future Directions: A Review. *Smart Cities* **2024**, *7*, 1346–1389. [CrossRef]
5. Hashmi, E.; Yamin, M.M.; Yayilgan, S.Y. Securing tomorrow: A comprehensive survey on the synergy of Artificial Intelligence and information security. *AI Ethics* **2025**, *5*, 1911–1929. [CrossRef]
6. Yang, X.; Song, Z.; King, I.; Xu, Z. A Survey on Deep Semi-Supervised Learning. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 8934–8954. [CrossRef]
7. Nguyen, A.T.; Raff, E.; Nicholas, C.; Holt, J. Leveraging uncertainty for improved static malware detection under extreme false positive constraints. *arXiv* **2021**, arXiv:2108.04081. [CrossRef]
8. Ucci, D.; Aniello, L.; Baldoni, R. Survey of machine learning techniques for malware analysis. *Comput. Secur.* **2019**, *81*, 123–147. [CrossRef]
9. Alhogail, A.; Alharbi, R.A. Effective ML-Based Android Malware Detection and Categorization. *Electronics* **2025**, *14*, 1486. [CrossRef]
10. Huang, W.; Stokes, J.W. MtNet: A multi-task neural network for dynamic malware classification. In Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, San Sebastián, Spain, 7–8 July 2016; pp. 399–418.
11. Ahmad, R.; Alsmadi, I.; Alhamdani, W.; Tawalbeh, L.A. Zero-day attack detection: A systematic literature review. *Artif. Intell. Rev.* **2023**, *56*, 10733–10811. [CrossRef]
12. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv* **2017**, arXiv:1703.01780. [CrossRef]
13. Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; Raffel, C.A. Mixmatch: A holistic approach to semi-supervised learning. *arXiv* **2019**, arXiv:1905.02249. [CrossRef]
14. Berthelot, D.; Carlini, N.; Cubuk, E.D.; Kurakin, A.; Sohn, K.; Zhang, H.; Raffel, C. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv* **2019**, arXiv:1911.09785. [CrossRef]
15. Xie, Q.; Dai, Z.; Hovy, E.; Luong, M.-T.; Le, Q.V. Unsupervised data augmentation for consistency training. *arXiv* **2019**, arXiv:1904.12848. [CrossRef]
16. Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C.A.; Cubuk, E.D.; Kurakin, A.; Li, C.L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv* **2020**, arXiv:2001.07685. [CrossRef]
17. Xie, Q.; Luong, M.T.; Hovy, E.; Le, Q.V. Self-training with noisy student improves imagenet classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 10687–10698. [CrossRef]
18. Oliver, A.; Odena, A.; Raffel, C.A.; Cubuk, E.D.; Goodfellow, I. Realistic evaluation of deep semi-supervised learning algorithms. *arXiv* **2018**, arXiv:1804.09170. [CrossRef]

19. Arazo, E.; Ortego, D.; Albert, P.; O'Connor, N.E.; McGuinness, K. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8. [CrossRef]
20. Mvula, P.K.; Branco, P.; Jourdan, G.V.; Viktor, H.L. A Survey on the Applications of Semi-supervised Learning to Cyber-security. *ACM Comput. Surv.* **2024**, *56*, 1–41. [CrossRef]
21. Memon, M.; Unar, A.A.; Ahmed, S.S.; Daudpoto, G.H.; Jaffari, R. Feature-Based Semi-Supervised Learning Approach to Android Malware Detection. *Eng. Proc.* **2023**, *32*, 6. [CrossRef]
22. Liu, M.; Yang, Q.; Wang, W.; Liu, S. Semi-Supervised Encrypted Malicious Traffic Detection Based on Multimodal Traffic Characteristics. *Sensors* **2024**, *24*, 6507. [CrossRef]
23. Chin, M.; Corizzo, R. Continual Semi-Supervised Malware Detection. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 2829–2854. [CrossRef]
24. Li, J.; Zhang, Y.; Huang, Y.; Leach, K. Malmixer: Few-shot malware classification with retrieval-augmented semi-supervised learning. *arXiv* **2024**, arXiv:2409.13213.
25. Shu, R.; Xia, T.; Tu, H.; Williams, L.; Menzies, T. Reducing the Cost of Training Security Classifier (via Optimized Semi-Supervised Learning). *arXiv* **2022**, arXiv:2205.00665. [CrossRef]
26. Zheng, X.; Yang, S.; Wang, X. SF-IDS: An Imbalanced Semi-Supervised Learning Framework for Fine-Grained Intrusion Detection. In Proceedings of the IEEE International Conference on Communications, Rome, Italy, 28 May–1 June 2023. [CrossRef]
27. Belkin, M.; Niyogi, P.; Sindhwani, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.* **2006**, *7*, 2399–2434.
28. Grandvalet, Y.; Bengio, Y. Semi-supervised learning by entropy minimization. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 1 December 2004.
29. Dang, Z.; Zheng, Y.; Lin, X.; Peng, C.; Chen, Q.; Gao, X. Semi-Supervised Learning for Anomaly Traffic Detection via Bidirectional Normalizing Flows. *arXiv* **2024**, arXiv:2403.10550. [CrossRef]
30. Williams, B.; Qian, L. Semi-Supervised Learning for Intrusion Detection in Large Computer Networks. *Appl. Sci.* **2025**, *15*, 5930. [CrossRef]
31. Yuan, Y.; Huang, Y.; Zeng, X.; Mei, H.; Cheng, G. M3S-UPD: Efficient Multi-Stage Self-Supervised Learning for Fine-Grained Encrypted Traffic Classification with Unknown Pattern Discovery. *arXiv* **2025**, arXiv:2505.21462.
32. Sun, J.; Zhang, X.; Wang, Y.; Jin, S. CoMDet: A Contrastive Multimodal Pre-Training Approach to Encrypted Malicious Traffic Detection. In Proceedings of the 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC), Osaka, Japan, 2–4 July 2024; pp. 1118–1125. [CrossRef]
33. Perales Gómez, Á.L.; Fernández Maimó, L.; Huertas Celdrán, A.; García Clemente, F.J. An interpretable semi-supervised system for detecting cyberattacks using anomaly detection in industrial scenarios. *IET Inf. Secur.* **2023**, *17*, 553–566. [CrossRef]
34. Krajewska, A.; Niewiadomska-Szynkiewicz, E. Clustering Network Traffic Using Semi-Supervised Learning. *Electronics* **2024**, *13*, 2769. [CrossRef]
35. Nataraj, L.; Karthikeyan, S.; Jacob, G.; Manjunath, B.S. Malware images: Visualization and automatic classification. In Proceedings of the 8th International Symposium on Visualization for Cyber Security, Pittsburgh, PA, USA, 20 July 2011; pp. 1–7. [CrossRef]
36. Seneviratne, S.; Shariffdeen, R.; Rasnayaka, S.; Kasthuriarachchi, N. Self-Supervised Vision Transformers for Malware Detection. *IEEE Access* **2022**, *10*, 103121–103135. [CrossRef]
37. Chen, L. C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
38. Wang, Z.; Zhao, Z.; Xing, X.; Xu, D.; Kong, X.; Zhou, L. Conflict-based cross-view consistency for semi-supervised semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 19585–19595. [CrossRef]
39. Android Malware Dataset (CICMalDroid 2020). Available online: https://www.unb.ca/cic/datasets/maldroid-2020.html (accessed on 5 August 2025).
40. Lee, S. Distributed Detection of Malicious Android Apps While Preserving Privacy Using Federated Learning. *Sensors* **2023**, *23*, 2198. [CrossRef]
41. PyTorch. Available online: https://pytorch.org/ (accessed on 5 August 2025).

*electronics*

MDPI

*Article*

# Distributed Partial Label Learning for Missing Data Classification

**Zhen Xu [1,2,*,†] and Zushou Chen [1,†]**

[1] College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325006, China; 24451352006@stu.wzu.edu.cn
[2] The Metaverse and Artificial Intelligence Institute of Wenzhou University, Wenzhou 325006, China
* Correspondence: 20200588@wzu.edu.cn
† These authors contributed equally to this work.

**Abstract:** Distributed learning (DL), in which multiple nodes in an inner-connected network collaboratively induce a predictive model using their local data and some information communicated across neighboring nodes, has received significant research interest in recent years. Yet, it is challenging to achieve excellent performance in scenarios when training data instances have incomplete features and ambiguous labels. In such cases, it is essential to develop an efficient method to jointly perform the tasks of missing feature imputation and credible label recovery. Considering this, in this article, a distributed partial label missing data classification (dPMDC) algorithm is proposed. In the proposed algorithm, an integrated framework is formulated, which takes the ideas of both generative and discriminative learning into account. Firstly, by exploiting the weakly supervised information of ambiguous labels, a distributed probabilistic information-theoretic imputation method is designed to distributively fill in the missing features. Secondly, based on the imputed feature vectors, the classifier modeled by the random feature map of the $\chi^2$ kernel function can be learned. Two iterative steps constitute the dPMDC algorithm, which can be used to handle dispersed, distributed data with partially missing features and ambiguous labels. Experiments on several datasets show the superiority of the suggested algorithm from many viewpoints.

**Keywords:** distributed processing; partial label classification; missing data classification; random feature map of $\chi^2$ kernel

## 1. Introduction

Nowadays, with the advancement of distributed hardware systems, substantial data are typically collected and stored at multiple nodes over different geographical regions [1–10]. To handle these kinds of data, distributed learning (DL), where multiple nodes collaboratively perform the global-like task based on their own local data and limited information provided by one-hop neighboring nodes, has been developed and has attracted much research attention. DL is commonly used in many areas, such as anomaly detection [3], the industrial Internet of Things [4], environmental monitoring [8], and data mining [7,10], due to its excellent learning performance and adaptability to node failures.

For these DL algorithms, having a sufficient number of high-quality data with complete features is a precondition for obtaining satisfactory learning performance. However, due to various causes, such as absent features or acquisition failures, the collected data vectors often contain a certain number of missing features [6]. A lack of high-quality data can degrade classification performance. Recent years have witnessed some efforts to tackle

this problem. Most current missing data classification (MDC) methods address the tasks of missing feature imputation and predictive model induction independently. To be specific, they firstly utilize some imputation methods, including mean imputation [11], knn imputation [12,13], logistic regression imputation [14], and auto-encoder imputation [15–17], to induce an imputed model to fill in the missing features in the early stage and then learn the classifier based on the recovered features. Although extensive experiments have shown that these data imputation methods can boost learning performance to some extent, a certain amount of training data is complete features and precise labels are required during the induction of the imputed model, which may be infeasible in many real applications. In addition to the above methods, in the literature [18–20], a probabilistic generative model was designed to seek the optimal completion solution based on the learned model. For such a method, although a complete data sample was not required to learn the imputation model, the missing features of some training data needed to be pre-imputed before training the classifier. Since the accuracy of pre-imputation is heavily dependent on accurate supervision information, it is difficult to obtain good learning performance when a large amount of training data is unlabeled or ambiguously labeled. Another strategy suggested in [21] was to lessen the negative impact of missing features on classification performance by reducing the importance of training data that have many missing features. However, this method does not consider the information about data distribution when evaluating the weight of training data, which may lead to the degradation of the induced classifier's performance. Lately, a few novel MDC methods have been proposed [6,22,23], which jointly address the tasks of missing feature imputation and classifier induction in an integral framework. These MDC approaches usually require accurate label information to impute missing features and induce classifiers, which includes an underlying assumption that all the available labels of incomplete data are error-free. However, this assumption is not valid in many scenarios.

Actually, gathering a substantial quantity of data samples with missing features is simple, but labeling these incomplete data without any ambiguous information is an expensive/time-consuming process. It is more likely that only partially labeled data annotated with a series of ambiguous labels can be obtained. Therefore, it is preferable to use the valuable information from ambiguous labels to perform missing feature imputation in such cases.

Recently, partial label learning (PLL), which induces the classifier based on training data annotated with ambiguous labels, has emerged as a new approach in machine learning. Most traditional PLL algorithms are designed for multi-class classification (MCC) [24–26], which typically employs disambiguation procedures to recover the correct label from the candidate label set and then train the classifier based on the recovered labels. For example, recently, two novel instance-dependent PLL algorithms were proposed in [27], which characterize the latent label distribution to disambiguate the ambiguous labels by inferring variational posterior density and exploiting the mutual information between label and feature space. In [9], a distributed, semi-supervised PLL was developed in which the model parameters, labeling confidence, and weights of training data are iteratively updated in a collaborative manner. Recently, PLL was extended to deal with the problem of multi-label classification (MLC) [28–33]. For example, in [33], a distributed, partial, multi-label label method was introduced that identifies reliable labeling information from a series of ambiguous labels based on globally common basic data and induces a predictive model by making full use of identified credible label information. Although these approaches have been shown to be effective, a significant limitation is that they do not account for the effect of the missing features on the performance of the disambiguation strategy.

Jointly taking the above consideration into account, in this article, as shown in Figure 1, the problem of distributed classification of partially labeled incomplete data is considered. For this study, an integrated framework was designed that could address jointly the tasks of missing feature imputation and predictive classifier induction over a network. Specifically, the main contributions of this paper are summarized as follows:

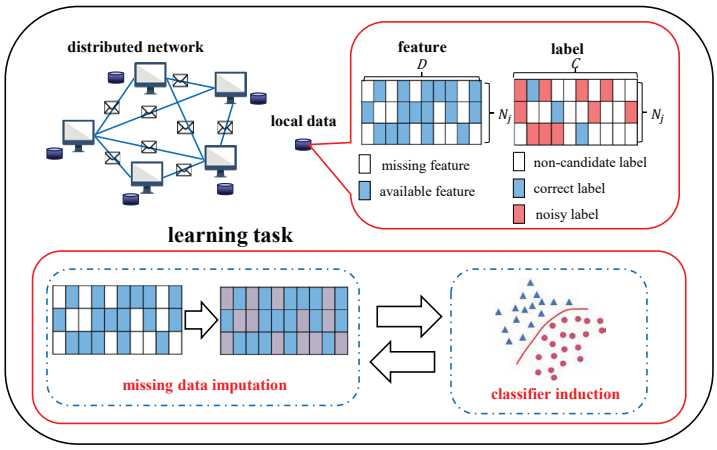



**Figure 1.** An example of the topology of a connected network and a flowchart of the proposed dPMDC algorithm.

1. In the proposed algorithm, a distributed, information-theoretic, learning-based (ITL-based) data imputation method is developed based on the Gaussian mixture model (GMM), which exploits weakly supervised information of ambiguous labels to guide model parameter estimation. Then, the missing features can be imputed by computing the conditional expectation based on the observed features and estimated parameters.

2. To induce the classifier based on the imputed data, the information-theoretic measures, including logistic loss with respect to imputed data and the mutual information with respect to cluster centers of the Gaussian components, are used to design the cost function. By using the random feature map to replace the kernel feature map for the discriminant function construction, a non-linear multi-class classifier can be distributively learned. Moreover, in order to make the estimated labeling confidence more suitable for guiding missing feature imputation and model induction, we introduced a novel normalized sigmoid function to scale the value of labeling confidence.

3. We alternately established two steps in a collaborative manner and developed the dPMDC algorithm, which can address the issue of the distributed classification of training data presented by partially available features annotated with ambiguous labels.

The subsequent sections of this article are organized as follows: Section 2 formulates the issue of the distributed classification of partially labeled incomplete data and presents relevant preliminaries. Then, Section 3 describes the technical details of the proposed dPMDC algorithm. Following this, Section 4 reports the experimental results of the dPMDC algorithm and state-of-the-art methods on multiple datasets. Finally, we conclude this paper in Section 5.

## 2. Problem Formulation and Preliminaries

In this section, the issue of the distributed classification of training data represented by partially available features annotated with a series of ambiguous labels over a network is formulated. To ensure that this paper is self-contained, some fundamental preliminaries should be briefly introduced. To improve the readability of this article, we present the explanation of professional terms and notation in Tables 1 and 2, respectively.

**Table 1.** Explanation of notations.

| Variable | Meaning | Variable | Meaning |
|---|---|---|---|
| $\{\boldsymbol{x}_{j,n}\}_{n=1}^{N_j}$ | training data with complete features | $\{\boldsymbol{\Omega}_{j,n}\boldsymbol{x}_{j,n}\}_{n=1}^{N_j}$ | observed training data with incomplete features |
| $\{\boldsymbol{y}_{j,n}\}_{n=1}^{N_j}$ | class label | $k(\boldsymbol{x}_h, \boldsymbol{x}_n)$ | kernel function |
| $\boldsymbol{w}_c$ | weight vector with respect to the $c$-th class | $f_c(\boldsymbol{x})$ | output of the discriminant function with respect to the $c$-th class |
| $\phi(\boldsymbol{x}_n)$ | kernel feature map | $p(\boldsymbol{x}_n)$ | distribution of training data $\boldsymbol{x}_n$ |
| $\pi_k$ | mixing parameter | $\boldsymbol{\mu}_k$ | mean vector of the $k$-th Gaussian component |
| $\boldsymbol{\Sigma}_k$ | covariance matrix of the $k$-th Gaussian component | $r_{ck}$ | probability that the $k$-th Gaussian component belongs to the $c$-th class |
| $p(\boldsymbol{x}_n\|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ | distribution of $\boldsymbol{x}_n$ with respect to the $k$-th Gaussian component | $g_{j,n,c}$ | confidence of the $c$-th candidate label |
| $\gamma_{j,k,n}$ | posterior probability belongs to the $k$-th component | $v_{ji}$ | cooperative coefficient |
| $\boldsymbol{\Xi}_{j,k}$ | inverse matrix of $\boldsymbol{\Sigma}_{j,k}$ | $\boldsymbol{\Omega}_{j,n}^{'}$, $\boldsymbol{\Lambda}_{j,k}$ | auxiliary variable matrices |
| $p(c\|\boldsymbol{x}_{j,n})$ | conditional probability belongs to the $c$-th class | $I_W(\boldsymbol{a}; K)$ | mutual information between Gaussian component centers and class labels |
| $p(c)$ | empirical distribution of the $c$-th class | $h(\cdot)$ | a scaling function |
| $\lambda_A$ | weight parameter with respect to logistical loss | $\lambda_B$ | weight parameter with respect to mutual information |

**Table 2.** Explanation of technical terms.

| Term | Meaning | Term | Meaning |
|---|---|---|---|
| data imputation | filling in missing features in training data | ambiguous labels | the label is not unique and candidate labels consist of the correct label and noisy labels |
| discriminant function | describing the boundary between separate classes | kernel function | a tool for non-linear classification problems |
| random feature map | a promising tool for approximating kernel functions | Gaussian mixture model (GMM) | a technique for estimating the probability of data belonging to a cluster |
| expectation-maximization (EM) procedures | an iterative method to seek the maximum likelihood estimates | probability density function (pdf) | a function for determining the probability of the variable falling within a range of values |

**Table 2.** *Cont.*

| Term | Meaning | Term | Meaning |
|---|---|---|---|
| marginal probability | the unconditional probability of an event occurring | conditional probability | the probability of an event occurring on the basis that the other event occurs |
| mutual information | a tool for measuring the amount of dependence between two variables | missing completely at random (MCAR) | the missing data are independent of the observed and unobserved data |
| missing at random | the missing data are related to the observed data but not the unobserved data | missing not at random (MNAR) | the missing data are related to the unobserved data |

### *2.1. Problem Formulation*

In this paper, a network including $J$ inter-connected nodes spread across a geographically dispersed region is considered. Without loss of generality, we model the considered network using an undirected graph $\mathcal{G} = (\mathcal{J}, \mathcal{E})$, where $\mathcal{J}$ indicates the set of the nodes and $\mathcal{E}$ indicates the set of the edges. Each node $j$ in this network performs the global-like computation by exchanging the information with its neighboring nodes $i \in \mathcal{B}_j$, where $\mathcal{B}_j$ stands for the set of neighboring nodes composed of all the one-hop neighbors and node $j$ itself.

Assuming that the features of the training data are missing completely at random (MCAR), there exist $N_j$ partially labeled incomplete data $\{\mathbf{\Omega}_{j,n}\boldsymbol{x}_{j,n}, \boldsymbol{y}_{j,n}\}_{n=1}^{N_j}$ at each individual node $j$, just as shown in Figure 1. Here, $\mathbf{\Omega}_{j,n}\boldsymbol{x}_{j,n} \in \mathcal{X}^D$ represents the observed input vector and $\boldsymbol{y}_{j,n}$ denotes the collected candidate label vector. To represent these kinds of data, we use $\boldsymbol{x}_{j,n}$ to denote a $D$-dimensional originally input vector with complete features and introduce $\mathbf{\Omega}_{j,n}$ to stand for a $D$-dimensional diagonal matrix, where its $d$-th diagonal element $\mathbf{\Omega}_{j,n,d} = 1$ if the corresponding feature is accessible; otherwise, $\mathbf{\Omega}_{j,n,d} = 0$. So, in the observed feature vector $\mathbf{\Omega}_{j,n}\boldsymbol{x}_{j,n}$, all accessible features are preserved, while the absent features are assigned to be zeros. The candidate label vector $\boldsymbol{y}_{j,n}$ is a $C$-dimensional vector, which is composed of $C$ specific classes, with the $c$-th entry being 1 if the $c$-th label is a candidate label and 0 otherwise.

Assuming that the discriminant function is non-linear, we can express the value of the output of the discriminant function with respect to the $c$-th class as

$$f_c(\boldsymbol{x}_n) = \sum_h \alpha_{c,h} k(\boldsymbol{x}_h, \boldsymbol{x}_n), \tag{1}$$

where $k(\boldsymbol{x}_h, \boldsymbol{x}_n) =< \phi(\boldsymbol{x}_h), \phi(\boldsymbol{x}_n) >$ denotes the kernel function, with $\phi(\cdot)$ being the infinite-dimensional kernel map in the reproducing kernel Hilbert space. $\alpha_{c,h}$ denotes the weighted coefficient with respect to the $c$-th class. Based on the kernel feature map $\phi(\cdot)$, we can rewrite (1) as

$$f_c(\boldsymbol{x}_n) = \boldsymbol{w}_c^T \cdot \phi(\boldsymbol{x}_n), \tag{2}$$

where $\boldsymbol{w}_c$ denotes the weight parameter with respect to the $c$-th class, which can be calculated by $\boldsymbol{w}_c = \sum_h \alpha_{c,h}\phi(\boldsymbol{x}_h)$.

The objective of the proposed algorithm is to seek the optimal weight parameter so that the high-precision classifier can be obtained.

### *2.2. Random Feature Map*

Since the kernel feature map $\phi(\cdot)$ cannot be explicitly expressed, the weight vector $\boldsymbol{w}_c$ composed of a linear combination of the kernel feature maps cannot be adaptively updated and freely exchanged among neighboring nodes.

To address this issue, we utilize a limited-dimensional random feature map $\hat{\phi}(\cdot)$ to substitute the original kernel feature map $\phi(\cdot)$ for model parameter construction. But, calculating the values of some common kernel functions, including the Gaussian kernel and Laplacian kernel, requires the components of the feature vectors to be complete. If we apply the random feature map of a Gaussian kernel or Laplacian kernel to the incomplete data, then the approximation error of missing feature imputation may increase and the properties inside the approximated feature space may be deteriorated. Taking this into account, the $\chi^2$ kernel, whose features are independent of each other, is employed in this paper. Specifically, the kernel function of the $\chi^2$ kernel is calculated by [34]

$$k_{\chi^2}(\boldsymbol{x}_h, \boldsymbol{x}_n) = \sum_{d=1}^{D} \frac{2x_{h,d}x_{n,d}}{x_{h,d} + x_{n,d}}, \tag{3}$$

Correspondingly, the random feature map with respect to the $\chi^2$ kernel function can be constructed by

$$\hat{\phi}(\boldsymbol{x}_n) = [\hat{\phi}(x_{n,1})^T, ..., \hat{\phi}(x_{n,D})^T]^T, \tag{4}$$

with each element

$$[\hat{\phi}(x_{n,d})]_k = \begin{cases} \sqrt{x_{n,d}H\hat{\kappa}_0}, & k = 0, \\ \sqrt{2x_{n,d}H\hat{\kappa}_{\frac{k+1}{2}}}\cos(\frac{k+1}{2}H\log x_{n,d}), & k > 0, \quad \text{odd}, \\ \sqrt{2x_{n,d}H\hat{\kappa}_{\frac{k}{2}}}\sin(\frac{k}{2}H\log x_{n,d}), & k > 0, \quad \text{even}, \end{cases} \tag{5}$$

where $\hat{\kappa}_k, k = 1,\ldots,U$ denotes the discrete spectrum, which is sampled from the corresponding continuous function $\hat{\kappa}_\omega$. $H$ stands for a parameter whose value depends on particular facts. Readers can refer to [34] for a detailed discussion.

## 3. dPMDC Algorithm

This section presents the technical details of the dPMDC algorithm. In general, our proposed algorithm consists of two main parts. For the first part, we present an ITL-based missing feature imputation approach that can fill in missing features by fully using the weakly supervised information of ambiguous labels. In the second step, based on imputed data features, a multi-class classifier can be distributively induced at each individual node, which can be used to update the labeling confidences. We perform these two steps alternately until convergence.

### *3.1. Missing Feature Imputation*

In this subsection, a fully decentralized ITL-based missing feature imputation method is described based on the GMM model. At first, the parameters of the Gaussian mixture components can be adaptively estimated via fully decentralized expectation-maximization (EM) procedures. Then, the missing features of each training datum can be imputed based on the estimated parameters of the GMM and the partially observed features.

To be specific, referring to the theory of the GMM model [20], the distribution of each training datum with full features may be described as a mixture of $K$ Gaussian components, as indicated by

$$p(\boldsymbol{x}_n) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{6}$$

where $\pi_k$ denotes the mixing parameter. The probability density function (pdf) $p(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ characterizes the distribution of training data $\boldsymbol{x}_n$ with respect to the $k$-th component in the GMM, which can be expressed by

$$p(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{\exp(-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k))}{(2\pi)^{\frac{D}{2}} |\boldsymbol{\Sigma}_k|^{\frac{1}{2}}}, \tag{7}$$

with $\boldsymbol{\mu}_k$ denoting the mean vector of the $k$-th Gaussian component and $\boldsymbol{\Sigma}_k$ representing the covariance matrix of the $k$-th Gaussian component, respectively.

In this approach, weakly supervised information from ambiguous labels is incorporated into the GMM model and then the data distribution (6) can be reformulated as [35]

$$p(\boldsymbol{x}_n) = \sum_{c=1}^{C} \sum_{k=1}^{K} r_{ck} \pi_k p(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{8}$$

where $r_{ck}$ denotes the probability that the $k$-th Gaussian component belongs to the $c$-th class.

To fill in the missing features, we formulate the decentralized framework of missing feature imputation, which is composed of the log-likelihood function $Q(\boldsymbol{\theta})$ and a series of consensus-based constraints, i.e.,

$$\begin{aligned} &Q(\boldsymbol{\theta}) = \sum_{j=1}^{J} Q_j(\boldsymbol{\theta}_j) = \sum_{j=1}^{J} \sum_{c=1}^{C} \sum_{n=1}^{N_j^c} g_{j,n,c,t} \log \sum_{c=1}^{C} \sum_{k=1}^{K} r_{j,ck} \pi_{j,k} p(\boldsymbol{x}_{j,n}|\boldsymbol{\mu}_{j,k}, \boldsymbol{\Sigma}_{j,k}), \\ &\text{s.t.} \quad \boldsymbol{\mu}_{j,k} = \boldsymbol{\mu}_{i,k}, \quad \boldsymbol{\Sigma}_{j,k} = \boldsymbol{\Sigma}_{i,k}, \quad r_{j,ck} = r_{i,ck}, \quad \pi_{j,k} = \pi_{i,k}, j \in \mathcal{J}, i \in \mathcal{B}_j, \end{aligned} \tag{9}$$

where $g_{j,n,c,t}$ stands for the confidence of the $c$-th candidate label with respect to $\boldsymbol{x}_{j,n}$ at iteration $t$ and $N_j^c$ denotes the amount of training data belonging to the $c$-th class (their candidate label sets contain the $c$-th class) at node $j$. For simplicity, we use $\boldsymbol{\theta}_j$ to represent the parameter set of the GMM, where $\boldsymbol{\theta}_j = \{\boldsymbol{\mu}_{j,k}, \boldsymbol{\Sigma}_{j,k}, \pi_{j,k}, r_{j,ck} | k = 1, ..., K\}$.

Referring to the general GMM [36], we would like to utilize the EM method to obtain the optimal solution of (9). But, since partial components in the data vector $\boldsymbol{x}_{j,n}$ are absent, the EM procedures cannot be directly executed. To address this problem, we use the marginal probability distribution with respect to the observable features to substitute the original one $p(\boldsymbol{x}_{j,n}|\boldsymbol{\mu}_{j,k}, \boldsymbol{\Sigma}_{j,k})$ for parameter estimation. To explicitly express this marginal probability distribution, we divide the whole training data sample as $\boldsymbol{x}_{j,n} = [\boldsymbol{x}_{j,n}^{o,T}, \boldsymbol{x}_{j,n}^{m,T}]^T$, where $\boldsymbol{x}_{j,n}^{o}$ and $\boldsymbol{x}_{j,n}^{m}$ respectively denote the observed and missing features of $\boldsymbol{x}_{j,n}$. Correspondingly, we have the new forms of $\boldsymbol{\mu}_{j,k}$ and $\boldsymbol{\Sigma}_{j,k}$ as follows:

$$\boldsymbol{\mu}_{j,k} = [\boldsymbol{\mu}_{j,k}^{o,T}, \boldsymbol{\mu}_{j,k}^{m,T}]^T, \quad \boldsymbol{\Sigma}_{j,k} = \begin{bmatrix} \boldsymbol{\Sigma}_{j,k}^{oo} & \boldsymbol{\Sigma}_{j,k}^{om} \\ \boldsymbol{\Sigma}_{j,k}^{mo} & \boldsymbol{\Sigma}_{j,k}^{mm} \end{bmatrix}. \tag{10}$$

By employing the above marginal probability distribution, we reformulate the loss function (9) as follows:

$$\begin{aligned} Q^{'}(\boldsymbol{\theta}) &= \sum_{j=1}^{J}\sum_{c=1}^{C}\sum_{n=1}^{N_j^c} g_{j,n,c,t}\log\sum_{c=1}^{C}\sum_{k=1}^{K} r_{j,ck}\pi_{j,k}p(\boldsymbol{x}_{j,n}^{o}|\boldsymbol{\mu}_{j,k}^{o},\boldsymbol{\Sigma}_{j,k}^{oo}), \\ \text{s.t.}\quad & \boldsymbol{\mu}_{j,k}=\boldsymbol{\mu}_{i,k},\quad \boldsymbol{\Sigma}_{j,k}=\boldsymbol{\Sigma}_{i,k},\quad r_{j,ck}=r_{i,ck},\quad \pi_{j,k}=\pi_{i,k},\quad j\in\mathcal{J}, i\in\mathcal{B}_j. \end{aligned} \tag{11}$$

Update of mean vector $\mu_{j,k,t+1}$: We can update the mean vector $\boldsymbol{\mu}_{j,k}$ by solving the following optimization problem:

$$\begin{aligned} \boldsymbol{\mu}_{j,k,t+1} &= \arg\max Q^{'}(\boldsymbol{\theta}|\boldsymbol{\theta}_t) \\ &= \arg\max \sum_{c=1}^{C}\sum_{n=1}^{N_j^c} g_{j,n,c,t}\log\sum_{c=1}^{C}\sum_{k=1}^{K} r_{j,ck,t}\pi_{j,k,t}p(\boldsymbol{x}_{j,n}^{o}|\boldsymbol{\mu}_{j,k}^{o},\boldsymbol{\Sigma}_{j,k,t}^{oo}) \\ &= \arg\max \sum_{c=1}^{C}\sum_{n=1}^{N_j^c} g_{j,n,c,t}\log\sum_{c=1}^{C}\sum_{k=1}^{K} \frac{r_{j,ck,t}\pi_{j,k,t}}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}_{j,k,t}^{oo}|^{\frac{1}{2}}} \\ &\qquad \cdot\exp(-\frac{1}{2}(\boldsymbol{x}_{j,n}^{o}-\boldsymbol{\mu}_{j,k}^{o})^{T}\boldsymbol{\Sigma}_{j,k,t}^{oo,-1}(\boldsymbol{x}_{j,n}^{o}-\boldsymbol{\mu}_{j,k}^{o})), \\ \text{s.t.}\quad & \boldsymbol{\mu}_{j,k}=\boldsymbol{\mu}_{i,k},\quad j\in\mathcal{J},\quad i\in\mathcal{B}_j. \end{aligned} \tag{12}$$

The aforementioned objective function (12) can be equivalently expressed in a full vector form, i.e.,

$$\begin{aligned} \boldsymbol{\mu}_{j,k,t+1} &= \arg\max \sum_{c=1}^{C}\sum_{n=1}^{N_j^c} g_{j,n,c,t}\log\sum_{c=1}^{C}\sum_{k=1}^{K} \frac{r_{j,ck,t}\pi_{j,k,t}|\boldsymbol{\Sigma}_{j,k,t}^{oo}|^{-\frac{1}{2}}}{(2\pi)^{\frac{D}{2}}} \\ &\quad\cdot\exp(-\frac{1}{2}(\boldsymbol{x}_{j,n}-\boldsymbol{\mu}_{j,k})^{T}\boldsymbol{\Omega}_{j,n}(\boldsymbol{\Xi}_{j,k,t}-\boldsymbol{\Lambda}_{j,k,t})\boldsymbol{\Omega}_{j,n}(\boldsymbol{x}_{j,n}-\boldsymbol{\mu}_{j,k})), \\ &\quad\text{s.t.}\quad \boldsymbol{\mu}_{j,k}=\boldsymbol{\mu}_{i,k},\quad j\in\mathcal{J},\quad i\in\mathcal{B}_j, \end{aligned} \tag{13}$$

where $\boldsymbol{\Xi}_{j,k}\in\mathbb{R}^{D\times D}$ is the inverse matrix of $\boldsymbol{\Sigma}_{j,k}$, which is composed of four parts, i.e., $\boldsymbol{\Xi}_{j,k}=\begin{bmatrix}\boldsymbol{\Xi}_{j,k}^{oo} & \boldsymbol{\Xi}_{j,k}^{om}\\ \boldsymbol{\Xi}_{j,k}^{mo} & \boldsymbol{\Xi}_{j,k}^{mm}\end{bmatrix}$. Considering that the value of $\boldsymbol{\Xi}_{j,k}^{oo}$ is not equivalent to the value of $\boldsymbol{\Sigma}_{j,k}^{oo,-1}$, the matrix $\boldsymbol{\Lambda}_{j,k}$ is introduced to solve this problem, which is given by $\boldsymbol{\Lambda}_{j,k}=\begin{bmatrix}\boldsymbol{\Sigma}_{j,k}^{oo,-1}\boldsymbol{\Sigma}_{j,k}^{om}\boldsymbol{\Xi}_{j,k}^{mm}\boldsymbol{\Sigma}_{j,k}^{mo}\boldsymbol{\Sigma}_{j,k}^{oo,-1} & \mathbf{0}_{om}\\ \mathbf{0}_{mo} & \mathbf{0}_{mm}\end{bmatrix}$.

To obtain a global consensus estimation of the mean vector, referring to the diffusion cooperative strategy in [10], the update process can be established by two steps, adaptation and cooperation. In the adaptation step, the mean vector can be locally estimated depending on the local data. In the cooperation step, we can update the mean vector by combining the instantaneous estimations exchanged from neighboring nodes with the cooperative coefficient.

To be specific, taking the partial derivative of (13) and setting the results to 0, we have

$$\sum_{c=1}^{C}\sum_{n=1}^{N_j^c} g_{j,n,c,t}\gamma_{j,k,n,t}\boldsymbol{\Omega}_{j,n}(\boldsymbol{\Xi}_{j,k}-\boldsymbol{\Lambda}_{j,k})\boldsymbol{\Omega}_{j,n}(\boldsymbol{x}_{j,n}-\boldsymbol{\mu}_{j,k})=\mathbf{0}_{D\times 1}. \tag{14}$$

It is noted that, for simplicity, $\gamma_{j,k,n,t}$ is introduced to denote the posterior probability belonging to the $k$-th component using the current estimations of parameters $\{\boldsymbol{\mu}^o_{j,k,t}, \boldsymbol{\Sigma}^{oo}_{j,k,t}\}$, which can be calculated by

$$\gamma_{j,k,n,t} = \frac{\sum_{c=1}^{C} r_{j,ck,t}\pi_{j,k,t}p(\boldsymbol{x}^o_{j,n}|\boldsymbol{\mu}^o_{j,k,t}, \boldsymbol{\Sigma}^{oo}_{j,k,t})}{\sum_{c=1}^{C}\sum_{k=1}^{K} r_{j,ck,t}\pi_{j,k,t}p(\boldsymbol{x}^o_{j,n}|\boldsymbol{\mu}^o_{j,k,t}, \boldsymbol{\Sigma}^{oo}_{j,k,t})}. \tag{15}$$

Based on (14), each component of $\boldsymbol{\mu}_{j,k}$ can be locally updated using the local observed features and the posterior probability $\gamma_{j,k,n,t}$, i.e.,

$$\boldsymbol{\mu}^{'}_{j,k,t+1}(d) = \frac{\sum_{c=1}^{C}\sum_{n=1}^{N_j^c} g_{j,n,c,t}\gamma_{j,k,n,t}\boldsymbol{x}_{j,n}(d)}{\sum_{c=1}^{C}\sum_{n=1}^{N_j^c} g_{j,n,c,t}\gamma_{j,k,n,t}\boldsymbol{\Omega}_{j,n}(d)}, \quad d = 1, ..., D. \tag{16}$$

Then, by combining the estimation exchanged from neighbors, we have

$$\boldsymbol{\mu}_{j,k,t+1} = \sum_{i\in\mathcal{B}_j} v_{ji}\boldsymbol{\mu}^{'}_{i,k,t+1}, \tag{17}$$

where $v_{ji}$ represents the cooperative coefficient, which can be designed according to the Metropolis rule [10].

Update of covariance matrix $\boldsymbol{\Sigma}_{j,k}$: We can derive the update equation of the covariance matrix $\boldsymbol{\Sigma}_{j,k}$ by addressing the subsequent optimization problem:

$$\begin{aligned}
\boldsymbol{\Sigma}_{j,k,t+1} &= \arg\max Q^{'}(\boldsymbol{\theta}|\boldsymbol{\theta}_t) \\
&= \arg\max \sum_{j=1}^{J}\sum_{c=1}^{C}\sum_{n=1}^{N_j^c} g_{j,n,c,t}\log\sum_{c=1}^{C}\sum_{k=1}^{K} r_{j,ck,t}\pi_{j,k,t}p(\boldsymbol{x}^o_{j,n}|\boldsymbol{\mu}^o_{j,k,t}, \boldsymbol{\Sigma}^{oo}_{j,k}) \\
&= \arg\max \sum_{j=1}^{J}\sum_{c=1}^{C}\sum_{n=1}^{N_j^c} g_{j,n,c,t}\log\sum_{c=1}^{C}\sum_{k=1}^{K} r_{j,ck,t}\pi_{j,k,t}\frac{|\boldsymbol{\Sigma}^{oo}_{j,k}|^{-\frac{1}{2}}}{(2\pi)^{\frac{D}{2}}} \\
&\qquad \cdot \exp(-\frac{1}{2}(\boldsymbol{x}_{j,n} - \boldsymbol{\mu}_{j,k,t})^T\boldsymbol{\Omega}_{j,n}(\Xi_{j,k} - \boldsymbol{\Lambda}_{j,k,t})\boldsymbol{\Omega}_{j,n}(\boldsymbol{x}_{j,n} - \boldsymbol{\mu}_{j,k,t})), \\
&\text{s.t.} \quad \boldsymbol{\Sigma}_{j,k} = \boldsymbol{\Sigma}_{i,k}, \quad j \in \mathcal{J}, \quad i \in \mathcal{B}_j.
\end{aligned} \tag{18}$$

Taking the partial derivative of (18) with respect to $\boldsymbol{\Sigma}_{j,k}$ and letting its results equal zero, we have

$$\sum_{c=1}^{C}\sum_{n=1}^{N_j^c} g_{j,n,c,t}\gamma_{j,k,n,t}\boldsymbol{\Omega}^{'}_{j,n} \circ (\boldsymbol{\Lambda}^{'}_{j,k,t} - \boldsymbol{\Sigma}^{-1}_{j,k}\boldsymbol{\Omega}_{j,n}(\boldsymbol{x}_{j,n} - \boldsymbol{\mu}_{j,k,t})(\boldsymbol{x}_{j,n} - \boldsymbol{\mu}_{j,k,t})^T\boldsymbol{\Omega}_{j,n}\boldsymbol{\Sigma}^{-1}_{j,k}) = \mathbf{0}_{D\times D}, \tag{19}$$

where $\circ$ denotes the Hadamard product and $\mathbf{1}_{D\times D}$ is a $D \times D$ dimensional square matrix, with all the entries being 1. The auxiliary variable matrices $\boldsymbol{\Omega}^{'}_{j,n} = \boldsymbol{\Omega}_{j,n}\mathbf{1}_{D\times D}\boldsymbol{\Omega}_{j,n}$ and $\boldsymbol{\Lambda}^{'}_{j,k} = \begin{bmatrix} \boldsymbol{\Lambda}^{'oo}_{j,k} & \boldsymbol{\Lambda}^{'om}_{j,k} \\ \boldsymbol{\Lambda}^{'mo}_{j,k} & \boldsymbol{\Lambda}^{'mm}_{j,k} \end{bmatrix}$ with

$$\begin{aligned}
\boldsymbol{\Lambda}^{'oo}_{j,k} =& \boldsymbol{\Sigma}^{oo,-1}_{j,k} - \boldsymbol{\Sigma}^{oo,-1}_{j,k}\boldsymbol{\Sigma}^{om}_{j,k}\Xi^{mo}_{j,k}(\boldsymbol{x}^o_{j,n} - \boldsymbol{\mu}^o_{j,k})(\boldsymbol{x}^o_{j,n} - \boldsymbol{\mu}^o_{j,k})^T\boldsymbol{\Sigma}^{oo,-1}_{j,k} - \boldsymbol{\Sigma}^{oo,-1}_{j,k}(\boldsymbol{x}^o_{j,n} - \boldsymbol{\mu}^o_{j,k})(\boldsymbol{x}^o_{j,n} - \boldsymbol{\mu}^o_{j,k})^T \\
&\cdot \Xi^{om}_{j,k}\boldsymbol{\Sigma}^{mo}_{j,k}\boldsymbol{\Sigma}^{oo,-1}_{j,k} - \boldsymbol{\Sigma}^{oo,-1}_{j,k}\boldsymbol{\Sigma}^{om}_{j,k}\Xi^{mo}_{j,k}(\boldsymbol{x}^o_{j,n} - \boldsymbol{\mu}^o_{j,k})(\boldsymbol{x}^o_{j,n} - \boldsymbol{\mu}^o_{j,k})^T\Xi^{om}_{j,k}\boldsymbol{\Sigma}^{mo}_{j,k}\boldsymbol{\Sigma}^{oo,-1}_{j,k}.
\end{aligned}$$

These auxiliary variables $\boldsymbol{\Lambda}^{'om}_{j,k}$, $\boldsymbol{\Lambda}^{'mo}_{j,k}$, and $\boldsymbol{\Lambda}^{'mm}_{j,k}$ are introduced to recover the complete matrix $\boldsymbol{\Sigma}_{j,k}$. It is noted that regardless of what values are assigned to these auxiliary

variables, equation (19) remains valid since their corresponding Hadamard coefficients in $\mathbf{\Omega}_{j,n}^{'}$ are zeros.

Using $\mathbf{\Sigma}_{j,k}$ to pre-multiply and post-multiply both sides of (19), we can obtain

$$\sum_{c=1}^{C}\sum_{n=1}^{N_j^c} g_{j,n,c,t}\gamma_{j,k,n,t}\mathbf{\Omega}_{j,n}^{'} \circ (\mathbf{\Sigma}_{j,k}\mathbf{\Lambda}_{j,k,t}^{'}\mathbf{\Sigma}_{j,k} - \mathbf{\Omega}_{j,n}(\boldsymbol{x}_{j,n}-\boldsymbol{\mu}_{j,k,t})(\boldsymbol{x}_{j,n}-\boldsymbol{\mu}_{j,k,t})^T\mathbf{\Omega}_{j,n}) = \mathbf{0}_{D\times D} \tag{20}$$

The update equation of the $(d_1, d_2)$-th element of $\mathbf{\Sigma}_{j,k,t+1}$ can be obtained based on the local data, which are given by

$$\mathbf{\Sigma}_{j,k,t+1}^{'}(d_1,d_2) = \frac{\sum_{c=1}^{C}\sum_{n=1}^{N_j^c} g_{j,n,c,t}\gamma_{j,k,n,t}\boldsymbol{S}_{j,n,t}(d_1,d_2)}{\sum_{c=1}^{C}\sum_{n=1}^{N_j^c} g_{j,n,c,t}\gamma_{j,k,n,t}\mathbf{\Omega}_{j,n}^{'}(d_1,d_2)}, \tag{21}$$

where $\boldsymbol{S}_{j,n,t} = \mathbf{\Omega}_{j,n}^{'} \circ (\boldsymbol{x}_{j,n}-\boldsymbol{\mu}_{j,k,t})(\boldsymbol{x}_{j,n}-\boldsymbol{\mu}_{j,k,t})^T + \begin{bmatrix} \mathbf{\Sigma}_{j,k,t}^{oo}\mathbf{\Lambda}_{j,k,t}^{'oo}\mathbf{\Sigma}_{j,k,t}^{oo} - \boldsymbol{I}_{oo} & \mathbf{0}_{om} \\ \mathbf{0}_{mo} & \mathbf{0}_{mm} \end{bmatrix}$.

Similar to the update of the mean vector $\boldsymbol{\mu}_{j,k}$, we can update the covariance matrix at node $j$ by combining the instantaneous estimation exchanged from neighboring nodes $i \in \mathcal{B}_j$,

$$\mathbf{\Sigma}_{j,k,t+1} = \sum_{i\in\mathcal{B}_j} v_{ji}\mathbf{\Sigma}_{i,k,t+1}^{'} \tag{22}$$

Update of mixing parameter $\pi_{j,k}$: Based on the latest estimation $\boldsymbol{\mu}_{j,k,t+1}$ and $\mathbf{\Sigma}_{j,k,t+1}$, we can update the mixing parameter $\pi_{j,k}$ via adaptation and cooperation steps:

$$\pi_{j,k,t+1}^{'} = \frac{\sum_{c=1}^{C}\sum_{n=1}^{N_j^c} g_{j,n,c,t}\gamma_{j,k,n,t+1}}{\sum_{k=1}^{K}\sum_{c=1}^{C}\sum_{n=1}^{N_j^c} g_{j,n,c,t}\gamma_{j,k,n,t+1}}, \tag{23}$$

$$\pi_{j,k,t+1} = \sum_{i\in\mathcal{B}_j} v_{ji}\pi_{i,k,t+1}^{'}. \tag{24}$$

Update of probability $r_{j,ck}$: Similarly, by keeping the other parameters unchanged, we can update the probability $r_{j,ck}$ as follows:

$$r_{j,ck,t+1}^{'} = \frac{\sum_{n=1}^{N_j^c} g_{j,n,c,t}\delta_{j,c,n,k,t+1}}{\sum_{c=1}^{C}\sum_{n=1}^{N_j^c} g_{j,n,c,t}\delta_{j,c,n,k,t+1}}, \tag{25}$$

$$r_{j,ck,t+1} = \sum_{i\in\mathcal{B}_j} v_{ji}r_{i,ck,t+1}^{'}, \tag{26}$$

where the probability $\delta_{j,c,n,k,t+1} = \frac{\pi_{j,k,t}r_{j,ck,t}p(\boldsymbol{x}_{j,n}^o|\boldsymbol{\mu}_{j,k,t+1}^o,\mathbf{\Sigma}_{j,k,t+1}^{oo})}{\sum_{c=1}^{C}\sum_{k=1}^{K}\pi_{j,k,t}r_{j,ck,t}p(\boldsymbol{x}_{j,n}^o|\boldsymbol{\mu}_{j,k,t+1}^o,\mathbf{\Sigma}_{j,k,t+1}^{oo})}$ denotes the posterior probability of the $k$-th component of the $n$-th training data belonging to the $c$-th class.

Then, based on the observed features $\boldsymbol{x}_{j,n}^o$ and the parameters of GMM, we compute the conditional mean of the missing features $\hat{\boldsymbol{x}}_{j,n}^{m|o}$ to impute the missing feature $\boldsymbol{x}_{j,n}^m$ [37], which is given by

$$\begin{aligned}\hat{\boldsymbol{x}}_{j,n,t+1}^{m|o} &= \sum_{k=1}^{K}\gamma_{j,k,n,t+1}\boldsymbol{\mu}_{j,k,t+1}^{m|o} \\ &= \sum_{k=1}^{K}\gamma_{j,k,n,t+1}\cdot[\boldsymbol{\mu}_{j,k,t+1}^{m} + \mathbf{\Sigma}_{j,k,t+1}^{mo}\mathbf{\Sigma}_{j,k,t+1}^{oo,-1}(\boldsymbol{x}_{j,n}^o - \boldsymbol{\mu}_{j,k,t+1}^o)],\end{aligned} \tag{27}$$

where $\boldsymbol{\mu}_{j,k,t+1}^{m|o}$ denotes the corresponding conditional expectation.

### *3.2. Classifier Induction*

In this subsection, we describe the induction of the classifier using the imputed data and the learned Gaussian component centers.

To be specific, the logistic loss of all the imputed data is designed to leverage the supervised information from the ambiguous labels. Subsequently, the objective function can be expressed as follows:

$$\begin{aligned} &\min_{\boldsymbol{w}_c} -\sum_{j=1}^{J}\sum_{n=1}^{N_j}\lambda_A \log\sum_{c=1}^{C} y_{j,n,c}g_{j,n,c,t}p(c|\hat{\boldsymbol{x}}_{j,n,t+1}), \\ &\text{s.t.} \quad \boldsymbol{w}_{j,c} = \boldsymbol{w}_{i,c}, \quad j \in \mathcal{J}, i \in \mathcal{B}_j, \end{aligned} \tag{28}$$

where $\lambda_A$ stands for the weighted parameter with respect to the logistical loss of training data. The conditional probability $p(c|\hat{\boldsymbol{x}}_{j,n,t+1})$ can be characterized by the kernel logistic regression functions modeled by the random feature map, which is given by

$$p(c|\hat{\boldsymbol{x}}_{j,n,t+1}) \propto \exp\Big(\boldsymbol{w}_c^T \cdot \hat{\phi}(\hat{\boldsymbol{x}}_{j,n,t+1})\Big). \tag{29}$$

Then, the mutual information between the Gaussian component centers and their corresponding class label is used to construct a regularization term, which helps explore the hidden structure of data and group data items into the corresponding classes [38], as shown below:

$$I_W(\boldsymbol{a}; K) = \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{c=1}^{C}\frac{1}{JK}p(c|\boldsymbol{a}_{j,k,t+1})\log\frac{p(c|\boldsymbol{a}_{j,k,t+1})}{p(c)}, \tag{30}$$

where the empirical distribution of class labels can be estimated based on $K$ learned Gaussian components, i.e., $p(c) = p_j(c) = \frac{1}{K}\sum_{k=1}^{K} p(c|\boldsymbol{a}_{j,k,t+1})$ for $j \in \mathcal{J}$. Here, $\boldsymbol{a}_{j,k,t+1}$ is the estimated center of the $k$-th learned Gaussian component, which can be obtained by the mean vector of imputed training data belonging to the $k$-th Gaussian component.

Incorporating the mutual information regularization term into the objective function, we have

$$\begin{aligned} \min_{W} F = \sum_{j=1}^{J} F_j = &-\sum_{j=1}^{J}\sum_{n=1}^{N_j}\lambda_A \log\sum_{c=1}^{C} y_{j,n,c}g_{j,n,c,t}p(c|\hat{\boldsymbol{x}}_{j,n,t+1}) \\ &-\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{c=1}^{C}\frac{\lambda_B}{JK}p(c|\boldsymbol{a}_{j,k,t+1})\log\frac{p(c|\boldsymbol{a}_{j,k,t+1})}{p(c)}, \\ \text{s.t.} \quad &\boldsymbol{w}_{j,c} = \boldsymbol{w}_{i,c}, \quad j \in \mathcal{J}, i \in \mathcal{B}_j, \end{aligned} \tag{31}$$

where $\lambda_B$ denotes the weighted parameter with respect to the mutual information regularization term.

Update of model parameter $\boldsymbol{w}_{j,c}$: Considering that the objective function is too complicated to obtain the closed-form solution, we optimize the model parameter $\{\boldsymbol{w}_{j,c}\}_{c=1}^{C}$ using the steepest gradient descent (SGD) method and diffusion cooperative approach.

To be specific, at the iteration $t > 0$, we have the update equation of the model parameter $\{\boldsymbol{w}_{j,c}\}_c$:

$$\boldsymbol{w}'_{j,c,t+1} = \boldsymbol{w}_{j,c,t} - \zeta_{t+1} \nabla_{\boldsymbol{w}_{j,c}} F_{j,t}, \tag{32a}$$

$$\boldsymbol{w}_{j,c,t+1} = \sum_{i \in \mathcal{B}_j} v_{ji} \boldsymbol{w}'_{j,c,t+1}, \tag{32b}$$

where $\zeta_{t+1}$ denotes the time-varying learning rate. The gradient $\nabla_{\boldsymbol{w}_{j,c}} F_j$

$$\nabla_{\boldsymbol{w}_{j,c}} F_j = -\sum_{n=1}^{N_j} \lambda_A \left( \frac{y_{j,n,c} g_{j,n,c,t} p_{j,n,c}(1 - p_{j,n,c}) - \sum_{c \neq h} y_{j,n,h} g_{j,n,h} p_{j,n,h} p_{j,n,c}}{\sum_{c=1}^{C} y_{j,n,c} g_{j,n,c,t} p_{j,n,c}} \right) \cdot \hat{\phi}(\hat{\boldsymbol{x}}_{j,n,t+1})$$
$$- \sum_{k=1}^{K} \frac{\lambda_B}{KJ} p^a_{j,k,c} \left( \log \frac{p^a_{j,k,c}}{p_c} - \sum_{h=1}^{C} p^a_{j,k,h} \log \frac{p^a_{j,k,h}}{p_h} \right) \cdot \hat{\phi}(\boldsymbol{a}_{j,k,t+1}).$$

For clarity, we use $p_{j,n,c}$, $p^a_{j,k,c}$, and $p_c$ to denote the abbreviations of $p(c|\hat{\boldsymbol{x}}_{j,n})$, $p(c|\boldsymbol{a}_{j,k})$, and $p(c)$, respectively.

Update of labeling confidence $g_{j,n,c}$: In this approach, labeling confidence $g_{j,n,c}$ plays two important roles. First of all, it may be applied directly to the imputation of missing features. Secondly, it also serves as a guide for model induction. Obviously, the value of the labeling confidence will heavily affect the imputed features as well as the induced classifier.

The detailed update processes of labeling confidence $g_{j,n,c}$ are presented as follows.

At the initial state $t = 0$, the value of labeling confidence of candidate labels can be set as

$$g_{j,n,c,0} = \begin{cases} \frac{1}{\sum_{c=1}^{C} y_{j,n,c}} & \text{if } y_{j,n,c} = 1, \\ 0 & \text{if } y_{j,n,c} = 0. \end{cases} \tag{33}$$

At the following iterations $t > 0$, the labeling confidence can be updated based on the conditional probability $p_{j,n,c,t}$, which is given by

$$g_{j,n,c,t+1} = \begin{cases} \frac{h_{j,n,c,t+1}}{\sum_{l=1}^{C} h_{j,n,l,t+1}} & \text{if } y_{j,n,c} = 1, \\ 0 & \text{if } y_{j,n,c} = 0. \end{cases} \tag{34}$$

In this article, to make the labeling confidence more suitable for guiding data imputation and classifier induction, we utilize a novel map $h(\cdot)$ to adjust its value as follows:

$$h_{j,n,c,t+1} = \frac{1/(1 + \exp(-\nu(p_{j,n,c,t+1} - 0.5))) - 1/(1 + \exp(0.5\nu))}{1/(1 + \exp(-0.5\nu)) - 1/(1 + \exp(0.5\nu))},$$

with $\nu$ denoting a time-varying scaling factor. Initially, the performance of the induced classifier may be coarse due to the impact of noisy labels. At this point, a small value of $\nu$ is employed to assign low confidence to the induced classifier. Through a sufficient number of iterations, the impact of noisy labels diminishes, yielding more reliable classification performance. At this point, a large value of $\nu$ is utilized to give a high confidence to the classification result of the induced model. Considering this, we set the parameter $\nu = \log(t + 1)$.

By alternatively executing the steps of missing feature imputation and model induction, we can obtain the optimal classifier. For clarity, the pseudo-code of the proposed dPMDC algorithm is summarized in Algorithm 1.

For illustration, the diagram of the main steps of the proposed dPMDC algorithm is presented in Figure 2.

**Algorithm 1** dPMDC algorithm

**Require:** Input partially labeled incomplete data $\{\boldsymbol{\Omega}_{j,n}\boldsymbol{x}_{j,n}, \boldsymbol{y}_{j,n}\}_{n=1}^{N_j}$, and initialize $\boldsymbol{w}_{j,c,0} = \mathbf{0}_{DU}$ for each node $j$.

**for** $t = 0, \ldots$ **do**
  **for** $j \in \mathcal{J}$ **do**
    Calculate $\boldsymbol{\mu}'_{j,k,t+1}$ and $\boldsymbol{\Sigma}'_{j,k,t+1}$ via (16), (21), and exchange them with the neighbors $\mathcal{B}_j$.
  **end for**
  **for** $j \in \mathcal{J}$ **do**
    Calculate $\boldsymbol{\mu}_{j,k,t+1}$, $\boldsymbol{\Sigma}_{j,k,t+1}$ via (17) and (22).
  **end for**
  **for** $j \in \mathcal{J}$ **do**
    Calculate $\pi'_{j,k,t+1}$ and $r'_{j,ck,t+1}$ via (23) and (25), and exchange them with the neighbors $\mathcal{B}_j$.
  **end for**
  **for** $j \in \mathcal{J}$ **do**
    Calculate $\pi_{j,k,t+1}$ and $r_{j,ck,t+1}$ via (24) and (26).
  **end for**
  **for** $j \in \mathcal{J}$ **do**
    Impute the missing features of the training data $\boldsymbol{x}^m_{j,n,t+1}$ via (27).
    Obtain the random feature map $\hat{\phi}(\hat{\boldsymbol{x}}_{j,n,t+1})$ via (4).
  **end for**
  **for** $j \in \mathcal{J}$ **do**
    Calculate $\boldsymbol{w}'_{j,c,t+1}$ via (32a) and exchange it with neighbors $i \in \mathcal{B}_j$.
  **end for**
  **for** $j \in \mathcal{J}$ **do**
    Calculate $\boldsymbol{w}_{j,c,t+1}$ via (32b).
  **end for**
  **for** $j \in \mathcal{J}$ **do**
    Calculate the new confidence $g_{j,n,c,t+1}$ via (34).
  **end for**
**end for**
**Output:** Optimal multi-classifier $\boldsymbol{f}^*_c$.

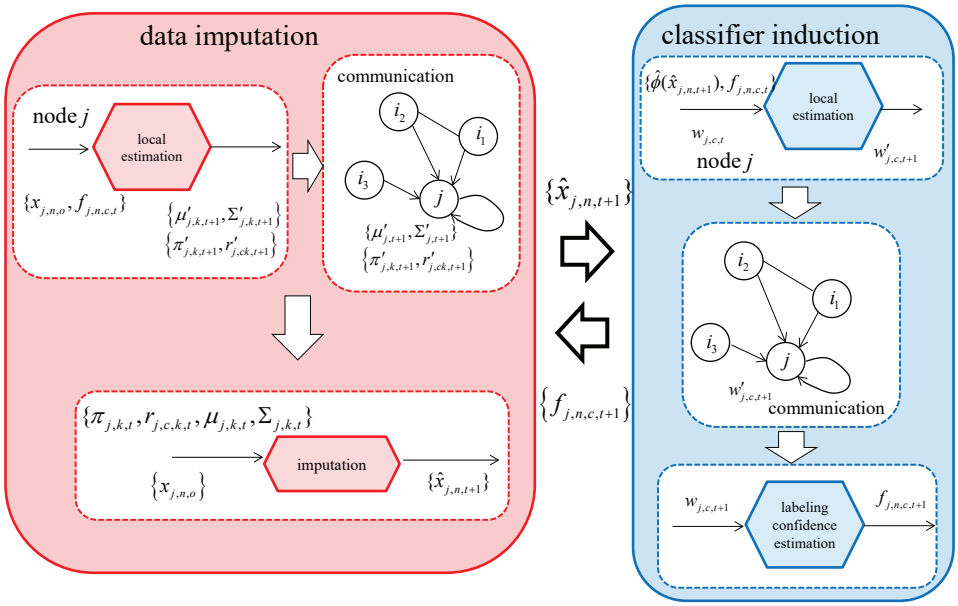



**Figure 2.** Diagram of the main steps of proposed dPMDC algorithm.

*3.3. Performance Analysis*

In this subsection, we conduct a theoretical analysis of the convergence, the computational complexity, and the communication cost of the proposed algorithm.

We first present some common assumptions before conducting the convergence analysis.

**Assumption 1.** *For a connected network $\mathcal{G}$, the cooperative matrix $V$ with its element $V_{ji} = v_{ji}$ satisfies the following conditions: $V\mathbf{1}_J = \mathbf{1}_J$ and $\mathbf{1}_J^T V = \mathbf{1}_J^T$. Additionally, the spectrum norm of the matrix $V - (1/J)\mathbf{1}_J\mathbf{1}_J^T$ is no larger than 1.*

**Lemma 1.** *For a connected, distributed network, if Assumption 1 holds, then all the local estimates of model parameters at each individual node $j$, including $\boldsymbol{\mu}_{j,k,t}$, $\boldsymbol{\Sigma}_{j,k,t}$, $\pi_{j,k,t}$, $r_{j,ck,t}$, and $W_{j,t}$, converge to their average values through a sufficient number of iterations, i.e., $\lim_{t\to\infty} ||\boldsymbol{\mu}_{j,k,t} - \bar{\boldsymbol{\mu}}_{k,t}||_2^2 = 0$, $\lim_{t\to\infty} ||\boldsymbol{\Sigma}_{j,k,t} - \bar{\boldsymbol{\Sigma}}_{k,t}||_F^2 = 0$, $\lim_{t\to\infty} |\pi_{j,k,t} - \bar{\pi}_{k,t}|^2 = 0$, $\lim_{t\to\infty} |r_{j,ck,t} - \bar{r}_{ck,t}|^2 = 0$, and $\lim_{t\to\infty} ||\boldsymbol{w}_{j,c,t} - \bar{\boldsymbol{w}}_{c,t}||_F^2 = 0$. It should be noted that $\bar{\boldsymbol{\mu}}_{k,t}$, $\bar{\boldsymbol{\Sigma}}_{k,t}$, $\bar{\pi}_{k,t}$, $\bar{r}_{ck,t}$, and $\bar{\boldsymbol{w}}_{c,t}$ denote the average values of $\{\boldsymbol{\mu}_{j,k,t}\}_{j=1}^J$, $\{\boldsymbol{\Sigma}_{j,k,t}\}_{j=1}^J$, $\{\pi_{j,k,t}\}_{j=1}^J$, $\{r_{j,ck,t}\}_{j=1}^J$, and $\{\boldsymbol{w}_{j,c,t}\}_{j=1}^J$, respectively.*

**Proof.** See [39]. □

**Theorem 1.** *For a connected network $\mathcal{G}$, if Assumption 1 and Lemma 1 hold, the parameters of the GMM $\boldsymbol{\theta}_{j,t} = \{\boldsymbol{\mu}_{j,k,t}, \boldsymbol{\Sigma}_{j,k,t}, \pi_{j,k,t}, r_{j,ck,t}\}_k$ can converge to the optimal value $\boldsymbol{\theta}^*$ as $t$ tends to infinity.*

**Proof.** See Appendix A of [40]. □

**Theorem 2.** *For a connected network $\mathcal{G}$, if Assumption 1 and Lemma 1 hold, the model parameter of the multi-class classifier $\boldsymbol{W}_{j,t}$ can converge to its optimal value $\boldsymbol{W}^*$ when $t \to \infty$.*

**Proof.** See Appendix A. □

Based on Theorems 1 and 2, we know that all the local estimates of variables at each individual node can converge to the optimal values through a sufficient number of iterations, provided that the distributed network keeps connected, which verifies the effectiveness of the dPMDC method in theory.

The computational complexity of the algorithm is measured by the number of addition operations (AOs) and multiplication operations (MOs) at each node at each iteration. Table 3 summarizes the amount of the addition and multiplication operations of the proposed algorithm. To clearly illustrate the calculation methods of AO and MO, we cite two relevant examples:

**Example 1.** *Taking the addition of two $N * N$ matrices as an example, the whole process requires 0 multiplication operations and $N^2$ addition operations.*

**Example 2.** *Taking the multiplication of two $N * N$ matrices as an example, the whole process requires $N^3$ multiplication operations and $N^2(N-1)$ addition operations.*

By observing Table 3, we can see that the computational complexity of the algorithm is related to multiple factors, including the proportion of missing features $P_m$ at MCAR assumption, the dimension of non-linear mapping $DU$, the number of Gaussian components $K$, and the network topology, in addition to the characteristics of the data themselves. Therefore, when the number of neighbor nodes in the network is moderate, the computational complexity of the algorithm is acceptable as long as the value of $K$ and $DU$ is controlled within a reasonable range.

In addition, we also analyzed the communication complexity of the algorithm. At each iteration $t$, each node $j$ needed to exchange $K(D + D^2 + 1 + C) + CDU$ scalars to its neighboring nodes. So, the communication cost of the proposed dPMDC was deemed to be acceptable.

**Table 3.** Computation cost in terms of number of MOs and AOs per iteration $t$ per node $j$.

| | | |
|---|---|---|
| $\{\gamma_{j,k,n}\}_{k,n}$ | MO | $\frac{1}{2}(1-P_m)^3D^3 + (1-P_m)^2D^2 + (1-P_m)D + C + CK$ |
| | AO | $\frac{1}{2}(1-P_m)^3D^3 + (1-P_m)^2D^2 + 2(1-P_m)D + C + CK$ |
| $\{\boldsymbol{\mu}_{j,k}\}_k$ | MO | $D(\lvert\mathcal{B}_j\rvert + 2N_jC)$ |
| | AO | $D(\lvert\mathcal{B}_j\rvert + 4N_jC)$ |
| $\{\boldsymbol{\Sigma}_{j,k}\}_k$ | MO | $D^2(\lvert\mathcal{B}_j\rvert + 2N_jC) + (1-P_m)^2P_mD^3 + 4(1-P_m)^3D^3$ |
| | AO | $D^2(\lvert\mathcal{B}_j\rvert + 4N_jC) + (1-P_m)^2P_mD^3 + 4(1-P_m)^3D^3$ |
| $\{\boldsymbol{\pi}_{j,k}\}_k$ | MO | $KN_jC + \lvert\mathcal{B}_j\rvert$ |
| | AO | $KN_jC + N_jC + \lvert\mathcal{B}_j\rvert$ |
| $\{r_{j,c,k}\}_{c,k}$ | MO | $2CKN_j + N_jC + \lvert\mathcal{B}_j\rvert$ |
| | AO | $CKN_j + N_jC + N_j + \lvert\mathcal{B}_j\rvert$ |
| $\{\hat{\boldsymbol{x}}_{j,n}^{m\lvert o}\}_n$ | MO | $KP_mD + \frac{1}{2}(1-P_m)^3D^3 + P_m(1-P_m)^2D^3 + P_m(1-P_m)D^2$ |
| | AO | $KP_mD + \frac{1}{2}(1-P_m)^3D^3 + P_m(1-P_m)^2D^3 + P_m(1-P_m)D^2$ |
| $\{\boldsymbol{w}_{j,c}\}_c$ | MO | $T[DU(\lvert\mathcal{B}_j\rvert + 4N_jC + KC)]$ |
| | AO | $T[DU(\lvert\mathcal{B}_j\rvert + 6N_jC + KC)]$ |
| $\{g_{j,n,c}\}_{n,c}$ | MO | $CDU + 6$ |
| | AO | $CDU + 2C + 6$ |

## 4. Experiment

In this section, to validate the efficacy of the proposed approach, a series of experiments on several artificial and real PLL datasets are described, including the Double Moon [9], mHealth [41], Gas Drift [41], Pendigits [41], Segmentation [41], Ecoli [41], Vertebral [41], Lost [42], Birdsong [43], and MSRCv2 [44] datasets.

The profiles of these utilized datasets are shown in Table 4. It is noted that seven artificial PLL datasets (Double Moon, mHealth, Gas Drift, Pendigits, Segmentation, Vertebral, and Ecoli datasets) were generated by adding a series of noisy labels into the set of candidate labels under the configuration of two controlled parameters, $s$ and $\epsilon$ [9]. In this context, $s$ represents the number of noisy labels inside the candidate label set and $\epsilon$ represents the co-occurrence probability between a coupling noisy label and the correct label. That is, for each partially labeled data sample, a randomly selected coupling noisy label and the correct label occurred in a pair with probability $\epsilon$. For three real-world PLL datasets (Lost, Birdsong, and MSRCv2 datasets), the original labels of the training data were ambiguous and, thus, no extra noisy labels were added into the set of candidate labels.

To investigate the impact of the different proportions of missing features on classification performance under the MCAR assumption, a metric $P_m$, namely, the percentage of missing features relative to the total number of features in the training dataset, was defined.

**Table 4.** Summary of characteristics of used datasets.

| Dataset | No. Training Data | No. Testing Data | No. Dimension | No. Class |
|---|---|---|---|---|
| Double Moon | 20,000 | 5000 | 2 | 4 |
| mHealth | 16,000 | 4000 | 23 | 4 |
| Gas Drift | 11,120 | 2790 | 129 | 6 |
| Pendigits | 2800 | 698 | 16 | 10 |
| Segmentation | 3700 | 928 | 19 | 7 |
| Ecoli | 270 | 66 | 8 | 8 |
| Vertebral | 250 | 60 | 6 | 3 |
| Lost | 900 | 222 | 108 | 16 |
| Birdsong | 4000 | 998 | 38 | 13 |
| MRSCv2 | 1410 | 348 | 48 | 23 |

For each experiment, a total of 50 Monte Carlo cross-validation simulations were conducted, and the average results from these simulations are reported herein. Furthermore, all datasets utilized in each Monte Carlo simulation were arbitrarily partitioned into 10 folds; the training phase utilized 8 folds, while the testing phase employed the remaining 2 folds. To simulate the performance of a distributed network, here, an interconnected network consisting of 10 nodes and 23 edges was randomly generated. All training data were completely randomly partitioned into $J$ parts with equal size and allocated to these nodes. To conduct the following experiments, the data instances needed to be preprocessed at the initial state, i.e., the values of attributes were normalized into [0, 1].

At each trial of simulation, for the proposed dPMDC, the parameters were set as $\lambda_A = 1$, $\lambda_B = 0.1$, and $K = 30$, $U = 4$; the step size was set as $\zeta_{t+1} = 0.25/t^{0.5}$; and the weight parameters $W$ are set as $\boldsymbol{w}_{j,c,0} = \mathbf{0}_{DU}$ initially. The mean vector could be randomly initialized to a vector uniformly distributed in the range (0, 0.5) and the covariance matrix could be initialized to a $D$-dimensional identity matrix $I_D$. Furthermore, in the initial state, we had the mixing parameter $\pi_{j,k,0} = \frac{1}{K}$ and probability $r_{j,ck,0} = \frac{1}{C}$.

Taking "mHealth" and "Gas Drift" as representatives of all the considered datasets, we depict the learning curves of the imputation error and classification accuracy of the proposed dPMDC algorithm in Figures 3 and 4. In order to test the robustness of the algorithm under different network sizes, we also compare the changing curves of imputation error and classification accuracy of the proposed dPMDC on multiple distributed networks. It should be noted that in this experiment, the controlled parameters of the distributed network topology were characterized by the number of nodes $J$ and the number of edges $|\mathcal{E}|$.

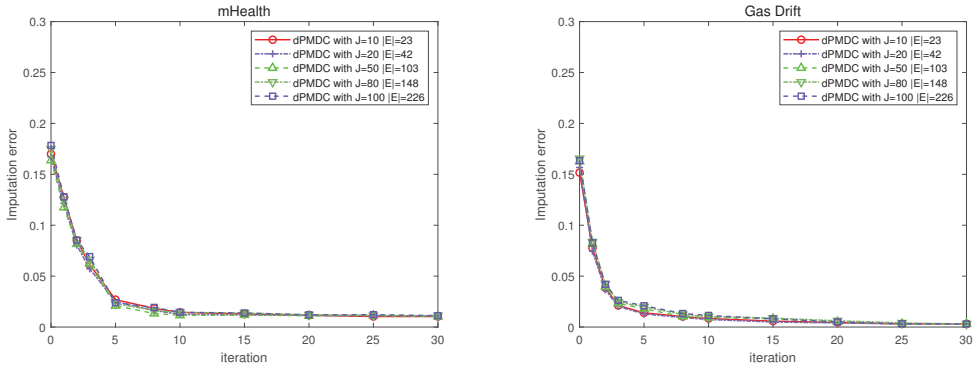



**Figure 3.** Learning curve of imputation error of dPMDC using different networks on "mHealth" and "Gas Drift" datasets.

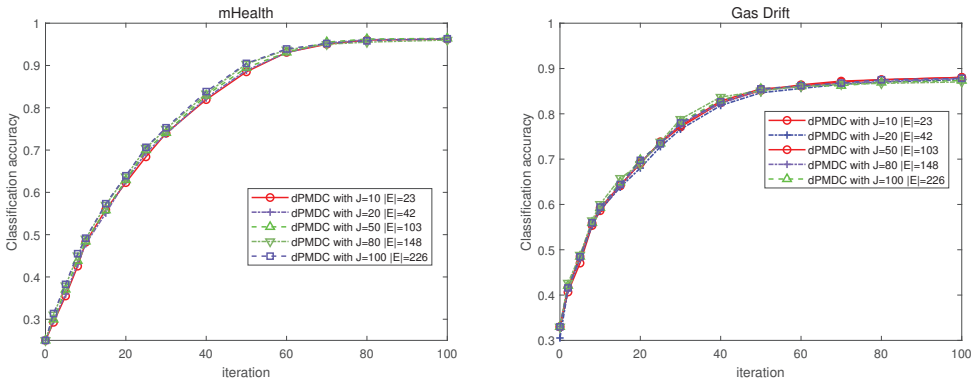



**Figure 4.** Learning curve of classification accuracy of dPMDC using different networks on "mHealth" and "Gas Drift" datasets.

By observing the simulation results presented in Figures 3 and 4, we can notice that the learning curves of imputation error converged significantly faster than those of classification accuracy. The values of imputation error rapidly decreased at the initial 15 iterations and converged to the stable state after about 15 iterations. The learning curve of classification accuracy was relatively smooth. During the first 50 iterations, the values of classification accuracy steadily increased. After about 70 iterations, it gradually converged to the optimal value. We also can observe that the learning curves of the proposed algorithm using different network topologies, either the imputation error or the classification accuracy, were very close to each other. The simulation results show that the size of the network did not significantly affect the learning performance of our proposed algorithm.

Furthermore, we also compare the CPU times of the proposed algorithm at each individual node under different networks on the "mHealth" and "Gas Drift" datasets in Figure 5. To ensure fairness in this experiment, we set the amount of training data at each individual node to be the same. From Figure 5, we can see that the CPU times of the proposed algorithm at each individual node remained nearly unchanged. Such a result indicates that different network topologies could not affect the computational efficiency of the algorithm, as long as the data size of a single node remained unchanged.

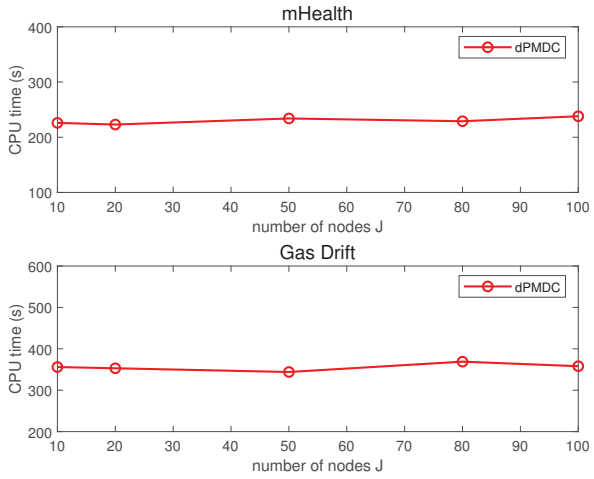



**Figure 5.** CPU times of dPMDC algorithm versus different network topologies.

Additionally, to simulate the robustness of the proposed algorithm against the initial setting of the model parameters, we compared the learning performance of the proposed algorithm with different parameter settings. To be specific, two different cases were taken into consideration.

Case 1: We maintained the original settings. That is, the mean vector of each Gaussian component could be randomly initialized to a vector uniformly distributed in the range (0,0.5), the covariance matrix of each Gaussian component could be initialized to a $D$-dimensional identity matrix $I_D$, and the weight parameter could be initialized to $\boldsymbol{w}_{j,c,0} = \mathbf{0}_{DU}$.

Case 2: We reset the initialized settings. That is, the mean vector of each Gaussian component was randomly initialized to the training data with complete attributes of the local node and the covariance matrix of each Gaussian component was initialized as $0.5I_D$. We set the initialized state of the weight parameter as $\boldsymbol{w}_{j,c,0} = 0.01 \times \mathbf{1}_{DU}$.

We depict the learning curves of the imputation error and the classification accuracy of the proposed algorithm with different initial settings in Figures 6 and 7. It should be noted that in order to distinguish them from each other, we name case 1 dPMDC with originally initial setting and case 2 as dPMDC with newly initial setting. From the simulation results in Figures 6 and 7, we can observe that the changing curves of dPMDC with originally initial setting and dPMDC with newly initial setting were almost overlapping, indicating that our proposed algorithm was insensitive to the initial setting of the model parameters.

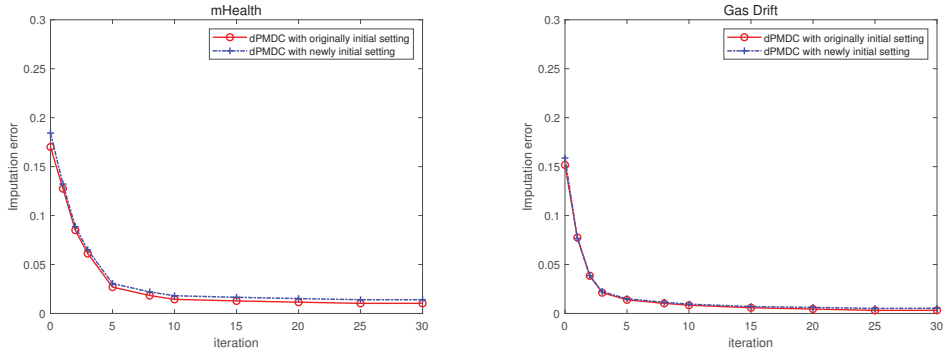



**Figure 6.** Learning curve of imputation error of dPMDC with different initial settings on "mHealth" and "Gas Drift" datasets.

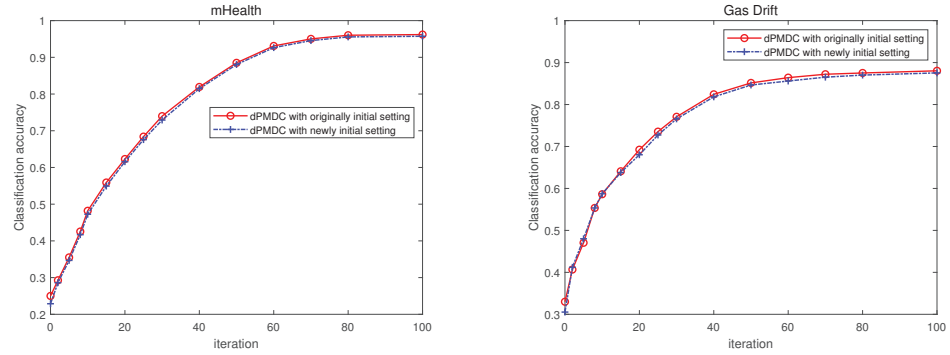



**Figure 7.** Learning curve of classification accuracy of dPMDC with different initial settings on "mHealth" and "Gas Drift" datasets.

Moreover, we investigated the impact of different values of parameters $\lambda_A$ and $\lambda_B$ and the discretization level of random feature map $U$ on the classification performance of our proposed algorithm using the "mHealth" and "Gas Drift" datasets. In this experiment, we investigated the performance changing of the proposed algorithm by varying the value of one parameter while keeping the other parameters unchanged. We can see that the changing trends of the parameters $\lambda_A$ and $\lambda_B$ were similar. The simulation results presented in Figure 8 indicate that as long as the values of parameters $\lambda_A$ and $\lambda_B$ were set within [0.1, 1] and [0.01, 0.1], good learning performance could be obtained. We can see that the classification accuracy of the proposed method gradually improved as the discretization

level of random feature map $U$ increased. The possible reasons were analyzed and are presented as follows. When the value of $U$ increased, the random feature map could give a more precise approximation for the kernel feature map, which boosted the classification performance of the induced classifier to some extent. When the value of $U$ exceeded 4, the performance improvement resulting from the increment of $U$ diminished progressively. Since larger values of $U$ led to higher computational complexity and communication cost, an appropriate choice for $U$ could be set as 4 in order to achieve a balance between classification performance and computational complexity.

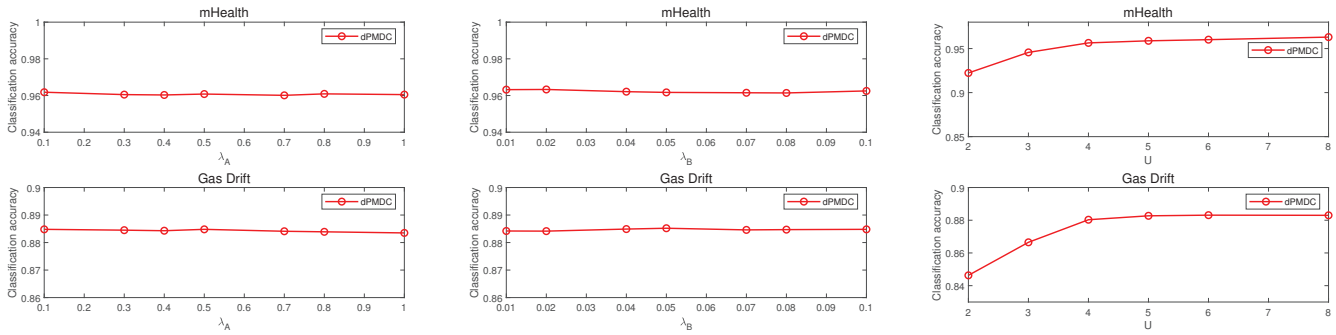



**Figure 8.** Classification accuracy of dPMDC algorithm versus weight parameters $\lambda_A$ and $\lambda_B$ and discretization level of random feature map $U$ on "mHealth" and "Gas Drift" datasets.

Considering that the value of $K$ had a significant effect on both imputation error and classification accuracy, we investigated the changing trends of imputation error and classification accuracy versus $K$, shown in Figure 9. The simulation results indicate that the learning performance of the proposed dPMDC gradually improved as the number of Gaussian mixture components $K$ escalated. When the value of $K$ was greater than 30, the extent of learning performance improvement rapidly decreased. These changing trends were similar to those of $U$. Therefore, we set the number of Gaussian mixture components $K$ to 30 to strike a balance between computational complexity and learning performance.

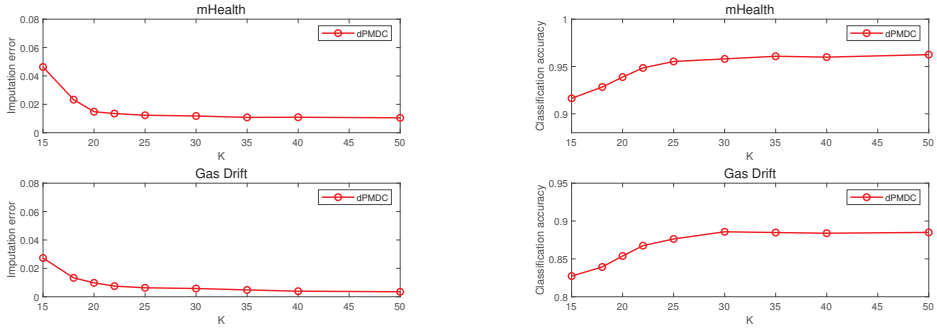



**Figure 9.** Imputation error and classification accuracy of dPMDC versus number of Gaussian components $K$ on "mHealth" and "Gas Drift" datasets.

Furthermore, we investigated the learning performance of the proposed dPMDC algorithm for different distribution data. Given the challenge of characterizing the distribution of existing real datasets, we adopted a common synthetic dataset known as "Double Moon". Referring to the operations in [9], we randomly generated 20,000 training data and divided the upper and the lower moon into two classes, as shown in Figure 10. Then, a special number of noisy labels were added into candidate labels, such that the values of the controlled parameters were $s = 1$ and $\epsilon = 0.3$. To simulate the learning performance of the proposed algorithm, we added noise from different distributions to the training data. Specifically, three cases were considered.

Case 1: We added zero-mean Gaussian noise to the training data so that the signal-to-noise ratio was equivalent to 15 dB.

Case 2: We added 0–1 noise with a magnitude of 0.3 and a probability of 0.5 to the training data.

Case 3: We added uniformly distributed noise with a magnitude of 0.3 to the training data.

For clarity, we depict the training data after adding noise in Figure 10.

The learning curves of the imputation error and classification accuracy of the proposed dPMDC for three cases are presented in Figure 11. We can see that the learning curves were quite similar except for the first few iterations and all converged to ideal levels, indicating that the GMM could effectively characterize the training data with different distributions.

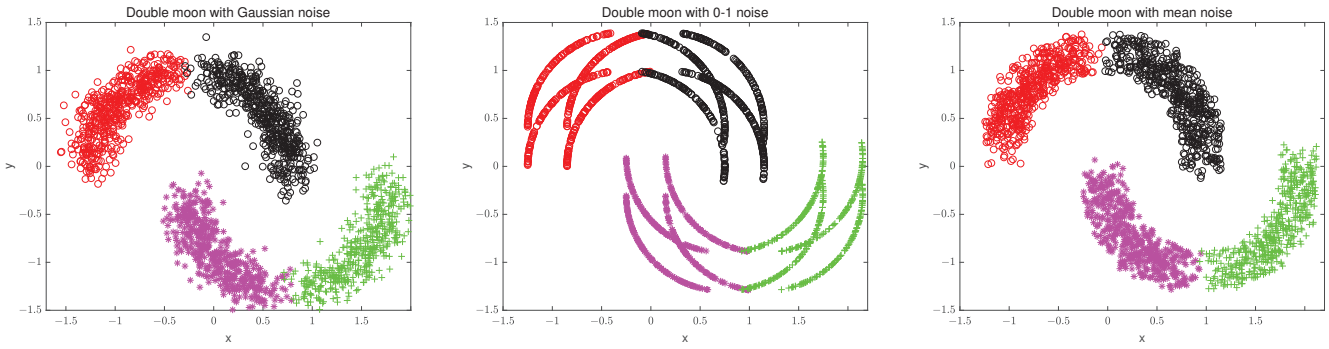



**Figure 10.** Diagram of "Double-moon" dataset with different types of noise.

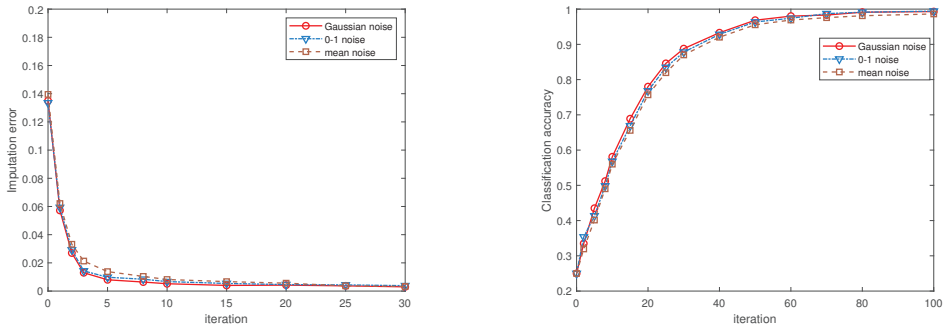



**Figure 11.** Learning curve of imputation error and classification accuracy of dPMDC on "Double Moon" dataset with different types of noise.

To validate the efficacy of the proposed approach in imputing the missing features, we investigated the imputation error of the proposed dPMDC algorithm under the different types of missing data, including MCAR, missing at random (MAR), and missing not at random (MNAR).

MCAR: The missing data features were completely independent of the variable value. In the following experiments, we used $P_m$ to measure the probability of missing values.

MAR: The probability of missing data features was related to the observed variables and unrelated to the characteristics of the unobserved data. In the following experiments, we randomly selected a pair of strongly correlated features. When one feature was larger than threshold $\delta$, the coupling feature was missing with probability $P_c$.

MNAR: The missing data features entirely depended on the unobserved variable itself. In the following experiments, we assumed that when the value of an attribute was greater than the threshold $\delta$, it was missing with probability 100%.

In the following experiments, for all the artificially generated PLL datasets (Double Moon, mHealth, Gas Drift, Pendigits, Ecoli, Segmentation, and Vertebral datasets), a

special number of noisy labels were added into candidate labels, such that the value of the controlled parameters the $s = 1$ and $\epsilon = 0.3$. For three real PLL datasets (Lost, Birdsong, and MRSCv2 datasets), no extra noisy label was added.

For the purpose of comparison, the imputation errors of the other existing imputation methods, including kNN imputation [13], subspace learning (SL) imputation [6], logistic regression (LR) imputation [14], extreme learning machine auto-encoder (ELM-AE) imputation [15], multi-layer auto-encoder (MAE) imputation [17], support vector regression imputation-based support vector machine (SVR-SVM) [22], and missing-data-importance-weighted auto-encoder imputation (MIWAE) [23] under the MACR, MAR, and MNAR assumptions, were also evaluated. All the simulation results are shown in Tables 5–7.

**Table 5.** Imputation error of different algorithms versus $P_m$ and $\epsilon$ on 10 used datasets under MCAR assumption, with the best performances shown in bold face.

| **Imputation Error** | | | | | | |
|---|---|---|---|---|---|---|
| **Dataset** | **Double Moon** | **mHealth** | **Gas Drift** | **Ecoli** | **Segmentation** | **Pendigits** |
| $P_m$ | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 |
| $\epsilon$ | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 |
| dPMDC | **0.003** | **0.010** | **0.004** | **0.005** | **0.005** | **0.009** |
| kNN | 0.007 | 0.037 | 0.042 | 0.012 | 0.052 | 0.062 |
| LR | 0.006 | 0.028 | 0.031 | 0.009 | 0.036 | 0.043 |
| SL | 0.006 | 0.022 | 0.018 | 0.009 | 0.026 | 0.018 |
| AE-ELM | 0.007 | 0.047 | 0.041 | 0.008 | 0.044 | 0.035 |
| MAE | **0.003** | 0.015 | 0.009 | 0.006 | 0.016 | 0.018 |
| SVR-SVM | 0.007 | 0.037 | 0.024 | 0.018 | 0.026 | 0.029 |
| MIWAE | 0.005 | 0.016 | 0.013 | 0.010 | 0.019 | 0.021 |
| Imputation Error | | | | | | |
| Dataset | Vertebral | Lost | MSRCv2 | Birdsong | | |
| $P_m$ | 0.3 | 0.3 | 0.3 | 0.3 | | |
| $\epsilon$ | 0.3 | / | / | / | | |
| dPMDC | **0.006** | **0.013** | **0.002** | **0.002** | | |
| kNN | 0.012 | 0.022 | 0.011 | 0.033 | | |
| LR | 0.015 | 0.020 | 0.006 | 0.022 | | |
| SL | 0.012 | 0.019 | 0.004 | 0.020 | | |
| AE-ELM | 0.013 | 0.018 | 0.006 | 0.026 | | |
| MAE | 0.010 | 0.015 | 0.004 | 0.007 | | |
| SVR-SVM | 0.013 | 0.021 | 0.009 | 0.028 | | |
| MIWAE | 0.010 | 0.017 | 0.006 | 0.012 | | |

**Table 6.** Imputation error of different algorithms versus $P_m$, $P_c$, and $\epsilon$ on 10 used datasets under MAR assumption, with the best performances shown in bold face.

| **Imputation Error** | | | | | | |
|---|---|---|---|---|---|---|
| **Dataset** | **Double Moon** | **mHealth** | **Gas Drift** | **Ecoli** | **Segmentation** | **Pendigits** |
| $\delta/\ P_c$ | 0.8/0.5 | 0.8/0.5 | 0.8/0.5 | 0.8/0.5 | 0.8/0.5 | 0.8/0.5 |
| $\epsilon$ | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 |
| dPMDC | **0.005** | **0.018** | **0.009** | **0.011** | **0.014** | **0.009** |
| kNN | 0.009 | 0.048 | 0.041 | 0.018 | 0.066 | 0.068 |
| LR | 0.008 | 0.035 | 0.028 | 0.023 | 0.048 | 0.048 |
| SL | 0.010 | 0.026 | 0.024 | 0.017 | 0.035 | 0.024 |
| AE-ELM | 0.011 | 0.055 | 0.040 | 0.026 | 0.053 | 0.047 |
| MAE | **0.005** | 0.019 | 0.013 | 0.020 | 0.023 | 0.024 |
| SVR-SVM | 0.007 | 0.027 | 0.034 | 0.022 | 0.038 | 0.039 |
| MIWAE | 0.009 | 0.024 | 0.018 | 0.019 | 0.026 | 0.026 |
| Imputation Error | | | | | | |
| Dataset | Vertebral | Lost | MSRCv2 | Birdsong | | |
| $P_m/P_c$ | 0.8/0.5 | 0.8/0.5 | 0.8/0.5 | 0.8/0.5 | | |
| $\epsilon$ | 0.3 | / | / | / | | |
| dPMDC | **0.012** | **0.016** | **0.008** | **0.007** | | |
| kNN | 0.022 | 0.028 | 0.017 | 0.039 | | |
| LR | 0.019 | 0.024 | 0.011 | 0.031 | | |
| SL | 0.020 | 0.022 | **0.008** | 0.026 | | |
| AE-ELM | 0.019 | 0.021 | 0.012 | 0.040 | | |
| MAE | 0.014 | 0.019 | 0.009 | 0.015 | | |
| SVR-SVM | 0.019 | 0.028 | 0.016 | 0.035 | | |
| MIWAE | 0.015 | 0.024 | 0.010 | 0.012 | | |

**Table 7.** Imputation error of different algorithms versus $\delta$ and $\epsilon$ on 10 used datasets under MNAR assumption, with the best performances shown in bold face.

| Imputation Error | | | | | | |
|---|---|---|---|---|---|---|
| **Dataset** | **Double Moon** | **mHealth** | **Gas Drift** | **Ecoli** | **Segmentation** | **Pendigits** |
| $\delta$ | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 |
| $\epsilon$ | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 |
| dPMDC | 0.037 | **0.031** | 0.029 | 0.031 | 0.034 | **0.029** |
| kNN | 0.069 | 0.078 | 0.068 | 0.078 | 0.081 | 0.088 |
| LR | 0.055 | 0.062 | 0.057 | 0.064 | 0.057 | 0.063 |
| SL | 0.063 | 0.056 | 0.064 | 0.041 | 0.046 | 0.041 |
| AE-ELM | 0.043 | 0.057 | 0.061 | 0.063 | 0.071 | 0.068 |
| MAE | 0.035 | 0.039 | 0.033 | 0.050 | **0.033** | 0.044 |
| SVR-SVM | 0.054 | 0.061 | 0.054 | 0.052 | 0.067 | 0.059 |
| MIWAE | **0.029** | 0.034 | **0.026** | **0.029** | 0.038 | 0.031 |
| Imputation Error | | | | | | |
| Dataset | Vertebral | Lost | MSRCv2 | Birdsong | | |
| $\delta$ | 0.9 | 0.9 | 0.9 | 0.9 | | |
| $\epsilon$ | 0.3 | / | / | / | | |
| dPMDC | **0.022** | 0.027 | **0.029** | 0.036 | | |
| kNN | 0.072 | 0.076 | 0.067 | 0.079 | | |
| LR | 0.046 | 0.054 | 0.051 | 0.057 | | |
| SL | 0.035 | 0.039 | 0.038 | 0.046 | | |
| AE-ELM | 0.042 | 0.046 | 0.051 | 0.054 | | |
| MAE | 0.035 | **0.025** | 0.037 | 0.035 | | |
| SVR-SVM | 0.052 | 0.064 | 0.045 | 0.058 | | |
| MIWAE | **0.022** | 0.029 | 0.031 | **0.028** | | |

From Tables 5 and 6, it can be observed that the comparison algorithms performed similar imputation performances under the MCAR and MAR assumptions. The following observations can be made: First, with the same degree of $\epsilon$, the kNN imputation method performed the worst in most cases, which indicated that simply exploiting the attribute values of neighboring data could not accurately describe the global data distribution. SL, LR, AL-ELM, and SVR-SVM imputation methods induced the imputed model based on global data distribution, and, thus, they performed better than the kNN imputation method. The MAE imputation method is based on an auto-encoder framework and can achieve relatively good imputation accuracy. However, as a supervised algorithm, it requires sufficient complete data with unambiguous labels to train the auto-encoder network to fine-tune the imputation results. When the value of $\epsilon$ was equivalent to 0.3, the ambiguous labels could have a negative effect on its imputation accuracy. Similarly, MIWAE outperformed most comparison algorithms by inducing the imputation model based on the auto-encoder framework. Even so, its imputation performance was still inferior to our suggested algorithm due to the coupling noise labels. Moreover, our suggested dPMDC performed significantly better than the other existing imputation approaches. Such a result indicates the superiority of the proposed imputation methods in characterizing the missing feature distribution under the MCAR and MAR assumptions.

By observing Table 7, we notice that, unlike the results in Tables 5 and 6, our proposed algorithm had no significant performance advantage when the MNAR assumption was used. Specifically, the imputation error of the proposed dPMDC algorithm was lower than that of kNN, SVR-SVM, LR, AE-ELM, and SL for all ten datasets. The imputation performance of the proposed dPMDC algorithm was better than that of MAE for six datasets and better than that of MIWAE for five datasets. We analyze the possible reasons in detail below. Under the MNAR assumption, all the features with values larger than the threshold were missing, making it challenging to characterize the distribution of these features. Therefore, it was difficult for our algorithm to achieve such excellent learning results, as in the cases of MCAR and MAR. Nevertheless, owing to the exploitation of weakly supervised information, the performance of our proposed algorithm was better

than those of most comparison algorithms and close to that of the MAE and MIWAE methods.

In order to highlight the superiority of the proposed algorithm in imputing the missing features, the Friedman test was implemented to determine whether there was a substantial distinction among the comparison algorithms [45]. Based on the amount of comparison methods $k_c = 8$ and the amount of used datasets $N_u = 10$, we computed the Friedman statistic $F_s = 40.81$ and its associated critical value 2.16. Obviously, the value of the Friedman statistic was larger than the critical value. This result indicated that the null hypothesis, stating that there was no significant difference among the evaluated comparison algorithms, was rejected at the 0.05 significance level.

Additionally, the Bonferroni–Dunn test was employed to determine whether our suggested method significantly surpassed the other comparative algorithms [45]. Figure 12 illustrates the average rankings of all comparison algorithms depicted by black lines, with the proposed dPMDC serving as the controlled algorithm in this evaluation.

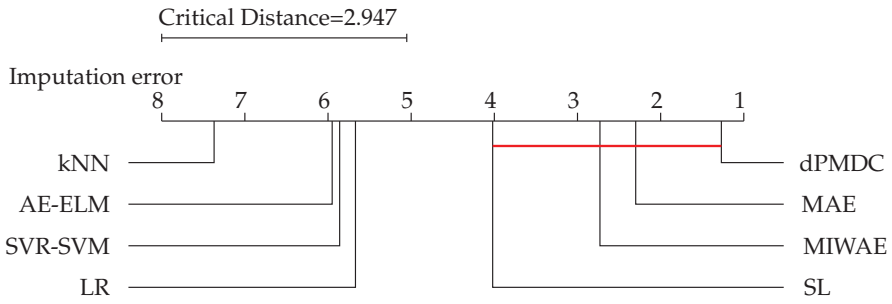



**Figure 12.** Comparison of dPMDC algorithm (the control algorithm) in contrast to other imputation methods using the Bonferroni–Dunn test.

The comparison algorithms whose average ranks fell within one CD compared to the controlled algorithms at a significance level of 0.05 and a critical difference (CD) of performance $CD = 2.576$ are connected by red lines. Otherwise, no line links them. Figure 12 illustrates that our suggested method achieved the highest ranking among all comparative algorithms. Furthermore, the results of the Bonferroni–Dunn test demonstrated that the performance of our proposed method was significantly superior to that of the LR, AE-ELM, SVR-SVM, and kNN imputation methods.

To assess the classification performance of the proposed dPMDC, we tested the classification accuracy of the proposed dPMDC method under the MCAR, MAR, and MNAR assumptions on ten datasets. We also evaluated the classification performance of four MDC methods, including dS$^2$MDC [6], ALS-SVM [21], SVR-SVM [22], and MIWAE [23], for comparison. Moreover, to highlight the advantages of the proposed algorithm in simultaneously handling the ambiguous labels and missing features, the performances of three novel, state-of-the-art PLL algorithms, including dS$^2$PLL [9], LWS [25], and VALEN [27], were simulated based on the complete features imputed by the existing imputation method. According to the results in the previous experiments, the imputation error of MAE was second only to our proposed method and significantly outperformed the other comparison methods. So, we utilized MAE to impute the missing features of the training data in the following simulations. In order to distinguish them from the original algorithms, we respectively use im-dS$^2$PLL, im-LWS, and im-VALEN to denote them. All the simulation results are given in Tables 8–10 and the average ranks of all the comparison algorithms are depicted in Figure 13.

**Table 8.** Classification accuracy of different algorithms versus $P_m$ and $\epsilon$ on 10 used datasets under MCAR assumption, with the best performances shown in bold face.

| Classification Accuracy | | | | | | |
|---|---|---|---|---|---|---|
| **Dataset** | **Double Moon** | **mHealth** | **Gas Drift** | **Ecoli** | **Segmentation** | **Pendigits** |
| $P_m$ | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 |
| $\epsilon$ | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 |
| dPMDC | **0.983** | **0.960** | **0.884** | **0.793** | **0.765** | **0.855** |
| im-dS$^2$PLL | 0.975 | 0.932 | 0.862 | 0.779 | 0.723 | 0.809 |
| dS$^2$MDC | 0.966 | 0.916 | 0.831 | 0.756 | 0.714 | 0.822 |
| ALS-SVM | 0.926 | 0.903 | 0.818 | 0.749 | 0.697 | 0.782 |
| im-LWS | 0.976 | 0.940 | 0.861 | 0.776 | 0.747 | 0.827 |
| im-VALEN | 0.980 | 0.937 | 0.857 | 0.778 | 0.752 | 0.819 |
| SVR-SVM | 0.967 | 0.939 | 0.827 | 0.766 | 0.736 | 0.805 |
| MIWAE | 0.973 | 0.946 | 0.839 | 0.781 | 0.759 | 0.825 |
| Classification Accuracy | | | | | | |
| Dataset | Vertebral | Lost | MSRCv2 | Birdsong | | |
| $P_m$ | 0.3 | 0.3 | 0.3 | 0.3 | | |
| $\epsilon$ | 0.3 | / | / | / | | |
| dPMDC | **0.822** | **0.528** | **0.443** | **0.628** | | |
| im-dS$^2$PLL | 0.795 | 0.501 | 0.425 | 0.609 | | |
| dS$^2$MDC | 0.776 | 0.491 | 0.431 | 0.601 | | |
| ALS-SVM | 0.763 | 0.375 | 0.395 | 0.573 | | |
| im-LWS | 0.807 | 0.504 | 0.414 | 0.602 | | |
| im-VALEN | 0.804 | 0.489 | 0.427 | 0.582 | | |
| SVR-SVM | 0.798 | 0.438 | 0.409 | 0.575 | | |
| MIWAE | 0.802 | 0.492 | 0.416 | 0.604 | | |

**Table 9.** Classification accuracy of different algorithms versus $P_m$, $P_c$, and $\epsilon$ on 10 used datasets under MAR assumption, with the best performances shown in bold face.

| Classification Accuracy | | | | | | |
|---|---|---|---|---|---|---|
| **Dataset** | **Double Moon** | **mHealth** | **Gas Drift** | **Ecoli** | **Segmentation** | **Pendigits** |
| $\delta/\ P_c$ | 0.8/0.5 | 0.8/0.5 | 0.8/0.5 | 0.8/0.5 | 0.8/0.5 | 0.8/0.5 |
| $\epsilon$ | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 |
| dPMDC | **0.975** | **0.957** | **0.867** | **0.786** | **0.759** | **0.850** |
| im-dS$^2$PLL | 0.968 | 0.930 | 0.840 | 0.771 | 0.709 | 0.802 |
| dS$^2$MDC | 0.960 | 0.909 | 0.815 | 0.747 | 0.701 | 0.816 |
| ALS-SVM | 0.913 | 0.887 | 0.798 | 0.741 | 0.685 | 0.772 |
| im-LWS | 0.965 | 0.922 | 0.842 | 0.768 | 0.740 | 0.821 |
| im-VALEN | 0.974 | 0.913 | 0.835 | 0.770 | 0.746 | 0.812 |
| SVR-SVM | 0.960 | 0.921 | 0.809 | 0.760 | 0.731 | 0.793 |
| MIWAE | 0.968 | 0.932 | 0.821 | 0.774 | 0.746 | 0.814 |
| Classification Accuracy | | | | | | |
| Dataset | Vertebral | Lost | MSRCv2 | Birdsong | | |
| $\delta/\ P_c$ | 0.8/0.5 | 0.8/0.5 | 0.8/0.5 | 0.8/0.5 | | |
| $\epsilon$ | 0.3 | / | / | / | | |
| dPMDC | **0.815** | **0.514** | **0.443** | **0.605** | | |
| im-dS$^2$PLL | 0.769 | 0.493 | 0.425 | 0.601 | | |
| dS$^2$MDC | 0.766 | 0.486 | 0.431 | 0.584 | | |
| ALS-SVM | 0.741 | 0.370 | 0.395 | 0.565 | | |
| im-LWS | 0.793 | 0.493 | 0.414 | 0.594 | | |
| im-VALEN | 0.789 | 0.480 | 0.427 | 0.572 | | |
| SVR-SVM | 0.780 | 0.421 | 0.409 | 0.570 | | |
| MIWAE | 0.791 | 0.485 | 0.416 | 0.590 | | |

The simulation results in Tables 8–10 indicate that the performance of ALS-SVM was significantly inferior to that of the other comparison algorithms. The main reason was that ALS-SVM only reduced the weight of data with a high proportion of missing attributes without filling in the missing attributes. Different from ALS-SVM, SVR-SVM trained the multi-class classifier based on the training data imputed by the induced SVR model, making its performance significantly better than the ALS-SVM. However, the simulation results in Tables 5–7 show that the performance of the induced SVR imputation model was relatively poor, which, in turn, had a negative impact on the performance of the SVM classifier. The dS$^2$MDC algorithm performed better than the ALS-SVM since it benefited from the

interplay of missing feature imputation and model induction in the learned subspace. On the other hand, by suffering from the negative effect of noisy labels, its performance was worse than that of our proposed algorithm. It can also be observed that the dS$^2$PLL, LWS, and VALEN algorithms achieved superior performance compared to ALS-SVM by imputing the missing features via the MAE imputation method and eliminating the effect of ambiguous labels using their disambiguation strategies. But, their performances were still poorer than that of the proposed dPMDC algorithm. MIWAE designed an integral framework that jointly addressed missing data imputation and classifier induction. Benefiting from this, the performance of the MIWAE algorithm ranked second among all the comparison algorithms. Additionally, our proposed dPMDC algorithm performed best among eight comparison algorithms in almost all cases. Although our algorithm had a relatively high imputation error under the MNAR assumption, the classifier's performance still maintained a high level due to the interaction between the imputation model and the classifier.

**Table 10.** Classification accuracy of different algorithms versus $\delta$ and $\epsilon$ on 10 used datasets under MNAR assumption, with the best performances shown in bold face.

| Classification Accuracy | | | | | | |
|---|---|---|---|---|---|---|
| **Dataset** | **Double Moon** | **mHealth** | **Gas Drift** | **Ecoli** | **Segmentation** | **Pendigits** |
| $\delta$ | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 |
| $\epsilon$ | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 |
| dPMDC | **0.971** | **0.953** | **0.832** | 0.731 | **0.721** | **0.823** |
| im-dS$^2$PLL | 0.960 | 0.926 | 0.812 | 0.716 | 0.673 | 0.791 |
| dS$^2$MDC | 0.953 | 0.901 | 0.789 | 0.705 | 0.661 | 0.786 |
| ALS-SVM | 0.910 | 0.872 | 0.770 | 0.693 | 0.648 | 0.757 |
| im-LWS | 0.964 | 0.908 | 0.825 | 0.714 | 0.703 | 0.793 |
| im-VALEN | 0.968 | 0.912 | 0.803 | 0.705 | 0.706 | 0.795 |
| SVR-SVM | 0.951 | 0.919 | 0.779 | 0.697 | 0.684 | 0.786 |
| MIWAE | 0.963 | 0.937 | 0.816 | **0.738** | 0.706 | 0.802 |

| Classification Accuracy | | | | |
|---|---|---|---|---|
| Dataset | Vertebral | Lost | MSRCv2 | Birdsong |
| $\delta$ | 0.9 | 0.9 | 0.9 | 0.9 |
| $\epsilon$ | 0.3 | / | / | / |
| dPMDC | **0.775** | **0.505** | **0.435** | **0.583** |
| im-dS$^2$PLL | 0.734 | 0.490 | 0.410 | 0.556 |
| dS$^2$MDC | 0.736 | 0.489 | 0.413 | 0.561 |
| ALS-SVM | 0.704 | 0.361 | 0.382 | 0.552 |
| im-LWS | 0.752 | 0.496 | 0.410 | 0.570 |
| im-VALEN | 0.757 | 0.474 | 0.408 | 0.557 |
| SVR-SVM | 0.750 | 0.410 | 0.392 | 0.528 |
| MIWAE | 0.773 | 0.487 | 0.403 | 0.575 |

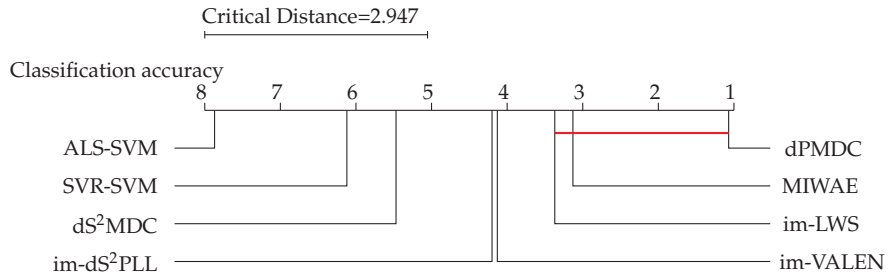



**Figure 13.** Comparison of dPMDC algorithm (the control algorithm) in contrast to other comparing classification algorithms using the Bonferroni–Dunn test.

Similar to the previous experiment, we used the Friedman test [45] to justify the performance difference among the eight compared algorithms. According to the theory of the Friedman test, when the significance level was set as 0.05, we could calculate the Friedman statistic value $F_s = 40.38$ and its associated critical value 2.16. We noticed that

the Friedman statistic was significantly larger than the critical value, indicating that there existed a significant difference among the evaluated comparison algorithms.

Additionally, the Bonferroni–Dunn test was employed to compare the performance difference between our suggested method and the other comparative algorithms [45]. All the results of the Bonferroni–Dunn test are depicted in Figure 13. Figure 13 illustrates that the learning performance of our suggested method was significantly better than that of the im-VALEN, dS$^2$MDC, im-dS$^2$PLL, SVR-SVM, and ALS-SVM algorithms.

To testify to the robustness of the proposed algorithm against coupling noisy labels, we compared the classification accuracy of the comparison algorithms versus $\epsilon$ on four artificial PLL datasets (Pendigits, Ecoli, Segmentation, and Vertebral datasets). From Figures 14 and 15, we can see that as the probability of co-occurring between the correct label and another coupling noisy label $\epsilon$ gradually increased, the classification performances of all the considered algorithms gradually deteriorated. Among all the comparison algorithms, we found that our proposed algorithm demonstrated superior performance, especially when the value of $\epsilon$ was smaller than 0.3. When the proportion of coupling noisy labels was larger than 0.3, the performance significance of the dPMDC algorithm gradually reduced. Such a simulation result validates the benefits of our proposed algorithm in handling a small amount of coupling noisy labels.

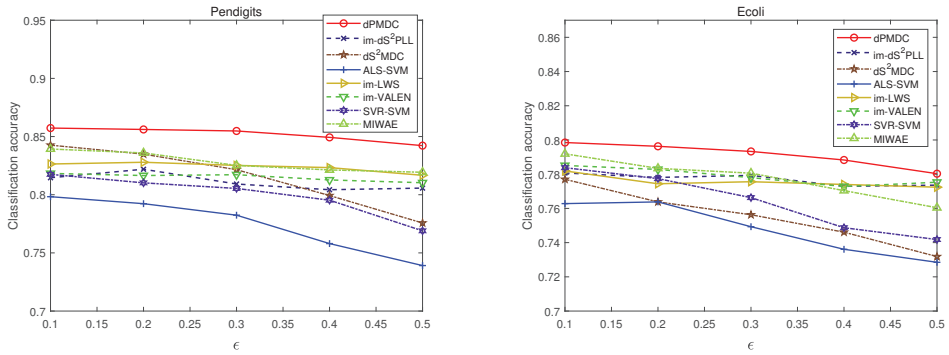



**Figure 14.** Classification accuracy of dPMDC versus the co-occurrence probability between a coupling noisy label and the correct label $\epsilon$ on "Pendigits" and "Ecoli" datasets.

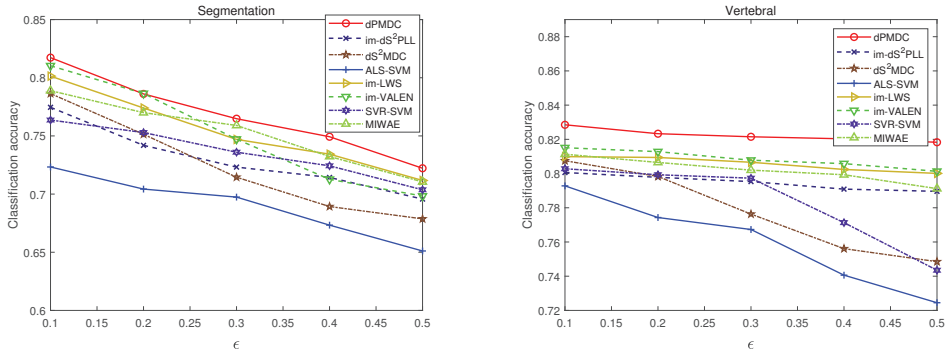



**Figure 15.** Classification accuracy of dPMDC versus the co-occurrence probability between a coupling noisy label and the correct label $\epsilon$ on "Segmentation" and "Vertebral" datasets.

Finally, we evaluated the CPU time of all the considered algorithms on the ten used datasets, shown in Table 11. It should be noted that for the distributed learning algorithms, all the computation operations were performed at the $J$ node over a network. For centralized learning algorithms, all the computation operations were centralized at a single fusion node. Owing to the distributed parallel computation, the CPU times of all the distributed learning methods were significantly lower than those of the centralized learning methods. Compared with im-dS$^2$PLL, our proposed dPMDC and the dS$^2$MDC required fewer com-

putations during the process of imputation model induction. Therefore, the CPU times of the proposed dPMDC and dS$^2$MDC were significantly shorter than those of im-dS$^2$PLL.

**Table 11.** CPU times of different algorithms on 10 used datasets under MNAR assumption, with the lowest CPU times performances shown in bold face.

| CPU Times | | | | | | |
|---|---|---|---|---|---|---|
| **Dataset** | **Double Moon** | **mHealth** | **Gas Drift** | **Ecoli** | **Segmentation** | **Pendigits** |
| dPMDC | **142.42** | 226.07 | **520.80** | **23.31** | **120.67** | 136.88 |
| im-dS$^2$PLL | 463.79 | 665.24 | 1568.06 | 97.60 | 437.70 | 462.88 |
| dS$^2$MDC | 147.83 | **219.29** | 535.80 | 23.85 | 124.29 | **134.14** |
| ALS-SVM | 484.23 | 779.94 | 1754.70 | 72.26 | 434.16 | 483.92 |
| im-LWS | 2493.65 | 4075.60 | 7434.66 | 550.53 | 2484.90 | 2528.03 |
| im-VALEN | 2562.82 | 4110.87 | 7353.02 | 435.44 | 2469.20 | 2520.75 |
| SVR-SVM | 539.50 | 874.93 | 1963.65 | 86.71 | 502.92 | 546.81 |
| MIWAE | 2569.84 | 4032.80 | 7622.61 | 430.96 | 2445.47 | 2385.89 |
| CPU Times | | | | | | |
| Dataset | Vertebral | Lost | MSRCv2 | Birdsong | | |
| dPMDC | **20.74** | **85.80** | **154.62** | **61.54** | | |
| im-dS$^2$PLL | 114.08 | 401.24 | 477.25 | 356.93 | | |
| dS$^2$MDC | 21.26 | 88.37 | 160.97 | 65.63 | | |
| ALS-SVM | 85.03 | 384.71 | 620.38 | 220.45 | | |
| im-LWS | 416.14 | 1980.48 | 2457.54 | 1207.67 | | |
| im-VALEN | 418.32 | 1876.09 | 2427.24 | 1498.37 | | |
| SVR-SVM | 96.93 | 438.56 | 670.23 | 245.30 | | |
| MIWAE | 421.86 | 1866.07 | 2525.24 | 1289.14 | | |

## 5. Conclusions

In this article, we addressed the issue of the distributed classification of partial label incomplete data and proposed the dPMDC algorithm. In our proposed algorithm, by jointly exploiting information from the estimated labeling confidence and partially observable features, a GMM-based imputed model is learned, which can distributively impute the partially missing features. Based on the imputed data, a high-precision classifier modeled by the non-linear random feature map is induced. A series of simulations were performed to validate the efficacy of the suggested dPMDC method. Simulation results across several datasets indicate that our proposed approach surpasses current imputation methods in imputation accuracy and outperforms state-of-the-art PLL algorithms using complete features imputed by the existing methods.

A potential limitation of this algorithm is that it can only be used to handle the problems of MDC in static networks. Therefore, developing algorithms for dynamic network topologies is a possible future direction. Considering that the rich semantics of the training data are simultaneously characterized by multiple label variables in many real applications, as opposed to a single label variable, we would like to extend the partial label missing data classification method into the domain of multi-dimensional classification. Furthermore, developing an imputation method for the case of MNAR is also an interesting research direction for the future.

**Author Contributions:** Conceptualization, Z.X.; Methodology, Z.X.; Software, Z.C.; Validation, Z.C.; Formal analysis, Z.X.; Investigation, Z.C.; Resources, Z.C.; Data curation, Z.C.; Writing—original draft, Z.C.; Writing—review and editing, Z.C.; Visualization, Z.C.; Supervision, Z.X.; Project administration, Z.C.; Funding acquisition, Z.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partly supported by the National Natural Science Foundation of China (grant no. 62201398).

**Data Availability Statement:** The data presented in this study are openly available in UCI Repository of Machine Learning Databases at http://archive.ics.uci.edu/ml/datasets.html [41].

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

## Appendix A. Proof of Theorem 2

For convenience, we combine the objective function of the imputation model and the multi-class classifier, i.e.,

$$\mathcal{L}_{j,t} = -Q^{'}(\boldsymbol{\theta}_j|\boldsymbol{\theta}_{j,t}) + F_{j,t} \tag{A1}$$

Based on Lemma 1, we know that all the local estimates $\boldsymbol{\theta}_{j,t}$ and $\boldsymbol{W}_{j,t}$ can converge to the corresponding average values $\bar{\boldsymbol{\theta}}_t$ and $\bar{\boldsymbol{W}}_t$ through a sufficient number of iterations. It should be noted that here, in order to make the subsequent proof clearer, we merge all the parameters of the GMM into one variable $\bar{\boldsymbol{\theta}}$ and combine all the weight vectors $\{\bar{\boldsymbol{w}}_c\}_c$ into a uniform matrix $\bar{W}$.

Based on this, we can rewrite the main update equation of the proposed dPMDC as follows:

$$\bar{\boldsymbol{\theta}}_{t+1} = \arg\min \mathcal{L}_j(\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{W}}_t), \tag{A2a}$$

$$\bar{\boldsymbol{W}}_{t+1} = \bar{\boldsymbol{W}}_t - \zeta_{t+1}\nabla_{\bar{\boldsymbol{W}}}\mathcal{L}_j(\bar{\boldsymbol{\theta}}_{t+1}, \bar{\boldsymbol{W}}), \tag{A2b}$$

Let $\mathcal{L}_{j,t} = \mathcal{L}_j(\bar{\boldsymbol{\theta}}_t, \bar{\boldsymbol{W}}_t)$ and $\mathcal{L}^{'}_{j,t} = \mathcal{L}_j(\bar{\boldsymbol{\theta}}_{t+1}, \bar{\boldsymbol{W}}_t)$; then, we can obtain

$$\mathcal{L}^{'}_{j,t} - \mathcal{L}^{'}_{j,t+1} = \mathcal{L}_j(\bar{\boldsymbol{\theta}}_{t+1}, \bar{W}_t - \zeta_{t+1}\nabla_{\bar{\boldsymbol{W}}}\mathcal{L}^{'}_{j,t}). \tag{A3}$$

Referring to [6], we can see that for the time-varying variable $\boldsymbol{s}_{t+1} = \boldsymbol{s}_t - \zeta_{t+1}\nabla g(\boldsymbol{s}_t)$, we have the following inequality:

$$g(\boldsymbol{s}_{t+1}) \leq g(\boldsymbol{s}_t) + < \nabla g(\boldsymbol{s}_t), \boldsymbol{s}_{t+1} - \boldsymbol{s}_t > + \frac{M}{2}||\boldsymbol{s}_{t+1} - \boldsymbol{s}_t||_2^2. \tag{A4}$$

By combining the above equations, we can obtain

$$g(\boldsymbol{s}_t) - g(\boldsymbol{s}_{t+1}) \geq (\zeta_{t+1} - \frac{\zeta_{t+1}^2 M}{2})||\boldsymbol{s}_{t+1} - \boldsymbol{s}_t||_2^2. \tag{A5}$$

Here, when $\zeta_{t+1} < 2/M$, the coefficient $\zeta_{t+1} - \frac{\zeta_{t+1}^2 M}{2}$ is positive.

By induction, we know that

$$\mathcal{L}^{'}_{j,t} - \mathcal{L}^{'}_{j,t+1} \geq Z||\nabla_{\bar{\boldsymbol{W}}}\mathcal{L}^{'}_{j,t}||_F^2, \tag{A6}$$

where the coefficient $Z = \zeta_{t+1} - \frac{\zeta_{t+1}^2 M}{2} > 0$.

On the basis of the assumption that $\mathcal{L}$ is convex, we can further obtain

$$\begin{aligned} \mathcal{L}^{'}_{j,t} - \mathcal{L}^{'*}_{j} &\leq \nabla_{\bar{\boldsymbol{W}}}\mathcal{L}^{'}_{j,t}(\bar{\boldsymbol{W}}_t - \bar{\boldsymbol{W}}^*) \\ &\leq ||\bar{\boldsymbol{W}}_t - \bar{\boldsymbol{W}}^*||_F||\nabla_{\bar{\boldsymbol{W}}}\mathcal{L}^{'}_{j,t}||_F^2 \\ &\leq G||\bar{\boldsymbol{W}}_t - \bar{\boldsymbol{W}}^*||_F \end{aligned} \tag{A7}$$

where $G$ denotes the upper bound of $||\nabla_{\bar{\boldsymbol{W}}}\mathcal{L}^{'}_{j,t}||$.

Based on the above equations, we can obtain

$$\begin{aligned}\mathcal{L}_t^{'} - \mathcal{L}_{t+1}^{'} &\geq Z||\nabla_{\bar{W}}\mathcal{L}_{j,t}^{'}||_F^2 \\ &\geq \frac{Z}{||\bar{\boldsymbol{W}}_t - \bar{\boldsymbol{W}}^*||_F^2}(\bar{\boldsymbol{W}}_t - \bar{\boldsymbol{W}}^*)^2 \\ &\geq \frac{Z}{R^2}(\bar{\boldsymbol{W}}_t - \bar{\boldsymbol{W}}^*)^2\end{aligned} \tag{A8}$$

where $R = \max\{||\bar{\boldsymbol{W}}_t - \bar{\boldsymbol{W}}^*||_F\}$.

In order to carry out the following analysis, a lemma in [6] should be given.

**Lemma A1.** *For a non-negative sequence $\beta_t$ satisfying the conditions $\beta_t - \beta_{t+1} \geq a\beta_t^2$ and $\beta_0 \leq \frac{1}{ai}$, we have*

$$\beta_t \leq \frac{1}{a(t+i)}, \tag{A9}$$

*where a and i are positive coefficients.*

Recalling Lemma A1, we let $\beta_t = \mathcal{L}_{j,t}^{'} - \mathcal{L}_{j,t}^{'*}$, $a = Z/R^2$ and $i = R/(ZG)$ and then obtain

$$\mathcal{L}_{j,t}^{'} - \mathcal{L}_{j}^{'*} \leq \frac{R^2}{Z(t + \frac{R}{ZG})}. \tag{A10}$$

Therefore, we can derive that

$$\lim_{t\to\infty} \mathcal{L}_{j,t}^{'} - \mathcal{L}_{j}^{'*} = 0. \tag{A11}$$

Based on (A11), we have $\lim_{t\to\infty} \bar{\boldsymbol{W}}_t - \boldsymbol{W}* = 0$. The proof is complete.

## References


1. Shen, X.; Liu, Y. Privacy-preserving distributed estimation over multitask networks. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 1953–1965. [CrossRef]
2. Chen, S.; Liu, Y. Robust distributed parameter estimation of nonlinear systems with missing data over networks. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 2228–2244. [CrossRef]
3. Miao, X.D.; Liu, Y.; Zhao, H.Q.; Li, C.G. Distributed online one-class support vector machine for anomaly detection over networks. *IEEE Trans. Cybern.* **2019**, *49*, 1475–1488. [CrossRef]
4. Liu, M.; Yang, K.; Zhao, N.; Chen, Y.; Song, H.; Gong, F. Intelligent signal classification in industrial distributed wireless sensor networks based industrial internet of things. *IEEE Trans. Ind. Inf.* **2020**, *17*, 4946–4956. [CrossRef]
5. Shen, X.; Liu, Y. Distributed differential utility/cost analysis for privacy protection. *IEEE Signal Process. Let.* **2019**, *26*, 1436–1440. [CrossRef]
6. Xu, Z.; Liu, Y.; Li, C. Distributed semi-supervised learning with missing data. *IEEE Trans. Cybern.* **2021**, *51*, 6165–6178. [CrossRef] [PubMed]
7. Liu, Y.; Xu, Z.; Li, C. Distributed online semi-supervised support vector machine. *Inf. Sci.* **2018**, *466*, 236–257. [CrossRef]
8. Carminati, M.; Kanoun, O.; Ullo, S.L.; Marcuccio, S. Prospects of distributed wireless sensor networks for urban environmental monitoring. *IEEE Aerosp. Electron. Syst. Mag.* **2019**, *34*, 44–52. [CrossRef]
9. Liu, Y.; Xu, Z.; Zhang, C. Distributed semi-supervised partial label learning over networks. *IEEE Trans. Artif. Intell.* **2022**, *3*, 414–425. [CrossRef]
10. Xu, Z.; Liu, Y.; Li, C. Distributed information theoretic semi-supervised learning for multi-label classification. *IEEE Trans. Cybern.* **2022**, *52*, 821–835. [CrossRef]
11. Garcia-Laencina, P.J.; Sancho-Gomez, J.L.; Figueiras-Vidal, A.R. Pattern classification with missing data: A review. *Neural Comput. Appl.* **2010**, *19*, 263–282. [CrossRef]
12. Emmanuel, T.; Maupong, T.; Mpoeleng, D.; Semong, T.; Tabona, O. A survey on missing data in machine learning. *J. Big Data* **2021**, *8*, 1–37. [CrossRef] [PubMed]

13. Murti, D.M.P.; Pujianto, U.; Wibawa, A.P.; Akbar, M.I. K-nearest neighbor (k-nn) based missing data imputation. In Proceedings of the 5th International Conference on Science in Information Technology (ICSITech), Yogyakarta, Indonesia, 23–24 October 2019; pp. 83–88.
14. Alabadla, M.; Sidi, F.; Ishak, I.; Ibrahim, H.; Affendey, L.; Ani, Z.; Jabar, M.; Bukar, U.; Devaraj, N.; Muda, A.; et al. Systematic review of using machine learning in imputing missing values. *IEEE Access* **2022**, *10*, 44483–44502 [CrossRef]
15. Lu, C.; Mei, Y. An imputation method for missing data based on an extreme learning machine auto-encoder. *IEEE Access* **2018**, *6*, 52930–52935. [CrossRef]
16. Pan, Z.; Wang, Y.; Wang, K.; Chen, H.; Yang, C.; Gui, W. Imputation of missing values in time series using an adaptive-learned median-filled deep autoencoder. *IEEE Trans. Cybern.* **2022**, *53*, 695–706. [CrossRef]
17. Lai, X.; Wu, X.; Zhang, L. Autoencoder-based multi-task learning for imputation and classification of incomplete data. *Appl. Soft Comput.* **2021**, *98*, 106838. [CrossRef]
18. Zhang, Y.; Li, M.; Wang, S.; Dai, S. Gaussian mixture model clustering with incomplete data. *ACM Trans. Multimed. Comput. Commun. Appl.* **2021**, *17*, 1–14. [CrossRef]
19. Sang, H.; Kim, J.; Lee, D. Semiparametric fractional imputation using gaussian mixture models for handling multivariate missing data. *J. Am. Stat. Asso.* **2022**, *117*, 654–663. [CrossRef]
20. Chen, S.; Liu, Y. Distributed multi-kernel learning based on Gaussian mixture model with missing Data. In Proceedings of the 2nd International Conference on Signal Processing, Computer Networks, and Communications (SPCNC 2023), Xiamen, China, 8–10 December 2023; pp. 106–111.
21. Wang, G.; Deng, Z.; Choi, K.S. Tackling missing data in community health studies using additive LS-SVM classifier. *IEEE J. Biomed. Health Inform.* **2018**, *22*, 579–587. [CrossRef]
22. Palanivinayagam, A.; Damasevicius, R. Effective handling of missing values in datasets for classification using machine learning methods. *Information* **2023**, *14*, 92. [CrossRef]
23. Kim, S.; Kim, H.; Yun, E.; Lee, H.; Lee, J.; Lee, J. Probabilistic imputation for time-series classification with missing data. In Proceedings of the 40th International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; pp. 16654–16667.
24. Wang, W.; Zhang, M.-L. Semi-supervised partial label learning via confidence-rated margin maximization. In Proceedings of the 34th Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020; pp. 6982–6993.
25. Wen, H.; Cui, J.; Hang, H.; Liu, J.; Wang, Y.; Lin, Z. Leveraged weighted loss for partial label learning. In Proceedings of the 38th International Conference on Machine Learning, Virtual Event, 18–24 July 2021; pp. 11091–11100.
26. Yu, X.R.; Wang, D.B.; Zhang, M.L. Dimensionality reduction for partial label learning: A unified and adaptive approach. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 3765–3782. [CrossRef]
27. Xu, N.; Qiao, C.; Zhao, Y.; Geng, X.; Zhang, M.-L. Variational label enhancement for instance-dependent partial label learning. *IEEE Trans. Patt. Anal. Mach. Intell.* **2024**, *46*, 11298–11313. [CrossRef]
28. Fang, J.P.; Zhang, M.L. Partial multi-label learning via credible label elicitation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 3587–3599. [CrossRef]
29. Xie, M.-K.; Huang, S. Semi-supervised partial multi-label learning. In Proceedings of the 2020 IEEE International Conference on Data Mining, Sorrento, Italy, 17–20 November 2020; pp. 691–700.
30. Yu, T.; Yu, G.; Wang, J.; Domeniconi, C.; Zhang, X. Partial multi-label learning using label compression. In Proceedings of the 2020 IEEE International Conference on Data Mining, Sorrento, Italy, 17–20 November 2020; pp. 761–770.
31. Xie, M.K.; Huang, S.J. Partial multi-label learning with noisy label identification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3676–3687.
32. Liu, B.Q.; Jia, B.B.; Zhang, M.-L. Towards enabling binary decomposition for partial multi-label learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 13203–13217. [CrossRef] [PubMed]
33. Xu, Z.; Chen, W. Distributed Semi-Supervised Partial Multi-Label Learning over Networks. *Electronics* **2024**, *13*, 4754. [CrossRef]
34. Vedaldi, A.; Zisserman, A. Efficient additive kernels via explicit feature maps. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 480–492. [CrossRef]
35. Pu, X.; Li, C. Probabilistic Information-Theoretic Discriminant Analysis for Industrial Label-Noise Fault Diagnosis. *IEEE Trans. Ind. Inf.* **2021**, *17*, 2664–2674. [CrossRef]
36. Hero, A.O.; Fessler, J.A. Convergence in norm for alternating expectation maximization (em) type algorithms. *Stat. Sin.* **1995**, *5*, 41–54.
37. Petersen, K.B.; Pedersen, M.S. *The Matrix Cookbook*; Technical University of Denmark Press: Kongens Lyngby, Denmark, 2012.
38. Niu, G.; Dai, B.; Yamada, M.; Sugiyama, M. Information-theoretic semi-supervised metric learning via entropy regularization. *Neural Comput.* **2014**, *26*, 1717–1762. [CrossRef]
39. Wang, S.; Li, C. Distributed stochastic algorithm for global optimization in networked system. *J. Optim. Theory Appl.* **2018**, *179*, 1001–1007. [CrossRef]

40. Chen, S.; Liu, Y. Distributed Personalized Imputation Based on Gaussian Mixture Model for Missing Data. *Neural Comput. Appl.* **2024**, *36*, 14237–14250. [CrossRef]
41. Blake, C.; Merz, C. UCI Repository of Machine Learning Databases. Available online: http://archive.ics.uci.edu/ml/datasets.html. (accessed on 1 July 2024).
42. Cour, T.; Sapp, B.; Jordan, C.; Taskar, B. Learning from ambiguously labeled images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09), Miami, FL, USA, 20–25 June, 2009 pp. 919–926.
43. Briggs, F.; Fern, X.Z.; Raich, R. Rank-loss support instance machines for MIML instance annotation. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12), Beijing, China, 12–16 August 2012; pp. 534–542.
44. Liu, L.; Dietterich, T. A conditional multinomial mixture model for superset label learning. In Proceedings of the 26th Neural Information Processing Systems. (NeurIPS'12), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 557–565.
45. Demsar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.

*electronics*

MDPI

*Article*

# Distributed Partial Label Multi-Dimensional Classification via Label Space Decomposition

**Zhen Xu [1,*,†] and Sicong Chen [2,†]**

[1] College of Computer Science and Artificial Intelligence Engineering, Wenzhou University, Wenzhou 325006, China
[2] Kasco Signal Co., Ltd., Shanghai 200072, China; 3120100951@zju.edu.cn
* Correspondence: 20200588@wzu.edu.cn
† These authors contributed equally to this work.

**Abstract:** Multi-dimensional classification (MDC), in which the training data are concurrently associated with numerous label variables across many dimensions, has garnered significant interest recently. Most of the current MDC methods are based on the framework of supervised learning, which induces a predictive model from a large amount of precisely labeled data. So, they are challenged to obtain satisfactory learning results in the situation where the training data are not annotated with precise labels but assigned with ambiguous labels. Besides, the current MDC algorithms only consider the scenario of centralized learning, where all training data are handled at a single node for the purpose of classifier induction. However, in some real applications, the training data are not consolidated at a single fusion center, but rather are dispersedly distributed among multiple nodes. In this study, we focus on the problem of decentralized classification involving partial multi-dimensional data that have partially accessible candidate labels, and develop a distributed method called dPL-MDC for learning with these partial labels. In this algorithm, we conduct one-vs.-one decomposition on the originally heterogeneous multi-dimensional output space, such that the problem of partial MDC can be transformed into the issue of distributed partial multi-label learning. Then, by using several shared anchor data to characterize the global distribution of label variables, we propose a novel distributed approach to learn the label confidence of the training data. Under the supervision of recovered credible labels, the classifier can be induced by exploiting the high-order label dependencies from a common low-dimensional subspace. Experiments performed on various datasets indicate that our proposed method is capable of achieving learning performance in distributed partial MDC.

**Keywords:** distributed processing; partial label learning; multi-dimensional classification

## 1. Introduction

The learning problems of traditional classification methods are usually formalized under the frameworks of binary classification [1–4] and multi-class classification [5], which learn the classifier under the supervision of a single label variable. In numerous real-world scenarios, the complex semantics of training data are conveyed through multiple heterogeneous label variables rather than a single variable. For example, in the e-commerce platforms, smartphones can be classified by brand dimension (with potential classes Apple, Huawei, Xiaomi, etc.), price dimension (with potential classes high grade, medium grade, and low grade), and color dimension (with potential classes white, black, blue, etc.). In recent years, a range of multi-dimensional classification (MDC) methods has been created to

address this type of data and has been extensively utilized across numerous domains [6–20], such as text categorization [7], image processing [8], biomedical engineering [9], etc.

Generally speaking, there exist several primary categories of MDC methods. The simple and straightforward solution is to ignore the dependencies between multiple heterogeneous class spaces, and then decompose the problem of MDC into several independent problems of multi-class classification for classifier induction. Nevertheless, this strategy fails to consider label dependencies, potentially resulting in unsatisfactory learning performances. The key to building an effective MDC approach lies in making full use of label dependencies in an appropriate way. In recent years, several strategies for exploiting label dependencies have been developed, including grouping distinct combinations of class variables into new super-class variables [6], capturing the dependencies between a pair of labels [12], learning a directed cyclic graph for class spaces [17,18], utilizing classifier chaining to exploit the high-order label correlations [14,19,20], etc.

Nevertheless, there are generally two main limitations in the existing MDC methods. Firstly, most contemporary MDC algorithms rely on a supervised learning framework, which needs a lot of labeled data with correct information to ensure excellent learning results. However, since label acquisition is a costly and time-consuming process, such a requirement hardly holds in many real applications. Typically, only partially labeled data that are associated with a set of candidate labels composed of correct labels and noisy labels are accessible. Many experiments in previous studies [5], Refs. [21–26] have demonstrated that noisy labels give incorrect guidance, which negatively impacts the learning performance of algorithms. Therefore, the primary challenge in this paper is to eliminate the impact of noisy labels.

Additionally, the aforementioned MDC techniques exclusively focus on centralized processing, necessitating the aggregation of the entire training dataset to a single fusion node for processing. Nonetheless, in several practical applications, such as anomaly detection [27], industrial Internet of Things [28], and traffic control systems [29,30], the training data samples are usually distributed at different nodes and cannot be centralized into one node due to various reasons [27–32]. Therefore, how to make the algorithm adapt to the distributed network and perform distributed learning without the transmission of original data is the second major challenge to be addressed in this paper.

In this paper, we develop the distributed partial label learning algorithm for multi-dimensional classification (dPL-MDC) by jointly considering the influence of noisy labels and the distributed storage of training data. The primary contributions are delineated as follows:

1. By using the one-vs.-one decomposition method, we transform the original distributed partial MDC problem into a distributed partial multi-label learning problem, which facilitates the subsequent exploitation of label correlations and the disambiguation of partial labels.

2. By leveraging weakly supervised information from ambiguous labels and ensuring constant similarity between label and feature spaces based on several anchor data, a distributed label recovery method is devised to estimate the label confidence of training data.

3. By adaptively updating the label confidence and model parameter of the classifier, we learn a multi-dimensional classifier while eliminating the influence of noisy labels. During the procedure of classifier induction, to exploit the high-order dependencies among newly transformed classes for classifier induction, we alternately learn the model parameters and the globally common predictive structure in the low-dimensional subspace by employing a computationally inexpensive and energy-saving distributed estimation method.

4. Simulations performed on several real datasets indicate that our proposed approach surpasses the existing MDC techniques.

The subsequent sections of this work are organized as follows. Section 2 provides a concise overview of the frameworks related to dPL-MDC over a network. Subsequently, Section 3 elaborates on the technical specifics of the dPL-MDC algorithm. After that, Section 4 presents the experimental results of the proposed algorithm alongside the existing MDC approaches. This paper concludes in Section 5.

## 2. Related Works

As far as we know, no prior research has tackled the issue of distributed classification of multi-dimensional data with ambiguous labels. In this section, a brief introduction related to three popular frameworks of dPL-MDC is presented, including MDC, distributed learning, and PLL.

The framework of MDC is similar to that of the widely researched multi-label classification (MLC) and multi-class classification (MCC). When the number of potential classes in each dimensional class space is equivalent to 2, the issue of MDC is transformed into the issue of MLC. If the dimension of output space is restricted to 1, then the issue of MDC degenerates into the issue of MCC. For MDC, the intuitive solution is to directly decompose the problem of MDC into several problems of MCC based on the dimensions of output space, and independently solve these problems dimension by dimension. This straightforward technique completely disregards the label dependencies across several class spaces, perhaps resulting in poor classification results. Therefore, how to exploit the label dependencies efficiently and induce a unified predictive model for multiple heterogeneous class spaces is the key to designing the MDC methods.

In recent years, a series of label dependence exploitation methods have also been proposed. An easy and straightforward way to exploit label dependence is to convert the label space of MDC into several new class spaces and treat each distinctly combined class variable as a new class variable for model induction [6]. This method has a significant flaw; they are unable to train classifiers for categories that have not appeared in the training set, which greatly restricts their applicability. Another strategy is to construct a directed acyclic graph (DAG) to characterize the distribution of the classes, and then induce the MDC model using a multi-dimensional Bayesian network [17,18]. However, since constructing Bayesian networks requires substantial computations, the performance of these methods is constrained by high computational costs, especially when dealing with large-scale data. Additionally, the issue of MDC can also be solved by a chain of induced multi-class classifiers [14,19,20], where the output of the previous classifier can be regarded as the extra features of the subsequent classifier. Obviously, for these algorithms, obtaining a proper chaining order is a precondition for achieving good learning performance. The absence of prior knowledge makes it challenging to determine the optimal order. Recently, two novel MDC methods have been developed based on decomposed label encoding [12,15]. These two approaches employ a one-vs.-one decomposition strategy to formulate a sequence of binary classification tasks inside a multi-dimensional output space. Then, they use the manifold structure [15] or covariance regularization [12] to exploit the relationship between a pair of newly transformed labels for the induction of MDC classifier. Nonetheless, as stated in Section 1, the existing MDC methods have two limitations. First, they fail to extract effective information from partially labeled data annotated with noisy labels to train classification models. Second, they cannot directly handle data stored distributively across distributed networks. These two aspects thus constitute the two key issues to be addressed in this paper.

Distributed learning is another novel branch of machine learning frameworks, where the training data is distributed among several nodes interconnected by a network [3,5,31,32]. By utilizing consensus-based or diffusion-based strategies to perform information fusion among nodes, these distributed learning methods can achieve learning performance that is nearly as excellent as the corresponding centralized learning methods. Recently, several distributed learning methods have been proposed for MCC and MLC [5,31,32]. For example, in [31], two distributed information-theoretic semi-supervised MLC methods have been developed, which introduces a new decentralized regularization term to take advantage of the relationships between label variables across a network. In [32], a distributed subspace structure learning algorithm has been proposed to distributively exploit the similarity of model parameter vectors by exploiting a common predictive structure in depth. Lately, a novel distributed partial MCC method has been developed, where a high-precision multi-class classifier is trained from partial labeled data by collaboratively learning the label confidence and the data instance weights to disambiguate the ambiguous labels [5]. Compared with existing distributed classification algorithms, the problems addressed in this paper are more complex. Owing to the fact that the output space in multi-dimensional classification problems is composed of multiple heterogeneous class spaces, the output values of multi-dimensional classification models are not comparable. As a result, traditional label dependency exploitation strategies, such as label–pair correlation learning and subspace-based high-order correlation learning, cannot be directly applied in this context.

PLL has become a novel learning paradigm in machine learning recently, which learns the classifier from a series of candidate labels [5,21–26,33–37]. Most of the existing PLL methods are usually formalized based on the framework of MCC [21,22,26]. These PLL methods assign a proper label for each training data by correcting the ambiguous labels using disambiguation strategies [22,26] or employing the loss-based decoding method to decode the transformed binary classifier's predictive outputs [21]. Lately, several novel PLL methods have been developed for MLC [33–37], which induce the predictive model by eliciting credible labels [33,34,37] or identifying noisy labels [35,36] from multiple candidate labels. Nevertheless, the majority of current PLL methods belong to centralized learning, which is impractical for distributed networks. So, developing a distributed PLL for MDC, where global classification can be performed across many nodes using dispersedly distributed partial multi-dimensional data, is the better option.

## 3. dPL-MDC Algorithm

This section formulates the issue of fully decentralized classification of partial multi-dimensional data and presents the technical specifics of the dPL-MDC algorithm.

### *3.1. Problem Formulation*

An interconnected network consisting of $J$ geographically dispersed nodes over a region is the subject of this paper. The total $N$ partially multi-dimensional data are dispersedly distributed over $J$ nodes. For the sake of generality, we use an undirected graph called $\mathcal{G} = (\mathcal{J}, \mathcal{E})$ to describe this network, where $\mathcal{J}$ is the set of nodes and $\mathcal{E}$ is the set of connections between them. For each node $j$, all the neighboring nodes and itself constitute the neighboring node set $\mathcal{B}_j$.

Let $\mathcal{X}$ denote the input vector space, and let $\mathcal{Y} = \mathcal{S}_1 \times \ldots \times \mathcal{S}_Q$ stand for the output space composed of a total of $Q$ heterogeneous class spaces, where each class space $\mathcal{S}_q = \{s_{q,1}, \ldots, s_{q,K_q}\}$ contains $K_q$ possible classes. At each node $j$, there exist $N_j$ locally partial multi-dimensional data annotated with the candidate labels at observable class spaces $\{\boldsymbol{x}_{j,n}, \Omega_j \boldsymbol{y}_{j,n}\}_{n=1}^{N_j}$, $(N = \sum_{j=1}^{J} N_j)$, where $\boldsymbol{x}_{j,n} \in \mathcal{X}^d$ denotes the input vector of the features, and $\Omega_j \boldsymbol{y}_{j,n} \in \mathcal{Y}_j \subset \mathcal{Y}$ denotes the collected labels. A $\sum_{q=1}^{Q} K_q$-dimensional vector $\boldsymbol{y}_{j,n} =$

$[\boldsymbol{y}_{j,n,1}^T, \ldots, \boldsymbol{y}_{j,n,Q}^T]^T \in \{+1,-1\}^{\sum_{q=1}^{Q} K_q}$ is utilized to represent the collected label vector and denote the original candidate label vector. In the $q$-th dimensional label vector $\boldsymbol{y}_{j,n,q} \in \{+1,-1\}^{K_q}$, the $k$-th element $y_{j,n,q,k}$ is assigned a value of 1 if the $k$-th label is included in the candidate label set, and -1 if it is excluded. Furthermore, a $\sum_{q=1}^{Q} K_q \times \sum_{q=1}^{Q} K_q$-dimensional diagonal matrix $\Omega_j$ is designed, whose diagonal element equals 1 if the corresponding label is accessible and 0 otherwise. So, in the observed label vector $\Omega_j \boldsymbol{y}_{j,n}$, all the accessible labels are kept unchanged, and all the missing labels are set to zero.

Each node $j$ is tasked with learning the globally optimum classifier using its local data and the discriminant information from neighboring nodes $i \in \mathcal{B}_j$, ensuring that unseen data can be accurately classified into the appropriate categories.

### *3.2. Output Space Decomposition*

For MDC, the main challenge is that the output space is composed of multiple heterogeneous class spaces, which makes the outputs of the predicted model across different class spaces not comparable with each other [12,15]. To tackle this problem, referring to the concept of decomposed label encoding strategy [12,15], we would like to conduct one-vs.-one decomposition on each dimensional class space of MDC, such that the problem of distributed partial MDC can be transformed into the problem of distributed homogeneous partial multi-label learning.

To be specific, for a label vector $\Omega_j \boldsymbol{y}_{j,n} = \Omega_j [\boldsymbol{y}_{j,n,1}^T, \ldots, \boldsymbol{y}_{j,n,Q}^T]^T$, supposing that the $q$-th class space composed of $K_q$ possible classes is observable, i.e., $q \in \mathcal{Y}_j$, we can transform a $K_q$-dimensional label vector $\boldsymbol{y}_{j,n,q}$ into a $K_q^{'}$-dimensional ternary label vector $\boldsymbol{y}_{j,n,q}^{'}$ via one-vs.-one decomposition, where $K_q^{'} = \frac{K_q!}{2!}$ with $N!$ being the factorial of $N$. Furthermore, if the $m$-th class space can not be accessed, i.e., $q \notin \mathcal{Y}_j$, then the newly transformed label vectors can be represented as a $K_q^{'}$-dimensional vector consisting of all zero elements. Correspondingly, the originally collected label vector $\Omega_j \boldsymbol{y}_{j,n}$ at total $Q$ class spaces can be decomposed as a $\sum_{q=1}^{Q} K_q^{'}$-dimensional vector $\Omega_j^{'} \boldsymbol{y}_{j,n}^{'} = \Omega_j^{'} [\boldsymbol{y}_{j,n,1}^{'T}, \ldots, \boldsymbol{y}_{j,n,Q}^{'T}]^T$, where $\Omega_j^{'}$ denotes a $\sum_{q=1}^{Q} K_q^{'} \times \sum_{q=1}^{Q} K_q^{'}$-dimensional diagonal matrix with its diagonal element being 1 if the corresponding labels are accessible and 0 otherwise.

To clarify the process of one-vs.-one decomposition, two cases are presented as follows:

Case 1: For a collected data sample annotated with correct labels, supposing that the $k$-th class variable in the $q$-th dimensional label vector $\boldsymbol{y}_{j,n,q}$ is the ground truth, each $r$-th element in the encoded ternary label vector $\boldsymbol{y}_{j,n,q}^{'}$ equals

$$\boldsymbol{y}_{j,n,q,r}^{'} = \begin{cases} +1, & \text{if} \quad r \in \mathcal{P}_{j,n,q,k}, \\ -1, & \text{if} \quad r \in \mathcal{N}_{j,n,q,k}, \\ 0. & \text{otherwise}, \end{cases} \tag{1}$$

where the positive set $\mathcal{P}_{j,n,q,k} = \{(k-1)K_q + 1 - \sum_{i=1}^{k} i - 1\}$ for $1 \leq k < K_q$, and the negative set $\mathcal{N}_{j,n,q,k} = \{k - 1 + \sum_{i=0}^{s} K_q - 2 - i\}$ for $0 \leq s \leq k-3$, $k \geq 3$, or $\mathcal{N}_{j,n,q,k} = \{1\}$ for $k = 2$.

**Example 1.** *Given four data with correct labels* $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), (\boldsymbol{x}_3, \boldsymbol{y}_3), (\boldsymbol{x}_4, \boldsymbol{y}_4)\}$*, supposing that the q-th dimensional original label vectors* $\boldsymbol{y}_{1,q} = [+1,-1,-1,-1]^T$*,* $\boldsymbol{y}_{2,q} = [-1,+1,-1,-1]^T$*,* $\boldsymbol{y}_{3,q} = [-1,-1,+1,-1]^T$ *and* $\boldsymbol{y}_{4,q} = [-1,-1,-1,+1]^T$ *for simplicity, the transformed ternary labels for these data samples can be obtained according to (1):*

$$\boldsymbol{y}_{1,q}^{'} = [+1,+1,+1,0,0,0]^T, \qquad \boldsymbol{y}_{2,q}^{'} = [-1,0,0,+1,+1,0]^T,$$
$$\boldsymbol{y}_{3,q}^{'} = [0,-1,0,-1,0,+1]^T, \qquad \boldsymbol{y}_{4,q}^{'} = [0,0,-1,0,-1,-1]^T.$$

Case 2: For a collected partial multi-dimensional data sample annotated with several candidate labels, supposing that the candidate label set in the $q$-th dimensional class space consists of a total of $m$ candidate labels (i.e., the $k_1$,...,$k_m$-th class variables in $\boldsymbol{y}_{j,n,q}$ are candidate labels), each $r$-th element in the ternary label vector $\boldsymbol{y}'_{j,n,q}$ equals

$$\boldsymbol{y}'_{j,n,q,r} = \begin{cases} +1, & \text{if } r \in \mathcal{P}_{j,n,q} \text{ and } r \notin \mathcal{N}_{j,n,q}, \\ -1, & \text{if } r \in \mathcal{N}_{j,n,q} \text{ and } r \notin \mathcal{P}_{j,n,q}, \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where the positive set $\mathcal{P}_{j,n,q} = \{\mathcal{P}_{j,n,q,k_1}, \ldots, \mathcal{P}_{j,n,q,k_m}\}$. Furthermore, the corresponding negative set $\mathcal{N}_{j,n,q} = \{\mathcal{N}_{j,n,q,k_1}, \ldots, \mathcal{N}_{j,n,q,k_m}\}$. The calculation of $\mathcal{P}_{j,n,q,k}$ and $\mathcal{N}_{j,n,q,k}$ is similar to those in case 1.

**Example 2.** *Given four data with candidate labels* $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), (\boldsymbol{x}_3, \boldsymbol{y}_3), (\boldsymbol{x}_4, \boldsymbol{y}_4)\}$*, supposing that the candidate labels in the q-th dimensional class space are composed of the correct label and an additional noisy label for simplicity, i.e.,* $\boldsymbol{y}_{1,m} = [+1, +1, -1, -1]^T$, $\boldsymbol{y}_{2,m} = [-1, +1, +1, -1]^T$, $\boldsymbol{y}_{3,m} = [-1, -1, +1, +1]^T$ *and* $\boldsymbol{y}_{4,m} = [+1, -1, -1, +1]^T$*, we have the transformed ternary labels as follows:*

$$\boldsymbol{y}'_{1,q} = [0, +1, +1, +1, +1, 0]^T, \qquad \boldsymbol{y}'_{2,q} = [-1, -1, 0, 0, +1, +1]^T,$$
$$\boldsymbol{y}'_{3,q} = [0, -1, -1, -1, -1, 0]^T, \qquad \boldsymbol{y}'_{4,q} = [+1, +1, 0, 0, -1, -1]^T.$$

### *3.3. Label Recovery*

In this study, it is imperative to devise an effective technique to mitigate the adverse effects of noisy labels and to recover the accurate labels from the candidate set.

Based on the underlying assumption in manifold learning that there exist similar structures in feature and label spaces [1], we would like to recover the correct labels by exploiting the similarity among the whole training data set. Nevertheless, because of the random distribution of training data across different nodes, the global distribution of data samples cannot be precisely characterized. To solve this problem, we randomly select or employ a decentralized vector quantization method [38] to select $C$ global common anchor data $\{\boldsymbol{c}_l\}_{l=1}^{C}$ as the representatives of the whole dataset. The research in [31] indicates that, given a sufficient amount of anchor data $C$, these quantized anchor data can cover all high-density regions. Furthermore, the similarity degree among the whole training dataset can be roughly measured by employing the quantized anchor data.

Following the analysis in Section 1, acquiring a significant quantity of precisely labeled anchor data samples necessitates extensive professional expertise. Consequently, we examine the typical scenario in which all anchor data samples are unlabeled. Now, the key to label recovery is to accurately estimate the labels of anchor data. To achieve this, a straightforward weighted voting method is used, which is given by

$$Z_j^c = \Phi_j \circ (R_j Y_j'), \tag{3}$$

where $Y_j' \in \{+1, 0, -1\}^{N_j \times \sum_{q=1}^{Q} K_q'}$ represents the transformed label matrix of $N_j$ local training data, and $Z_j^c = [\boldsymbol{z}_{j,1}^c, \ldots, \boldsymbol{z}_{j,C}^c]^T \in \mathbb{R}^{C \times \sum_{q=1}^{Q} K_q'}$ represents the recovered label matrix of $C$ anchor data. Furthermore, $R_j \in \mathbb{R}^{C \times N_j}$ denotes the weighted parameter for measuring the similarity between $C$ anchor data and $N_j$ training data at node $j$, in which each element $R_{j,c,n}$ can be calculated by the Gaussian kernel function. Furthermore, $\circ$ denotes the

Hadamard product, and $\Phi_j$ is the $N_j \times \sum_{q=1}^{Q} K_q^{'}$-dimensional matrix, which is computed by $\Phi_j = \mathbf{1}_{N_j}\mathbf{1}^T_{\sum_{q=1}^{Q} K_q^{'}}\Omega_j^{'}$.

According to the basic setting in the considered network, only the label variable of anchor data in available class spaces (i.e., $q \in \mathcal{Y}_j$) can be directly computed via (3). Moreover, owing to the existence of noisy labels, the predicted labels across various nodes for a specific class may be different. To address this issue, inspired by the concept of LMaFit presented in [39,40] , we convert the problem of label recovery into the problem of decentralized matrix completion. To be specific, we factorize the label matrix of anchor data $Z_j^c$ into two low-dimensional factorized matrices $A_j \in \mathbb{R}^{C\times\epsilon}$ and $B_j \in \mathbb{R}^{\epsilon\times\sum_{q=1}^{Q} K_q^{'}}$ ($\epsilon$ denotes the rank of the matrix $Z_j^c$), and then complete the matrix $Z_j^c$ by adaptively updating the factorized matrices. The decentralized framework of label recovery can be formulated as follows:

$$\begin{aligned} \min_{A_j,B_j,Z_j^c} \quad & ||\Phi_j \circ (Z_j^c - \tilde{Z}_j^c)||_F^2 + ||A_j \cdot B_j - Z_j^c||_F^2, \\ \text{s.t.} \quad & B_j = B_i, \quad j \in \mathcal{J}, \quad i \in \mathcal{B}_j. \end{aligned} \tag{4}$$

The objective function (4) consists of three terms.

The first term is utilized to reduce the reconstruction error between the recovered labels of anchor data and training data in partially accessible class spaces. Here, $\tilde{Z}_j^c$ denotes the initial estimation for matrix $Z_j^c$. The second term is targeted for imputing the missing entries in $Z^c$ based on two factorized matrices $A$ and $B_j$.

The process of label recovery is composed of three main steps:

Initialization: A small loop indexed by $\tau$ is set. When $\tau = 0$, we calculate $\tilde{Z}_j^{c,0}$ by (3) and set $Z_{j,0}^c = \tilde{Z}_j^c$.

Update of $A_{j,\tau+1}$: The local estimation of $A_j$ can be obtained by the linear combination of local estimations among one-hop neighboring nodes from its neighbors $i \in \mathcal{B}_j$

$$A_{j,\tau+1}^{'} = A_{j,\tau} - s_\tau(A_{j,\tau}B_{j,\tau} - Z_{j,\tau}^c)B_{j,\tau}^T, \tag{5}$$

$$A_{j,\tau+1} = \sum_{i\in\mathcal{B}_j} \zeta_{ji} A_{i,\tau+1}^{'}. \tag{6}$$

where $s_\tau$ denotes learning rate, and $\zeta_{ji}$ denotes the cooperative coefficient, which is followed by the Metropolis cooperation rule [32].

Update of $B_{j,\tau+1}$: The local estimation of $B_j$ can be obtained by the linear combination of local predictions from its neighbors $i \in \mathcal{B}_j$, which is given by

$$B_{j,\tau+1}^{'} = (A_{j,\tau+1}^T A_{j,\tau+1})^{-1} A_{j,\tau+1}^T Z_{j,\tau}^c, \tag{7}$$

$$B_{j,\tau+1} = \sum_{i\in\mathcal{B}_j} \zeta_{ji} B_{i,\tau+1}^{'}. \tag{8}$$

Update of $Z_{j,\tau+1}^c$: Based on the current estimations $B_{j,\tau+1}$ and $Z_{j,\tau+1}$, we have the updated equation of recovered labels of anchor data $Z_{j,\tau+1}^c$

$$Z_{j,\tau+1}^c = A_{j,\tau+1}B_{j,\tau+1} + \beta_{\tau+1}\{\Phi_j \circ [\tilde{Z}_j^c - A_{j,\tau+1}B_{j,\tau+1}]\}. \tag{9}$$

As for the update of elements in $Z_{j,\tau+1}^c$, there exist two different cases. The elements that cannot be locally estimated can be imputed using the product of two factorized matrices $A_{j,\tau+1}$ and $B_{j,\tau+1}$. The elements that can be locally predicted by $\tilde{Z}_j^c$ are derived from a weighted combination of the global estimation $A_jB_j$ and the local estimation $\tilde{Z}_j^c$, utilizing a time-varying weighted coefficient $\beta_{\tau+1}$. At the start, the global estimation $A_{j,\tau+1}B_{j,\tau+1}$ is rough, since the information fusion among neighboring nodes is insufficient. A substantial

value of $\beta_{\tau+1}$ is employed to provide low confidence to $A_{j,\tau+1}B_{j,\tau+1}$. After a sufficient number of iterations, more and more information shared by one node diffuses over the whole network, which makes the global estimation $A_{j,\tau+1}B_{j,\tau+1}$ consistent with each other. So, a small value of $\beta_{\tau+1}$ is employed. Consequently, we define $\beta_{\tau+1} = \exp(-\nu\tau)$, with $\nu$ being a positive coefficient.

The primary processes of label recovery are summarized in Algorithm 1 for clarity.

**Algorithm 1** Label recovery method

**Require:** Initialize $Z^c_{j,0} = \tilde{Z}^c_j$.
**for** $\tau = 0, \ldots, T_{\max}$ **do**
  **for** $j = 1 \in \mathcal{J}$ **do**
    Update $A'_{j,\tau+1}$ via (5) and transmit $A'_{j,\tau}$ to the neighbors $\mathcal{B}_j$.
  **end for**
  **for** $j = 1 \in \mathcal{J}$ **do**
    Update $A_{j,\tau+1}$ via (6).
    Update $B'_{j,\tau+1}$ via (7) and transmit $B'_{j,\tau}$ to the neighbors $\mathcal{B}_j$.
  **end for**
  **for** $j = 1 \in \mathcal{J}$ **do**
    Update $B_{j,\tau+1}$ via (8).
    Update $Z^c_{j,\tau+1}$ via (9).
  **end for**
**end for** Return the recovered label matrix of anchor points $Z^c_j$.

### *3.4. Classifier Induction*

This subsection describes the induction of the classifier based on the label recovery.

We can express the output of the discriminant function for the $k$-th class variable in the $q$-th dimensional class space as a linear combination of kernel functions, under the assumption that the discriminant function is non-linear. This is represented by the following:

$$f_{q,k}(\boldsymbol{x}_n) = \sum_l \alpha_{q,k,l} k(\boldsymbol{x}_n, \boldsymbol{x}_l), \tag{10}$$

where $k(\cdot,\cdot)$ denotes a kernel function, and $\alpha_{q,k,l}$ denotes the weight coefficient.

By introducing the infinite-dimensional kernel feature map $\phi(\cdot)$, we have

$$f_{q,k}(\boldsymbol{x}_n) = \boldsymbol{w}^T_{q,k} \cdot \phi(\boldsymbol{x}_n), \tag{11}$$

where the weight vector $\boldsymbol{w}_{q,k} = \sum_l \alpha_{q,k,l}\phi(\boldsymbol{x}_l)$.

Nevertheless, it is evident from (11) that the weight vectors $\boldsymbol{w}_{q,k}$, which are composed of a succession of kernel feature maps, cannot be explicitly expressed or freely exchanged among neighbors, as the infinite-dimensional kernel feature map $\phi(\boldsymbol{x}_l)$ is unknown.

To resolve this issue, we project the training data into a feature space with a constrained number of dimensions, thereafter substituting the kernel feature map $\phi(\cdot)$ with the random feature map $\hat{\phi}(\cdot)$ for the construction of model parameters. Consequently, the kernel function may be approximated as follows: [41,42]

$$k(\boldsymbol{x}_n, \boldsymbol{x}_l) \approx < \hat{\phi}(\boldsymbol{x}_n), \hat{\phi}(\boldsymbol{x}_l) > .$$

It is noted that we take the Gaussian kernel function into account here. According to the theory in [41,42], we have

$$[\hat{\phi}(\boldsymbol{x}_n)]_l = \begin{cases} \frac{1}{\sqrt{0.5D}} \cos\left(\boldsymbol{\kappa}_{\frac{l+1}{2}}{}^T \boldsymbol{x}_n\right), & 0 < l < D \quad \text{odd}, \\ \frac{1}{\sqrt{0.5D}} \sin\left(\boldsymbol{\kappa}_{\frac{l}{2}}{}^T \boldsymbol{x}_n\right), & 0 < l < D \quad \text{even}, \end{cases}$$

where $\boldsymbol{\kappa}$ is stochastically drawn from the distribution $\mathbf{p}(\boldsymbol{\kappa}) = (2\pi)^{-(D/2)} \exp(-||\boldsymbol{\kappa}||_2^2/2)$.

Based on this, we can explicitly express the weight vector $\boldsymbol{w}_{q,k}$ as a $D$-dimensional vector, denoted as $\boldsymbol{w}_{q,k} = \sum_l \alpha_{q,k,l} \hat{\phi}(\boldsymbol{x}_l)$.

Moreover, in order to exploit the high-order dependencies among the total $\sum_{m=1}^{q} K_q^{'}$ recovered label variables, we suppose that there exists a low-dimensional common predictive structure $\Theta \in \mathbb{R}^{h \times D}$ among all the label variables [43]. Consequently, the weight vector $\boldsymbol{w}_{q,k}$ can be rewritten as

$$\boldsymbol{w}_{q,k} = \boldsymbol{u}_{q,k} + \Theta^T \boldsymbol{v}_{q,k}, \tag{12}$$

where $\boldsymbol{u}_{q,k} \in \mathbb{R}^D$ and $\boldsymbol{r}_{q,k} \in \mathbb{R}^h$ ($h < D$) denote the private weight vectors with respect to (w.r.t.) the $k$-th class variable in the $q$-th dimensional class space, and $\Theta$ denotes the common predictive structure. Furthermore, to reduce the complexity of the predictive structure, we assume that there exists an orthogonality property, i.e., $\Theta\Theta^T = I_h$.

The global optimization problem can be formulated as follows by taking into account the aforementioned considerations:

$$\begin{aligned} \min_{z_{n,q,k}, \boldsymbol{w}_{q,k}, \boldsymbol{v}_{q,k}, \Theta} \mathcal{F} &= \sum_{n=1}^{N} \sum_{q \in \mathcal{Y}_j} \sum_{k=1}^{K_q^{'}} \frac{\gamma_A}{2N} (z_{n,q,k} - y_{n,q,k})^2 + \sum_{n=1}^{N} \sum_{q=1}^{Q} \sum_{k=1}^{K_q^{'}} \frac{\gamma_B}{2N} (z_{n,q,k} - r_{n,l} z_{j,l}^c)^2 \\ &+ \sum_{n=1}^{N} \sum_{q=1}^{Q} \sum_{k=1}^{K_q^{'}} \frac{\gamma_C}{2N} (z_{n,q,k} - f_{q,k}(\boldsymbol{x}_n))^2 + \sum_{q=1}^{Q} \sum_{k=1}^{K_q^{'}} (\frac{\gamma_D}{2} ||\boldsymbol{w}_{q,k} - \Theta^T \boldsymbol{v}_{q,k}||_2^2 + \frac{\gamma_E}{2} ||\boldsymbol{w}_{q,k}||_2^2), \\ \text{s.t.} \quad & \Theta\Theta^T = I_h. \end{aligned} \tag{13}$$

In the above global optimization problem, the first term is utilized to extract the weakly supervised information from the transformed label variables. Furthermore, $\gamma_A$ denotes the weight parameter, and $z_{n,q,k}$ denotes the $k$-th recovered label of the $n$-th training data in the $q$-th dimensional class space. The second term is targeted for seeking the optimal solution by minimizing the reconstruction error between the recovered labels of anchor data and training data in partly observable class spaces with weight parameter $\gamma_B$. The third term is the loss function with weight parameter $\gamma_C$, where $f_{q,k}(\boldsymbol{x}_n)$ stands for the output value of the discriminant function. The fourth term is the regularization term weighted by the parameter $\gamma_D$, which serves the purpose of reducing the complexity of components out of the subspace. The fifth term represents an additional regularization term with the weight parameter $\gamma_E$, utilized to enhance the generalization of the learned classifier.

To render the global optimization issue (13) suitable for the distributed network, we decentralize it by utilizing the local estimations $\{\boldsymbol{w}_{j,q,k}, \boldsymbol{v}_{j,q,k}, \Theta_j\}$ to replace the global ones and adding two consensus constraints, which are given by

$$\begin{aligned} \min_{z_{j,n,q,k}, \boldsymbol{w}_{j,q,k}, \boldsymbol{v}_{j,q,k}, \Theta_j} \mathcal{F} &= \sum_{j=1}^{J} \Bigg\{ \sum_{n=1}^{N_j} \sum_{q \in \mathcal{Y}_j} \sum_{k=1}^{K_q^{'}} \frac{\gamma_A}{2N} (z_{j,n,q,k} - y_{j,n,q,k})^2 + \sum_{n=1}^{N_j} \sum_{q=1}^{Q} \sum_{k=1}^{K_q^{'}} \\ &\frac{\gamma_B}{2N} (z_{j,n,q,k} - r_{n,l} z_{j,l}^c)^2 + \sum_{n=1}^{N_j} \sum_{q=1}^{Q} \sum_{k=1}^{K_q^{'}} \frac{\gamma_C}{2N} (z_{j,n,q,k} - f_{q,k}(\boldsymbol{x}_{j,n}))^2 \\ &+ \sum_{q=1}^{Q} \sum_{k=1}^{K_q^{'}} (\frac{\gamma_D}{2} ||\boldsymbol{w}_{j,q,k} - \Theta_j^T \boldsymbol{v}_{j,q,k}||_2^2 + \frac{\gamma_E}{2} ||\boldsymbol{w}_{j,q,k}||_2^2), \Bigg\} \\ &= \sum_{j=1}^{J} \mathcal{F}_j \\ \text{s.t.} \quad & \Theta_j \Theta_j^T = I_h, \quad \Theta_j = \Theta_i, \quad \boldsymbol{w}_{j,q,k} = \boldsymbol{w}_{i,q,k}, \quad j \in \mathcal{J}, \quad i \in \mathcal{B}_j. \end{aligned} \tag{14}$$

Two consensus constraints are incorporated into the optimization for forcing all the local estimations among different nodes to agree on the consensus value.

Optimized procedure of $z_{j,n,q,k}$: Given fixed $\boldsymbol{w}_{j,q,k,t}$, we have the sub-optimized problem

$$\{z_{j,n,q,k,t+1}\} = \arg\min_{z_{j,n,q,k}} \mathcal{F}_j(z_{j,n,q,k}, \boldsymbol{w}_{j,q,k,t}, \boldsymbol{v}_{j,q,k,t}, \Theta_{j,t}). \tag{15}$$

By employing the steepest gradient descent (SGD) method, we have the update equation of $z_{j,n,q,k}$ as follows:

$$z_{j,n,q,k,t+1} = z_{j,n,q,k,t} - \mu_{1,t+1} \nabla_{z_{j,n,q,k}} \mathcal{F}_j(z_{j,n,q,k}, \boldsymbol{w}_{j,q,k,t}, \boldsymbol{v}_{j,q,k,t}, \Theta_{j,t}), \tag{16}$$

where $\mu_{1,t+1}$ denotes the step size, and the gradient

$$\begin{aligned} &\nabla_{z_{j,n,q,k}} \mathcal{F}_j(z_{j,n,q,k}, \boldsymbol{w}_{j,q,k,t}, \boldsymbol{v}_{j,q,k,t}, \Theta_{j,t}) \\ =&\frac{\gamma_A}{N}\delta \cdot (z_{n,q,k} - y_{n,q,k}) + \frac{\gamma_B}{N}(z_{n,q,k} - r_{n,l} z^c_{j,l}) + \frac{\gamma_C}{N}(z_{n,q,k} - f_{q,k}(\boldsymbol{x}_{j,n})), \end{aligned}$$

with the indicator variable $\delta = 1$ if $q \in \mathcal{Y}_j$; otherwise, $\delta = 0$.

Optimized procedure of $\boldsymbol{w}_{j,q,k}$: For fixed $\Theta_{j,t}$ and $\boldsymbol{v}_{j,q,k,t} = \Theta_{j,t}\boldsymbol{w}_{j,q,k,t}$, the sub-optimized problem reduces to

$$\begin{aligned} &\{\boldsymbol{w}_{j,q,k,t+1}\} = \arg\min_{\boldsymbol{w}_{j,q,k}} \mathcal{F}_j(z_{j,n,q,k,t+1}, \boldsymbol{w}_{j,q,k}, \boldsymbol{v}_{j,q,k,t}, \Theta_{j,t}), \\ &\text{s.t.} \quad \boldsymbol{w}_{j,q,k} = \boldsymbol{w}_{i,q,k}, \quad j \in \mathcal{J}, \quad i \in \mathcal{B}_j. \end{aligned} \tag{17}$$

Referring to the update process of distributed learning [3,31], we have the updated equation of $\boldsymbol{w}_{j,q,k}$, which is composed of two parts (adaptive and cooperative steps):

$$\boldsymbol{w}'_{j,q,k,t+1} = \boldsymbol{w}_{j,q,k,t} - \mu_{2,t+1} \nabla_{\boldsymbol{w}_{j,q,k}} \mathcal{F}_j(z_{j,n,q,k,t+1}, \boldsymbol{w}_{j,q,k}, \boldsymbol{v}_{j,q,k,t}, \Theta_{j,t}), \tag{18}$$

$$\boldsymbol{w}_{j,q,k,t+1} = \sum_{i \in \mathcal{B}_j} \zeta_{ji} \boldsymbol{w}'_{i,q,k,t+1} \tag{19}$$

where $\mu_{2,t+1}$ denotes the step size, and the gradient

$$\begin{aligned} &\nabla_{\boldsymbol{w}_{j,q,k}} \mathcal{F}_j(z_{j,n,q,k,t+1}, \boldsymbol{w}_{j,q,k}, \boldsymbol{v}_{j,q,k,t}, \Theta_{j,t}) \\ =&-\sum_{n=1}^{N_j} \frac{\gamma_C}{N}(z_{j,n,q,k} - f_{q,k}(\boldsymbol{x}_{j,n}))\hat{\phi}(\boldsymbol{x}_{j,n}) + \frac{\gamma_D}{J}(\boldsymbol{w}_{j,q,k,t} - \Theta^T_{j,t}\boldsymbol{v}_{j,q,k,t}) + \frac{\gamma_E}{J}\boldsymbol{w}_{j,q,k,t}. \end{aligned}$$

Optimized procedure of $\boldsymbol{v}_{j,q,k,t}$: Incorporating the fixed $\boldsymbol{w}_{j,q,k,t+1}$ and $\Theta_{j,t}$ into the decentralized optimization problem (14), we have the update equation of $\boldsymbol{v}_{j,q,k,t+1}$

$$\boldsymbol{v}_{j,q,k,t+1} = \Theta_{j,t}\boldsymbol{w}_{j,q,k,t+1}. \tag{20}$$

Optimized procedure of $\Theta_{j,t}$: Based on the updated $\boldsymbol{v}_{j,q,k,t+1}$, the optimization problem (14) can be reduced to

$$\begin{aligned} \Theta_{j,t+1} &= \arg\min_{\Theta_j} \mathcal{F}_j(\boldsymbol{w}_{j,q,k,t+1}, \boldsymbol{v}_{j,q,k,t+1}, \Theta_j), \\ &= \arg\max_{\Theta_j} \operatorname{tr}\left[\Theta_j(\textstyle\sum_{q=1}^{Q}\sum_{k=1}^{K'_q} \boldsymbol{w}_{j,q,k,t+1} \cdot \boldsymbol{w}^T_{j,q,k,t+1})\Theta^T_j\right] \\ &= \arg\max_{\Theta_j} \operatorname{tr}\left[\Theta_j \Gamma_{j,t+1} \Theta^T_j\right], \\ \text{s.t.} \quad &\Theta_j\Theta^T_j = I_h, \quad \Theta_j = \Theta_i, \quad j \in \mathcal{J}, \quad i \in \mathcal{B}_j. \end{aligned} \tag{21}$$

Based on the current consensus estimations $\{\boldsymbol{w}_{j,q,k,t+1}\}$, we can seek the optimal solution of sub-optimization (21) by employing eigenvalue decomposition (ED) of $\Gamma_{j,t+1} = \sum_{q=1}^{Q}\sum_{k=1}^{K_q'} \boldsymbol{w}_{j,q,k,t+1} \cdot \boldsymbol{w}_{j,q,k,t+1}^T$. Specifically, the rows of $\Theta_{j,t+1}$ are determined by the eigenvectors w.r.t. the largest $h$ eigenvalues of $\Gamma_{j,t+1}$.

The ideal predictive model may be achieved by iteratively optimizing the four aforementioned sub-optimization issues until convergence. Subsequently, for an unseen instance $\boldsymbol{x}^*$, the outputs of the discriminant function $\boldsymbol{f}^*$ may be derived from the trained classifier. Correspondingly, the predicted label vector $\boldsymbol{y}^{*'}$ in the transformed output space can be calculated by

$$\boldsymbol{y}^{*'} = \text{sign}(\boldsymbol{f}^*), \tag{22}$$

where $\text{sign}(\cdot)$ denotes the element-wise signed function. Now, the predicted label vector $\boldsymbol{y}^{*'}$ can be expressed as a $\sum_{q=1}^{Q} K_q'$-dimensional binary vector consisting of $\pm 1$ elements.

To obtain the $\sum_{q=1}^{Q} K_q$-dimensional label vector $\boldsymbol{y}^*$ in the original output space, a one-vs.-one decoding rule is applied to the $\sum_{q=1}^{Q} K_q'$-dimensional transformed label vector $\boldsymbol{y}^{*'}$. To be specific, referring to (1), we count the elements in binary vector $\boldsymbol{y}^{*'}$ whose serial number belongs to sets $\mathcal{P}_{j,n,q,k}$ and $\mathcal{N}_{j,n,q,k}$, where $\mathcal{P}_{j,n,q,k} = \{(k-1)K_q + 1 - \sum_{i=1}^{k} i - 1\}$ for $1 \leq k < K_q$, and $\mathcal{N}_{j,n,q,k} = \{k - 1 + \sum_{i=0}^{s} K_q - 2 - i\}$ for $0 \leq s \leq k-3$, $k \geq 3$, or $\mathcal{N}_{j,n,q,k} = \{1\}$ for $k = 2$. Based on this, we utilize $p_{q,k}$ and $n_{q,k}$ to denote, respectively, the count number w.r.t. positive set $\mathcal{P}_{j,n,q,k}$ and negative set $\mathcal{N}_{j,n,q,k}$, and then compute the value of the $k$-th class variable in the $q$-th dimensional original label vectors $\boldsymbol{y}_q$ using the majority voting method, i.e.,

$$y_{q,k}^* = \begin{cases} +1, & \text{if } k = \arg\max\limits_{1 \leq r \leq K_q} p_{q,r} + n_{p,r}, \\ -1, & \text{otherwise,} \end{cases} \quad k = 1, ..., K_q. \tag{23}$$

Similarly, the predicted label vector of unseen data in all the $Q$ class spaces $\boldsymbol{y}^* \in \{\pm 1\}^{\sum_{q=1}^{Q} K_q}$ can be obtained.

The flowchart of the proposed dPL-MDC algorithm is depicted in Figure 1, and the essential steps of the dPL-MDC algorithm are encapsulated in Algorithm 2.

**Algorithm 2** dPL-MDC algorithm

**Require:** Input training data $\{\boldsymbol{x}_{j,n}, \boldsymbol{y}_{j,n}\}_{n=1}^{N_j}$. Initialize $\Theta_{j,0} = \boldsymbol{0}_{h \times D}$ and $\boldsymbol{w}_{j,q,k,0} = \boldsymbol{0}_D$.

```
Obtain the transformed labels for training data via (1) or (2).
Obtain the recovered label matrix of anchor points {Z_j^c}_{j=1}^J via Algorithm 1.
for t = 0, 1, ... do
    for j = 1 ∈ 𝒥 do
        Update w'_{j,q,k,t+1} via (18).
        Transmit w'_{j,q,k,t+1} to its neighbors ß ∈ ℬ_j.
    end for
    for j = 1 ∈ 𝒥 do
        Update w_{j,q,k,t+1} via (19).
    end for
    for j ∈ 𝒥 do
        Update v_{j,q,k,t+1} via (20).
        Update Θ_{j,t+1} by ED of matrix Γ_{j,t+1}.
    end for
end for
Obtain binary label vector y^{*'} for an unseen data x^*.
Return y^* by applying one-vs.-one decoding rule over y^{*'}.
```

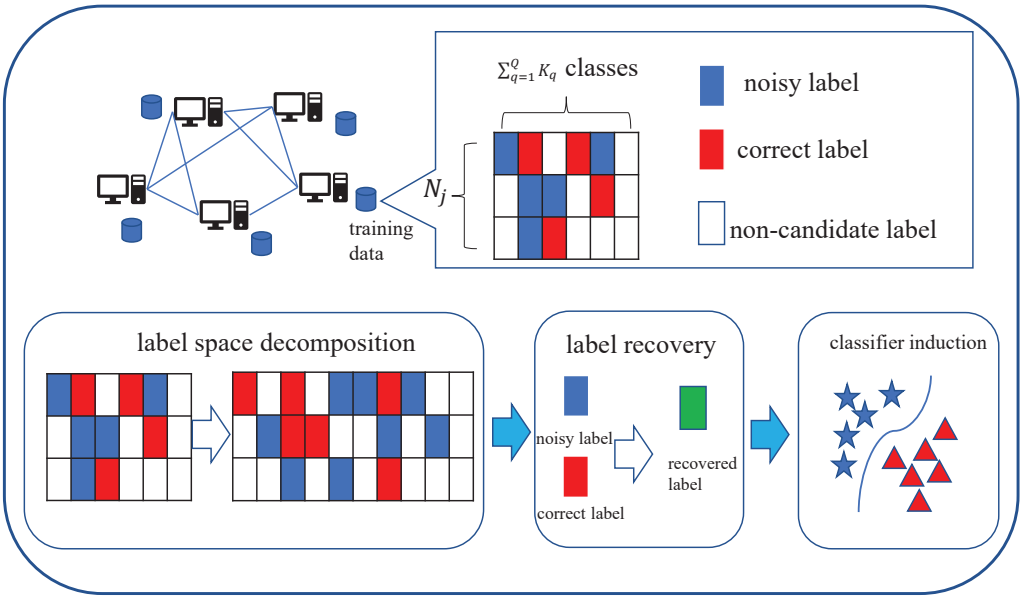



**Figure 1.** Diagram of the proposed dPL-MDC algorithm.

*3.5. Performance Analysis*

This subsection analyzes the performance of the proposed dPL-MDC algorithm.

Convergence Analysis: To carry out the following analysis, a frequently employed assumption is implemented.

**Assumption 1.** *The cooperative matrix* $\Lambda$ *with its element* $\Lambda_{ji} = \zeta_{ji}$ *in the distributed learning methods satisfies the following conditions:*

*(1)* $\Lambda \mathbf{1}_J = \mathbf{1}_J$, $\mathbf{1}_J^T \Lambda = \mathbf{1}_J^T$.

*(2) The spectrum norm of the matrix* $\Lambda - (1/J)\mathbf{1}_J \mathbf{1}_J^T$ *is no more than 1.*

**Theorem 1.** *If Assumption 1 is valid, the estimations of the model parameters* $\{\boldsymbol{w}_{j,q,k}\}$, $\{\boldsymbol{v}_{j,q,k}\}$, *and* $\Theta_j$ *at each node* $j$ *will converge to their optimal values as* $t$ *approaches* $+\infty$, *i.e.,* $\lim_{t\to\infty} ||\boldsymbol{w}_{j,q,k,t} - \boldsymbol{w}_{q,k}^*||_2^2 = 0$, $\lim_{t\to\infty} ||\boldsymbol{v}_{j,q,k,t} - \boldsymbol{v}_{q,k}^*||_2^2 = 0$ *and* $\lim_{t\to\infty} ||\Theta_{j,t} - \Theta^*||_F^2 = 0$, *where* $\boldsymbol{v}_{q,k}^*$ *and* $\Theta^*$ *denote their optimal values, respectively.*

**Proof.** See [2,3]. □

Theorem 1 may be demonstrated by adhering to the proof presented in [1–3]. Consequently, we do not provide the comprehensive proof herein.

Complexity Analysis: The complexity analysis of the suggested method is then reported. This section uses the amount of multiplication and addition operations at each node throughout each iteration to assess computational complexity. The detailed results are summarized in Table 1.

Each node $j$ must send $\epsilon \times \sum_m K_m$ scalars to its one-hop neighbors $i \in \mathcal{B}_j$ at each iteration $\tau$ of the label recovery procedure. In addition, the quantity of scalars exchanged among $|\mathcal{B}_j|$ neighboring nodes during the process of classifier induction is $D\sum_{q=1}^{Q} K_q'$ at each iteration $t$.

The analysis indicates that the complexity of the proposed algorithm is contingent upon the total number of potential classes $\sum_{q=1}^{Q} K_q'$, the rank $\epsilon$, the amount of anchor points $C$, and the dimension $D$, with the value of $\epsilon$ being associated with $\sum_{q=1}^{Q} K_q'$. In actual classification situations, the total number of label classes $\sum_{q=1}^{Q} K_q'$ is constrained, resulting

in a modest value of $\epsilon$. So, as long as the dimension $D$ and the value $C$ are adjusted to a reasonable level, the complexity of our suggested approach is practically manageable.

**Table 1.** Computational complexity of the proposed dS$^2$PMDL algorithm.

| Variables | Multiplication Operations | Addition Operations |
|---|---|---|
| $B_j$ | $2C\epsilon^2+\epsilon^3+C\epsilon\sum_{q=1}^{Q}K_q^{'}$ $+\epsilon\lvert\mathcal{B}_j\rvert\sum_{q}^{Q}K_q^{'}$ | $2C\epsilon^2+\epsilon^3+C\epsilon\sum_{q=1}^{Q}K_q^{'}$ $+\epsilon\lvert\mathcal{B}_j\rvert\sum_{q=1}^{Q}K_q^{'}$ |
| $Z_j^c$ | $(\epsilon+1)N_j\sum_{q=1}^{Q}K_q^{'}$ | $(\epsilon+2)N_j\sum_{q=1}^{Q}K_q^{'}$ |
| $\{z_{j,n,q,k}\}_{n,q,k}$ | $5N_j\sum_{q=1}^{Q}K_q^{'}$ | $3N_j\sum_{q=1}^{Q}K_q^{'}$ |
| $\{\boldsymbol{w}_{j,q,k}\}_{q,k}$ | $(N_j+3)D\sum_{q=1}^{Q}K_q^{'}$ | $(2N_j+3)D\sum_{q=1}^{Q}K_q^{'}$ |
| $\{\boldsymbol{v}_{j,q,k}\}_{q,k}$ | $Dh\sum_{q=1}^{Q}K_q^{'}$ | $Dh\sum_{q=1}^{Q}K_q^{'}$ |
| $\Theta_j$ | $D^3+D^2\sum_{q=1}^{Q}K_q^{'}$ | $D^3+D^2\sum_{q=1}^{Q}K_q^{'}$ |

## 4. Experiments

This section conducts a series of experiments with real MDC datasets to evaluate the efficacy of the proposed method. In these experiments, we utilize MATLAB 2024a to conduct all experiments on an identical workstation featuring a 12th Gen Intel Core (2.10-GHz) CPU, 32 GB of RAM, (Intel Corporation, Santa Clara, CA, USA) and the Windows 11 operating system.

We should first present some related descriptions before conducting the experiments. This experiment examines a distributed network with 20 nodes and 38 edges, with all training data randomly allocated among these nodes. In our configuration, each MDC data sample is allocated a collection of candidate labels inside partially observed class spaces. To produce this type of data, we incorporate additional false positive labels into the candidate labels within the available class spaces while designating the labels in other inaccessible spaces as missing. To assess the influence of different quantities of noisy labels on the learning efficacy of the proposed method, the Average Number of Noisy Labels at each available class space (ANL) is defined. Furthermore, to examine the influence of varying quantities of accessible class spaces on learning performance, the amount of available class spaces at each node $j$, denoted as $|\mathcal{Y}_j|$, is specified. Furthermore, to facilitate an easy comparison of the effectiveness of various comparison algorithms across several aspects, several widely utilized metrics in MDC, including Hamming loss, exact match, and sub-exact match, may be employed. For their detailed definition, please refer to [10,13].

### *4.1. Verification Experiment on Convergence of dPL-MDC Algorithm*

The learning performance of the dPL-MDC method is initially assessed utilizing the "Edm" dataset. In this experiment, we assign the values of the weight parameters $\gamma_A = 0.1$, $\gamma_B = 0.1$, $\gamma_C = 1$, $\gamma_D = 0.001$, $\gamma_E = 0.1$, $C = 30$, the dimension $D = 200$, and the step sizes $\mu_{1,t+1} = \mu_{2,t+1} = 0.5/t^{0.5}$, respectively (available code: https://github.com/yujue-xuzhen/dpl-mdc access on 25 June 2025). Taking the Hamming loss as a representative for performance evaluation, and setting the values of $|\mathcal{Y}_j| = 1$ and ANL=0.2, we illustrate the learning curve of the proposed dPL-MDC method in Figure 2. To assess the efficacy of the suggested method with varying quantities of accessible class spaces, we additionally simulate the Hamming loss of the proposed dPL-MDC with $|\mathcal{Y}_j| = 2$. To differentiate the previous simulation with $|\mathcal{Y}_j| = 1$ (denoted as dPL-MDC case 1), we call this simulation dPL-MDC case 2. Furthermore, we also simulate the Hamming loss of the centralized partial label learning for MDC (denoted by "cPL-MDC", that is, one fusion center centrally

processes all the training data), and the non-cooperative partial label learning for MDC (denoted by "ncPL-MDC", that is, each node in the considered network independently trains classifiers without any information fusion) in Figure 2 for comparison. In fairness to these comparison algorithms, we set $|\mathcal{Y}_j| = 2$ for cPL-MDC and ncPL-MDC, that is, all the training data are assigned with complete label information in all the considered class spaces.

Figure 2 demonstrates that all of the learning curves exhibit comparable tendencies in their evolution. In the initial period, all learning curves experience a rapid decline. All of the learning curves progressively converge and remain unaltered after approximately 150 iterations. These simulation results intuitively illustrate that the dPL-MDC method can obtain classification results that are virtually equivalent to those of the corresponding centralized learning method, and significantly outperform the corresponding non-cooperative method. Additionally, it can also be noticed that the learning performance of the dPL-MDC case 1 is slightly worse than that of the dPL-MDC case 2. Such a simulation result indicates that as the value of $|\mathcal{Y}_j|$ decreases, the number of available labels becomes smaller, which leads to a slight performance deterioration. Owing to the effectiveness of the dPL-MDC algorithm, the performance deterioration can be controlled in an acceptable region.

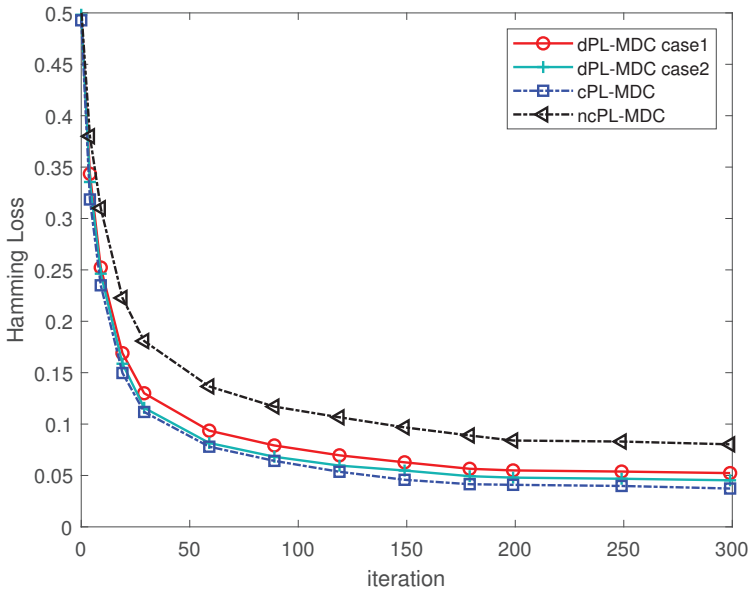



**Figure 2.** The learning curves of different algorithms on "Edm" dataset.

### *4.2. Investigation on Parameter Sensitivity of dPL-MDC Algorithm*

The impacts of the varying values of weight coefficients $\gamma_A$, $\gamma_B$, $\gamma_C$, $\gamma_D$, and $\gamma_E$ on the Hamming loss of the proposed algorithm are investigated in Figure 3. From Figure 3, it can be observed that all parameters have their applicable ranges. Once the parameters are less than the lower bound or greater than the upper bound of the applicable range, the algorithm performs poorly. Conversely, when parameters are selected within the applicable range, the algorithm can achieve superior performance. Consequently, we can ascertain that the appropriate selection of parameters is set as $\gamma_A \in [10^0, 10^1]$, $\gamma_B \in [10^{-1}, 10^0]$, $\gamma_C \in [10^0, 10^1]$, $\gamma_D \in [10^{-2}, 10^{-1}]$ and $\gamma_E \in [10^{-1}, 10^0]$, respectively.

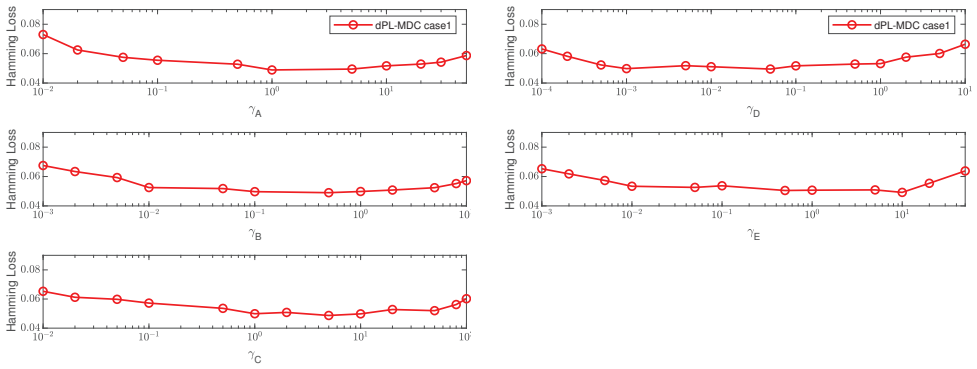



**Figure 3.** Hamming loss of dPL-MDC under varying values of parameters $\gamma_A$, $\gamma_B$, $\gamma_C$, $\gamma_D$, and $\gamma_E$ on "Edm" dataset.

The learning performance of the proposed method is examined in relation to varying numbers of anchor points $C$ and varied dimensions $D$ on the "Edm" dataset. From the simulation results in the upper sub-figure of Figure 4, it can be observed that the classification performance of the algorithm gradually improves as the value of $C$ increases. The analysis reveals that a deficient number of anchor points results in inadequate distribution to encompass all high-density areas. So, its ability to characterize the global data distribution is limited, which in turn affects the ability to eliminate noisy labels. Correspondingly, when the number of anchor points increases to more than 30, their ability to characterize data distribution significantly improves, thereby enhancing the capability to eliminate noisy labels. When the value of $C$ exceeds 30, the performance improvement diminishes progressively. So, setting the number of anchor points $C$ to 30 is suitable.

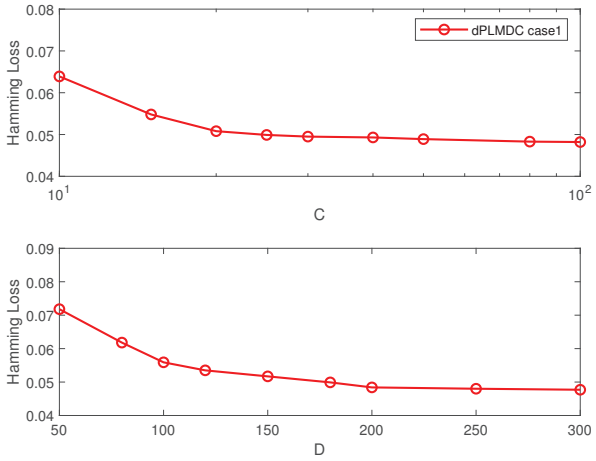



**Figure 4.** Hamming loss of dPL-MDC under varying numbers of anchor points $C$ and dimensions of random feature map $D$ on "Edm" dataset.

Moreover, the Hamming loss of the proposed methods gradually decreases as the value of $D$ increases. When the dimension $D$ is too small, the approximation error of the kernel function by random feature mapping is relatively large, thus affecting the performance of the classifier. As $D$ increases, the random feature map yields a more precise approximation of the kernel feature map, hence enhancing classification performance. A larger dimension of the model parameter implies high computation and communication complexity. To achieve a balance between learning performance and computational complexity, we establish the value of $D$ at 200.

*4.3. Performance Comparison Among Multiple Contrast Algorithms*

In order to further demonstrate the generality of the dPL-MDC algorithm, we conduct more simulations to evaluate its learning performance on a series of MDC datasets, including Edm [15], Song [12], WQani. [12], WQpla. [12], Jura [15], Flare [12] and Music-emo. [33]. The detailed profiles of the used datasets are summarized in Table 2. To examine the effect of different numbers of noisy labels on the learning efficacy of the proposed algorithm, we compare three evaluation metrics of the dPL-MDC algorithm with varying levels of ANL. Furthermore, to emphasize the superiority of our suggested algorithm, the performances of other prominent state-of-the-art algorithms are also evaluated. Given that the challenge of distributed classification of multi-dimensional data with ambiguous labels remains unresolved, we simulate a distributed partial single-dimensional classification method, namely, dS2PLL [31], and four centralized MDC methods, including ECC [19], ESC [6], M3MDC [13], and DLEM [15], for the purpose of comparison.

**Table 2.** Detailed profiles of used datasets.

| Dataset | ♯ Training Exam. | ♯ Testing Exam. | ♯ Feature | ♯ Lab./Dim. |
|---|---|---|---|---|
| Edm | 1230 | 310 | 2 | 3, 3 |
| Song | 3140 | 785 | 3 | 3, 3, 3 |
| WQpla. | 4240 | 1060 | 7 | 4, 4, 4, 4, 4, 4, 4 |
| WQani. | 4240 | 1060 | 7 | 4, 4, 4, 4, 4, 4, 4 |
| Jura | 2870 | 720 | 2 | 4, 5 |
| Flare | 2580 | 650 | 3 | 3, 4, 2 |
| Music-emo. | 5466 | 1367 | 11 | 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 |

To be specific, dS2PLL independently induces a multi-class classifier for each dimensional output space by exploiting the useful information from candidate labels [31]. ECC trains the prediction model by training a sequence of multi-class classifiers inside each dimensional class space, where the output of the preceding classifier is considered as additional characteristics for the subsequent classifier [19]. ESC firstly groups the multi-dimensional class labels into a series of superclass labels, and learns the classifier based on these super-class label variables [6]. M3MDC maximizes the margins between each label pair, and exploits the label dependencies using a covariance regularization term [13]. DLEM initially decomposes the output space of MDC into an encoded label space, thereafter employing the manifold structure to leverage the relationship between pairs of encoded labels [15]. PLEM extracts intrinsic information within the encoded label space by maintaining consistency between attribute distributions and label distributions [16]. It is noted that the hyperparameters for these comparison methods are the same as those used in the referenced publication.

Furthermore, in order to comprehensively assess the efficacy of the dPL-MDC algorithm in dealing with a small amount of missing labels, we also compare the three evaluated metrics of the dPL-MDC algorithm with different values of $|\mathcal{Y}_j|$. Similar to the last experiment, we use the dPL-MDC case 1 to denote the dPL-MDC with $|\mathcal{Y}_j| = Q - 1$, and call the dPL-MDC with $|\mathcal{Y}_j| = Q$ as dPL-MDC case 2 for simplicity. The classification performance of the cPL-MDC algorithm is evaluated as a benchmark.

Initially, we evaluate the learning performance of several algorithms versus the ANL on the "Edm", "Song", "WQani", and "WQpla" datasets, as illustrated in Figures 5–7. It is noted that for the "Edm" and "Song" datasets, only one noisy label at most can be

added into candidate label sets. Furthermore, for the "WQani." and "WQpla." datasets, the maximum number of noisy labels incorporated into the candidate label set is two. Therefore, in this experiment, the range of ANL is, respectively, set as [0.1, 0.9] for the "Edm" and "Song" datasets, and set as [0.1, 1.8] for the "WQani." and "WQpla." datasets. Analysis of the simulation results depicted in these figures reveals a clear trend; as the ANL grows, an increasing number of noisy labels are integrated into the training dataset. The efficacy of all comparison algorithms steadily declines due to the adverse impact of the noisy labels. Furthermore, our proposed dPL-MDC case 1, dPL-MDC case 2, and cPL-MDC algorithm achieve the best three performances among the comparison algorithms in all the metrics, indicating that our proposed algorithm has good generality in solving the MDC problem with a proportion of ambiguous labels. Furthermore, we can see that dPL-MDC case 2 shows better learning performance compared to dPL-MDC case 1. The simulation results demonstrate that more label information can enhance the learning performance of the proposed method to a certain degree.

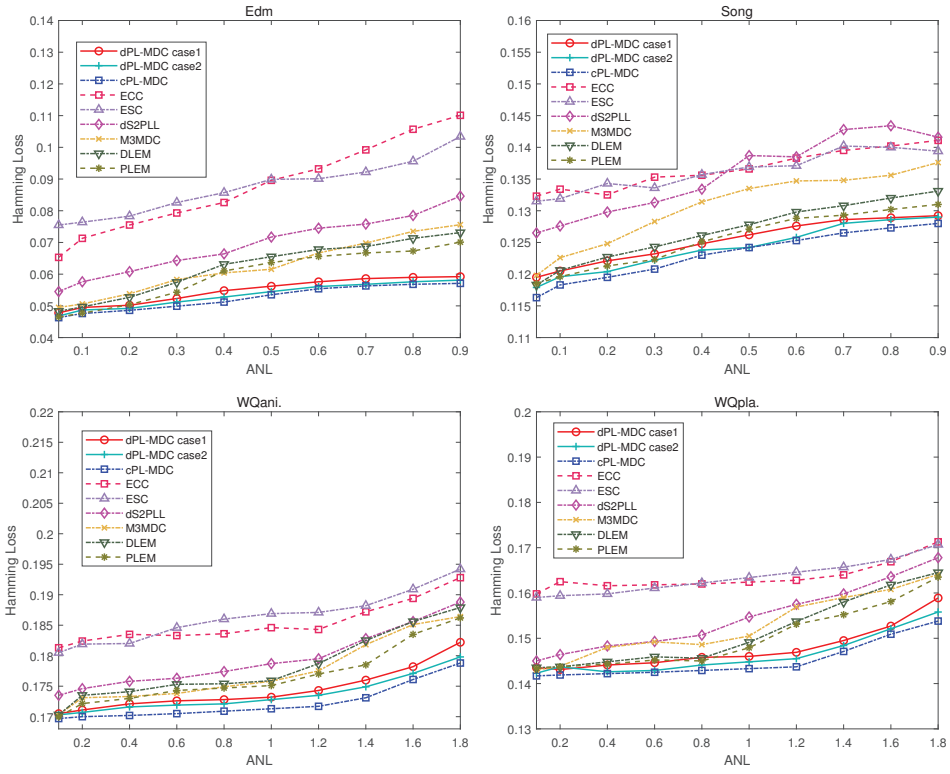



**Figure 5.** Hamming loss of different comparison algorithms versus the ANL on "Edm", "Song", "WQani." and "WQpla." datasets.

1. We can see that the ECC and the ESC perform significantly worse than the other comparison algorithms. The possible reasons are analyzed as follows. During the training of ECC, the output from the preceding classifier is employed as an additional feature in the subsequent classifier. The accuracy of the former classifiers is influenced by the noisy labels, and then the prediction error also expands through the classifier chain propagation, which leads to unsatisfactory classification results. Furthermore, since ESC learns the classifier supervised by the superclasses, its learning performance heavily depends on the accuracy of the grouped superclasses. However, negatively affected by noisy labels, the grouped superclasses may be inaccurate, which seriously deteriorates the classification performance of ESC.

By comparing the learning performance of different comparison algorithms, we can observe the following phenomena.

2. The DLEM algorithm and M3MDC algorithm share similar training patterns, both employing the one-versus-one decomposition method to achieve problem transformation for multi-dimensional learning and leveraging correlations between label pairs to enhance model performance. Benefiting from the effectiveness of the label dependency exploitation strategy, the M3MDC and DLEM perform significantly better than the ESC and ECC. Nevertheless, due to the influence of noisy labels, the learning performances of M3MDC and DLEM are inferior to our proposed algorithm.

3. Furthermore, the dS2PLL initially clarifies the candidate labels by assessing label confidence, and subsequently trains the classifier under the supervision of reliable labels. So, it performs better than ESC and ECC. However, due to ignoring the potential label dependencies among multiple heterogeneous class spaces, the performance of dS2PLL is still inferior to the dPL-MDC algorithm.

4. Furthermore, the PLEM algorithm uses manifold structures to characterize the distributions of feature and label spaces, and explicitly leverages label correlation information for model training. Nevertheless, due to the adverse impact of noisy labels on the characterization of label distributions, the performance of the trained model is degraded. Overall, the PLEM algorithm does not perform as well as our proposed algorithm.

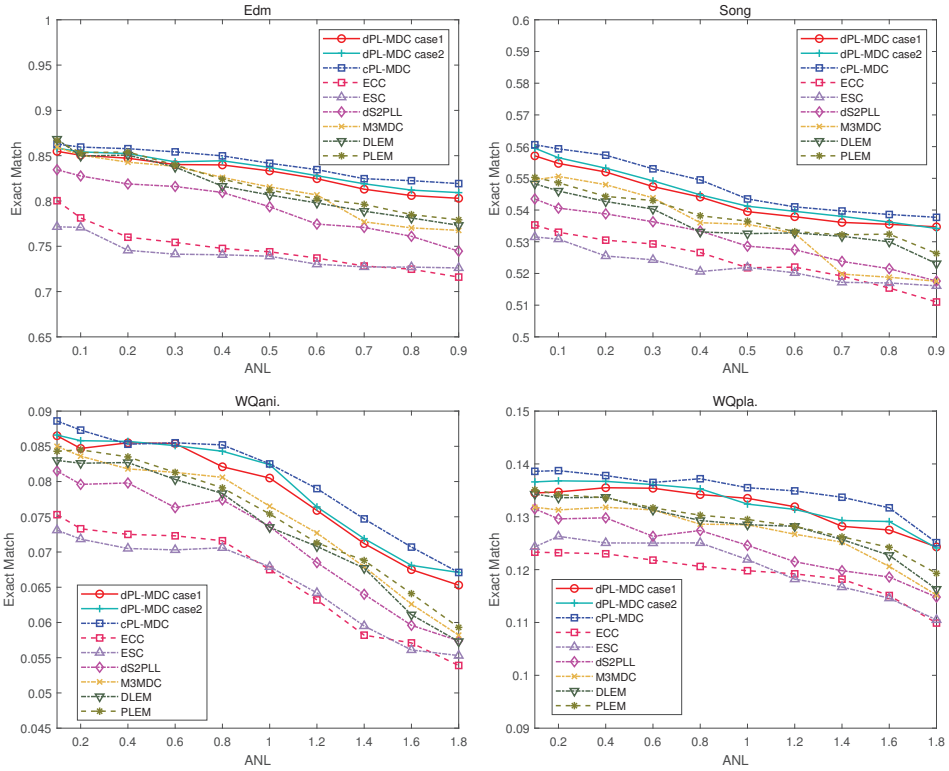



**Figure 6.** Exact match of different comparison algorithms versus the ANL on "Edm", "Song", "WQani." and "WQpla." datasets.

5. The proposed dPL-MDC algorithm demonstrates the best performance among all compared algorithms, and the possible reasons are as follows. When the parameters are set within appropriate ranges, the algorithm exhibits strong capabilities in eliminating noisy labels and exploiting label dependencies in heterogeneous class spaces. This enables the algorithm to significantly outperform other comparison algorithms when the value of ANL is equivalent to 0.8.

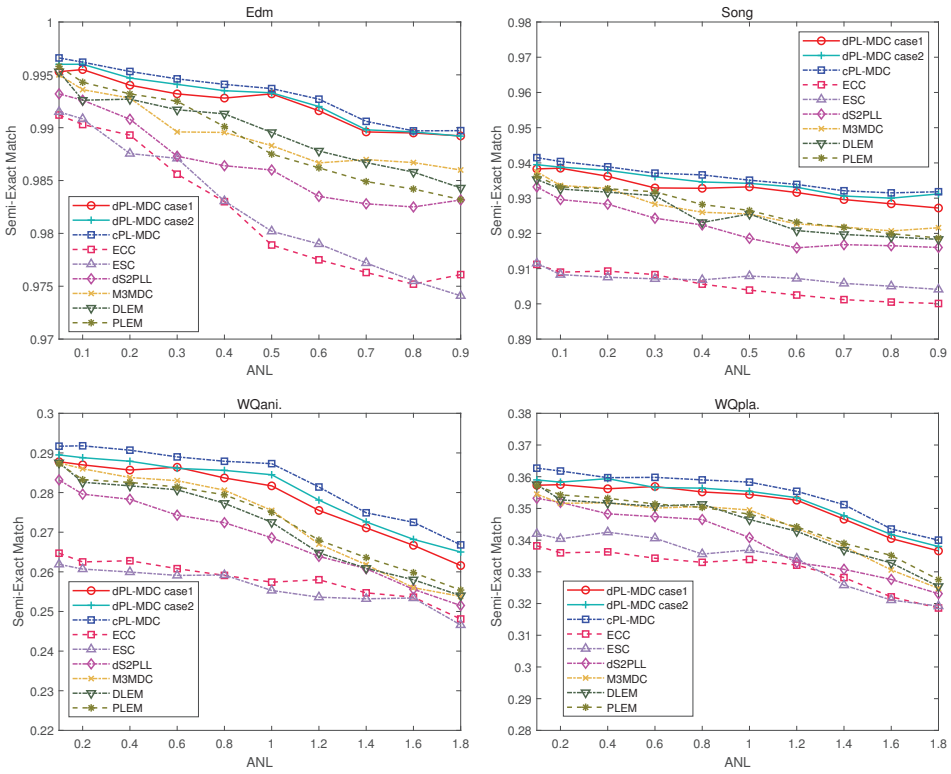



**Figure 7.** Semi-exact match of different comparison algorithms versus the ANL on "Edm", "Song", "WQani." and "WQpla." datasets.

Additionally, we assess the efficacy of all comparison algorithms on the other datasets and obtain comparable results. The pertinent figures are not included in this section due to a page limitation. Instead, we present all the simulation results in terms of three evaluation metrics in Tables 3–5. By observing Tables 3–5, we can draw a similar conclusion as those made above.

**Table 3.** Hamming loss of different algorithms versus ANL on MDC datasets.

| | **Hamming Loss** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Dataset** | **Edm** | **Song** | **WQani.** | **WQpla.** | **Jura** | **Flare** | **Music-Emo.** |
| **ANL** | **0.8** | **0.8** | **0.8** | **0.8** | **0.8** | **0.8** | **0.481** |
| dPL-MDC case 1 | 0.0590 | 0.1289 | 0.1728 | 0.1458 | 0.0587 | 0.0526 | 0.2521 |
| dPL-MDC case 2 | 0.0576 | 0.1286 | 0.1719 | 0.1441 | 0.0570 | **0.0515** | 0.2504 |
| cPL-MDC | **0.0568** | **0.1273** | **0.1708** | **0.1430** | **0.0552** | 0.0519 | **0.2472** |
| $dS^2PLL$ | 0.0785 | 0.1434 | 0.1774 | 0.1507 | 0.0702 | 0.0536 | 0.2913 |
| M3MDC | 0.0735 | 0.1356 | 0.1745 | 0.1484 | 0.0735 | 0.0540 | 0.2924 |
| DLEM | 0.0713 | 0.1320 | 0.1757 | 0.1455 | 0.0707 | 0.0558 | 0.2933 |
| ECC | 0.1057 | 0.1402 | 0.1836 | 0.1621 | 0.0787 | 0.0621 | 0.3101 |
| ESC | 0.0956 | 0.1398 | 0.1870 | 0.1627 | 0.0775 | 0.0605 | 0.3175 |
| PLEM | 0.0685 | 0.1304 | 0.1744 | 0.1446 | 0.0685 | 0.0543 | 0.2872 |

**Table 4.** Exact match of different algorithms versus ANL on MDC datasets.

| | Exact Match | | | | | | |
|---|---|---|---|---|---|---|---|
| **Dataset** | **Edm** | **Song** | **WQani.** | **WQpla.** | **Jura** | **Flare** | **Music-Emo.** |
| **ANL** | **0.8** | **0.8** | **0.8** | **0.8** | **0.8** | **0.8** | **0.481** |
| dPL-MDC case 1 | 0.8091 | 0.5355 | 0.0821 | 0.1342 | 0.7611 | 0.8227 | 0.2552 |
| dPL-MDC case 2 | 0.8102 | 0.5364 | 0.0843 | 0.1354 | 0.7646 | **0.8245** | 0.2564 |
| cPL-MDC | **0.8224** | **0.5387** | **0.0854** | **0.1370** | **0.7706** | 0.8236 | **0.2577** |
| dS$^2$PLL | 0.7612 | 0.5215 | 0.0774 | 0.1274 | 0.7055 | 0.8182 | 0.2087 |
| M3MDC | 0.7703 | 0.5188 | 0.0806 | 0.1285 | 0.7305 | 0.8140 | 0.2047 |
| DLEM | 0.7812 | 0.5300 | 0.0784 | 0.1278 | 0.7412 | 0.8201 | 0.2052 |
| ECC | 0.7247 | 0.5155 | 0.0716 | 0.1126 | 0.7070 | 0.7850 | 0.1880 |
| ESC | 0.7273 | 0.5170 | 0.0707 | 0.1171 | 0.7005 | 0.7826 | 0.1872 |
| PLEM | 0.7850 | 0.5324 | 0.0787 | 0.1303 | 0.7342 | 0.8198 | 0.2126 |

**Table 5.** Semi-exact match of different algorithms versus ANL on MDC datasets.

| | Semi-Exact Match | | | | | | |
|---|---|---|---|---|---|---|---|
| **Dataset** | **Edm** | **Song** | **WQani.** | **WQpla.** | **Jura** | **Flare** | **Music-Emo.** |
| **ANL** | **0.8** | **0.8** | **0.8** | **0.8** | **0.8** | **0.8** | **0.481** |
| dPL-MDC case 1 | 0.9895 | 0.9285 | 0.2842 | 0.3562 | 0.9702 | 0.9524 | 0.3756 |
| dPL-MDC case 2 | **0.9897** | 0.9301 | 0.2856 | 0.3572 | 0.9689 | 0.9552 | 0.3779 |
| cPL-MDC | 0.9896 | **0.9315** | **0.2869** | **0.3590** | **0.9736** | **0.9566** | **0.3785** |
| dS$^2$PLL | 0.9825 | 0.9165 | 0.2724 | 0.3465 | 0.9585 | 0.9368 | 0.3465 |
| M3MDC | 0.9867 | 0.9207 | 0.2806 | 0.3506 | 0.9608 | 0.9482 | 0.3551 |
| DLEM | 0.9858 | 0.9188 | 0.2773 | 0.3513 | 0.9598 | 0.9501 | 0.3568 |
| ECC | 0.9763 | 0.9005 | 0.2556 | 0.3360 | 0.9410 | 0.9257 | 0.3401 |
| ESC | 0.9755 | 0.9050 | 0.2568 | 0.3401 | 0.9487 | 0.9242 | 0.3385 |
| PLEM | 0.9842 | 0.9195 | 0.2794 | 0.3503 | 0.9625 | 0.9517 | 0.3593 |

We rank all the comparison algorithms in Figure 8 in terms of three evaluation metrics to emphasize the advantages of the proposed algorithm from a macro perspective. Our proposed dPL-MDC case 1 and dPL-MDC case 2 algorithms are ranked lower than cPL-MDC. Furthermore, the rankings of the proposed dPL-MDC case 1 and dPL-MDC case 2 are significantly higher than the other comparison algorithms in each evaluation metric. This experimental result further corroborates the efficacy of the proposed method.

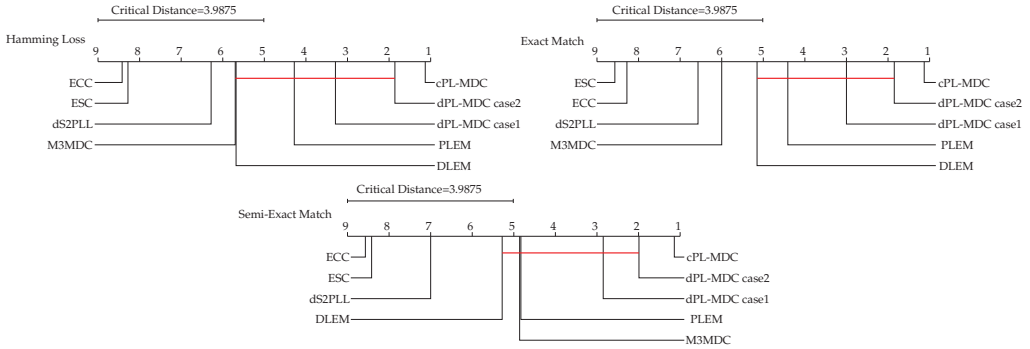



**Figure 8.** Comparison of dPL-MDC case 2 (the control algorithms) in contrast to other comparing algorithms using the Bonferroni–Dunn test.

### *4.4. Performance Comparative Analysis of Different Algorithms*

Furthermore, to further measure the relative performance difference across all comparison methods, the Friedman test is employed [44]. According to its theory, we have the

number of comparison algorithms $k_A = 9$ and the number of data sets $N_D = 7$. Based on $k_A$ and $N_D$, we can calculate the critical value w.r.t. each evaluation metric and the corresponding value of the Friedman statistic $F_F$. The detailed statistical results are presented in Table 6. Obviously, the null hypothesis of no distinguishable performance difference across all the comparison algorithms is rejected for each evaluation metric, with a significant threshold of $\alpha = 0.05$.

**Table 6.** Summary of the Friedman statistics $F_F$ and the critical value in teams of Hamming loss, exact match and semi-exact match.

| Metric | $F_F$ | Critical value ($\alpha = 0.05$) |
|---|---|---|
| Hamming loss | 44.000 | |
| Exact match | 49.814 | 2.14 |
| Semi-exact match | 61.605 | |

Finally, the Bonferroni–Dunn test is used to quantify the relative performance difference between a pair of comparison algorithms [44]. In this experiment, dPL-MDC case 2 is set as the controlled method. We calculate the value of the critical difference (CD) at a significance level $\alpha = 0.05$, and depict the CD diagrams in Figure 8. In each sub-figure of Figure 8, the average rankings of all the comparison algorithms are depicted by black lines, and any comparison algorithm whose average rank is within one CD to that of the controlled method is connected using a red line. From Figure 8, the dPL-MDC instance 2 markedly surpasses the dS2PLL, ECC, and ESC across all evaluation metrics, proving the superiority of our proposed method.

## 5. Conclusions

This study has tackled the challenge of distributed classification of multi-dimensional data labeled with candidate labels inside partially accessible class spaces over a network, and developed the dPL-MDC algorithm. The proposed method has employed one-vs.-one decomposition on the original multi-dimensional output space, which converts the issue of partial multi-dimensional classification (MDC) into a series of problems related to distributed partial multi-label learning. Then, a distributed label recovery approach has been devised to assess the label confidence of the training data. Under the supervision of the recovered labels, by exploiting high-order label dependencies from a common predictive structure in the subspace, the classifier has been trained. A number of simulations using multiple MDC datasets have been conducted to validate the efficacy of the proposed approach. Existing experimental results show that as long as the parameters of the proposed algorithm are set within a reasonable range, it can significantly outperform existing comparison algorithms. Especially when the proportion of noisy labels is high, the performance advantage of our proposed algorithm becomes more significant.

Nonetheless, the proposed algorithm still has some limitations. For example, the proposed algorithm currently relies on manual selection of hyperparameters. Therefore, in the future, we would like to integrate swarm intelligence algorithms to assist in automatically setting the values of hyperparameters. Additionally, the proposed algorithm is currently more suitable for small-scale networks. In the future, developing a new information cooperation model between nodes to adapt to large-scale networks is also an interesting direction. Moreover, considering the excellent performance of deep learning in many fields, leveraging related deep learning technologies to achieve continual learning is also a potential research direction in the future.

**Author Contributions:** Conceptualization, Z.X.; Methodology, Z.X.; Software, S.C.; Validation, S.C.; Formal analysis, Z.X.; Investigation, S.C.; Resources, S.C.; Data curation, S.C.; Writing—original draft, Z.X.; Writing—review & editing, Z.X.; Supervision, Z.X.; Project administration, Z.X.; Funding acquisition, Z.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partly supported by the National Natural Science Foundation of China (Grant No. 62201398).

**Data Availability Statement:** All data generated or analyzed during this study are included in this published article.

**Conflicts of Interest:** Author Sicong Chen was employed by the company Kasco Signal 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. Liu, Y.; Xu, Z.; Li, C. Distributed online semi-supervised support vector machine. *Inf. Sci.* **2018**, *466*, 236–257. [CrossRef]
2. Hua, J.; Li, C.; Shen H. Distributed learning of predictive structures from multiple tasks over networks. *IEEE Trans. Ind. Electron.* **2017**, 5, 4246–4256. [CrossRef]
3. Xu, Z.; Liu, Y.; Li, C. Distributed semi-supervised learning with missing data. *IEEE Trans. Cybern.* **2021**, *51*, 6165–6178. [CrossRef] [PubMed]
4. Forero, P.A.; Cano, A.; Giannakis, G.B. Consensus-based distributed support vector machines. *J. Mach. Learn. Res.* **2010**, *11*, 1663–1707.
5. Liu, Y.; Xu, Z.; Zhang, C. Distributed semi-supervised partial label learning over networks. *IEEE Trans. Artif. Intell.* **2022**, *3*, 414–425. [CrossRef]
6. Read, J.; Bielza, C.; Larranaga, P. Multi-dimensional classification with super-classes. *IEEE Trans. Knowl. Data Eng* **2014**, *26*, 1720–1733. [CrossRef]
7. Serafino, F.; Pio, G.; Ceci, M.; Malerba, D. Hierarchical multi-dimensional classification of web documents with multiwebclass. In Proceedings of the 18th International Conference on Discovery Science, Banff, AB, Canada, 4–6 October 2015; pp. 236–250.
8. Yang, M.; Deng, C.; Nie, F. Adaptive-weighting discriminative regression for multi-view classification. *Pattern Recognit.* **2019**, *88*, 236–245. [CrossRef]
9. Borchani, H.; Bielza, C.; Toro, C.; Larranaga, P. Predicting human immunodeficiency virus inhibitors using multi-dimensional Bayesian network classifiers. *Artif. Intell. Med.* **2013**, *57*, 219–229. [CrossRef]
10. Jia, B.-B.; Zhang, M.-L. Multi-dimensional classification via kNN feature augmentation. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 3975–3982.
11. Wang, H.; Chen, C.; Liu, W.; Chen, K.; Hu, T.; Chen, G. Incorporating label embedding and feature augmentation for multi-dimensional classification. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 6178–6185.
12. Jia, B.-B.; Zhang, M.-L. Maximum margin multi-dimensional classification. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 4312–4319.
13. Jia, B.-B.; Zhang, M.-L. Multi-dimensional classification via stacked dependency exploitation. *Sci. China Inf. Sci.* **2020**, *63*, 222102. [CrossRef]
14. Jia, B.-B.; Zhang, M.-L. Decomposition-based classifier chains for multi-dimensional classification. *IEEE Trans. Artif. Intell.* **2022**, *3*, 176–191. [CrossRef]
15. Jia, B.-B.; Zhang, M.-L. Multi-dimensional classification via decomposed label encoding. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 1844–1856. [CrossRef]
16. Tang, J.; Chen, W.; Wang, K.; Zhang, Y.; Liang, D. Probability-based label enhancement fro multi-dimensional classification. *Inf. Sci.* **2024**, *653*, 119790. [CrossRef]
17. Zhu, M.; Liu, S.; Jiang, J. A hybrid method for learning multi-dimensional Bayesian network classifiers based on an optimization model. *Appl. Intell.* **2016**, *44*, 123–148. [CrossRef]
18. Bolt, J.; Gaag, L.C. Balanced sensitivity functions for tuning multi-dimensional Bayesian network classifiers. *Int. J. Approx. Reason.* **2017**, *80*, 361–376. [CrossRef]
19. Read, J.; Martino, L.; Luengo, D. Efficient monte carlo methods for multi-dimensional learning with classifier chains. *Pattern Recogn.* **2014**, *47*, 1535–1546. [CrossRef]

20. Zaragoza, J.; Sucar, L.; Morales, E.; Bielza, C.; Larranaga, P. Bayesian chain classifiers for multidimensional classification. In Proceeding of the 22nd International Joint Conference on Artificial Intelligence (IJCAI), Barcelona, Spain, 12–16 July 2011; pp. 2192–2197.
21. Zhang, M.-L.; Yu, F.; Tang, C. Disambiguation-free partial label learning. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2155–2167. [CrossRef]
22. Tang, C.; Zhang, M.-L. Confidence-rated discriminative partial label learning. In Proceeding of the 31st AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 2611–2617.
23. Feng, L.; Lv, J.; Han, B.; Xu, M.; Niu, G.; Geng, X.; An, B.; Sugiyama, M. Provably consistent partial-label learning. In Proceedings of the 33rd Neural Information Processing Systems (NeurIPS'20), Virtual Conference, 6–12 December 2020; pp. 10948–10960.
24. Lv, J.; Xu, M.; Feng, L.; Niu, G.; Geng, X.; Sugiyama, M. Progressive identification of true labels for partial-label learning. In Proceedings of the 37th International Conference on Machine Learning (PMLR), Virtual Conference, 13–18 July 2020; pp. 6500–6510.
25. Zhang, Q.; Zhu, Y.; Cordeiro, F.R.; Chen, Q. PSSCL: A progressive sample selection framework with contrastive loss designed for noisy labels. *Pattern Recogn.* **2025**, *161*, 111284. [CrossRef]
26. Wang, W.; Zhang, M.-L. Semi-supervised partial label learning via confidence-rated margin maximization. In Proceedings of the 33rd Neural Information Processing Systems (NeurIPS'20), Virtual Conference, 6–12 December 2020; pp. 6982–6993.
27. Miao, X.; Liu, Y.; Zhao, H.; Li, C. Distributed online one-class support vector machine for anomaly detection over networks. *IEEE Trans. Cybern.* **2019**, *49*, 1475–1488. [CrossRef]
28. Shen, X.; Liu, Y.; Zhang, Z. Performance-enhanced Federated Learning with Differential Privacy for Internet of Things. *IEEE Int. Things J.* **2022**, *9*, 24079–24094. [CrossRef]
29. Shen, X.; Liu, Y. Privacy-preserving distributed estimation over multitask networks. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 1953–1965. [CrossRef]
30. Chen, S.; Liu, Y. Robust distributed parameter estimation of nonlinear systems with missing data over networks. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 2228–2244. [CrossRef]
31. Xu, Z.; Liu, Y.; Li, C. Distributed information theoretic semisupervised learning for multilabel classification. *IEEE Trans. Cybern.* **2022**, *52*, 821–835. [CrossRef]
32. Xu, Z.; Zhai, Y.; Liu, Y. Distributed semi-supervised multi-label classification with quantized communication. In Proceedings of the 12th International Conference on Machine Learning and Computing, Shenzhen, China, 15–17 February 2020; pp. 57–62.
33. Fang, J.-P.; Zhang, M.-L. Partial multi-label learning via credible label elicitation. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 3518–3525.
34. Sun, L.; Feng, S.; Wang, T.; Lang, T.; Jin, Y. Partial multi-label learning by low-rank and sparse decomposition. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 5016–5023.
35. Yu, T.; Yu, G.; Wang, J.; Domeniconi, C.; Zhang, X. Partial multi-label learning using label compression. In Proceedings of 2020 IEEE International Conference on Data Mining (ICDM), Sorrento, Italy, 17–20 November 2020; pp. 761–770.
36. Xie, M.-K.; Huang, S. Semi-supervised partial multi-label learning. In Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM), Sorrento, Italy, 17–20 November 2020; pp. 691–700.
37. Xie, M.-K.; Huang, S.J. Partial multi-label learning with noisy label identification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3676–3687. [PubMed]
38. Li, C.; Luo, Y. Distributed vector quantization over sensor network. *Int. J. Dis. Sens. Netw.* **2014**, *10*, 189619. [CrossRef]
39. Wen, Z.; Yin, W.; Zhang, Y. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Math. Program. Comput.* **2012**, *4*, 333–361. [CrossRef]
40. Lin, A.Y.; Ling, Q. Decentralized and privacy-preserving low-rank matrix completion. *J. Oper. Res. Soc. China* **2015**, *3*, 1–17. [CrossRef]
41. Rahimi, A.; Recht, B. Random features for large-scale kernel machines. In Proceedings of the 21st International Conference on Neural Information Processing Systems (NeurIPS'07), Vancouver, BC, Canada, 3–6 December 2007; pp. 1177–1184.
42. Sreekanth, V.; Vedaldi, A.; Zisserman, A.; Jawahar, C. Generalized RBF feature maps for efficient detection. In Proceedings of the 21st British Machine Vision Conference, Aberystwyth, UK, 31 August–3 September 2010; pp. 1–11.
43. Luo, Y.; Tao, D.; Geng, B.; Xu, C.; Maybank, S.J. Manifold regularized multitask learning for semi-supervised multilabel image classification. *IEEE Trans. Image Process.* **2013**, *22*, 523–536. [CrossRef]
44. Demsar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.

*electronics*

MDPI

*Article*

# AgriTransformer: A Transformer-Based Model with Attention Mechanisms for Enhanced Multimodal Crop Yield Prediction

**Luis Jácome Galarza [1], Miguel Realpe [1], Marlon Santiago Viñán-Ludeña [2,*], María Fernanda Calderón [1] and Silvia Jaramillo [3]**

[1] Escuela Superior Politécnica del Litoral, ESPOL, Facultad de Ingeniería en Electricidad y Computación, CIDIS, Campus Gustavo Galindo, km 30.5 Vía Perimetral, P.O. Box 09-01-5863, Guayaquil 090112, Ecuador; ljacome@espol.edu.ec (L.J.G.); mrealpe@espol.edu.ec (M.R.); mafercal@espol.edu.ec (M.F.C.)
[2] Escuela de Ingeniería, Universidad Católica del Norte, Coquimbo 1781421, Chile
[3] Business School, Universidad Internacional del Ecuador, Loja 110111, Ecuador; sijaramillolu@uide.edu.ec
* Correspondence: marsantovi@gmail.com

**Abstract:** A more accurate crop yield estimation is essential for optimizing agricultural productivity and resource management. Traditional machine learning models, such as linear regression and convolutional neural networks (CNNs), often struggle to integrate multimodal data sources effectively, limiting their predictive accuracy. In this study, we propose the AgriTransformer model, a transformer-based model that enhances crop yield prediction by leveraging attention mechanisms for multimodal data fusion. The AgriTransformer model incorporates tabular agricultural data and vegetation indices (VI), allowing dynamic feature interaction and improved interpretability. Experimental results have demonstrated that AgriTransformer significantly outperforms conventional approaches, achieving an $R^2$ of 0.919, compared to 0.884 for the best-performing linear regression model. The findings highlight the importance of structured tabular data in yield estimation, while VI serves as a complementary feature that increases the prediction capability and confidence. This study highlights the potential of transformer-based architectures in precision agriculture, offering a scalable and adaptable framework for crop yield forecasting. The AgriTransformer model enhances predictive accuracy and generalization across diverse agricultural conditions by prioritizing relevant features through attention mechanisms.

**Keywords:** precision agriculture; transformers architecture; attention mechanism; deep learning; multimodal learning

## 1. Introduction

Estimating crop yield accurately remains a fundamental challenge in agriculture, having a big impact on assuring food security, planting and harvest planning, and the use of pesticides and fertilizers. However, many factors influence crop yield, including soil condition, weather, crop management practices, diseases, and pests. The presence of these elements causes the harvest to present great spatial-temporal variability and is difficult to model jointly, especially in extensive areas [1,2]. Moreover, precise estimation of agricultural production is essential for feeding our growing population [3]. reveals that food demand also refers to producing quality food that contains proteins, vitamins, or minerals. His work encourages us to change our perspective on sustainable agriculture; in these circumstances, forecasting food production is highly pertinent and recommendable.

Traditional prediction models, such as mathematical and statistical methods applied to weather, soil, and management data, or vegetation indices obtained from remote data,

often fail to generalize across different regions and crop types. These approaches usually rely on a single data modality and assume linear relationships between variables, limiting their predictive capacity. Following this approach [4], explain that vegetation indices are used with regression methods to forecast the yield of crops, and data mining techniques add information like weather, soil status, or management practices that improve the production estimation.

Likewise [5], explain that satellite images are improving the accuracy of crop production estimation. In their experiments, they used images with resolutions of 250 m, 500 m, and 1 km, where the resolution of 250 m means that a pixel represents an area of 250 $\times$ 250 m. The prediction model used regression to estimate the yield having the NDVI (Normalized Difference Vegetation Indices) values as the independent variable, obtaining a coefficient of determination $r^2$ value up to 0.615; this metric is improved when they use higher resolution images; however, it is explained that using satellite images faces challenges like the storage demand, cloudiness, or lack of high-quality images.

Recent studies have utilized machine learning and deep learning approaches, such as random forests, convolutional neural networks, and long short-term memory models, to improve yield estimation [6,7]. While these methods have shown improvements, they often miss the full potential of multimodal data fusion, which is combining structured tabular information (e.g., soil, climate, crop type, water information, or plant data) with unstructured sources like satellite imagery or vegetation indices [8]. Moreover, most models process these modalities independently or simply concatenate them, which fails to capture cross-modal interactions that are crucial for understanding plant development and health.

In this work, we propose AgriTransformer, a deep learning architecture based on cross-modal attention that jointly models tabular data and vegetation indices [9]. By employing a co-attention mechanism, the model can learn interdependencies between variables such as irrigation conditions, crop type, or management practices, and on the other hand, vegetation indices, enabling more accurate yield predictions. Our approach addresses key challenges in yield estimation, including:

- Integration of heterogeneous data sources.
- Modeling nonlinear relationships and latent interactions.
- Enhancing robustness across diverse field conditions.

The AgriTransformer model was evaluated using real-world data from Telangana, India, and demonstrated that it significantly outperforms baseline models in terms of mean squared error and coefficient of determination ($R^2$).

### *1.1. Related Work*

Crop yield estimation has been studied exhaustively using statistical, machine learning, and remote sensing techniques. Early approaches relied on linear regression models using climatic and agronomic variables [10], but these methods usually failed to capture the complex, nonlinear interactions between soil, weather, and crop characteristics.

In order to improve the accuracy of yield estimation, researchers are using machine learning models such as Random Forest [11], Support Vector Regression [12], and Gradient Boosting Machines [13]. Even though these models capture more complex patterns than traditional statistical techniques, they require extensive feature engineering and still operate mainly on tabular data, without taking advantage of spatial and temporal information of imagery from remote sensors.

On the other hand, methods that utilize remote sensor data have been used to monitor crop conditions through vegetation indices such as NDVI, EVI, and SAVI [14]. These indices help assess plant vigor and photosynthetic activity. Although these models provide useful

information, they suffer from saturation in dense canopies and are sensitive to atmospheric noise, cloud cover, and low-quality images.

Furthermore, recent deep learning approaches have improved modeling complex data modalities. Convolutional Neural Networks (CNNs) have been applied to satellite imagery for yield prediction [15,16], while Recurrent Neural Networks (RNNs) and LSTMs have been used to capture temporal patterns [17]. However, these methods often treat different modalities (such as images and tabular features) independently or merge them using simple concatenation, ignoring the latent interactions of diverse data modalities.

### *1.2. The Attention Mechanism in AgriTransformer*

The attention mechanism is a key technique in deep learning models like the Transformer architecture, and it assigns different weights to the most relevant parts of the input information. In the context of crop yield estimation, the attention mechanism is used to merge multiple sources of data, such as satellite images, field sensors, and time-series weather data [18].

#### 1.2.1. Types of Attention in Deep Learning

First, we have the Self-Attention mechanism, which is used in models such as Vision Transformers (ViTs) and BERT, which has been widely used in social media data processing, and it allows each input element to interact with every other element, assigning weights according to their importance. The key technique is the scaled-dot product attention, which computes the relevance of each element [19].

Next, we have the Cross-Modal Attention, which has emerged as a powerful solution to fuse multimodal data by learning where and how different modalities influence the outcome [20]. The formula for obtaining cross-attention is similar to the formula of self-attention, with the difference that in cross-attention, the information comes from different sources, where the model uses the decoder's output to generate the query (Q), and it looks for keys (K) and values (V) that come from the encoder to obtain the answers.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V \quad (1)$$

Certain studies have used the Transformer architecture with agricultural tasks, but their application to yield estimation remains limited. Our work extends this line of research by introducing a co-attention mechanism adapted to the agricultural domain, enabling the model to learn dynamic relationships between vegetation indices and tabular features, such as crop type, irrigation, and crop management practices. This approach allows the model to learn

#### 1.2.2. Comparison with Traditional Methods

Unlike traditional methods that analyze each modality separately, AgriTransformer uses Cross-Modal Attention in order to learn complex relationships between diverse modalities of data, improving the accuracy of yield estimation and being more adaptive to changes in weather and geographic conditions.

This paper continues as follows: in Section 2, we explain the methodology that were used in the project, describe the dataset, and explain the algorithms in detail; in Section 3, we expose the empirical results of the experiments; in Section 4, we analyze the results and support our ideas; finally, in Section 5, we present the conclusions of the research project.

## 2. Materials and Methods

### *2.1. Methodology*

According to [21], the process of machine learning-based crop yield prediction consists of stages like data collection, data pre-processing, building the machine learning model (in this case, it can be a regression model for estimating yield or a classification model for estimating the health of a crop), and making crop yield predictions in a real-world scenario. Figure 1 illustrates the steps taken in the present project for crop yield prediction using machine learning.

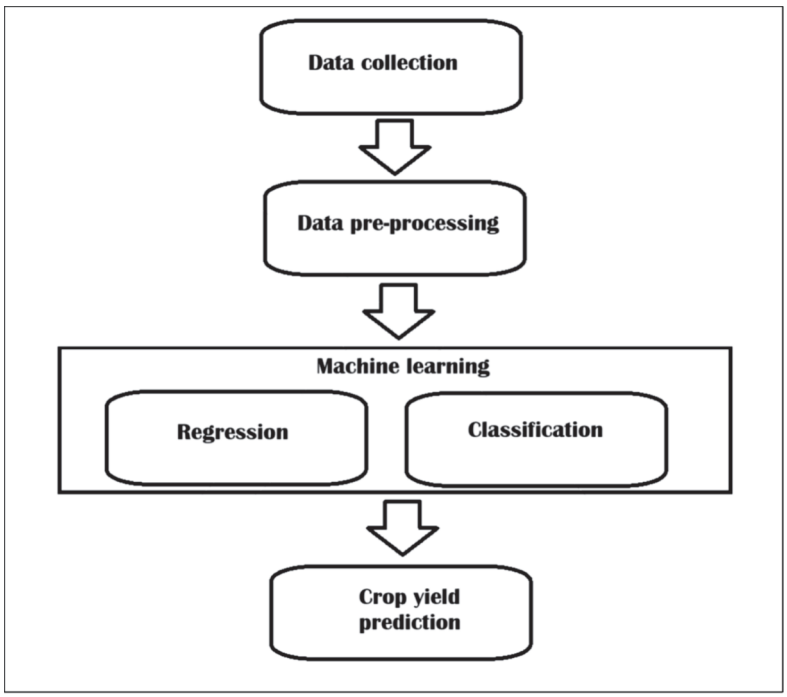
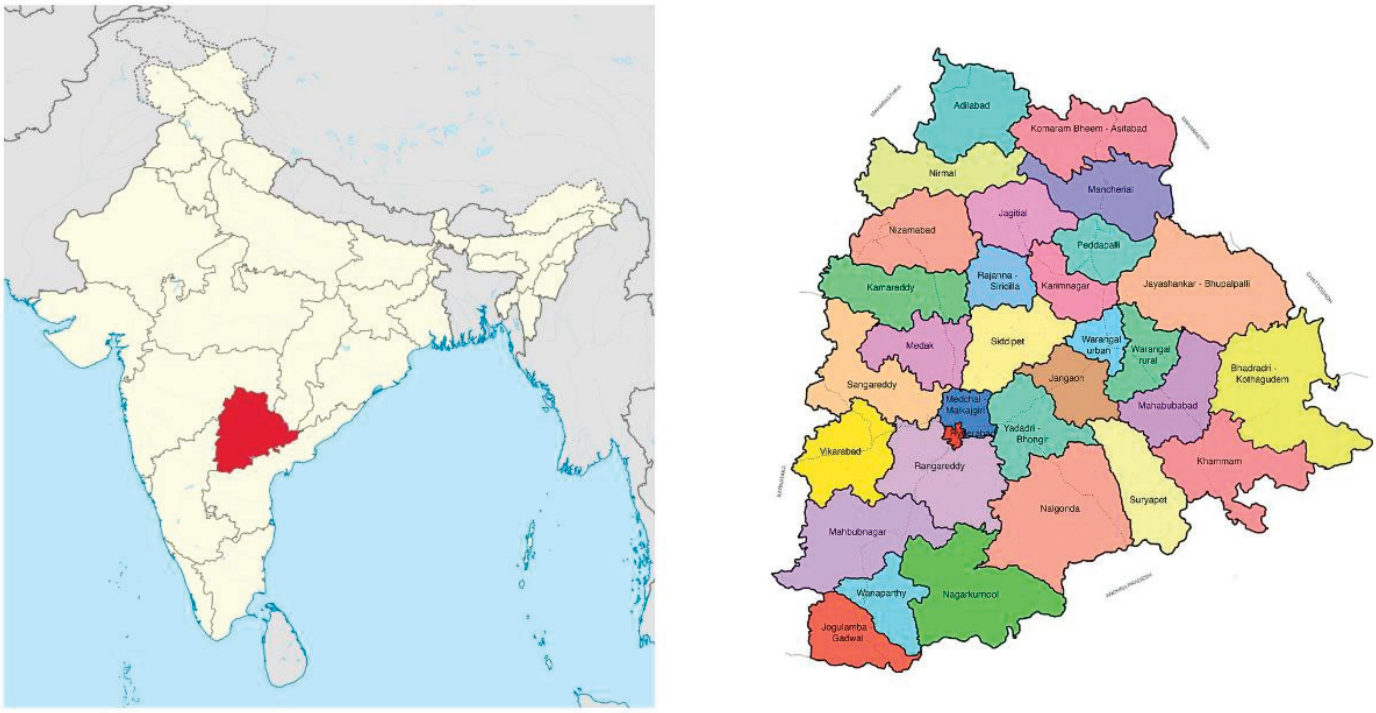



**Figure 1.** General architecture of machine learning-based crop yield prediction [21]. Methodology used in the present project.

### *2.2. Description of the Utilised Dataset*

The data used for the experiments is the dataset from the Telangana Crop Health Challenge, available on [22]. The dataset contains 10,037 rows with information about the same number of farms located in diverse locations in Telangana state, India (Figure 2).

The dataset contains, on the one hand, farm tabular data such as type of crop, crop cover area, sowing date, harvest date, crop height, condition of crop transpiration, type of irrigation method, irrigation type (possible values: drip, sprinkler, or surface), irrigation source (possible values: canal, borewell, or rainfall), number of times the farm has been irrigated, estimated percentage of the area covered with water due to irrigation, season in which the crop is cultivated, geographical data such as state, district, and sub-district. Among those fields, we considered the "expected yield" field as the target variable, which consists of a numeric value in the unit of hundred weight per acre, which is used in the United Kingdom, where 100 weight equals 112 pounds or 50.8 kg, and an acre equals 4047 square meters. On the other hand, the dataset has the geometry field, which contains the physical coordinates or spatial geometry of the farm location, as seen in Table 1.




(**a**) (**b**)

**Figure 2.** (**a**) The Telangana state in India and (**b**) the districts of the Telangana state.

**Table 1.** Field geometry description on the Telangana Crop Health Challenge dataset.

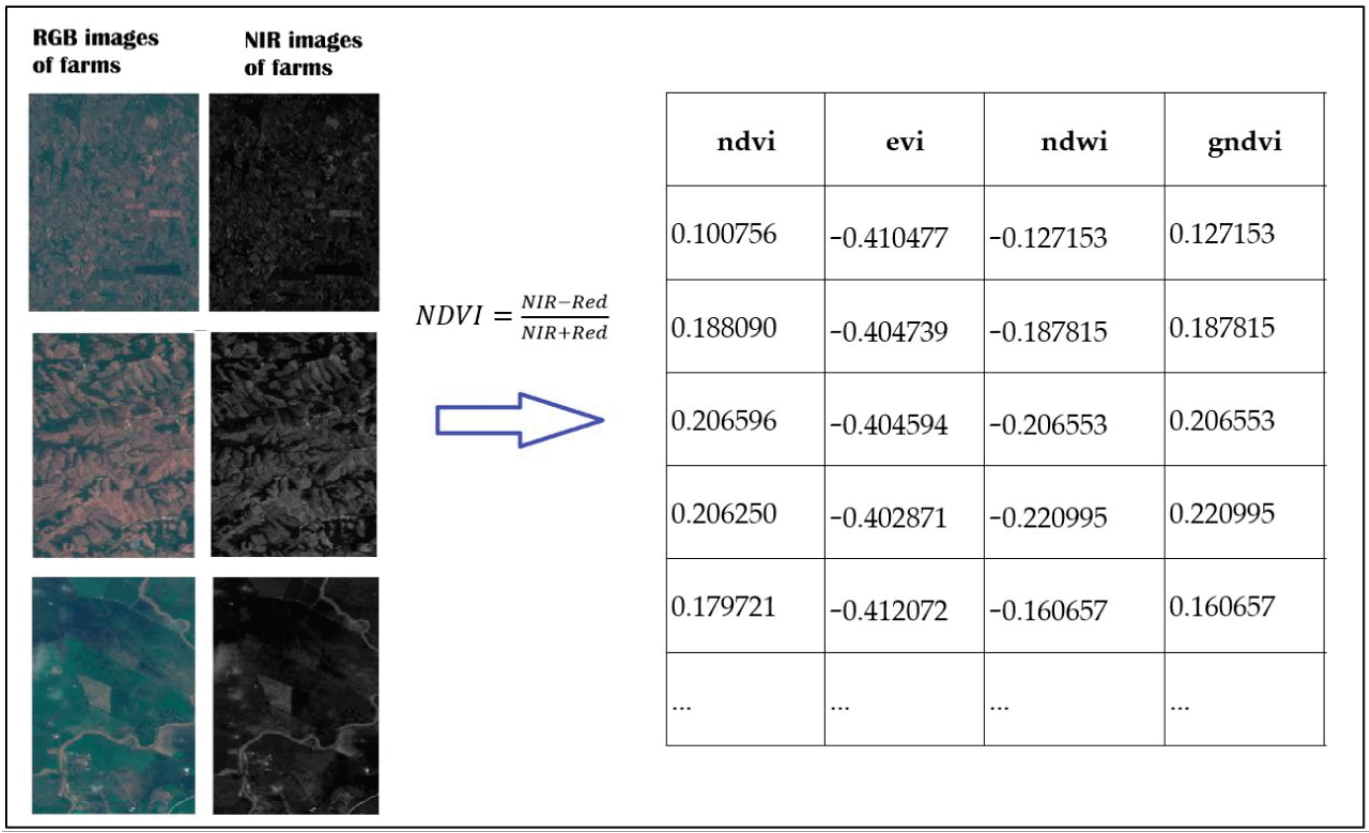
| Index | Geometry |
|---|---|
| 0 | POLYGON ((78.18143 17.97888, 78.18149 17.97899, 78.18175 17.97887, 78.18166 17.97873, 78.18143 17.97888)) |
| 1 | POLYGON ((78.17545 17.98107, 78.17578 17.98104, 78.17574 17.98086, 78.17545 17.98088, 78.17545 17.98107)) |
| 2 | POLYGON ((78.16914 17.97621, 78.1693 17.97619, 78.16928 17.97597, 78.16911 17.97597, 78.16914 17.97621)) |
| 3 | POLYGON ((78.16889 17.97461, 78.16916 17.97471, 78.16923 17.97456, 78.16895 17.97446, 78.16889 17.97461)) |
| 4 | POLYGON ((78.17264 17.96925, 78.17276 17.96926, 78.17276 17.96913, 78.17273 17.96905, 78.17264 17.96925)) |
| ... | ... |
| 8770 | POLYGON ((78.79225 19.7354, 78.79276 19.73531, 78.7927 19.73418, 78.79213 19.73423, 78.79225 19.7354)) |
| 8771 | POLYGON ((78.79762 19.75388, 78.79859 19.75375, 78.79853 19.75335, 78.79751 19.75337, 78.79762 19.75388)) |
| 8772 | POLYGON ((78.80798 19.75445, 78.80899 19.75448, 78.80895 19.75415, 78.80795 19.75412, 78.80798 19.75445)) |
| 8773 | POLYGON ((78.80939 19.75338, 78.81022 19.75344, 78.81018 19.75305, 78.80942 19.75302, 78.80939 19.75338)) |
| 8774 | POLYGON ((80.11489 17.37211, 80.11505 17.37208, 80.11508 17.37193, 80.11511 17.37158, 80.11489 17.37211)) |

With the use of the Shapely library 2.0.3 and the Python language 3.11.13, we could scale those geometries around their centroid point. Having those geographical coordinates

and shapes, we used the Google Earth Engine script to download the multispectral images from the Sentinel-2 satellite system. These multispectral images have a spatial resolution of 10 m, and the image for each farm is taken depending on its availability between the crop sowing and harvesting dates. The tabular data, which is a CSV file, was stored in Google Drive, and then we used Google Colaboratory to execute the aforementioned Python script for downloading the multispectral images and storing them in Google Drive.

### *2.3. Data Pre-Processing*

Once we have the data, it comes to the pre-processing stage. We proceeded to remove those rows that had null values and those that had invalid information about the geographical coordinates of the farm. After this step, our dataset contained 7796 rows with tabular information about the agricultural management of the crop and a link to the multispectral image of each row. Figure 3 shows some RGB images and near-infrared images of the farms in the dataset.




| ndvi | evi | ndwi | gndvi |
|---|---|---|---|
| 0.100756 | -0.410477 | -0.127153 | 0.127153 |
| 0.188090 | -0.404739 | -0.187815 | 0.187815 |
| 0.206596 | -0.404594 | -0.206553 | 0.206553 |
| 0.206250 | -0.402871 | -0.220995 | 0.220995 |
| 0.179721 | -0.412072 | -0.160657 | 0.160657 |
| ... | ... | ... | ... |

**Figure 3.** Samples of satellite RGB and NIR images of farms.

As seen in Figure 3, we conducted the processing of multispectral images for obtaining the vegetation indices and used the blue, red, green, and near-infrared channels to obtain the vegetation indices, as shown in Table 2.

After calculating the vegetation indices for each row of the dataset, we included those values and obtained the final dataset with 2 groups of fields: the management practices of the farm (Table 3) and the vegetation indices of that farm (Table 4). It is worth noting that these 2 groups of fields come from heterogeneous sources: tabular data and multispectral images.

**Table 2.** Description of vegetation indices that were used in the project.

| Vegetation Index | Use | Formula | |
|---|---|---|---|
| NDVI (Normalized Difference Vegetation Index) | It is used for assessing the health and density of vegetation [23] | $NDVI = \frac{NIR-Red}{NIR+Red}$ | (2) |
| EVI (Enhanced Vegetation Index) | It is used for adjusting the relation between vegetation and soil or when the NDVI index is not good [24] | $EVI = \frac{G*NIR-Red}{NIR+C1*Red-C2*Blue+L}$ | (3) |
| NDWI (Normalized Difference Water Index) | It is used to monitor the amount of water on the surface or moisture on the ground [25] | $NDWI = \frac{Green-NIR}{Green+NIR}$ | (4) |
| GNDVI (Green Normalized Difference Vegetation Index) | It is used to assess the health of vegetation, especially when the NDVI index is not sensitive enough [10,26,27] | $GNDVI = \frac{NIR-Green}{NIR+Green}$ | (5) |
| SAVI (Soil Adjusted Vegetation Index) | It is used when the soil is visible in satellite images in order to reduce the effect of soil in areas with a low amount of vegetation [28] | $SAVI = \frac{(NIR-Red)*(1+L)}{NIR+Red+L}$ | (6) |
| MSAVI (Modified Soil Vegetation Index) | It is used in areas with low vegetation amount and to obtain a more accurate assessment of the vegetation [29] | $MSAVI = \frac{2*NIR+1-\sqrt{(2*NIR+1)^2-8*(NIR-Red)}}{2}$ | (7) |

**Table 3.** The farm management fields of the dataset.

| Crop | State | District | Sub-District | CropCoveredArea | CHeight | IrriType | IrriSource | IrriCount | WaterCov |
|---|---|---|---|---|---|---|---|---|---|
| 5 | 0 | 5 | 61 | 97 | 54 | 1 | 1 | 4 | 87 |
| 5 | 0 | 5 | 61 | 82 | 58 | 1 | 0 | 5 | 94 |
| 5 | 0 | 5 | 61 | 92 | 91 | 1 | 0 | 3 | 99 |
| 5 | 0 | 5 | 61 | 91 | 52 | 1 | 0 | 5 | 92 |
| 5 | 0 | 5 | 61 | 94 | 55 | 1 | 0 | 5 | 97 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2 | 0 | 0 | 11 | 78 | 81 | 0 | 3 | 2 | 60 |
| 2 | 0 | 0 | 11 | 81 | 110 | 0 | 2 | 3 | 45 |
| 2 | 0 | 0 | 11 | 68 | 66 | 2 | 0 | 3 | 58 |
| 2 | 0 | 0 | 11 | 84 | 101 | 0 | 2 | 3 | 52 |
| 1 | 0 | 2 | 51 | 60 | 100 | 0 | 1 | 2 | 46 |

**Table 4.** Vegetation indices fields of the dataset and the ground truth (ExpYield).

| ndvi | evi | ndwi | gndvi | savi | msavi | ExpYield |
|---|---|---|---|---|---|---|
| 0.100756 | −0.410477 | −0.127153 | 0.127153 | 0.150938 | 0.182590 | 17 |
| 0.188090 | −0.404739 | −0.187815 | 0.187815 | 0.281782 | 0.316035 | 15 |
| 0.206596 | −0.404594 | −0.206553 | 0.206553 | 0.309491 | 0.341444 | 20 |
| 0.206250 | −0.402871 | −0.220995 | 0.220995 | 0.308917 | 0.340748 | 16 |
| 0.179721 | −0.412072 | −0.160657 | 0.160657 | 0.269242 | 0.304072 | 20 |
| ... | ... | ... | ... | ... | ... | ... |
| −0.004249 | −0.417536 | −0.014609 | 0.014609 | −0.006368 | −0.008525 | 18 |
| −0.006838 | −0.417692 | −0.013866 | 0.013866 | −0.010247 | −0.013755 | 11 |
| 0.059614 | −0.410222 | −0.099442 | 0.099442 | 0.089317 | 0.112032 | 14 |
| −0.013908 | −0.417783 | −0.005324 | 0.005324 | −0.020841 | −0.028154 | 20 |
| 0.191313 | −0.402399 | −0.205605 | 0.205605 | 0.286604 | 0.320355 | 9 |

### *2.4. AgriTransformer: Model Description*

AgriTransformer is a deep learning architecture designed to improve crop yield estimation by leveraging cross-modal attention [30] to integrate two distinct data sources: tabular agricultural features (e.g., soil, weather, and management) and vegetation indices such as NDVI. The model is inspired by transformer-based attention mechanisms,

which have demonstrated strong performance in capturing dependencies across complex, structured inputs.

Our core hypothesis is that joint modeling of these heterogeneous modalities can better capture the plant–environment interaction dynamics that influence yield. Unlike previous methods that treat these modalities independently or combine them through simple concatenation, AgriTransformer employs co-attention blocks to learn relationships between features dynamically.

#### 2.4.1. Input Modalities

- **Tabular data**: Structured features such as crop type, irrigation, and management practices.
- **Vegetation indices (VIs)**: Remotely sensed indices of vegetation (e.g., NDVI and EVI) derived from multispectral satellite imagery during the growing season.

#### 2.4.2. Architecture

The AgriTransformer consists of three main components:

(a) ***Embedding layers***

- The tabular features are passed through a dense layer to obtain a fixed-dimensional embedding.
- The vegetation indices were obtained in a pre-processing stage by the VI formulas described in Table 1.

These branches are processed separately at the first stages of the model.

(b) ***Cross-modal co-attention***

- The embeddings from both modalities are input into a co-attention module, adapted from the concept of cross-attention in vision-language models.
- The dot product calculates the similarities between the projections of each modality [31].
- Normalization with the softmax function is used, and then the model applies weights with the multiplication operation. This multiplication operation is crucial to determine the relevance of one modality with respect to the other.
- This mechanism allows the model to attend to relevant VI patterns conditioned on the tabular context, and vice versa.
- It enables learning interactions such as "how irrigation affects the relationship between NDVI values and final yield."

(c) ***Fusion and prediction***

- The attended representations are concatenated and passed through a feed-forward network (FFN).
- The model outputs a scalar value representing the predicted crop yield.

Figure 4 shows the AgriTransformer architecture, while Figure 5 shows the variants of the AgriTransformer model.

In Figure 5, the 3 variants of the AgriTransformer model are shown. The first implementation uses vegetation indices attention, in which the model calculates the attention of the tabular data over the vegetation indices data. In the second implementation, the model uses tabular data attention, which similarly obtains the attention of the vegetation indices data over tabular data. Finally, in the third implementation, the model uses co-attention, which combines both vegetation indices attention and tabular data attention.

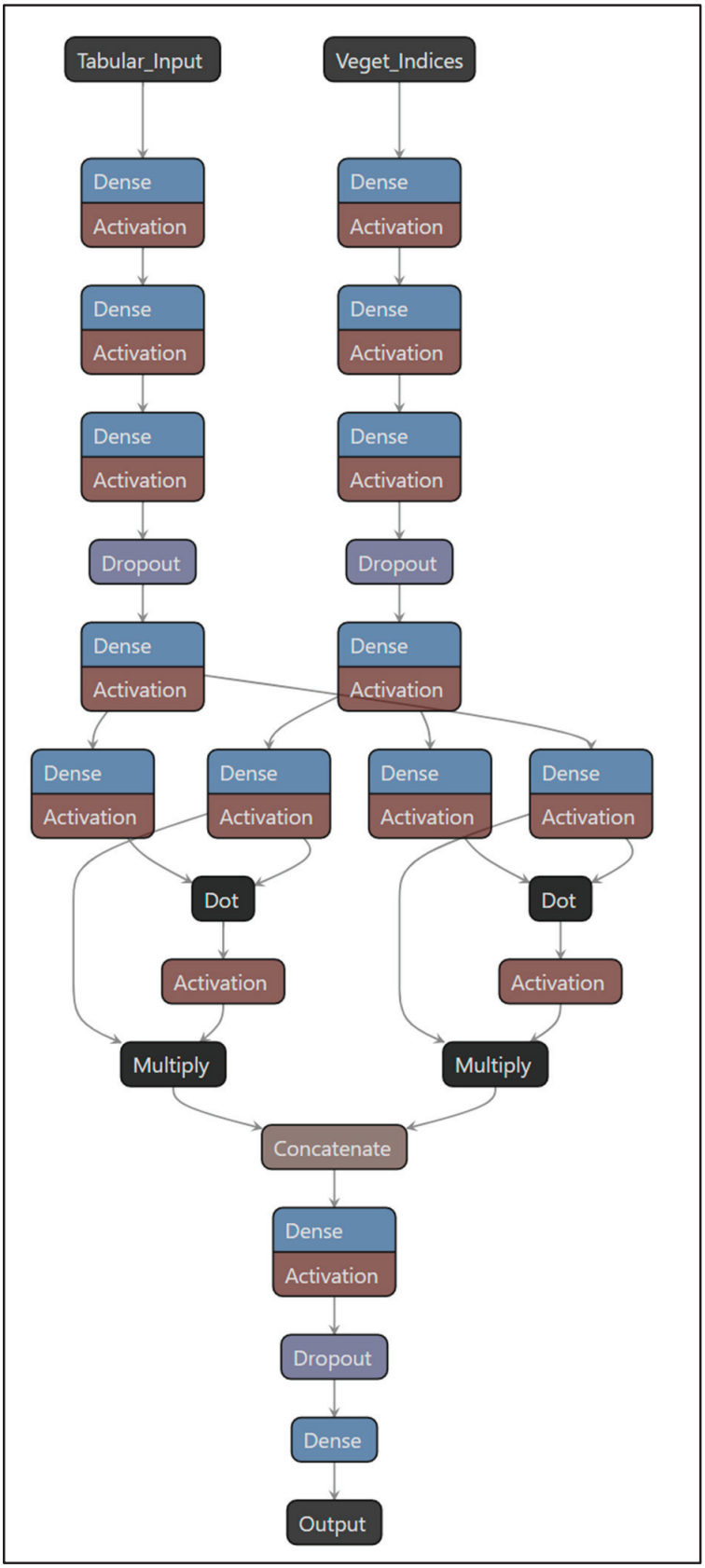



**Figure 4.** AgriTransformer model architecture.

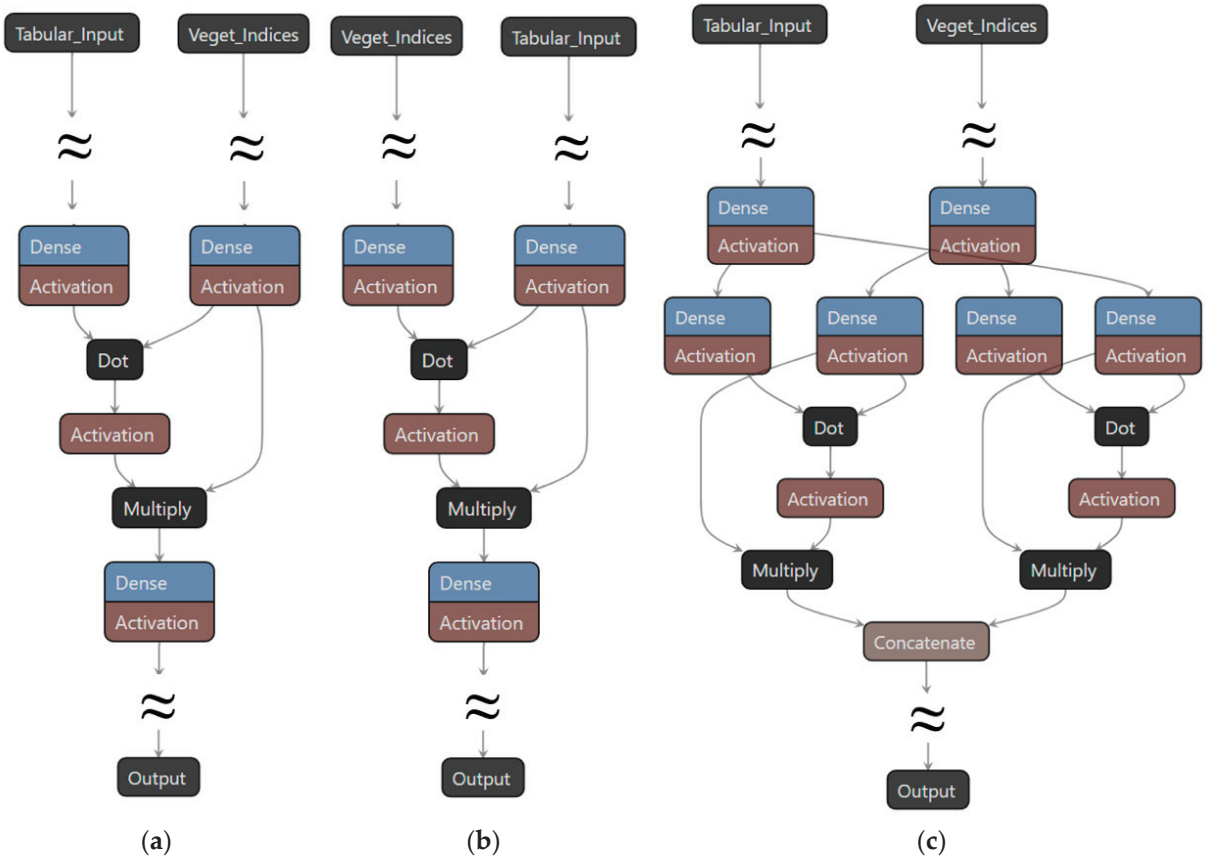



**Figure 5.** Variants of the AgriTransformer model. (**a**) Vegetation indices attention. (**b**) Tabular data attention. (**c**) Co-attention.

#### 2.4.3. Training Setup

The AgriTransformer model was trained and fine-tuned using the values shown in Table 5.

**Table 5.** Parameters of the AgriTransformer.

| Aspect | Value |
|---|---|
| Optimizer | Adam |
| Initial learning rate | 0.001 |
| Search technique | Random search |
| Batch sizes tested | 32, 16, 12 |
| Epoch numbers tested | 20, 50, 70 |
| Dropout rate | 0.10, 0.20, 0.25 |
| Hidden layers | 3, 4 |
| Random seed configuration | 42 |
| Dataset split ratio | 90% train, 10% test |
| Cross-validation | 10-fold cross-validation to ensure robustness |

We used the Google Colaboratory platform with the Python programming language and the TensorFlow and Keras frameworks for the experiment part. Moreover, we utilized Google Earth Engine Python to obtain the multispectral satellite images of the farms.

In addition, to compare the performance of the AgriTransformer model, we employed both linear regression models and deep learning models. Table 6 shows the parameters of the AgriTransformer and other deep learning models.

**Table 6.** Parameters of the AgriTransformer and other deep learning models.

| Deep Learning Model | Hidden Layers | Hidden Nodes | Activation Function | Loss Function |
|---|---|---|---|---|
| Dense neural networks | 3 | 128, 64, 32 | Relu | MSE |
| Convolutional neural networks (1D) | 3 | 64, 128, 64, kernel_size = 3, pool_size = 2 | Relu | MSE |
| AgriTransformer | 4 for each branch + 1 after fusion | 128, 64, 32, 16, 128 | Relu | MSE |

#### 2.4.4. Advantages

AgriTransformer enables cross-modal interaction by learning how features across different modalities influence one another, unlike models that process each modality separately. It also enhances interpretability, as attention mechanisms provide insight into which modality holds greater importance. Additionally, its modular design ensures transferability, allowing it to adapt across various crop types and geographical regions.

### *2.5. Evaluation Metrics of the Model*

The AgriTransformer and the baseline models were evaluated on a real-world dataset comprising both tabular agro-environmental features and vegetation indices for crop fields in Telangana, India. We used two standard regression metrics to measure the performance of the estimation models:

- **Mean Squared Error (MSE)**: Penalizes larger errors.

$$MSE = \frac{1}{n} * \sum_{i=0}^{n} (y_i - \hat{y}_i)^2 \tag{8}$$

- **Coefficient of Determination ($R^2$)**: Indicates the proportion of variance explained by the model.

$$\begin{gathered} r^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \overline{y})^2} \\ \text{where :} \\ \overline{y} = mean\ of\ actual\ values \end{gathered} \tag{9}$$

Each model was evaluated using 10-fold cross-validation to ensure generalizability. We report the mean and standard deviation of MSE and $R^2$ across folds. Additionally, we conducted paired *t*-tests to assess the statistical significance of differences between AgriTransformer and the best-performing baseline.

In the next section, we present the results of our experiments, which were conducted to evaluate the performance of the AgriTransformer model.

## 3. Results

### *3.1. Quantitative Results*

To evaluate the performance of the AgriTransformer model, we experimented and compared it with the linear regression baseline algorithm and deep learning models. The experiments with linear regression had three versions of the utilized data (tabular data + vegetation indices, tabular data only, and vegetation indices only), which helped compare the impact of a single modality of data against a multimodality of data. The two deep learning models, deep neural networks and convolutional neural networks, were evaluated using tabular and VI data. For its part, we evaluated the AgriTransformer model with three implementation variants that were described previously: co-attention (Tabular + VI), VI-attention (Tabular + VI), and Tabular-attention (Tabular + VI). Table 7 presents the results of the experimental stage.

**Table 7.** Vegetation indices fields of the dataset and the ground truth (ExpYield).

| Model | MSE (Media) | MSE (Std) | $R^2$ (Media) | $R^2$ (Std) |
|---|---|---|---|---|
| Linear reg. (Tabular only) | 9.399 | 0.406 | 0.703 | 0.024 |
| Linear reg. (VI only) | 31.516 | 2.008 | 0.007 | 0.009 |
| Linear reg. (Tabular + VI) | 9.364 | 0.402 | 0.704 | 0.024 |
| Dense neural networks (Tabular + VI) | 3.666 | 0.219 | 0.884 | 0.012 |
| Convolutional neural networks (Tabular + VI) | 4.726 | 1.289 | 0.849 | 0.054 |
| AgriTransformer with Tabular-attention (Tabular + VI) | 5.037 | 0.481 | 0.841 | 0.021 |
| AgriTransformer with VI-attention (Tabular + VI) | 31.832 | 1.833 | −0.002 | 0.003 |
| **AgriTransformer with co-attention (Tabular + VI)** | **2.598** | **0.816** | **0.919** | **0.022** |

The linear regression models exhibited limitations in predictive accuracy, particularly when using only vegetation indices (VI-only), having a high error of MSE = 31.516 and low explanatory power $R^2$ = 0.007. In contrast, combining tabular data with vegetation indices (Tabular + VI) improved the accuracy, with MSE = 9.364 and $R^2$ = 0.704, but it still had lower performance than more advanced models.

The application of deep learning led to a notable reduction in prediction error. The dense neural network achieved an MSE of 3.666 and an $R^2$ of 0.884, indicating a significantly better fit. The 1D convolutional neural network also performed well, though with greater variability in error: MSE = 4.726 and $R^2$ = 0.849.

The AgriTransformer model showed varying performance depending on the attention mechanism used. The Tabular-attention version maintained solid performance with MSE = 5.037 and $R^2$ = 0.841. The VI-attention version performed poorly, with a negative $R^2$ = −0.002, suggesting an inability to model the relationship between variables properly.

The co-attention version was the most effective, achieving the lowest error of MSE = 2.598 and the highest fit of $R^2$ = 0.919, showing its superior ability to integrate and process information efficiently.

The results indicate that deep learning models, particularly the AgriTransformer with co-attention, significantly outperform traditional methods in terms of accuracy and model fit. Moreover, the integration of tabular data and vegetation indices enhances performance in advanced neural networks.

Furthermore, Figure 6 shows graphically the yield prediction of the AgriTransformer model when it is applied to the images of the dataset. It is important to highlight that these predictions are more confident than vegetation indices alone since the forecast is based on richer and diverse information.

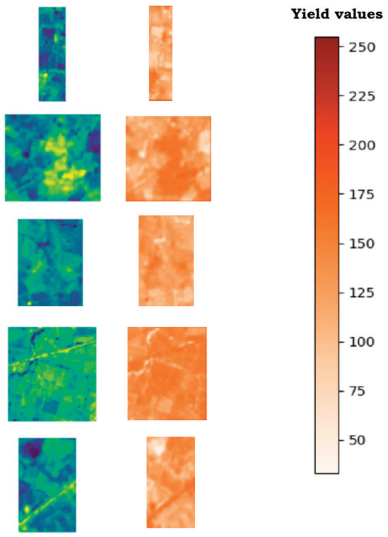



**Figure 6.** Yield predictions of the AgriTransformer model.

### *3.2. Statistical Significance*

To statistically validate the improvement provided by our AgriTransformer model (co-attention) over the best-performing model (dense neural networks), we performed paired hypothesis tests on the cross-validation results (k = 10), using both mean squared error (MSE) and coefficient of determination ($R^2$) as performance metrics.

- **MSE (Mean Squared Error)**
- **Paired *t*-test**: $p = 0.0023 < 0.01$
- **Wilcoxon signed-rank test**: $p = 0.0059 < 0.01$
- **$R^2$ (Coefficient of Determination)**
- **Paired *t*-test**: $p = 0.0032 < 0.01$
- **Wilcoxon signed-rank test**: $p = 0.0032 < 0.01$

### *3.3. Interpretation*

Both the parametric and non-parametric tests indicate highly significant differences ($p < 0.01$) for both metrics. This confirms that the co-attention variant of the AgroTransformer model, which leverages attention mechanisms over tabular data and vegetation indices, significantly outperforms the other best-performing model with dense neural networks, both in prediction accuracy (lower MSE) and explanatory power (higher $R^2$).

### *3.4. Data Interpretability*

To understand the importance of each feature of the dataset on the prediction model, we implemented the SHAP (SHapley Additive exPlanations) method, which is based on game theory and helps explain the predictions of machine learning models. It assigns a fair contribution to each feature in a model's prediction using Shapley values. Figure 7 shows the application of SHAP, and we can see the contribution of each feature (tabular features extend from 0 to 15, while VI features extend from 16 to 21). The image reveals that despite the variance of the features, both modalities add moderate to significant value to the prediction model.

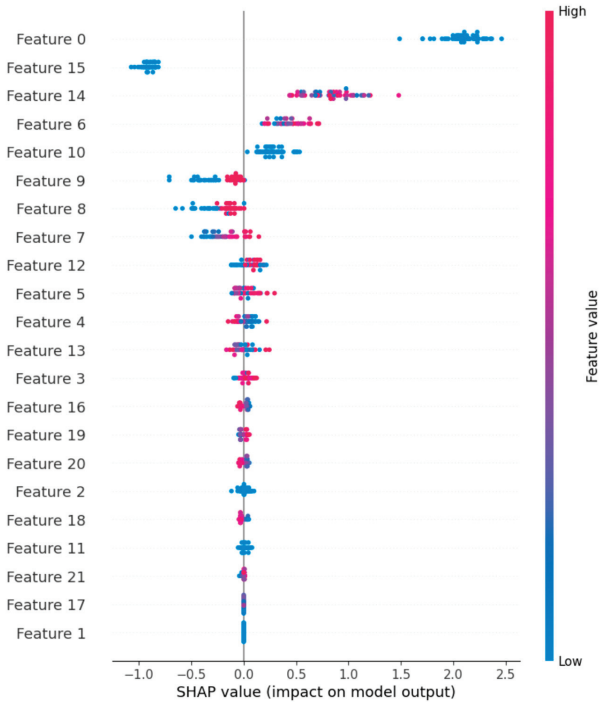



**Figure 7.** Importance of each feature in the predicting model using the SHAP method (Tabular features: 0–15, VI features: 16:21).

### *3.5. Deployment Validation*

The AgriTransformer model was evaluated with a different dataset, which was obtained by the ESPOL (Escuela Superior Politécnica del Litoral) University, located in Guayaquil, Ecuador. The experiments were conducted using 10-fold cross-validation to assess their robustness and generalization capability prior to deployment.

The average MSE across folds was low on average (0.0383), which indicates that the model's prediction error is moderate. However, the $R^2$ scores were negative across all folds, with an average of -0.5973, indicating that the model consistently underperformed compared to a baseline that simply predicts the mean target value.

Despite an acceptable average MSE, the consistently negative $R^2$ can be explained by the fact that this validation dataset did not have all the fields in which the AgriTransformer was initially trained. Moreover, it is worth noting that the farming conditions were different. Further steps include collecting more diverse samples across seasons, regions, and crop types to capture variability better. It is also important to include relevant agronomic, environmental, or remote sensing features that may improve predictive capability.

## 4. Discussion

The results demonstrate that the AgriTransformer model significantly improves crop yield estimation by effectively integrating multimodal data sources and cross-modal attention. Compared to traditional linear regression and other deep learning models, AgriTransformer exhibits lower prediction errors and higher explanatory power.

One of the key insights from this study is the role of attention mechanisms in multimodal data fusion. The AgriTransformer variant with co-attention achieved the best performance, suggesting that combining tabular data and vegetation indices provides a more reliable foundation for yield estimation than using single-modal data.

The superior performance of AgriTransformer with co-attention further supports the hypothesis that meaningful interactions between tabular data and vegetation indices enhance predictive accuracy. By allowing features from different modalities to influence one another dynamically, the co-attention mechanism enabled a more comprehensive representation of agricultural conditions. This approach contrasts with traditional models that process each modality separately, often failing to capture complex dependencies between environmental factors and crop health.

Moreover, our model demonstrated superior performance compared to previous approaches, including the work of [5], who utilized satellite imagery to predict crop production and achieved an $R^2$ value of 0.615. In contrast, AgriTransformer, particularly the variant with co-attention, achieved an $R^2$ of 0.919, representing a substantial improvement in predictive accuracy. This difference highlights the advantages of integrating multimodal data sources rather than relying solely on satellite imagery.

The limitations of this paper include that the variability in MSE and $R^2$ across different AgriTransformer variants suggests that attention mechanisms require careful tuning to optimize the performance of the model. Additionally, while AgriTransformer has proved strong generalization power across different crop types and geographical regions, it is necessary to do further validation on larger datasets to confirm its scalability. Future research should explore adaptive attention strategies that dynamically adjust the weighting of different modalities based on environmental conditions and specific characteristics of the crops.

## 5. Conclusions

In this work, we presented the AgriTransformer model, a novel deep learning architecture for crop yield estimation that utilizes cross-modal attention to integrate tabular

agro-environmental features with vegetation indices from satellite multispectral images. Our approach is motivated by the observation that yield is governed by complex, non-linear interactions between environmental conditions, management practices, and plant physiological responses, patterns that are difficult to model using unimodal approaches.

The ability of AgriTransformer to capture cross-modal interactions allows for a more comprehensive representation of agricultural conditions, improving generalization across different crop types and geographical regions. The modular design of the model ensures adaptability, making it a scalable solution for precision agriculture applications. Furthermore, the results emphasize the effectiveness of attention mechanisms in prioritizing relevant features, demonstrating that transformer-based architectures can outperform conventional machine learning approaches in agricultural modeling.

Overall, AgriTransformer offers a promising foundation for data-driven agricultural decision-making and contributes to a broader effort to develop robust, explainable, and generalizable AI models for sustainable food production.

**Author Contributions:** Conceptualization, L.J.G. and M.R.; Formal analysis, L.J.G., M.F.C. and M.S.V.-L.; Investigation, L.J.G., M.R. and M.F.C.; Methodology, S.J. and M.R.; Supervision, M.R., M.F.C. and M.S.V.-L.; Writing—original draft, L.J.G.; Writing—review editing, L.J.G., M.S.V.-L. and S.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** The author(s) declare that no financial support was received for the research and/or publication of this article.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available at https://github.com/lrobertojacomeg/multimodal (accessed on 6 June 2025).

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

## References

1. Ahn, D.; Kim, S.; Hong, H.; Ko, B. Star-transformer: A spatio-temporal cross attention transformer for human action recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision 2023, Waikoloa, HI, USA, 2–7 January 2023; pp. 3330–3339.
2. Ajith, S.; Vijayakumar, S.; Elakkiya, N. Yield prediction, pest and disease diagnosis, soil fertility mapping, precision irrigation scheduling, and food quality assessment using machine learning and deep learning algorithms. *Discov. Food* **2025**, *5*, 63.
3. Hobbs, P. Conservation agriculture: What is it and why is it important for future sustainable food production? *J. Agric. Sci.* **2007**, *145*, 127. [CrossRef]
4. Marshall, M.; Belgiu, M.; Boschetti, M.; Pepe, M.; Stein, A.; Nelson, A. Field-level crop yield estimation with PRISMA and Sentinel-2. *ISPRS J. Photogramm. Remote Sens.* **2022**, *187*, 191–210. [CrossRef]
5. Roznik, M.; Boyd, M.; Porth, L. Improving crop yield estimation by applying higher resolution satellite NDVI imagery and high-resolution cropland masks. *Remote Sens. Appl. Soc. Environ.* **2022**, *25*, 100693. [CrossRef]
6. Nikhil, U.; Pandiyan, A.; Raja, S.; Stamenkovic, Z. Machine learning-based crop yield prediction in south india: Performance analysis of various models. *Computers* **2024**, *13*, 137. [CrossRef]
7. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [CrossRef]
8. Oikonomidis, A.; Catal, C.; Kassahun, A. Deep learning for crop yield prediction: A systematic literature review. *N. Z. J. Crop Hortic. Sci.* **2023**, *51*, 1–26. [CrossRef]
9. Mingyong, L.; Yewen, L.; Mingyuan, G.; Longfei, M. CLIP-based fusion-modal reconstructing hashing for large-scale unsupervised cross-modal retrieval. *Int. J. Multimed. Inf. Retr.* **2023**, *12*, 2. [CrossRef]
10. Bhattacharyya, B.; Biswas, R.; Sujatha, K.; Chiphang, D. Linear regression model to study the effects of weather variables on crop yield in Manipur state. *Int. J. Agric. Stat. Sci.* **2021**, *17*, 317–320.
11. Dhillon, M.; Dahms, T.; Kuebert-Flock, C.; Rummler, T.; Arnault, J.; Steffan-Dewenter, I.; Ullmann, T. Integrating random forest and crop modeling improves the crop yield prediction of winter wheat and oil seed rape. *Front. Remote Sens.* **2023**, *3*, 1010978. [CrossRef]

12. Kok, Z.; Shariff, A.; Alfatni, M.; Khairunniza-Bejo, S. Support vector machine in precision agriculture: A review. *Comput. Electron. Agric.* **2021**, *191*, 106546. [CrossRef]
13. Mahesh, P.; Soundrapandiyan, R. Yield prediction for crops by gradient-based algorithms. *PLoS ONE* **2024**, *19*, e0291928. [CrossRef] [PubMed]
14. Anderson, K. Detecting Environmental Stress in Agriculture Using Satellite Imagery and Spectral Indices. Ph.D. Thesis, Obafemi Awolowo University: Ile-Ife, Nigeria, 2024.
15. Peng, M.; Liu, Y.; Khan, A.; Ahmed, B.; Sarker, S.; Ghadi, Y.; Ali, Y. Crop monitoring using remote sensing land use and land change data: Comparative analysis of deep learning methods using pre-trained CNN models. *Big Data Res.* **2024**, *36*, 100448. [CrossRef]
16. Petit, O.; Thome, N.; Rambour, C.; Themyr, L.; Collins, T.; Soler, L. U-net transformer: Self and cross attention for medical image segmentation. In *Machine Learning in Medical Imaging, Proceedings of the 12th International Workshop, MLMI 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, 27 September 2021, Proceedings 12*; Springer International Publishing: Cham, Switzerland, 2021; pp. 267–276.
17. Rahimi, E.; Jung, C. The efficiency of long short-term memory (LSTM) in phenology-based crop classification. *Korean J. Remote Sens.* **2024**, *40*, 57–69.
18. Dieten, J. Attention Mechanisms in Natural Language Processing. Bachelor's Thesis, University of Twente, Enschede, The Netherlands, 2024.
19. Guo, M.; Xu, T.; Liu, J.; Liu, Z.; Jiang, P.; Mu, T.; Hu, S. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [CrossRef]
20. Lin, H.; Cheng, X.; Wu, X.; Shen, D. Cat: Cross attention in vision transformer. In Proceedings of the 2022 IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, 18–22 July 2022; pp. 1–6.
21. Rashid, M.; Bari, B.; Yusup, Y.; Kamaruddin, M.; Khan, N. A comprehensive review of crop yield prediction using machine learning approaches with special emphasis on palm oil yield prediction. *IEEE Access* **2021**, *9*, 63406–63439. [CrossRef]
22. Kaggle. Telangana Crop Health Challenge. Kaggle. 2024. Available online: https://www.kaggle.com/datasets/adhittio/z-1-telangana-crop-health-challenge (accessed on 1 December 2024).
23. Pettorelli, N.; Vik, J.; Mysterud, A.; Gaillard, J.; Tucker, C.; Stenseth, N. Using the satellite-derived NDVI to assess ecological responses to environmental change. *Trends Ecol. Evol.* **2005**, *20*, 503–510. [CrossRef]
24. Gurung, R.; Breidt, F.; Dutin, A.; Ogle, S. Predicting Enhanced Vegetation Index (EVI) curves for ecosystem modeling applications. *Remote Sens. Environ.* **2009**, *113*, 2186–2193. [CrossRef]
25. Ashok, A.; Rani, H.; Jayakumar, K. Monitoring of dynamic wetland changes using NDVI and NDWI based landsat imagery. *Remote Sens. Appl. Soc. Environ.* **2021**, *23*, 100547. [CrossRef]
26. Basso, M.; Stocchero, D.; Ventura, R.; Vian, A.; Bredemeier, C.; Konzen, A.; Pignaton de Freitas, E. Proposal for an embedded system architecture using a GNDVI algorithm to support UAV-based agrochemical spraying. *Sensors* **2019**, *19*, 5397. [CrossRef]
27. Chen, Z.; Liu, H.; Zhang, L.; Liao, X. Multi-dimensional attention with similarity constraint for weakly-supervised temporal action localization. *IEEE Trans. Multimed.* **2022**, *25*, 4349–4360. [CrossRef]
28. Ren, H.; Zhou, G.; Zhang, F. Using negative soil adjustment factor in soil-adjusted vegetation index (SAVI) for aboveground living biomass estimation in arid grasslands. *Remote Sens. Environ.* **2018**, *209*, 439–445. [CrossRef]
29. Novando, G.; Arif, D. Comparison of soil adjusted vegetation index (SAVI) and modified soil adjusted vegetation index (MSAVI) methods to view vegetation density in padang city using landsat 8 image. *Int. Remote Sens. Appl. J.* **2021**, *2*, 31–36. [CrossRef]
30. Wen, Z.; Lin, W.; Wang, T.; Xu, G. Distract your attention: Multi-head cross attention network for facial expression recognition. *Biomimetics* **2023**, *8*, 199. [CrossRef]
31. Soydaner, D. Attention mechanism in neural networks: Where it comes and where it goes. *Neural Comput. Appl.* **2022**, *34*, 13371–13385. [CrossRef]

*electronics*

MDPI

*Article*

# Enhancing Basketball Team Strategies Through Predictive Analytics of Player Performance

**Roshan Chandru [1], Abhishek Kaushik [1,2,*] and Pranay Jaiswal [1,2,3]**

[1] Department of Computing Science and Mathematics, Dundalk Institute of Technology, A91 K584 Dundalk, Ireland
[2] Regulated Software Research Center, Dundalk Institute of Technology, A91 K584 Dundalk, Ireland
[3] Lero, the Research Ireland Centre for Software, V94 NYD3 Limerick, Ireland
* Correspondence: abhishek.kaushik@dkit.ie

**Abstract:** This study explores the application of predictive analytics in evaluating player performance in the National Basketball Association (NBA), focusing on rebounds per game (REB), an essential component for better performance and results in basketball. The research employs a comparative analysis of machine learning (ML) models by leveraging a detailed NBA dataset. A key novelty lies in integrating advanced hyperparameter tuning and feature selection, enabling these models to capture complex relationships within the dataset. The Gradient Boosting Regressor demonstrated superior predictive performance, achieving an $R^2$ score of 0.8749 after tuning, with Linear Regression following closely at 0.8668. This study also highlights the importance of model interpretability and scalability, emphasizing the balance between predictive accuracy and usability for real-world decision-making. By offering actionable insights for optimizing player strategies and team performance, this research contributes to the growing body of knowledge in data-driven sports analytics and paves the way for more advanced applications in professional basketball management.

**Keywords:** predictive analytics; player performance; machine learning; best linear regression; best gradient boosting regressor

## 1. Introduction

In professional sports, accurately predicting player performance is essential in effective decision-making and maximizing game outcomes. The experts can make better team and game plan choices by applying the mathematical theory of evidence, which allows them to express and integrate subjective beliefs to enhance decisions precisely [1]. Player performance analysis is transformed by shifting to complex statistical models, such as machine learning (ML) algorithms, which offer a dynamic and accurate framework for athlete assessment. The research on predictive saccades of athletes has revealed information regarding the ability of players to adapt to game changes, which can impact overall performance significantly [2]. The significance of their actions in determining outcomes, such as committee decisions, which are collectively made by a panel or team, is often influenced by the probabilistic prediction value of individual contributions. This gives an idea of the importance of personal contributions to team performance and strategy [3].

The traditional fitness–fatigue models are mathematical frameworks used in sports science to predict and explain how training impacts performance over time and are improved by using ML [4]. This is done by integrating physiological data and multivariate algorithms, which advance performance predictions and provide a deeper understanding of athlete potential [5].

Professional sports performance and strategic planning have greatly improved by incorporating ML techniques into sports analytics, allowing for more accurate forecasts and data-driven decision-making. ML is increasingly used to improve the team's training to prevent injuries and monitor performance in sports analytics [6,7].

ML can conduct video analysis for classifying strength and conditioning exercises, which helps achieve high accuracy with less engineering than deep learning. By achieving remarkable accuracy in predicting ski jump lengths, ML models have shown that there is a new opportunity for enhanced sports broadcasting and real-time performance analysis [8]. The heterogeneity in performance curves is formed by variability in basketball player performance measurements. This causes prediction challenges, requiring sophisticated methods to balance curve smoothness with covariate integration [9]. For the modeling of performance indicators' joint distribution, Bayesian networks are crucial. This provides deeper insights into factors influencing game outcomes and performance dynamics [10]. There is a significant evolution from traditional metrics by evaluating defensive performance through technologies such as optical player-tracking systems, which offer a more comprehensive assessment of player actions [11,12].

The more holistic approach to player evaluation is achieved by understanding the physiological factors impacting performance: body composition and nutrient intake [13]. They provide detailed insights into players' abilities and playing styles, informing coaching; player utilization strategies can be found by utilizing predictive models based on NBA tracking data [14,15]. The tailoring of game plans to player profiles is aided by discovering latent heterogeneity in shot selection through Bayesian nonparametric clustering [16]. Beyond sports, cognitive constraint modeling enables the optimization of digital and physical environments for improved human performance. This method of player evaluation and prediction highlights the increasing complexity and influence of sports analytics in improving professional sports operations and strategy [17].

In professional basketball, evaluating player performance is pivotal for optimizing team strategies and decision-making. Predictive analytics has emerged as a transformative tool in sports, enabling data-driven insights into key performance metrics such as rebounds per game (REB). One of the most crucial aspects of the basketball game is rebounding. A team gains possession of the basketball when a rebound is made, and the rebound is given to the player who catches the ball later on. The more rebounds there are, the higher the ball's possession. And each possession benefits the team's defense and offense, ultimately leading to a victory.

Despite its potential, traditional parametric models like Linear Regression are limited by their inability to account for nonlinear relationships and complex interactions inherent in player performance data [1]. On the other hand, advanced ML methods, such as Gradient Boosting Regressor, offer greater flexibility in capturing these dynamics while maintaining high predictive accuracy [8]. This research focuses on addressing the trade-offs between model simplicity and accuracy by exploring the effectiveness of parametric and nonparametric models in predicting REB. By leveraging techniques such as hyperparameter tuning and feature engineering, this study seeks to improve predictive accuracy and identify the factors that significantly influence performance outcomes [8]. Furthermore, it evaluates how these models can balance interpretability and scalability, ensuring their applicability in real-world decision-making scenarios [3,8]. Through robust ML methodologies, this study aims to contribute actionable insights to the growing field of sports analytics. Previous studies have demonstrated the potential of integrating ML techniques into sports analytics to optimize player utilization and enhance decision-making strategies [1,8].

The formulated hypothesis and related research questions of the study are as follows:

**Hypothesis 1.** *Advanced feature engineering, data preprocessing, and hyperparameter tuning significantly increase the ML models' ability to predict future average rebounds per game (REB) for NBA players.*

The research questions to investigate the hypothesis are:

1. What is the prediction accuracy level of parametric and non-parametric ML models when predicting REB for NBA players?
2. This question aims to evaluate and compare how effectively different types of ML models—both parametric and non-parametric—can predict average REB. By analyzing model performance using evaluation metrics such as $R^2$, MSE, and RMSE, we identify which models best capture the patterns in basketball performance data.
3. What type of model, parametric or nonparametric, benefits most from optimization by tuning hyperparameters to predict REB for NBA players?
4. This question investigates the impact of hyperparameter tuning on model performance. Specifically, it examines whether parametric or non-parametric models show greater improvements in prediction accuracy when optimized, providing insights into the models' adaptability and potential for enhancement in real-world applications.

The major contributions of this paper are as follows:

1. We conducted a comparative analysis of various parametric and non-parametric machine learning models to predict NBA player performance, with a focus on rebounds (REB).
2. We applied hyperparameter tuning techniques to optimize model performance, demonstrating that proper tuning significantly improves prediction accuracy in sports analytics.

The rest of the paper is structured as follows: Section 2 discusses the relevant literature in the analysis of predictive modeling techniques and optimization of predictive accuracy. Section 3 discusses the study's methodology, including dataset details, data preparation, exploratory data analysis, feature selection, and ML models. In Section 4, the results of the model training and evaluation are presented. Section 5 discusses our research hypothesis and research questions. Section 6 concludes the study with future guidance.

## 2. Literature Review

### *2.1. Analysis of Predictive Modeling Techniques: Applications and Evaluations in NBA Player Performance*

In recent years, predictive modeling has become a key component of sports analytics, offering data-driven insights into player performance. Accurately predicting metrics requires selecting appropriate models that can handle the complexity and variability of basketball data. This section introduces and evaluates a range of ML models commonly used in this domain, assessing their suitability and effectiveness for predicting NBA player performance.

Linear regression is very straightforward for modeling relationships between variables and is practical, making it suitable for predicting basketball player performance. Its simplicity allows even those without statistical training to grasp and use it efficiently. Because of the roles of explanatory variables and key concepts like r-squared, the statistical significance of regression coefficients is straightforward and intuitive. Thus, linear regression is a standard tool in educational settings for introducing statistical principles using basketball statistics [18]. Due to the low computational demands and efficiency, linear regression is highly valued across various sectors and finds applications beyond sports,

such as predicting food content. This versatility shows that it can be used in multiple fields, which provides effectiveness in predictive modeling [19].

Even though linear regression provides a solid foundation, it is found that non-linear models often yield higher accuracy in more complex systems. For example, while capturing the intricate dynamics of non-linear relationships between anthropometric predictors and basketball performance, non-linear regression is more precise [20]. The complexity of player performance assessment is illustrated when integrating personality traits into performance prediction models. It was found that traits such as agreeableness and conscientiousness are significant predictors of performance in the studies using automated language-based analyses. This adds another layer of complexity to the analysis [21].

The RF model is quite effective for analyzing complex datasets, such as basketball analytics, which consists of player statistics, game events, and biometric data. This model is very effective in selecting essential features. This is achieved by breaking the data into smaller groups using RF-based multi-round screening (RFMS) [22]. This can be further improved using techniques like Mutual forest impact (MFI) and Mutual impurity reduction (MIR). These techniques help to understand how different features interact, which in turn helps improve model performance [23]. The RF model can also handle various data types, such as categorical, time-series, and numerical. Methods such as random similarity forests ensure the model works well with different kinds of data [24]. Techniques such as correlation-based feature selection (CFS) increase the model's accuracy and efficiency using extensive data.

Gradient Boosting combines multiple weak learners, typically decision trees, to produce a strong predictive model. Its robustness comes from its iterative methodology, which learns from the residuals of previous models to minimize prediction errors. It can easily handle complex, non-linear relationships in sports analytics, improving accuracy with each iteration. Large datasets benefit significantly from the significant advancement in this field known as the gradient-boosted binary histogram ensemble (GBBHE), which increases convergence rates and computational efficiency [25].

In sports analytics, the KNN algorithm is very effective for making accurate predictions by utilizing the inherent characteristics of local data points. The performance metrics in sports data can vary significantly between contexts and conditions, and the algorithm's high adaptability to changes is essential [26]. It has been found that KNN can improve predictive performance by using effective forward selection of predictor variables. This is useful in sports analytics, as it can result in more accurate predictions by choosing the most pertinent features, such as player statistics and game conditions [27]. Additionally, it can be found that KNN integration with other techniques has demonstrated high accuracy rates. For example, protein sequence coding in biological contexts has improved accuracy, indicating that sports analytics could benefit from applying similar hybrid approaches to improve prediction models [28].

Many studies have demonstrated the effectiveness of the MLP and other deep learning models in identifying patterns in basketball player performance. The studies on performance statistics of the MLP model, specifically its neural network architecture with a 21-7-3 design, have provided high accuracy in NBA player classification. The complex data is easily handled, identifying patterns and classifying clusters by this neural network architecture [29]. Furthermore, the MP-LSTM algorithm has achieved 94% recognition accuracy.

After reviewing the strengths and limitations of various predictive modeling techniques, we recognized that different models handle complexity and data structure in distinct ways. This led us to formulate our first research question. This question aims to

evaluate which modeling approach better captures the variation of player performance, particularly in rebound prediction.

### 2.2. Optimizing Predictive Accuracy: The Critical Role of Hyperparameter Tuning in Sports Analytics

The model's efficacy and precision can be increased by figuring out the best mix of hyperparameters. It has been found that there is a significant advancement in convergence rates and recovery performance by utilizing neural network-based auto-tuners. These are important for precise and timely sports predictions [30]. Evidence suggests that various algorithms perform differently based on their hyperparameters. This includes Feedforward Neural Networks (FFNN), RF, and Extreme XGB. Adjusting these hyperparameters is essential to obtaining the best outcomes [31].

In a study, it has been demonstrated that the model's capacity to identify malware in Internet of Things devices correctly can be significantly impacted by modifying hyperparameters such as learning rate, neighborhood function, and number of neurons in the Self-organizing Maps (SOM) [32], which can also be applied for predictive analysis in sports.

Similarly, obtaining the best results from models such as RF and CNN strongly depends on particular hyperparameters. The hyperparameters of CNNs are learning rate, batch size, and the number of layers. The hyperparameter of RFs is several trees [33]. Overfitting or underfitting can be prevented, and the data can be effectively learned from the model by carefully adjusting hyperparameters such as the learning rate, maximum depth, and number of estimators for Extreme Gradient Boosting (XGB); the number of hidden layers and neurons for Feedforward Neural Networks (FFNN) also need to be adjusted, which in turn improves the accuracy and efficiency of the model [31].

Several hyperparameter optimization techniques can be used to obtain the best hyperparameter settings; for instance, Bayesian Optimization, Genetic Algorithms, and other metaheuristic optimization techniques perform better than manual tuning. These techniques offer reduced computational overhead and increased performance [34,35].

Through this review, we observed that model performance in sports analytics greatly depends on the careful tuning of hyperparameters, which can significantly impact accuracy and prevent issues like overfitting. This realization led us to our second research question. This question aims to explore which modeling approach gains more from hyperparameter tuning when applied to the task of rebound prediction.

## 3. Methodology

This section explains the methodology, covering the dataset used, data preparation, feature engineering, and exploratory data analysis to gain insights and prepare the data for modeling. The step-by-step procedure is shown in Figure 1.

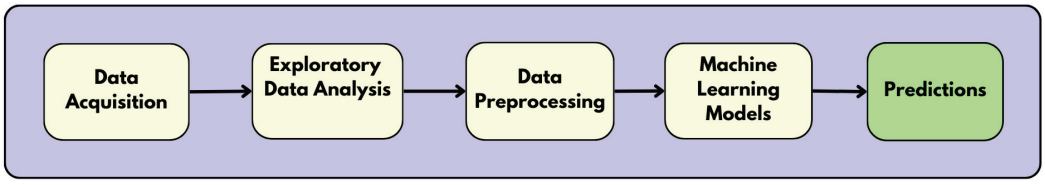



**Figure 1.** Procedure followed during this study.

### 3.1. Dataset

The National Basketball Association (NBA) dataset is an open-access dataset available in Kaggle [36]. The dataset contains 1340 entries and 22 features associated with players' performance, each representing an individual NBA player. The description of each feature

is given in Table 1. The data was collected over five years. The target feature is REB, which is the average rebounds per game.

**Table 1.** Description of NBA Player Statistics.

| Column Name | Column Description |
|---|---|
| name | NBA player's name |
| GP | Total games played |
| MIN | Average minutes played per game |
| PTS | Mean points scored per game |
| FGM | Average field goals successfully made per game |
| FGA | Mean field goal attempts per game |
| FG | Percentage of successful field goals per game |
| 3p_made | Average three-point shots successfully made per game |
| 3PA | Mean attempts for three-point shots per game |
| 3P | Success rate of three-point shots per game |
| FTM | Average number of free throws made per game |
| FTA | Mean free throw attempts per game |
| FT | Free throw success percentage |
| OREB | Mean offensive rebounds per game |
| DREB | Average defensive rebounds per game |
| AST | Mean assists per game |
| STL | Average number of steals per game |
| BLK | Mean blocks recorded per game |
| TOV | Average turnovers per game |
| target_5yrs | 1 if the player's career lasts at least 5 years, otherwise 0 |
| REB | Total average rebounds per game |

### *3.2. Exploratory Data Analysis*

First, we conducted exploratory data analysis on raw data to uncover patterns and relationships in the dataset. Table 2 provides the descriptive statistics for rebounds by type, and Figure 2 presents the distribution of rebounds per game (REB), offering a deeper understanding of the spread of REB, which comprises three key metrics: 'OREB' is the average number of offensive rebounds per game; 'DREB' is the average number of defensive rebounds per game; and 'REB' is the total average number of rebounds per game. The REB distribution is right-skewed, indicating that most players had lower rebound averages, with a few exhibiting exceptionally high values.

**Table 2.** Descriptive Statistics for Rebounds

| | OREB | DREB | REB |
|---|---|---|---|
| count | 1328.00 | 1328.00 | 1328.00 |
| mean | 1.01 | 2.03 | 3.04 |
| std | 0.78 | 1.36 | 2.06 |
| min | 0.00 | 0.20 | 0.30 |
| 25% | 0.40 | 1.00 | 1.50 |
| 50% | 0.80 | 1.70 | 2.50 |
| 75% | 1.40 | 2.60 | 4.00 |
| max | 5.30 | 9.60 | 13.90 |

Figure 3 is a scatter plot that provides details regarding the clear trend where players with more minutes per game tend to record higher REB. This pattern holds for both groups, mainly those who did and did not meet the target performance. It was found that the trend is more pronounced among players who achieved the target, which is

denoted by green dots, which shows that increased playing time often correlates with better rebounding performance.

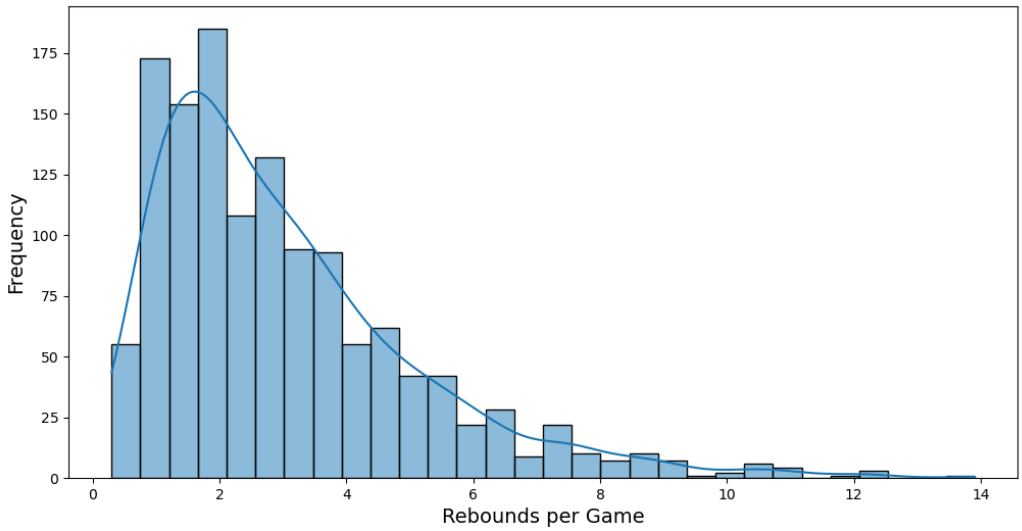



**Figure 2.** Distribution plot of REB.

Figure 4 is a box plot that compares the distribution of REBs between players who have met or have not met the target over five years. The players who met the target, which is denoted by the orange box or 1 on the x-axis, show higher median REB and a wider range of values compared to their counterparts. This gives the importance of rebounding in achieving long-term player success. These plots provide an understanding of the influence of minutes per game and efficiency on rebounding performance, and also give information regarding essential indicators for evaluating basketball players' success and potential.

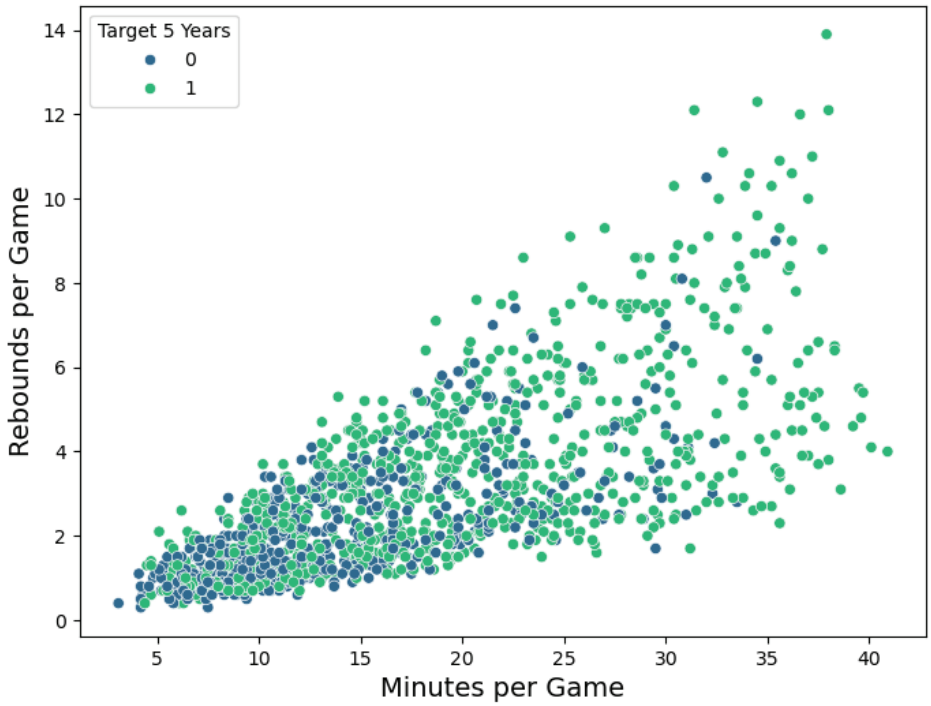



**Figure 3.** REB in relation to minutes per game.

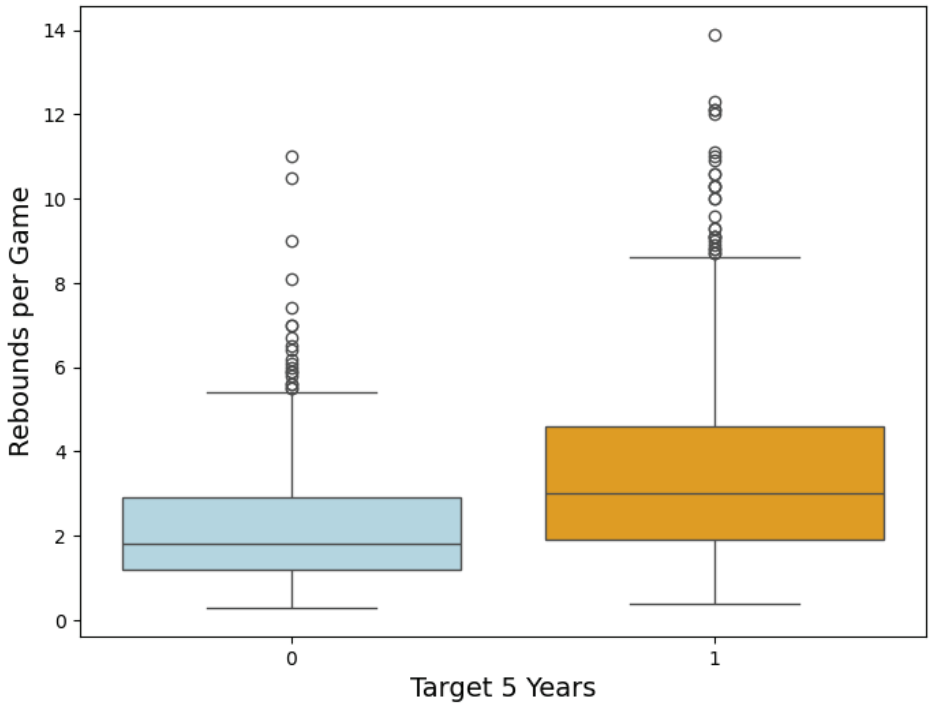



**Figure 4.** REB made by players who have played for 5 Years or not.

### *3.3. Data Preprocessing*

Several data preprocessing techniques have been applied to clean the data. First, we checked for missing values in the dataset, and no missing values were found. Then, we removed the duplicate rows with all the same entries, which resulted in 12 duplicate rows, and removed them. The next step involves checking the outliers. Figure 2 shows that the data is right-skewed, so we have used the Interquartile Range (IQR) method to remove the outliers [37]. First, we determined outlier thresholds using the IQR method by calculating the first and third quartiles and defining limits based on 1.5 times the IQR. Next, we identified outliers by checking whether data points fell outside these thresholds. Finally, we handled the detected outliers by replacing them with the respective threshold values. Figure 5 shows an example of detected outliers in the 'REB' feature.

By using this technique, the impact of extreme values, which could skew the analysis and the model's predictions, can be eliminated. This ensures that data remains robust by reducing the influence of extremes while retaining as much data as possible.

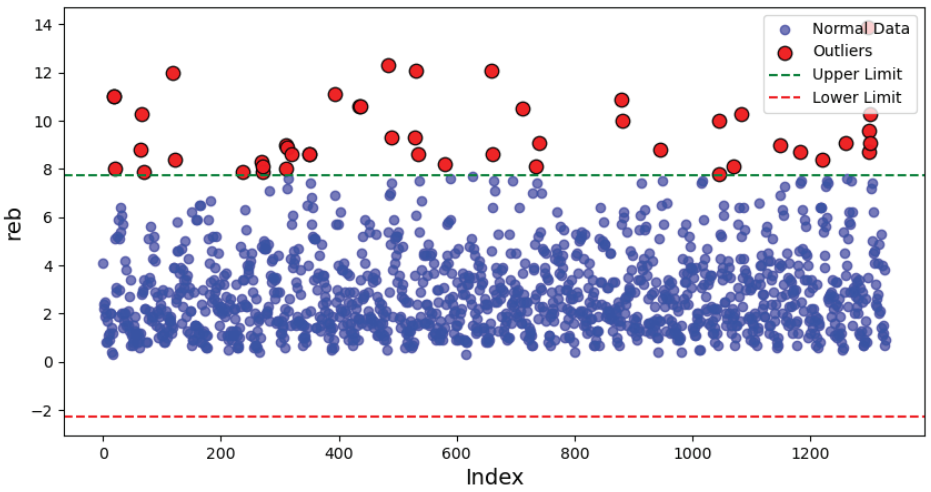



**Figure 5.** Detected outliers in the 'REB' feature.

Feature Selection

Feature selection was conducted to identify the most relevant features for predicting REB. We employed Recursive Feature Elimination (RFE) [38] using a Linear Regression estimator to select the most relevant features from our training dataset. RFE iteratively removed the least essential predictors until the top 10 features were retained, the selected features were PTS, FGM, 3p_made, FTM, FTA, OREB, DRAB, STL, BLK, and TOV. The selected features were then used to construct reduced training and testing sets, ensuring the model leveraged only the most informative predictors. Finally, all ML models were trained on these selected features and evaluated using the MSE and $R^2$ score.

### *3.4. Machine Learning Models*

In sports predictive analysis, various regression models are utilized to forecast outcomes such as player performance, game results, and injury probabilities. In our study, we applied multiple regression models—Linear Regression, RF Regressor, Gradient Boosting Regressor, K-Neighbors Regressor, and Multi-Layer Perceptron (MLP) Regressor—to the dataset to evaluate their predictive performance for enhancing basketball team strategies.

#### 3.4.1. Linear Regression

Linear Regression models the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data. The hyperparameter, 'fit_intercept=True', ensures the model includes an intercept term in the equation, allowing it to fit the data better if the target variable does not naturally pass through the origin [39].

#### 3.4.2. Random Forest Regressor

RF is an ensemble learning method that constructs multiple decision trees and outputs the average prediction. It has been used in sports to predict athlete performance metrics [40]. The hyperparameters n_estimators = 100, max_depth = 10, min_samples_split = 4, and min_samples_leaf = 2 help balance model complexity and overfitting. It performs well with high-dimensional data and captures intricate feature interactions, but its main drawback is its computational expense and reduced interpretability compared to simpler models [41].

#### 3.4.3. Gradient Boosting Regressor

XGB builds models sequentially, with each new model correcting errors made by previous ones. With hyperparameters n_estimators = 100, learning_rate = 0.01, and max_depth = 3, GBR balances accuracy and overfitting, making it effective for match outcome predictions and player valuation. It excels in handling complex, non-linear relationships but requires careful hyperparameter tuning to prevent excessive computational costs [42].

#### 3.4.4. K-Neighbors Regressor

The KNN predicts the value of a target variable based on the average of the 'k' nearest neighbors in the feature space. Its simplicity allows for easy implementation in sports analytics to predict outcomes based on similar historical instances. However, KNN can be sensitive to the choice of 'k' and the distance metric used, and it may struggle with high-dimensional data common in sports analytics [43].

#### 3.4.5. Multi-Layer Perceptron Regressor

An MLP Regressor is an artificial neural network consisting of multiple layers of nodes, including input, hidden, and output layers. Each node uses a nonlinear activation function, enabling the network to capture complex, nonlinear relationships in data [39,44,45].

Table 3 summarizes the ML models and their associated hyperparameters. This comparison highlights the specific hyperparameters considered for each model, showcasing the diversity in their configurations and optimization potential.

**Table 3.** Comparison of Models Used and Their Hyperparameters.

| Model | Hyperparameters Used |
|---|---|
| Linear Regression | fit_intercept = True |
| XGB Regressor | n_estimators = 100, learning_rate = 0.01, max_depth = 3 |
| RF Regressor | n_estimators = 100, max_depth = 10, min_samples_split = 4, min_samples_leaf = 2 |
| KNN | n_neighbors = 10, weights = uniform |
| MLP Regressor | hidden_layer_sizes = (50, 50), activation = relu, solver = adam, learning_rate = 0.001, max_iter = 200 |

## 4. Results

The results obtained after training and evaluating the ML models are discussed in this section. The dataset was split into training and testing sets using an 80/20 ratio. The selected models were trained on the training set, and their predictive accuracy was found using a test set.

### *4.1. Performance Matrix*

The model evaluation is carried out by using several performance metrics, such as mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared ($R^2$) score. These metrics offer a thorough evaluation of the performance of every model. The model precision is calculated by MSE and RMSE, which measures the average squared and absolute differences between the predicted and actual values, respectively. The MAE offered an average absolute error measure that provides the prediction error's magnitude. The $R^2$ score indicated the proportion of variance in the target variable explained by the model.

### *4.2. Model Training and Evaluation*

The trained model was evaluated on the test dataset. Table 4 provides the Comparison of Regression Models. Table 4 summarizes our experimental results for predicting REB. Notably, Linear Regression (MSE = 0.6133, RMSE = 0.7831, MAE = 0.5834, $R^2$ = 0.8668) and Gradient Boosting Regressor (MSE = 0.6282, RMSE = 0.7926, MAE = 0.5896, $R^2$ = 0.8636) are the best-performing models, exhibiting lower error metrics and higher $R^2$ scores. In comparison, while the RF Regressor shows competitive performance ($R^2$ = 0.8564), the K-Neighbors Regressor ($R^2$ = 0.5274) and MLP Regressor ($R^2$ = 0.8244) yield a higher number of errors, suggesting that more straightforward linear approaches or boosting methods may be more appropriate for this prediction task.

**Table 4.** Comparison of Regression Models.

| Model | MSE | RMSE | MAE | $R^2$ Score |
|---|---|---|---|---|
| Linear Regression | 0.6133 | 0.7831 | 0.5834 | 0.8668 |
| RF Regressor | 0.6615 | 0.8133 | 0.5804 | 0.8564 |
| XGB Regressor | 0.6282 | 0.7926 | 0.5896 | 0.8636 |
| KNN Regressor | 2.1765 | 1.4753 | 1.1240 | 0.5274 |
| MLP Regressor | 0.8089 | 0.8994 | 0.6674 | 0.8244 |

Initial evaluations showed that Linear Regression and Gradient Boosting Regressor performed well compared to other models, with high $R^2$ scores and low error metrics.

### 4.3. Hyperparameter Tuning with Grid Search

This section will discuss the details of hyperparameter tuning with a grid search to fine-tune the hyperparameter. Hyperparameter tuning optimizes the ML models and enhances their performance. This is done for the best-performing models: Linear Regression and Gradient Boosting Regressor. For hyperparameter tuning, 5-fold cross-validation was performed via GridSearchCV. For every set of hyperparameters, multiple models were trained as part of the Grid Search process, and their performance was assessed using the validation set's R-squared ($R^2$) score. This provides a robust understanding of how different hyperparameters influence model performance. The hyper-parameters that minimized the MSE, RMSE, and MAE and maximized the $R^2$ score were chosen as shown in Table 5.

**Table 5.** Hyperparameter Tuning Grids for Regression Models.

| Model | Parameter | Values |
|---|---|---|
| Linear Regression | fit_intercept | True, False |
| XGB Regressor | n_estimators | 100, 200, 300 |
| | learning_rate | 0.001, 0.01, 0.1 |
| | max_depth | 2, 3, 4 |

The grid search for optimal hyperparameters for Linear Regression ('fit_intercept = False') suggests that the dataset is well-preprocessed, eliminating the need for an intercept term, likely due to feature scaling or centering. For the XGB Regressor, the selected parameters ('learning_rate = 0.1', 'max_depth = 3', 'n_estimators = 300') indicate a balance between learning complexity and generalization. A moderate learning rate prevents overfitting while ensuring convergence, a tree depth of 3 captures essential feature interactions without excessive complexity, and 300 estimators provide sufficient iterations for robust predictions.

Table 6 provides a comparison of regression models after hyperparameter tuning. It is found that the best model configurations for Linear Regression are the optimal settings of fit_intercept and normalize and for Gradient Boosting Regressor, the optimal settings of n_estimators, learning_rate, and max_depth. These tuned models were then evaluated on the test set. It can be found that the model performance has increased compared to not-tuned models. The final models were made accurate and generalized by carefully adjusting the hyperparameters, meaning these approaches can be utilized in real-world scenarios effectively.

**Table 6.** Comparison of Regression Models after Hyper-parameter Tuning.

| Model | MSE | RMSE | MAE | $R^2$ Score |
|---|---|---|---|---|
| Best Linear Regression | 0.6133 | 0.7831 | 0.5834 | 0.8668 |
| Best Gradient Boosting Regressor | 0.5761 | 0.7590 | 0.5711 | 0.8749 |

## 5. Discussion

Our study investigated how well different parametric and nonparametric ML models could predict REB for NBA players. The prior studies discussed in the literature review align with our research, which supports the effectiveness of nonparametric models after hyperparameter tuning in capturing complex patterns that influence REB statistics [20].

Our findings are built on these concepts by measuring the predictive accuracy through various metrics, including $R^2$, MSE, RMSE, and MAE. In contrast, $R^2$ measures the proportion of variance explained by the model. Metrics such as MSE and RMSE quantify the average squared and absolute differences between predicted and actual values, offering insights into the magnitude of prediction errors. By combining these metrics, we

ensure a balanced evaluation of the model's ability to capture patterns in the data while minimizing error.

RQ1. What is the prediction accuracy level of parametric and non-parametric ML models when it comes to the prediction of REB for NBA players?

It was found that, among all nonparametric models, the Gradient Boosting Regressor and RF Regressor are best for predicting REB. The Gradient Boosting Regressor achieved an $R^2$ score of 0.8636, and the RF Regressor achieved an $R^2$ score of 0.8564. Interestingly, a parametric model, such as Linear Regression, achieved a slightly higher $R^2$ value of 0.8668, showing the ability to capture REB variance despite the model's simplicity effectively. These results underscore the fact that nonparametric models are typically more flexible in handling nonlinear relationships. However, parametric models can still perform well, depending on the nature of the data and the features used.

RQ2. What type of model, parametric or nonparametric, benefits most from optimizing hyperparameters to predict REB for NBA players?

The prediction accuracy of the ML models can be enhanced by hyperparameter tuning. It was found that, among all ML models, the ability of the Gradient Boosting Regressor in terms of $R^2$ value increased from 0.8636 to 0.8749 after optimization. This underscores the importance of model-specific fine-tuning, which was very beneficial for non-parametric ML models. The findings underscore how parameter adjustment can mitigate overfitting and underfitting, improving the model's predictive accuracy [31].

Our study employs "GridSearchCV" to perform a systematic and exhaustive search over predefined hyperparameter values, optimizing key hyperparameters such as 'n_estimators' and 'learning_rate' for the Gradient Boosting Regressor to enhance predictive performance. The parameter grid was carefully tailored to balance computational efficiency with model accuracy. This methodological innovation allows our models to predict outcomes with higher accuracy and can be utilized in real-world scenarios effectively. In our study, we compared the effectiveness of both parametric and nonparametric models in predicting NBA players' performance, such as REB.

The parametric Linear Regression model demonstrated a strong ability to predict outcomes with an $R^2$ score of 0.8668. This suggests that it can explain approximately 86.68% of the variance in REB from the variables used. The non-parametric XGB Regressor showed a lower initial $R^2$ value of 0.8636 but benefited from hyperparameter tuning, which improved its $R^2$ value to 0.8749.

Our study also deeply explains model applicability in sports in real time. The most precise of our predictive models was the Gradient Boosting Regressor with an $R^2$ value of 0.8749, which can help teams to make well-informed strategic decisions like optimizing player rotations and game tactics, which in turn minimize injury risks by more accurately predicting player fatigue levels [5]. In addition, predictive models can help create personalized training programs based on players' performance metrics. This helps to maximize player performance and overall team efficiency [6].

Together, the findings from RQ1 and RQ2 support our hypothesis: Advanced feature engineering, data preprocessing, and hyperparameter tuning significantly increase the ML models' ability to predict future average rebounds per game (REB) for NBA players. Our work demonstrates that while parametric models can perform well with proper data design, non-parametric models—especially when tuned—offer superior flexibility and predictive power in complex sports data environments.

## 6. Conclusions and Future Work

In this research, we explored the effectiveness of predictive analytics through various regression models to predict player performance, such as REB. Our findings indicate that

predictive analytics can enhance basketball team strategies by providing data-driven insights. Two models proved most effective after hyperparameter tuning: Gradient Boosting and Linear Regression models. Among these two models, the non-parametric Gradient Boosting model performed well.

The major limitation observed is that even after fine-tuning the hyperparameters, the performance of different models did not increase significantly. This highlights the need for a larger dataset to improve the model's performance and better generalization in predictive analysis.

Due to the low availability of the dataset, the ML models have been utilized in this study. Nevertheless, deep learning (DL) techniques can also be explored in larger datasets, which will help identify more complex patterns in future work. A comparative study between DL and ML models can be examined to better understand their implementation. Our analysis is mainly focused on predicting REBs; other features, such as the prediction of the number of years played, can be explored. Our study underscores the ability of ML to transform the sports field by offering a dynamic approach to player performance analysis. By accurately predicting REB, teams can optimize training, develop game strategies, and effectively utilize the game's players, enhancing team performance. As these approaches become more advanced and prevalent, a new age of data-driven sports management could emerge, reshaping the norms of operations and competition in the NBA and other sports leagues.

**Author Contributions:** Conceptualization, A.K.; Investigation, R.C.; Writing—original draft, R.C.; Writing—review & editing, R.C. and P.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research has been partially funded with the financial support of Taighde Éireann–Research Ireland under Grant number 13/RC/2094_2.

**Data Availability Statement:** The original data presented in the study are openly available in Kaggle at DATA.

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

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| RF | Random Forest |
| CNN | Convolutional Neural Networks |
| ML | Machine Learning |
| DL | Deep Learning |
| MLP | Multi-Layer Perceptron |
| MSE | Mean Squared Error |
| RMSE | Root Mean Squared Error |
| MAE | Mean Absolute Error |
| KNN | K-Neighbors |
| NBA | National Basketball Association |
| FFNN | Feedforward Neural Networks |
| SOM | Self-organizing Maps |
| XGB | Gradient Boosting |

## References


1. Biichner, A.G.; Dubitzky, W.; Schuster, A.; Lopes, P.; O'Doneghue, P.G.; Hughes, J.; Bell, D.A.; Adamson, K.; White, J.A.; Anderson, J.M.C.C.; et al. Corporate evidential decision making in performance prediction domains. *arXiv* **2013**, arXiv:1302.1523.
2. Vater, C.; Mann, D.L. Are predictive saccades linked to the processing of peripheral information? *Psychol. Res. Psychol. Forsch.* **2022**, *87*, 1501–1519. [CrossRef] [PubMed]
3. Koster, M.; Kurz, S.; Lindner, I.; Napel, S. The Prediction Value. *Soc. Choice Welf.* **2017**, *48*, 433–460. [CrossRef]
4. Wang, H.; Gao, S.; Wang, B.; Ma, Y.; Guo, Z.; Zhang, K.; Yang, Y.; Yue, X.; Hou, J.; Huang, H.; et al. Recent advances in machine learning-assisted fatigue life prediction of additive manufactured metallic materials: A review. *J. Mater. Sci. Technol.* **2024**, *198*, 111–136. [CrossRef]
5. Imbach, F.; Sutton-Charani, N.; Montmain, J.; Candau, R.; Perrey, S. The Use of Fitness-Fatigue Models for Sport Performance Modelling: Conceptual Issues and Contributions from Machine-Learning. *Sport Med. Open* **2022**, *8*, 29. [CrossRef]
6. Singh, A.; Bevilacqua, A.; Nguyen, T.L.; Hu, F.; McGuinness, K.; OReilly, M.; Whelan, D.; Caulfield, B.; Ifrim, G. Fast and robust video-based exercise classification via body pose tracking and scalable multivariate time series classifiers. *Data Min. Knowl. Discov.* **2022**, *37*, 873–912. [CrossRef]
7. Jaiswal, P.; Kaushik, A.; Lawless, F.; Malaquias, T.; McCaffery, F. Preliminary Investigation on Machine Learning and Deep Learning Models for Change of Direction Classification in Running. In *International Conference on Intelligent Data Engineering and Automated Learning*; Springer: Cham, Switzerland, 2024, pp. 180–191.
8. Rossi, A.; Perri, E.; Pappalardo, L.; Cintia, P.; Alberti, G.; Norman, D.; Iaia, F.M. Wellness Forecasting by External and Internal Workloads in Elite Soccer Players: A Machine Learning Approach. *Front. Physiol.* **2022**, *13*, 896928. [CrossRef]
9. Guo, J.; Czarnecki, K.; Apely, S.; Siegmundy, N.; Wasowski, A. Variability-aware performance prediction: a statistical learning approach. In Proceedings of the 2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE), Silicon Valley, CA, USA , 11–15 November 2013; pp. 301–311. [CrossRef]
10. Transtrum, M.K.; Machta, B.B.; Sethna, J.P. Why are nonlinear fits to data so challenging. *Phys. Rev. Lett.* **2010**, *104*, 060201. [CrossRef]
11. Sodhi, H.S. Kinanthropometry and performance of top ranking Indian basketball players. *Br. J. Sport Med.* **1980**, *14*, 139–144. [CrossRef]
12. Sampaio, J.; McGarry, T.; Calleja-González, J.; Sáiz, S.L.J.; i del Alcázar, X.S.; Balciunas, M. Exploring Game Performance in the National Basketball Association Using Player Tracking Data. *PLoS ONE* **2015**, *10*, e0132894. [CrossRef]
13. Nishisaka, M.M.; Zorn, S.; Kristo, A.S.; Sikalidis, A.K.; Reaves, S.K. Assessing Dietary Nutrient Adequacy and the Effect of Season—Long Training on Body Composition and Metabolic Rate in Collegiate Male Basketball Players. *Sports* **2022**, *10*, 127. [CrossRef] [PubMed]
14. Hauri, S.; Djuric, N.; Radosavljevic, V.; Vucetic, S. Multi-Modal Trajectory Prediction of NBA Players. In Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2021; pp. 1639–1648. [CrossRef]
15. Chang, J.C. Predictive Bayesian selection of multistep Markov chains, applied to the detection of the hot hand and other statistical dependencies in free throws. *R. Soc. Open Sci.* **2019**, *6* . [CrossRef] [PubMed]
16. Vaci, N.; Cocić, D.; Gula, B.; Bilalić, M. Large data and Bayesian modeling-aging curves of NBA players. *Behav. Res. Methods* **2019**, *51*, 1544–1564. [CrossRef]
17. Shen, W. Analysis of Professional Basketball Field Goal Attempts via a Bayesian Matrix Clustering Approach. *J. Comput. Graph. Stat.* **2022**, *32*, 49–60. [CrossRef]
18. Arnold, T.; Godbey, J.M. Introducing Linear Regression: An Example Using Basketball Statistics. *Soc. Sci. Res. Netw.* **2012** . [CrossRef]
19. Xie, X. Analysis on the Application of Linear Regression in Various Fields 2020. Available online: https://www.clausiuspress.com/conferences/LNEMSS/ICEMGD%202020/368.pdf (accessed on 26 April 2025).
20. Siemon, D.; Ahmad, R.; Huttner, J.P.; Robra-Bissantz, S. Predicting the Performance of Basketball Players Using Automated Personality Mining. In Proceedings of the Twenty-fourth Americas Conference on Information Systems, New Orleans, LA, USA, 16–18 August 2018.
21. Bavencoff, F.; Vanpeperstraete, J.M.; Cadre, J.P. Performance analysis of optimal data association within a linear regression framework. In Proceedings of the 2005 7th International Conference on Information Fusion, Philadelphia, PA, USA, 25–28 July 2005. [CrossRef]
22. Hanczár, G.; Stippinger, M.; Hanák, D.; Kurbucz, M.T.; Törteli, O.M.; Chripkó, Á.; Somogyvári, Z. Feature space reduction method for ultrahigh-dimensional, multiclass data: Random forest-based multiround screening (RFMS). *arXiv* **2023**. [CrossRef]
23. Voges, L.F.; Jarren, L.C.; Seifert, S. Opening the random forest black box by the analysis of the mutual impact of features. *arXiv* **2023**. [CrossRef]
24. Piernik, M. Random Similarity Forests. In *Lecture Notes in Computer Science*; Springer: Cham, Switzerland , 2023. [CrossRef]

25. Ustimenko, A.; Prokhorenkova, L.O.; Malinin, A. Uncertainty in Gradient Boosting via Ensembles. *arXiv* **2020**, arXiv:2006.10562.
26. Rittler, N.; Chaudhuri, K. A Two-Stage Active Learning Algorithm for *k*-Nearest Neighbors. *arXiv* **2022**. [CrossRef]
27. Pei, E.; Fokoué, E. Improving the Predictive Performances of *k* Nearest Neighbors Learning by Efficient Variable Selection. *arXiv* **2022**. [CrossRef]
28. Gui, Y.; Wang, X. Application of K-nearest neighbors in protein-protein interaction prediction. In *Highlights in Science, Engineering and Technology*; Darcy & Roy Press: Hillsboro, OR, USA, 2022. [CrossRef]
29. Chi, Y.N.; Chi, J. A Mixed Model for Performance-Based Classification of NBA Players: Performance-Based Classification of NBA Players. *Int. J. Data Sci. Adv. Anal.* **2021**, *3*, 36–46. [CrossRef]
30. Gao, D.; Guo, Q.; Jin, M.; Liao, G.; Eldar, Y. C. Hyper-Parameter Auto-Tuning for Sparse Bayesian Learning. *arXiv* **2022**. [CrossRef]
31. Bhattacharyya, A.; Vaughan, J.; Nair, V.N. Behavior of Hyper-Parameters for Selected Machine Learning Algorithms: An Empirical Investigation. *arXiv* **2022**. [CrossRef]
32. Nguyen, H.N. Tuning hyperparameters of self-organizing maps in combination with k-nearest neighbors for iot malware detection. *J. Sci. Technol.* **2023**, *12*. [CrossRef]
33. Raji, I.D.; Bello-Salau, H.; Umoh, I.J.; Onumanyi, A.J.; Adegboye, M.A.; Salawudeen, A.T. Simple Deterministic Selection-Based Genetic Algorithm for Hyperparameter Tuning of Machine Learning Models. *Appl. Sci.* **2022**, *12*, 1186. [CrossRef]
34. Eimer, T.; Lindauer, M.; Raileanu, R. Hyperparameters in Reinforcement Learning and How To Tune Them. *arXiv* **2023**, [CrossRef]
35. Roy, S.; Mehera, R.; Pal, R.K.; Bandyopadhyay, S.K. Hyperparameter Optimization for Deep NeuralNetwork Models: A Comprehensive Study onMethods and Techniques. *Res. Sq.* 2023, *preprint*. [CrossRef]
36. Yakhyojon. National Basketball Association (NBA) Dataset. Kaggle dataset. 2024. Available online: https://www.kaggle.com/datasets/yakhyojon/national-basketball-association-nba (accessed on 26 April 2025).
37. Dash, C.S.K.; Behera, A.K.; Dehuri, S.; Ghosh, A. An outliers detection and elimination framework in classification task of data mining. *Decis. Anal. J.* **2023**, *6*, 100164. [CrossRef]
38. Chen, X.w.; Jeong, J.C. Enhanced recursive feature elimination. In Proceedings of the Sixth international conference on machine learning and applications (ICMLA 2007), Cincinnati, OH, USA, 13–15 December 2007; pp. 429–435.
39. Baboota, R.; Kaur, H. Predictive analysis and modelling football results using machine learning approach for English Premier League. *Int. J. Forecast.* **2019**, *35*, 741–755. [CrossRef]
40. Sahin, E.K. Assessing the predictive capability of ensemble tree methods for landslide susceptibility mapping using XGBoost, gradient boosting machine, and random forest. *SN Appl. Sci.* **2020**, *2*, 1308. [CrossRef]
41. Salman, H.A.; Kalakech, A.; Steiti, A. Random forest algorithm overview. *Babylon. J. Mach. Learn.* **2024**, *2024*, 69–79. [CrossRef] [PubMed]
42. Zhang, Z.; Zhao, Y.; Canes, A.; Steinberg, D.; Lyashevska, O. Predictive analytics with gradient boosting in clinical medicine. *Ann. Transl. Med.* **2019**, *7*, 152. [CrossRef] [PubMed]
43. Halder, R.K.; Uddin, M.N.; Uddin, M.A.; Aryal, S.; Khraisat, A. Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. *J. Big Data* **2024**, *11*, 113. [CrossRef]
44. Sayed, E.H.; Alabrah, A.; Rahouma, K.H.; Zohaib, M.; Badry, R.M. Machine Learning and Deep Learning for Loan Prediction in Banking: Exploring Ensemble Methods and Data Balancing. *IEEE Access* **2024**, *12*, 193997–194019. [CrossRef]
45. Maszczyk, A.; Gołaś, A.; Pietraszewski, P.; Roczniok, R.; Zając, A.; Stanula, A. Application of neural and regression models in sports results prediction. *Procedia-Soc. Behav. Sci.* **2014**, *117*, 482–487. [CrossRef]

*electronics*

MDPI

*Article*

# LocRecNet: A Synergistic Framework for Table Localization and Rectification

**Zefeng Cai [1,†], Jie Feng [2,*,†], Zhaokun Hou [2,†], Haixiang Zhang [2,†] and Hanjie Ma [2,†]**

[1] School of Information Science and Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China; 202230704045@mails.zstu.edu.cn
[2] School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China; 2023220603025@mails.zstu.edu.cn (Z.H.); zhhx@zstu.edu.cn (H.Z.); mahanjie@zstu.edu.cn (H.M.)
* Correspondence: arlose@zstu.edu.cn
† These authors contributed equally to this work.

**Abstract:** This paper introduces LocRecNet, a deformation-aware network for table localization and correction, aimed at improving the recognition accuracy of complex table data. Conventional algorithms typically depend on table cell or line features for model training but exhibit limitations when processing real-world deformed table data. LocRecNet addresses these challenges by correcting deformations prior to table structure recognition, significantly enhancing model performance. The proposed network employs a novel keypoint detection method to precisely locate table edge points, enabling the efficient correction of deformed tables. Experimental results reveal that integrating LocRecNet substantially improves table recognition algorithms in terms of various key performance metrics, with recall rates increasing by up to 10% and F1 scores nearing 90%. Tests conducted on real-world datasets further validate its effectiveness, demonstrating a reasonable trade-off between computational cost and performance gains. Additionally, LocRecNet enhances performance even on standard table data, highlighting its strong generalizability and potential for broader application.

**Keywords:** LocRecNet; keypoint detection; table correction; TSR

## 1. Introduction

In the current era of big data, the ability to efficiently access and retrieve data, as well as extract meaningful insights from vast datasets, has emerged as a critical imperative across various sectors. Being a key medium for organizing data, tables are valued for their ability to efficiently consolidate information and clearly represent data relationships, which has led to their widespread use across industries. Table data are commonly used in documents to summarize and present information [1]. In the context of the ongoing digital transformation, there is a rapidly growing demand to extract tabular data from unstructured sources, such as images and PDF files [2]. Although this task may seem trivial to humans, the significant variability in table layouts and formats poses a considerable challenge for automated systems.

Table structure recognition (TSR) aims to interpret the structure of tables and represent them in machine-readable formats. However, due to variations in table layouts, presentation styles, and noise, this task continues to pose significant challenges. Accurately detecting and recognizing tables under such conditions has become a crucial aspect of table data processing. While numerous algorithms exist for the structural recognition of standard tables, they often require high-quality input table images [3]. In real-world

scenarios, however, table images often display various deformations or missing elements, compromising the effectiveness of earlier algorithms.

In this study, we propose a novel network architecture, LocRecNet, aimed at addressing deformations in table data to enhance extraction accuracy. To simulate table data representing various degrees of deformation in real-life situations, a deformation algorithm is designed to generate the input dataset. In the table structure recognition process, LocRecNet is incorporated into the preprocessing stage of existing recognition models, acting as a data correction module to accurately restore deformed tables and improve recognition performance. Unlike traditional approaches that rely on edge detection or similar techniques to define image boundaries, we introduce a novel keypoint detection algorithm specifically tailored for table image analysis. After being localized and corrected by LocRecNet, the table data are subsequently input into the table structure recognition model. Experimental results show that this method substantially improves table structure recognition performance. In conclusion, our contributions are reflected in the following three aspects:

1. A novel network architecture, LocRecNet, is proposed to effectively detect and correct deformations in table data. It precisely localizes key points and corrects deformations, ensuring a more reliable input for subsequent structure recognition, thereby improving both accuracy and robustness.
2. A new keypoint detection algorithm, tailored for table image analysis and serving as a preprocessing step for correcting deformations, efficiently detects and localizes tables of various types and structures. This approach addresses the limitations of existing methods in handling severely deformed or noisy tables, substantially enhancing processing and recognition capabilities.
3. Multiple deformed table datasets are generated using the algorithm, covering various table types, such as financial reports and forms, and incorporating different levels of geometric distortions, noise, and other real-world challenges. This fills a gap in current research, where comprehensive deformed datasets have been notably lacking.

## 2. Related Work

In recent years, the technology for recognizing table structures has rapidly developed, achieving significant results on datasets such as ICDAR-13 [4], SCITSR [5], PubTabNet [6], and WTW [7]. The progress in this field can be divided into two stages: Initially, it relied on traditional algorithms, including methods based on row and column segmentation, text detection and expansion, text block classification, and the integration of various approaches. These traditional methods performed well in simple scenarios but struggled in complex situations with high noise levels or low text density. With the advent of deep learning technologies, table structure recognition has made breakthroughs in adapting to complex scenarios. However, current algorithms primarily process data from PDFs or scans, which are relatively uniform and free of deformations. Therefore, the effectiveness of recognizing deformed table images in real-world applications still needs improvement.

The continuous exploration by researchers aimed at overcoming the limitations of existing technologies has led to the introduction of more advanced solutions. In 2021, Qiao and others launched the LGPMA [8] network, which employs a local and global pyramid mask alignment framework. It refines the prediction boundaries in local and global feature maps through a soft pyramid mask learning mechanism, enhancing the capability to locate and divide empty cells. Subsequently, Liu and his team attempted to integrate Transformer technology to learn more suitable inductive biases but faced challenges due to large data scales and unstable training. They further proposed FLAG-Net [9], combining a Transformer in an adaptive manner with a graph-based context aggregator,

to achieve end-to-end table element relation inference without the need for additional metadata or OCR information. Additionally, addressing the diversity issue in table structure recognition, Liu's team developed a new Neural Cooperative Graph Machine (NCGM) [10], alternating stacked cooperative blocks to extract intra-modal context and layering inter-modal interactions to represent the intra-modal relationships of table elements and adjust the cooperation mode among different modalities more accurately. In 2023, Xing and colleagues observed that many methods relied on heuristic rules to recover table structures, which not only required a large amount of training data but were also inefficient. Therefore, they proposed the LORE [11] framework, simplifying table structure recognition to a logical position regression problem. This approach is simpler, easier to train, and more accurate than traditional TSR methods.

We examined the current advancements and limitations in table structure recognition technology. While advanced algorithms like LGPMA and FLAG-Net have made significant progress in processing conventional table structures, their performance on diverse deformed tables remains suboptimal. This issue can be attributed to two factors: the lack of datasets with sufficient deformed tables and the immaturity of recognition techniques for deformed data. To address these challenges, this paper proposes an innovative deformation simulation algorithm designed to generate deformed table images simulating real-world scenarios, effectively augmenting existing datasets. Additionally, we introduce a novel recognition strategy that involves first localizing and correcting deformed tables and then performing structural recognition. This strategy integrates a localization and correction network at the front end of the recognition pipeline. Once the deformations are corrected, existing recognition techniques can be applied directly, significantly enhancing both accuracy and efficiency, as well as paving the way for the more effective recognition of complex deformed tables.

## 3. Methodology

### *3.1. Overall Architecture*

This section provides a detailed explanation of the proposed LocRecNet, designed to optimize table structure recognition models for various table formats. As shown in Figure 1, LocRecNet first employs a table edge point localization network to detect deformed tables, obtaining the key points needed for subsequent correction. In the correction network, based on the key points from table localization, curve fitting of the four boundary lines is performed, along with the calculation of boundary lengths, to determine the size of the restored table. The correction algorithm then uses the boundary curves and internal key points to restore the deformed table image, producing a normalized output. Finally, the corrected image is fed into the table structure recognition network for recognition. The following subsections introduce these key components in detail.

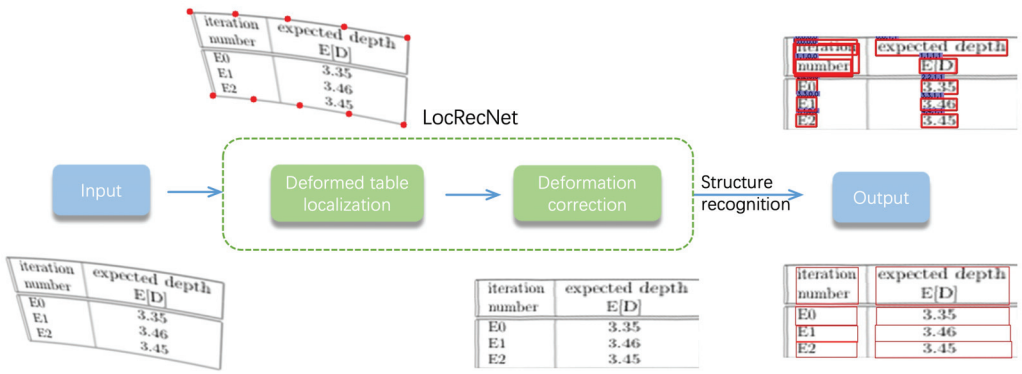



**Figure 1.** Overallstructure diagram.

### *3.2. Table Edge Point Localization*

In the LocRecNet framework, the precise localization of table boundary lines is a crucial component. For the design of the table edge point localization module, we conducted extensive analyses and experimentation. While many existing image localization techniques, such as boundary detection [12] and YOLO [13], demonstrate excellent accuracy, the information provided by these methods is limited for subsequent correction tasks, particularly in handling tables. These techniques generally only detect the corner points of tables, offering insufficient information for precise boundary curve fitting. To address this limitation, we propose a novel keypoint detection network specifically tailored for table detection.

Our algorithm is inspired by the lightweight keypoint detection network SimCC [14], with significant improvements to better accommodate the characteristics of table images. Unlike SimCC, our approach not only transforms the regression problem into a classification problem but also enhances both accuracy and efficiency by optimizing feature extraction and keypoint prediction. We specifically adopted an optimized version of HRNet [15] as the backbone network and introduced a new architecture, HRNet-s, as shown in Figure 2. By removing the fourth stage of HRNet and retaining only the first three, we reduced the model's complexity while preserving its strong feature extraction capabilities for table images.

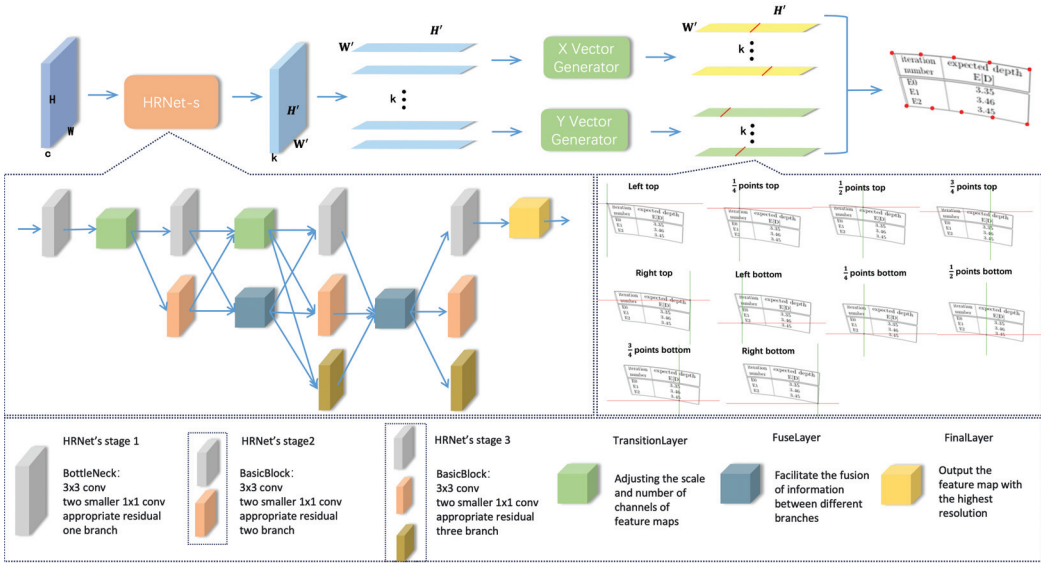



**Figure 2.** Keypointdetection network.

HRNet-s [16] extracts keypoint feature representations from the input table images, generating a feature map of shape (n, H′, W′). To facilitate subsequent classification operations, we flatten these features into a two-dimensional vector of shape (n, H′ × W′). Building upon this, we design a novel method to predict keypoint coordinates. Inspired by the classification strategy of SimCC, we employ separate classifiers for the horizontal and vertical coordinates, each consisting of a linear layer, to predict the x and y coordinates of each keypoint.

To further enhance localization accuracy, we adopt a method akin to that of SimCC, classifying each pixel into multiple bins to minimize the quantization error. This approach enables sub-pixel precision in keypoint detection, ultimately mapping precise coordinates through probability distributions. This foundation provides a solid basis for subsequent table correction and structural recognition.

### *3.3. Image Correction*

The image correction algorithm is an integral part of LocRecNet, developed to restore deformed tables and resolve recognition challenges caused by structural distortions. The correction process was designed with the careful consideration of computational costs as a preprocessing step for table structure recognition, resulting in a low-computation solution. Inspired by the thin-plate spline interpolation method [17], a traditional algorithm-based table correction approach was devised. This approach effectively manages deformed tables while enhancing the efficiency of the overall recognition process.

Based on the theory of thin-plate spline interpolation, it is necessary to define two terms initially: one is the fitting term $E_\Phi$, which measures the magnitude of the deformation of source points towards the target points; the second is the bending term $E_d$, which measures the amount of distortion of the surface. Therefore, we can derive the total loss function as follows:

$$E = E_\Phi + \lambda E_d \tag{1}$$

where $\lambda$ is a weighting coefficient that controls the degree to which non-rigid deformations are permitted, with different values of $\lambda$ being suitable for varying degrees of deformation. Specifically,

$$E_\Phi = \sum_{i=1}^{N} ||\Phi(p_i) - q_i||^2 \tag{2}$$

$$E_d = \iint_{R^2} \left( \left( \frac{\partial^2 \Phi}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 \Phi}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 \Phi}{\partial y^2} \right)^2 \right)^2 dx\, dy \tag{3}$$

In Equation (2), $N$ denotes the number of control points. Each control point $p_i \in R^2$ in the source image is mapped by the deformation function $\Phi$ and compared to its corresponding point $q_i \in R^2$ in the target image, forming the data fidelity term. Equation (3) defines the bending energy of the deformation function $\Phi$ over the entire domain, incorporating second-order partial derivatives with respect to spatial coordinates to quantify the smoothness of the transformation. By jointly minimizing these two energy terms, a unique closed-form solution for the deformation function $\Phi$ can be obtained, as presented in Equation (4):

$$\Phi(p) = M \cdot p + m_0 + \sum_{i=1}^{N} \omega_i U(||p - p_i||) \tag{4}$$

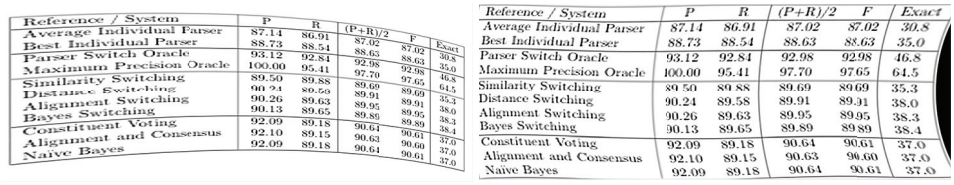
Let $p$ denote an arbitrary point on the surface, expressed as $p = (x, y)^T$, and let $p_i$ be the $i$-th control point in the domain. The term $\omega_i$ represents the weight associated with the radial basis function centered at $p_i$. The matrix $M = (m_1, m_2)$ is a $2 \times 2$ affine transformation matrix that models the global linear deformation, where $m_1$ and $m_2$ are its parameter vectors. The vector $m_0$ denotes the translation term, which controls the overall translation of the deformation. The function $U(\cdot)$ is the radial basis function, indicating that the deformation at any point on the surface is influenced by all control points. The complete formulation is given in Equation (5):

$$U(r) = r^2 \log r \tag{5}$$

where $r = ||p - p_i||$ represents the Euclidean distance between the point $p$ and the control point $p_i$. This function governs the influence of each control point on different locations on the surface, with the influence decreasing as the distance increases [18].

To address the problem of image deformation, this study proposes a novel image correction algorithm designed to restore the original structure and content of distorted

images by adjusting the control points. Inspired by thin-plate spline interpolation theory, the algorithm is optimized and customized for specific correction tasks, significantly enhancing its practicality and correction performance. Initially, the target dimensions of the corrected image are determined based on the keypoint detection results, with the primary objective of preserving the integrity of the table content. Based on the table localization outputs, 10 keypoints are selected as the initial control points, as their target transformation coordinates can be readily inferred. By calculating the length of the pre-correction table boundaries, the coordinates of the corner points and equally spaced division points along the top and bottom boundaries of the corrected table are derived. Using these points to construct the initial control set, the correction results (see Figure 3) demonstrate the algorithm's capability to recover the structural layout along the table boundaries. However, its performance in handling interior regions of the table or cases with severe deformations remains limited.



| Reference / System | P | R | (P+R)/2 | F | Exact |
|---|---|---|---|---|---|
| Average Individual Parser | 87.14 | 86.91 | 87.02 | 87.02 | 30.8 |
| Best Individual Parser | 88.73 | 88.54 | 88.63 | 88.63 | 35.0 |
| Parser Switch Oracle | 93.12 | 92.84 | 92.98 | 92.98 | 46.8 |
| Maximum Precision Oracle | 100.00 | 95.41 | 97.70 | 97.65 | 64.5 |
| Similarity Switching | 89.50 | 89.88 | 89.69 | 89.69 | 35.3 |
| Distance Switching | 90.24 | 89.58 | 89.91 | 89.91 | 38.0 |
| Alignment Switching | 90.26 | 89.63 | 89.95 | 89.95 | 38.3 |
| Bayes Switching | 90.13 | 89.65 | 89.89 | 89.89 | 38.4 |
| Constituent Voting | 92.09 | 89.18 | 90.64 | 90.61 | 37.0 |
| Alignment and Consensus | 92.10 | 89.15 | 90.63 | 90.60 | 37.0 |
| Naïve Bayes | 92.09 | 89.18 | 90.64 | 90.61 | 37.0 |

(**a**) (**b**)

**Figure 3.** Correction results using 10 control points. (**a**) Original deformed image; (**b**) image after correction.

To further improve correction accuracy, a control point enhancement strategy based on table structural features is proposed. Specifically, the left and right boundaries of the table are divided into four equal segments, excluding the existing corner points, thereby generating six additional evenly spaced points along the vertical edges. These enhance the algorithm's ability to handle vertical distortions. Furthermore, nine internal intersection points are derived by analyzing the equally spaced divisions across both the vertical and horizontal directions, providing more detailed and accurate references for deformation modeling. The coordinates of these points and their corresponding target positions in the corrected image can be directly computed from the predefined image dimensions. In total, 25 control points are constructed, comprising 10 points along the top and bottom boundaries (including corner points and evenly divided points), 6 points along the vertical boundaries (excluding corners), and 9 internal intersection points. The top and bottom boundary points primarily constrain overall structural deformation, the vertical boundary points enhance the correction of vertical distortions, and the internal points offer finer-grained geometric reference information, collectively improving the precision and robustness of the correction process. Two corresponding control point sets are ultimately established, denoted as sets $S$ and $T$, each containing 25 points.

$$S = \{(x_0, y_0), (x_0, y_1), \ldots, (x_4, y_4)\} \tag{6}$$

$$T = \{(x'_0, y'_0), (x'_0, y'_1), \ldots, (x'_4, y'_4)\} \tag{7}$$

where set $S$ and set $T$ correspond one-to-one, with set $S$ referred to as the template point set and set $T$ referred to as the target point set, which are the control points for thin-plate spline interpolation. The specific configuration of these point sets is illustrated in Figure 4.

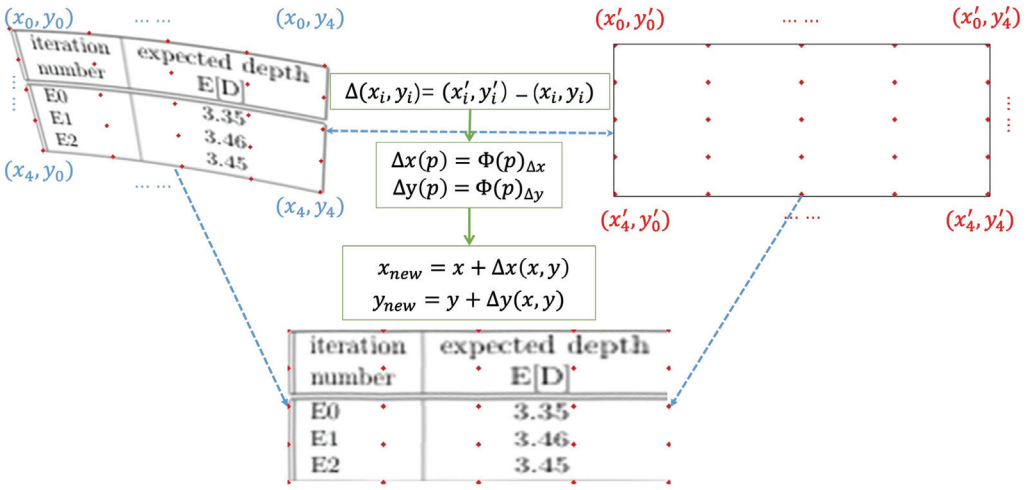



**Figure 4.** Correctionalgorithm architecture diagram.

Subsequently, utilizing the aforementioned 25 control points, the displacements $(\Delta x, \Delta y)$ can be derived through coordinate transformations, yielding $\Delta S = \{(\Delta x_0, \Delta y_0), \ldots, (\Delta x_4, \Delta y_4)\}$. Following this, it is necessary to establish a thin-plate spline interpolation model. By employing the control points and their corresponding displacements, two distinct thin-plate spline models are constructed: one for the interpolation of the horizontal displacement $\Delta x$ and the other for the interpolation of the vertical displacement $\Delta y$. The specific functional forms are as follows:

$$\Delta x(p) = \Phi(p)_{\Delta x} \tag{8}$$

$$\Delta y(p) = \Phi(p)_{\Delta y} \tag{9}$$

Thus, for each pixel point $(x, y)$ in the original image, the horizontal displacement $\Delta x(x, y)$ and vertical displacement $\Delta y(x, y)$ can be computed utilizing the thin-plate spline interpolation model. Consequently, as depicted in Figure 4, the updated pixel coordinates $(x_{\text{new}}, y_{\text{new}})$ can be derived using the following equations:

$$x_{\text{new}} = x + \Delta x(x, y) \tag{10}$$

$$y_{\text{new}} = y + \Delta y(x, y) \tag{11}$$

To obtain the corrected table image, we need to combine all the newly acquired pixel points. The implementation steps for the entire table image correction are shown in Algorithm 1. Ultimately, utilizing the aforementioned pixel processing techniques, we are able to achieve the correction of distorted tables, as illustrated in Figure 5.

| $n$ | $N_v(n)$ | $N_+(n)$ | Tot. | Tried |
|---|---|---|---|---|
| 1 | 0.1418 | 0.034 | 492 | 2,700 |
| 2 | 0.0203 | 0.0119 | 225 | 10,000 |
| 3 | 0.0028 | 0.0028 | 100 | 32,039 |
| 4 | $4.6\ 10^{-4}$ | $6.5\ 10^{-4}$ | 100 | 181,507 |
| 5 | $5.7\ 10^{-5}$ | $1.4\ 10^{-4}$ | 100 | $1.3\ 10^{6}$ |
| 6 | $8.6\ 10^{-6}$ | $2.9\ 10^{-5}$ | 100 | $7.3\ 10^{6}$ |
| 7 | $1.3\ 10^{-6}$ | $5.7\ 10^{-6}$ | 100 | $5.1\ 10^{7}$ |
| 8 | $1.8\ 10^{-7}$ | $1.1\ 10^{-6}$ | 34 | $1.0\ 10^{9}$ |

(**a**)

| $n$ | $N_v(n)$ | $N_+(n)$ | Tot. | Tried |
|---|---|---|---|---|
| 1 | 0.1418 | 0.034 | 492 | 2,700 |
| 2 | 0.0203 | 0.0119 | 225 | 10,000 |
| 3 | 0.0028 | 0.0028 | 100 | 32,039 |
| 4 | $4.6\ 10^{-4}$ | $6.5\ 10^{-4}$ | 100 | 181,507 |
| 5 | $5.7\ 10^{-5}$ | $1.4\ 10^{-4}$ | 100 | $1.3\ 10^{6}$ |
| 6 | $8.6\ 10^{-6}$ | $2.9\ 10^{-5}$ | 100 | $7.3\ 10^{6}$ |
| 7 | $1.3\ 10^{-6}$ | $5.7\ 10^{-6}$ | 100 | $5.1\ 10^{7}$ |
| 8 | $1.8\ 10^{-7}$ | $1.1\ 10^{-6}$ | 34 | $1.0\ 10^{9}$ |

(**b**)

**Figure 5.** Correction results using 25 control points. (**a**) Original deformed image; (**b**) image after correction.

**Algorithm 1** Table Image Correction Algorithm

**Require:** Deformed table image $Img_{deform}$, Control points set $S$, Target points set $T$
**Ensure:** Rectified table image $Img_{corrected}$

```
//Compute control points displacement
for i = 0 to length(S) − 1 do
    ΔS[i] = T[i] − S[i]
end for
//Establish TPS interpolation model
model_Δx = BuildRBFModel(S, ΔS_x component)
model_Δy = BuildRBFModel(S, ΔS_y component)
//Apply interpolation model to each pixel
for y = 0 to image_height(Img_deform) − 1 do
    for x = 0 to image_width(Img_deform) − 1 do
        Δx = Interpolate(model_Δx, x, y)
        Δy = Interpolate(model_Δy, x, y)
        x_new = x + Δx
        y_new = y + Δy
        Img_corrected[y_new][x_new] = GetPixel(Img_deform, x_new, y_new)
    end for
end for
return Img_corrected
```

### *3.4. Table Structure Recognition*

Numerous recognition algorithms have achieved significant success in processing standard tables. However, their performance remains limited when dealing with deformed tables. Although research efforts such as FLAG-Net and NCGM have sought to address this challenge, the absence of publicly available models and datasets has hindered in-depth performance comparisons. To overcome this limitation, the present study integrates open-source standard table structure recognition algorithms with the proposed LocRecNet to explore more effective methods for recognizing deformed table structures.

Following a comprehensive analysis, we select LORE and LGPMA as baseline algorithms for a performance evaluation, primarily due to their open-source nature and the validation of their original models on the SCITSR and PubTabNet datasets. As recent advancements in the field, both LORE and LGPMA show strong potential for handling complex deformed table structure recognition tasks. Consequently, we plan to use the pre-trained models of these two algorithms to recognize both original and deformed tables, incorporating LocRecNet for further optimization. This will allow us to assess the performance improvements brought by LocRecNet across different types of table images.

## 4. Experiments

### *4.1. Experimental Setting*

All experiments were conducted on a server equipped with an NVIDIA GeForce RTX 3090 GPU (24 GB VRAM) and an Intel Xeon Gold 6133 CPU @ 2.50 GHz, with 24 GB of RAM and a 2 TB SSD. The software environment consisted of Ubuntu 20.04.6, PyTorch 2.2.1, Python 3.9, and CUDA 12.1. Image processing and data preprocessing were performed using OpenCV and other related libraries.

For table keypoint detection, the SCITSR dataset was used for training. The input image sizes were set to 256 × 192 and 384 × 288 for different model scales. Training was performed for 150 epochs with a batch size of 16 using the Adam optimizer. The initial learning rate was $1\times 10^{-3}$, which was reduced to $1 \times 10^{-4}$ and $1 \times 10^{-5}$ at the 80th and 120th epochs, respectively. Label smoothing was employed to improve the generalization capability of the classification-based keypoint detection model. In the table structure

recognition task, the proposed LocRecNet was integrated into both the LORE and LGPMA frameworks. All models were trained and evaluated under identical settings to ensure fair comparisons and to validate the effectiveness and adaptability of the proposed method.

### *4.2. Dataset*

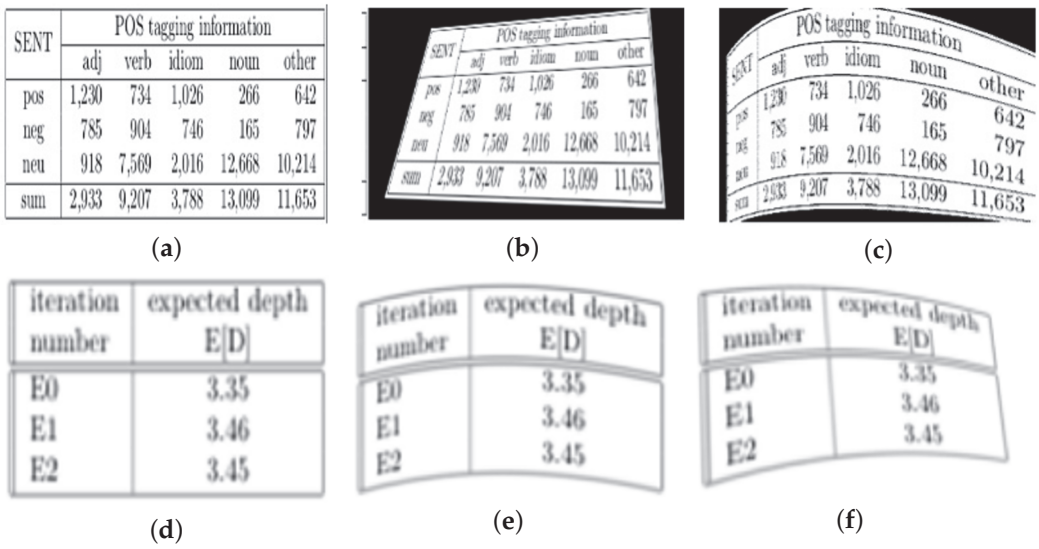
Currently, most publicly available table datasets primarily consist of standard table data obtained from scanned images or PDF screenshots, and, thus, they lack distortions or other artifacts. Although some datasets with deformed tables exist, they contain a very limited number of images. A review of the relevant literature showed that many datasets used in deformation-based table recognition studies are custom-created for specific experiments and are either not publicly available or restricted to internal access. This study applied a deformation algorithm to standard table image datasets, simulating the task of recognizing deformed tables.

The SCITSR and PubTabNet datasets were chosen as the foundation for the table deformation experiments. Renowned for their high-quality table images, comprehensive annotations, and extensive data volumes, these datasets are widely regarded as benchmarks in table structure recognition. They are frequently utilized to evaluate various table recognition algorithms, ensuring the generality and reliability of experimental results. To generate deformed data, an innovative algorithm combining Bézier curves [19] and perspective transformation [20] was developed. This approach integrates the nonlinear deformation characteristics of Bézier curves with the geometric deformation properties of perspective transformation, converting normal table data into deformed tables to effectively simulate the complex deformations encountered in real-world scenarios. Compared to methods such as NCGM, which rely on either Bézier curves or perspective transformations alone, the proposed combined approach generates more complex and realistic deformations. This provides more challenging and representative data for evaluating subsequent correction algorithms.

As shown in Figure 6, images b and c, generated using a single algorithm, exhibit simpler deformations. When the deformation extends beyond the image boundaries, black padding is typically applied. This not only affects structure recognition but also simplifies table localization, since the table boundaries are easily detectable through the black edges. In contrast, our method combines Bézier curves and perspective transformation, producing deformed table images (e.g., image f) that feature both bending and perspective effects. This results in more complex table structures that closely resemble real-world scenarios. Additionally, we improved the padding method by replacing the black padding with a white fill to match the table's background color. This adjustment enables the table to blend more seamlessly into the background, producing cleaner data and minimizing external noise.

Given that the aforementioned deformed table image datasets are primarily algorithmically generated, the WTW dataset was selected to further validate the algorithm's effectiveness in real-world scenarios. The WTW dataset is a public resource focused on table and text recognition in document images, containing table images from a variety of complex scenarios, offering high diversity and authenticity. To create the WTW-curved dataset, table samples with significant deformation were selected from the WTW dataset, primarily from ingredient lists on product packaging. The deformations in these samples are typically caused by surface distortions of the products or issues with the shooting angle, resulting in typical nonlinear deformation characteristics (as shown in Figure 7). During the selection process, efforts were made to ensure the representativeness of the samples, with some images cropped and cleaned to meet the experimental requirements. The introduction of the WTW-curved dataset was crucial for evaluating the adaptability

of the LocRecNet algorithm in real-world scenarios. Compared to algorithmically generated deformed data, these samples more accurately reflect the algorithm's performance in handling complex and irregular deformations. Experiments conducted on this dataset effectively analyzed the robustness and correction capabilities of LocRecNet in real-world settings, providing reliable evidence for its practical application.



| SENT | POS tagging information | | | | |
|---|---|---|---|---|---|
| | adj | verb | idiom | noun | other |
| pos | 1,230 | 734 | 1,026 | 266 | 642 |
| neg | 785 | 904 | 746 | 165 | 797 |
| neu | 918 | 7,569 | 2,016 | 12,668 | 10,214 |
| sum | 2,933 | 9,207 | 3,788 | 13,099 | 11,653 |

(**a**) (**b**) (**c**)

| iteration number | expected depth E[D] |
|---|---|
| E0 | 3.35 |
| E1 | 3.46 |
| E2 | 3.45 |

(**d**) (**e**) (**f**)

**Figure 6.** Datacomparison chart. (**a**) Original images; (**b**) Bezier transformation; (**c**) perspective transformation; (**d**) original images; (**e**) Bezier transformation; (**f**) perspective transformation on (**e**).


(**a**) (**b**) (**c**)

**Figure 7.** WTW-curved data. (**a**–**c**) are WTW-curved example images.

### *4.3. Evaluation Metrics*

Firstly, in assessing the performance of keypoint detection, we primarily use two metrics: average precision (AP) and average recall (AR). AP is a critical metric for measuring the accuracy of model predictions, calculated as the ratio of correctly predicted keypoints to the total number of keypoints predicted, demonstrating the precision of the model in predicting keypoints. AR measures the model's completeness, evaluated by the average recall rates. Recall itself is defined as the ratio of correctly predicted keypoints to the total number of actual keypoints, reflecting the model's ability to correctly identify keypoints. The calculations for AP and AR depend on Object Keypoint Similarity (OKS), which is based on the Euclidean distance between the predicted keypoints and their true coordinates. OKS is defined as follows:

$$OKS = \frac{\sum_i \exp\left(-\frac{d_i^2}{2s^2\hat{l}_i^2}\right)\sigma(v_i > 0)}{\sum_i \sigma(v_i > 0)} \tag{12}$$

where $d_i$ represents the Euclidean distance between the i-th predicted keypoint and its true coordinates, $s$ denotes the scale of the target, $\sigma$ is a fixed standard deviation, and $v_i$ indicates the visibility flag of the keypoint. Both AP and AR are calculated as the averages over multiple OKS thresholds, providing a comprehensive evaluation of model performance. This method of calculation ensures the completeness and accuracy of the evaluation results.

Secondly, for the determination of metrics for evaluating table structure, we chose the three most used metrics in table structure recognition work: precision, recall, and F1 score. Precision is defined as the ratio of the number of samples correctly identified as positive by the model to the total number of samples identified as positive by the model. It measures how many of the samples predicted as positive are truly positive. Recall is the ratio of the number of samples correctly identified as positive by the model to the total number of actual positive samples in the dataset. It assesses the model's ability to identify all positive samples. The F1 score is the harmonic mean of precision and recall, considering the performance of both. The F1 score ranges between 0 and 1, with values closer to 1 indicating better model performance. The specific calculation formulas are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{15}$$

where TP stands for true positives, FP stands for false positives, and FN stands for false negatives.

### *4.4. Experimental Results and Analysis*

#### 4.4.1. Performance Evaluation of LocRecNet

To validate the effectiveness of LocRecNet, we tested the LORE and LGPMA algorithms on the SCITSR-curved, PubTabNet-curved, and WTW-curved datasets, with the metrics presented in Table 1. The experimental results indicate that the inclusion of LocRecNet led to performance improvements for both algorithms across the three datasets.

On the SCITSR-curved and PubTabNet-curved datasets, LocRecNet significantly enhanced recognition performance. For the SCITSR-curved dataset, the recall rate of both algorithms increased by approximately 14% after integrating LocRecNet, with the F1 scores nearing 90%. On the PubTabNet-curved dataset, although the overall improvement was slightly lower than on the SCITSR-curved dataset, LocRecNet still yielded notable gains. In particular, the recall rate of the LGPMA algorithm improved by 10.5%, with an increase of 6.4% in the F1 score. These gains on deformed datasets highlight the substantial benefit of LocRecNet in precisely localizing table boundaries and structures, effectively enhancing both localization and recognition capabilities.

In the real-world scenario tests on the WTW-curved dataset, we first evaluated the LORE algorithm using its pre-trained model. The results showed that, after integrating LocRecNet, accuracy increased by 4%, and recall improved by 1.3%, with the F1 score reaching 97.9%. As for the LGPMA algorithm, which lacked a pre-trained model on the WTW dataset, the pre-trained model from PubTabNet was used for training. However, the experimental results revealed that the direct recognition performance of LGPMA was relatively weak, with an accuracy of 52.0% and an F1 score of only 64.1%. After integrating LocRecNet, the recall rate of LGPMA increased by just 5.3%, but accuracy surged by 36.5%, and the F1 score improved by 24.4%. The primary contribution of LocRecNet is its ability to effectively eliminate extraneous information outside the table, thereby preserving the table's key content. The results presented in Figure 8 show that the data processed by LocRecNet

were more streamlined, with the table's primary structure clearly retained, significantly reducing the difficulty of model recognition and enhancing overall performance.

**Table 1.** Performance comparison of LORE and LGPMA methods with/without LocRecNet.

| Method | Data | With/Without LocRecNet | P | R | F1 |
|---|---|---|---|---|---|
| LORE | SCITSR-curved | ×/✓ | 93.8% | 74.3% | 82.9% |
| | SCITSR-curved | ✓/× | 92.7% | 88.4% | 90.5% |
| | PubTabNet-curved | ×/✓ | 96.5% | 83.3% | 89.4% |
| | PubTabNet-curved | ✓/× | 97.2% | 86.7% | 91.6% |
| | WTW-curved | ×/✓ | 94.5% | 95.9% | 95.1% |
| | WTW-curved | ✓/× | 98.5% | 97.2% | 97.9% |
| LGPMA | SCITSR-curved | ×/✓ | 92.4% | 67.7% | 78.1% |
| | SCITSR-curved | ✓/× | 93.6% | 85.1% | 89.1% |
| | PubTabNet-curved | ×/✓ | 96.2% | 76.2% | 85.1% |
| | PubTabNet-curved | ✓/× | 96.8% | 86.7% | 91.5% |
| | WTW-curved | ×/✓ | 52.0% | 83.3% | 64.1% |
| | WTW-curved | ✓/× | 88.5% | 88.6% | 88.5% |


(**a**) (**b**)

(**c**) (**d**)

**Figure 8.** Results of LocRecNet on the WTW-curved dataset. (**a**) Original images; (**b**) LocRecNet result; (**c**) original images; (**d**) LocRecNet result.

#### 4.4.2. Computational Cost Analysis of LocRecNet

To evaluate the impact of introducing LocRecNet on computational cost, detailed tests are conducted on the performance of the keypoint detection and correction modules. The experimental results, summarized in Table 2, reveal notable differences in the processing time across datasets. These variations are hypothesized to stem from the differing characteristics of the table images in each dataset, which directly influence the localization and correction processes. Specifically, the WTW dataset, comprising primarily ingredient lists on product packaging, contains table images with complex color and background information, resulting in relatively longer processing times. In contrast, the SCITSR and PubTabNet datasets feature more structured and simpler table images, significantly reducing LocRecNet's processing time. For example, the average processing time for standard table images is 0.03 s, whereas processing a single table image from the WTW dataset re-

quires 0.2 s. While this processing time is relatively higher, it remains within an acceptable range. Notably, despite the slightly longer initial processing time for the WTW dataset, LocRecNet substantially enhances model performance in subsequent table recognition tasks, particularly in terms of recognition accuracy and stability. From a holistic perspective, the additional computational overhead introduced by LocRecNet is both reasonable and acceptable. Given the significant improvement in table recognition accuracy, the advantages of LocRecNet's performance are particularly compelling.

**Table 2.** LocRecNet's processing time.

| | SCITSR | PubTabNet | WTW |
|---|---|---|---|
| Table edge point localization | 0.008 s | 0.006 s | 0.069 |
| Image correction | 0.016 s | 0.029 s | 0.109 |

#### 4.4.3. Visualization of Results

To intuitively demonstrate the correction effectiveness of LocRecNet, representative images from different datasets were selected for a result comparison. The first row shows the recognition results on the original images, while the second row presents the results after table correction using LocRecNet. As illustrated in Figures 9 and 10, LocRecNet effectively rectifies table deformation in the SCITSR-curved and PubTabNet-curved datasets, significantly improving recognition accuracy. Additionally, the results on the WTW-curved dataset (Figure 11) further validate the advantages of LocRecNet, particularly in table edge point localization, where it demonstrates outstanding performance on complex and distorted tables. In the comparison images, the second column shows the recognition results of the LGPMA algorithm, while the third column displays those of LORE. These visual comparisons clearly highlight the superiority of LocRecNet in table structure recovery tasks.

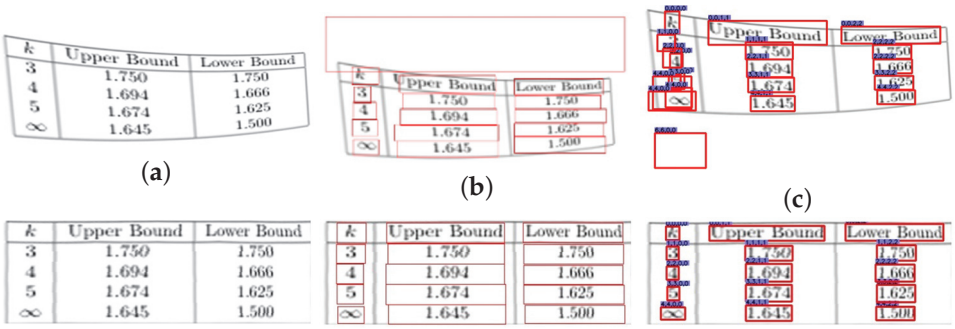



**Figure 9.** SCITSR-curved recognition results. (**a**) SCITSR-curved data; (**b**) LGPMA result; (**c**) LORE result; (**d**) LocRecNet result; (**e**) LGPMA result; (**f**) LORE result.

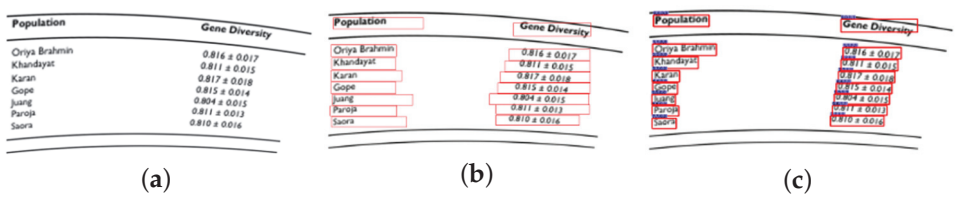
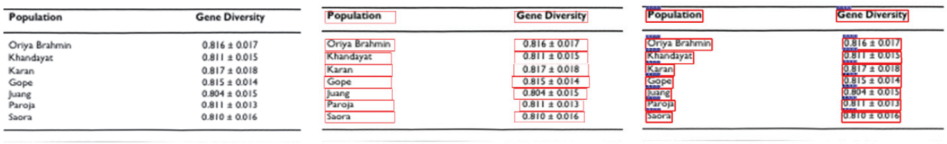
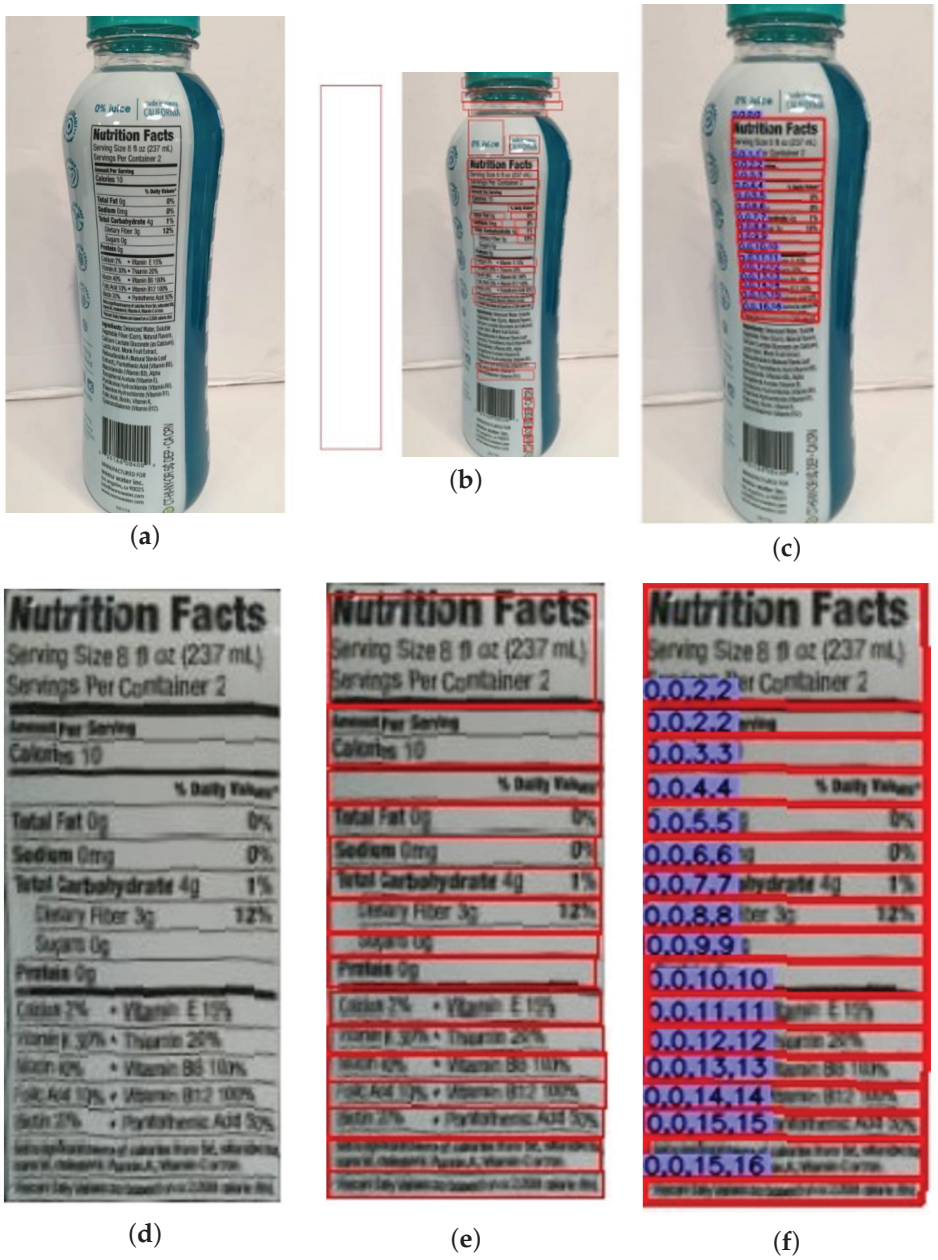



**Figure 10.** *Cont*.




(**d**) (**e**) (**f**)

**Figure 10.** PubTabNet-curvedrecognition results. (**a**) PubTabNet-curved data; (**b**) LGPMA result; (**c**) LORE result; (**d**) LocRecNet result; (**e**) LGPMA result; (**f**) LORE result.




(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

**Figure 11.** WTW-curved recognition results. (**a**) WTW-curved data; (**b**) LGPMA result; (**c**) LORE result; (**d**) LocRecNet result; (**e**) LGPMA result; (**f**) LORE result.

#### 4.4.4. Overall Performance

The experimental results demonstrate that LocRecNet significantly enhances the performance of two state-of-the-art table structure recognition algorithms, LORE and LGPMA, across three benchmark datasets featuring distorted tables—the SCITSR-curved, PubTabNet-curved, and WTW-curved datasets. Notably, LocRecNet exhibits a strong robustness and high recognition accuracy in scenarios involving structurally complex

and heavily deformed table images. By simultaneously improving table localization and structure reconstruction, the proposed method advances the overall recognition pipeline. Although the integration of LocRecNet introduces additional computational overhead—particularly a longer processing time for high-complexity samples in the WTW dataset—the trade-off is well justified by the substantial improvements in accuracy and result stability, underscoring the practical utility of the approach.

Nevertheless, despite its effectiveness, LocRecNet presents certain limitations under specific conditions. In particular, an experimental analysis reveals that the localization module may produce inaccurate predictions for tables lacking upper and lower boundary lines, thereby diminishing the effectiveness of the geometric correction stage. The absence of clear boundary cues in these borderless tables impairs the model's ability to accurately detect table contours, often resulting in cell segmentation errors or abnormal cell merging. As illustrated in Figure 12, such issues are evident: Figure 12a displays the manually annotated keypoints of the original table, while Figure 12b shows the predicted keypoints generated by LocRecNet, revealing noticeable positional deviations. These deviations propagate to the correction module, ultimately leading to a rectification output with visible residual distortions and structural misalignment, as seen in Figure 12c. These findings indicate that, in complex scenarios characterized by both ambiguous layouts and severe deformations, the current method still encounters challenges in boundary perception and geometric modeling. Future work will explore the integration of semantic guidance and visual attention mechanisms to enhance the model's adaptability and generalization capabilities in handling structurally ambiguous, borderless tables.

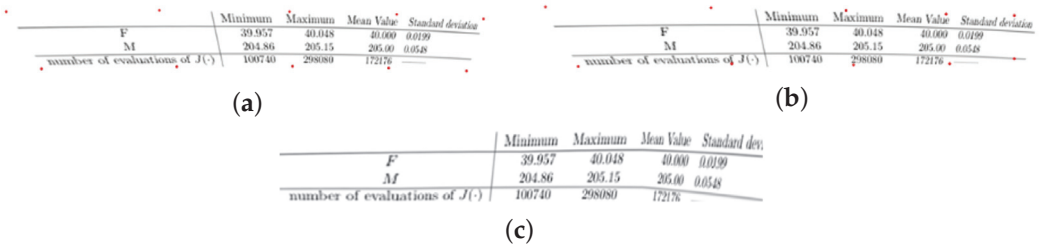



**Figure 12.** LocRecNet limitations on special tables. (**a**) Original table keypoint annotations; (**b**) predicted keypoints by LocRecNet; (**c**) corrected result by LocRecNet.

### *4.5. Ablation Study*

#### 4.5.1. LocRecNet Table Edge Point Localization

As a critical component of LocRecNet, the accuracy of table edge point localization directly affects the effectiveness of subsequent correction operations. To determine the optimal keypoint detection network for table images, we conducted multiple rounds of ablation experiments. These experiments evaluated the impact of various model architectures and algorithmic approaches on overall performance. To maintain fairness and scientific rigor, all experiments were conducted under identical training parameters for a consistent comparison.

Baseline Algorithm Selection Experiment: We explored the impact of different algorithmic approaches on model performance, focusing on a comparison between the SimCC method and the traditional Heatmap method. As shown in Table 3, we found that the SimCC method consistently outperformed the traditional Heatmap approach in terms of performance, regardless of the backbone network. Furthermore, after applying Gaussian label smoothing [21] at the final SimCC output stage, the model's performance improved significantly, especially in recognizing complex table structures. This validates the SimCC* method's effectiveness for keypoint detection and offers clear guidance for further algorithmic optimization.

Input Size Selection Experiment: To determine the optimal input size, we compared two resolutions: 256 × 192 and 384 × 288. As shown in Table 3, with the same backbone and algorithm, the model achieved higher AP and AR values with an input size of 384 × 288. We speculate that the larger input size offers a broader field of view, allowing the network to capture more contextual information. However, despite the performance improvements associated with the 384 × 288 input size, we also observed an increase in computational cost and model complexity. This may pose challenges for deployment and inference efficiency in resource-constrained applications.

Backbone Network Selection Experiment: In selecting the backbone network, we evaluated various models, including HRNet and ResNet [22]. We conducted extensive experiments under different algorithmic approaches, ensuring that each comparison was performed on the same baseline. The results indicated that HRNet consistently outperformed others regardless of the algorithm, demonstrating its superior performance in keypoint detection and establishing it as the ideal choice for our research. To mitigate the increased computational costs and model complexity associated with the larger input size, we optimized the HRNet structure by removing the fourth stage and retaining only the first three stages (referred to as HRNet-s) to simplify the model architecture. As shown in Tables 3 and A1, under the same input size and algorithm, the optimized HRNet exhibited a minimal decrease in AP and AR values, while the parameter count was reduced to 25% of the original, significantly decreasing the model size. This adjustment significantly reduced model complexity while maintaining acceptable performance, improving deployment efficiency and inference speed in practical applications.

**Table 3.** Table recognition metrics.

| Method | Representation | Input Size | AP | AR |
|---|---|---|---|---|
| HRNet | Heatmap | 256 × 192 | 77.3% | 79.7% |
| | | 384 × 288 | 82.5% | 84.2% |
| | SimCC | 256 × 192 | 84.0% | 86.9% |
| | SimCC* | 256 × 192 | 85.3% | 87.1% |
| | | 384 × 288 | 87.1% | 88.6% |
| HRNet-s | SimCC* | 256 × 192 | 83.4% | 85.1% |
| | | 384 × 288 | 86.1% | 87.4% |
| Res50 | Heatmap | 256 × 192 | 75.1% | 77.8% |
| | | 384 × 288 | 80.3% | 82.3% |
| | SimCC | 256 × 192 | 75.3% | 82.1% |
| | | 384 × 288 | 79.7% | 84.4% |
| | SimCC* | 384 × 288 | 85.0% | 87.1% |
| Res101 | Heatmap | 256 × 192 | 68.9% | 72.7% |
| | | 384 × 288 | 76.5% | 79.0% |
| | SimCC | 256 × 192 | 75.7% | 82.5% |
| | | 384 × 288 | 81.8% | 85.5% |
| Res152 | Heatmap | 256 × 192 | 75.8% | 78.8% |
| | | 384 × 288 | 81.4% | 83.3% |
| | SimCC | 384 × 288 | 81.2% | 85.4% |

SimCC* add Gaussian label smoothing before output.

#### 4.5.2. Impact of LocRecNet on Standard Table Data

To determine whether our designed LocRecNet impacts the recognition performance of standard tables, we conducted experiments on the SCITSR and PubTabNet datasets,

with the results presented in Table 4. We found that, for both algorithms, the addition of LocRecNet did not change the metrics for the PubTabNet dataset compared to direct recognition. However, on the SCITSR dataset, we observed that, while the accuracy of the LORE algorithm decreased by 0.2% after adding LocRecNet, both the recall and F1 scores increased by 0.7% and 0.2%, respectively. In the case of the LGPMA algorithm, the improvement was more pronounced, with no change in accuracy but increases of 0.7% and 0.4% in the recall and F1 scores, respectively. Overall, these experimental results indicate that the incorporation of the LocRecNet network leads to a noticeable improvement in recognition performance on standard table data, particularly reflected in the recall metrics.

**Table 4.** Performance comparison of LORE and LGPMA methods with/without LocRecNet.

| Method | Data | With/Without LocRecNet | P | R | F1 |
|---|---|---|---|---|---|
| LORE | SCITSR | ×/✓ | 94.3% | 90.9% | 92.6% |
| | SCITSR | ✓/× | 94.1% | 91.6% | 92.8% |
| | PubTabNet | ×/✓ | 97.9% | 88.2% | 92.8% |
| | PubTabNet | ✓/× | 97.9% | 88.2% | 92.8% |
| LGPMA | SCITSR | ×/✓ | 93.8% | 84.5% | 88.9% |
| | SCITSR | ✓/× | 93.8% | 85.2% | 89.3% |
| | PubTabNet | ×/✓ | 97.6% | 87.5% | 92.3% |
| | PubTabNet | ✓/× | 97.6% | 87.5% | 92.3% |

We analyzed the reasons behind the improved recognition performance of the algorithms on the SCITSR dataset after LocRecNet processing. Observations of the images revealed a significant presence of both tables and captions, which introduced noise and complicated the recognition process (Figure 13a). However, after processing with LocRecNet, the images (Figure 13b) contained only the table content. This enhancement increased the model's accuracy in locating the target, reducing missed detections and enabling it to capture more positive samples.

| Noise level | Easy | Hard | Tough |
|---|---|---|---|
| SIFT | 0.545 | 0.264 | 0.134 |
| TFeat | 0.551 | 0.344 | 0.189 |
| DeepCompare | 0.527 | 0.309 | 0.170 |
| DeepDesc | 0.561 | 0.374 | 0.228 |
| HardNet | 0.725 | 0.572 | 0.385 |
| SKAR-EgoSeg | 0.595 | 0.319 | 0.169 |
| SKAR-EgoSeg* | 0.605 | 0.344 | 0.189 |
| SKAR-HPatches | 0.756 | 0.621 | 0.440 |
| SKAR-HPatches* | **0.770** | **0.645** | **0.464** |

Table 1: Mean average precision on the retrieval task of the Hpatches benchmark ($10^4$ experiemnts of retrieving 5 patches corresponding to a query among $2 \times 10^4$ distractors).

(**a**)

| Noise level | Easy | Hard | Tough |
|---|---|---|---|
| SIFT | 0.545 | 0.264 | 0.134 |
| TFeat | 0.551 | 0.344 | 0.189 |
| DeepCompare | 0.527 | 0.309 | 0.170 |
| DeepDesc | 0.561 | 0.374 | 0.228 |
| HardNet | 0.725 | 0.572 | 0.385 |
| SKAR-EgoSeg | 0.595 | 0.319 | 0.169 |
| SKAR-EgoSeg* | 0.605 | 0.344 | 0.189 |
| SKAR-HPatches | 0.756 | 0.621 | 0.440 |
| SKAR-HPatches* | **0.770** | **0.645** | **0.464** |

(**b**)

**Figure 13.** LocRecNet processing results on SCITSR-curved data. (**a**) Original images; (**b**) LocRecNet result.

## 5. Conclusions

This paper proposes a novel table structure recognition framework, LocRecNet, designed to address the challenges posed by complex table images exhibiting geometric distortions. Centered on deformation correction, LocRecNet integrates keypoint detection and geometric rectification into a unified pipeline, enabling robust structure restoration, even under severe deformations or ambiguous layouts.

Extensive experiments conducted on three challenging benchmark datasets—the SCITSR-curved, PubTabNet-curved, and WTW-curved datasets—demonstrate that LocRecNet significantly outperforms state-of-the-art methods such as LORE and LGPMA in terms of both the recognition accuracy and F1 score. Notably, LocRecNet shows strong robustness and generalization capabilities when dealing with structurally complex or semantically

dense tables, validating its applicability in real-world scenarios. Although the introduction of LocRecNet results in a modest increase in computational cost when processing highly complex samples, the corresponding improvements in accuracy and output stability justify this trade-off. Moreover, an analysis of failure cases highlights areas for further enhancement, particularly in handling borderless tables and improving boundary perception under layout ambiguity.

In summary, LocRecNet achieves remarkable performance in the recognition of deformed table structures and introduces a new technical paradigm that combines deformation awareness with correction-driven design. This work lays a foundation for the development of more flexible, semantically guided table analysis systems and offers valuable insights for future research and practical deployment in the field of document image processing.

**Author Contributions:** Conceptualization, Z.C. and J.F.; methodology, Z.C. and J.F.; software, Z.C.; validation, Z.C. and Z.H.; formal analysis, Z.C.; investigation, Z.C.; resources, J.F., H.Z. and H.M.; data curation, Z.H.; writing—original draft preparation, Z.C.; writing—review and editing, J.F. and H.Z.; visualization, Z.H.; supervision, H.M. and H.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data that support the findings of this study are available from the corresponding author upon reasonable request.

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

## Appendix A

The parameter magnitude of the keypoint detection network backbone.

**Table A1.** Model parameter scale.

| Model | Parameter Magnitude |
|---|---|
| HRNet | ≈48.6 M |
| HRNet-s | ≈11.2 M |
| Res50 | ≈25.6 M |
| Res101 | ≈44.5 M |
| Res152 | ≈60.3 M |

## References

1. Hu, J.; Kashi, R.S.; Lopresti, D.P.; Wilfong, G. Table structure recognition and its evaluation. In *Document Recognition and Retrieval VIII, Proceedings of the 8th International Conference on Document Recognition and Retrieval, San Jose, CA, USA, 21 December 2000*; SPIE: Bellingham, WA, USA, 2000; pp. 44–55.
2. Deng, Y.; Rosenberg, D.; Mann, G. Challenges in end-to-end neural scientific table recognition. In Proceedings of the 16th International Conference on Document Analysis and Recognition (ICDAR), Sydney, NSW, Australia, 20–25 September 2019; IEEE: New York, NY, USA, 2019; pp. 894–901.
3. Alexiou, M.S.; Bourbakis, N.G. Pinakas: A methodology for deep analysis of tables in technical documents. *Int. J. Artif. Intell. Tools* **2023**, *32*, 2350042. [CrossRef]
4. Göbel, M.; Hassan, T.; Oro, E.; Orsi, G. ICDAR 2013 table competition. In Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR), Washington, DC, USA, 25–28 August 2013; IEEE: New York, NY, USA, 2013; pp. 1449–1453.
5. Desai, H.; Kayal, P.; Singh, M. TabLeX: A benchmark dataset for structure and content information extraction from scientific tables. In Proceedings of the 16th International Conference on Document Analysis and Recognition (ICDAR), Lausanne, Switzerland, 5–10 September 2021; Part II; Springer: Cham, Switzerland, 2021; pp. 554–569.

6. Zhong, X.; ShafieiBavani, E.; Jimeno Yepes, A. Image-based table recognition: Data, model, and evaluation. In *Computer Vision—ECCV 2020, Proceedings of the 16th European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer: Cham, Switzerland, 2020; pp. 569–585.
7. Long, R.; Wang, W.; Xue, N.; Gao, F.; Yang, Z.; Wang, Y.; Xia, G.S. Parsing table structures in the wild. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; IEEE: New York, NY, USA, 2021; pp. 944–952.
8. Qiao, L.; Li, Z.; Cheng, Z.; Zhang, P.; Pu, S.; Niu, Y.; Ren, W.; Tan, W.; Wu, F. LGPMA: Complicated table structure recognition with local and global pyramid mask alignment. In *Document Analysis and Recognition—ICDAR 2021, Proceedings of the 16th International Conference on Document Analysis and Recognition (ICDAR), Lausanne, Switzerland, 5–10 September 2021*; Lladós, J., Lopresti, D., Uchida, S., Eds.; Springer: Cham, Switzerland, 2021; pp. 67–73.
9. Liu, H.; Li, X.; Liu, B.; Jiang, D.; Liu, Y.; Ren, B.; Ji, R. Show, read and reason: Table structure recognition with flexible context aggregator. In Proceedings of the 29th ACM International Conference on Multimedia (MM '21), New York, NY, USA, 20–24 October 2021; ACM: New York, NY, USA, 2021; pp. 1084–1092.
10. Liu, H.; Li, X.; Liu, B.; Jiang, D.; Liu, Y.; Ren, B. Neural collaborative graph machines for table structure recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; IEEE: New York, NY, USA, 2022; pp. 4533–4542.
11. Xing, H.; Gao, F.; Long, R.; Bu, J.; Zheng, Q.; Li, L.; Yu, Z. LORE: Logical location regression network for table structure recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; AAAI Press: New York, NY, USA, 2023; Volume 37, pp. 2992–3000.
12. Zhang, Z.; Hu, P.; Ma, J.; Du, J.; Zhang, J.; Yin, B.; Liu, C. SEMv2: Table separation line detection based on instance segmentation. *Pattern Recognit.* **2024**, *149*, 110279. [CrossRef]
13. Huang, Y.; Yan, Q.; Li, Y.; Chen, Y.; Wang, X.; Gao, L.; Tang, Z. A YOLO-based table detection method. In Proceedings of the 16th International Conference on Document Analysis and Recognition (ICDAR), Sydney, NSW, Australia, 20–25 September 2019; IEEE: New York, NY, USA, 2019; pp. 813–818.
14. Li, Y.; Yang, S.; Liu, P.; Zhang, S.; Wang, Y.; Wang, Z.; Xia, S.-T. SimCC: A simple coordinate classification perspective for human pose estimation. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; Springer Nature Switzerland: Cham, Switzerland, 2022; pp. 89–106.
15. Yu, C.; Xiao, B.; Gao, C.; Yuan, L.; Zhang, L.; Sang, N.; Wang, J. Lite-hrnet: A lightweight high-resolution network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; IEEE: New York, NY, USA, 2021; pp. 10440–10450.
16. Yang, S.; Quan, Z.; Nie, M.; Yang, W. Transpose: Keypoint localization via transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; IEEE: New York, NY, USA, 2021; pp. 11802–11812.
17. Keller, W.; Borkowski, A. Thin plate spline interpolation. *J. Geod.* **2019**, *93*, 1251–1269. [CrossRef]
18. Wood, S.N. Thin plate regression splines. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2003**, *65*, 95–114. [CrossRef]
19. Prautzsch, H.; Boehm, W.; Paluszny, M. *Bézier and B-Spline Techniques*; Springer: Berlin/Heidelberg, Germany, 2002; Volume 6, pp. 25–41.
20. Jin, B.; Liu, Y.; Liu, D.; Qi, W.; Chen, Y.; Wang, S. Research on automatic correction of the document images based on perspective transformation. In Proceedings of the 2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI), Fuzhou, China, 24–26 September 2021; IEEE: New York, NY, USA, 2021; pp. 291–297.
21. Müller, R.; Kornblith, S.; Hinton, G.E. When does label smoothing help? *Adv. Neural Inf. Process. Syst.* **2019**. [CrossRef]
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: New York, NY, USA, 2016; pp. 770–778.

*electronics*

MDPI

*Article*

# Adapting a Previously Proposed Open-Set Recognition Method for Time-Series Data: A Biometric User Identification Case Study †

**András Pál Halász [1,*], Nawar Al Hemeary [1], Lóránt Szabolcs Daubner [1], János Juhász [1,2], Tamás Zsedrovits [1] and Kálmán Tornai [1]**

[1] Faculty of Information Technology and Bionics, Pázmány Péter Catholic University, 1083 Budapest, Hungary; al.hemeary@itk.ppke.hu (N.A.H.); daubner.lorant.szabolcs@itk.ppke.hu (L.S.D.); juhasz.janos@itk.ppke.hu (J.J.); zsedrovits.tamas@itk.ppke.hu (T.Z.); tornai.kalman@itk.ppke.hu (K.T.)
[2] Institute of Medical Microbiology, Semmelweis University, 1089 Budapest, Hungary
* Correspondence: halasz.andras@itk.ppke.hu
† This paper is an extended version of our paper published in Halász, A.; Daubner, L.; Al-Hemeary, N.; Juhász, J.; Zsedrovits, T.; Tornai, K. Adapting Open-Set Recognition Method to Various Time-Series Data. In Proceedings of the 19th International Conference on Web Information Systems and Technologies, Rome, Italy, 15–17 November 2023.

**Abstract:** Conventional classifiers are generally unable to identify samples from classes absent during the model's training. However, such samples frequently emerge in real-world scenarios, necessitating the extension of classifier capabilities. Open-Set Recognition (OSR) models are designed to address this challenge. Previously, we developed a robust OSR method that employs generated—"fake"—features to model the space of unknown classes encountered during deployment. Like most OSR models, this method was initially designed for image datasets. However, it is essential to extend OSR techniques to other data types, given their widespread use in practice. In this work, we adapt our model to time-series data while preserving its core efficiency advantage. Thanks to the model's modular design, only the feature extraction component required modification. We implemented three approaches: a one-dimensional convolutional network for accurate representation, a lightweight method based on predefined statistical features, and a frequency-domain neural network. Further, we evaluated combinations of these methods. Experiments on a biometric time-series dataset, used here as a case study, demonstrate that our model achieves excellent open-set detection and closed-set accuracy. Combining feature extraction strategies yields the best performance, while individual methods offer flexibility: CNNs deliver high accuracy, whereas handcrafted features enable resource-efficient deployment. This adaptability makes the proposed framework suitable for scenarios with varying computational constraints.

**Keywords:** open-set recognition; time-series analysis; biometric authentication

## 1. Introduction

Machine learning has made significant strides in various classification and recognition tasks, often surpassing human performance. For example, the current benchmark is a model that achieves an exceptionally low error rate of just 0.21% on the MNIST handwritten digit image dataset [1]. Although the field may appear to have overcome all challenges, these accomplishments are primarily confined to closed-set scenarios, where all classes are known during training. However, new classes can appear during real-world applications at the time of testing, necessitating models to make informed rejections in open-set scenarios.

Our previous research tackled this fundamental challenge by introducing an effective Open-Set Recognition (OSR) methodology [2]. The core of our approach was generating synthetic samples from actual data instances to represent the unknown space. A key finding was that training the model to identify and reject these artificial samples significantly enhances its ability to reject genuine unknown samples during testing.

Our approach diverges from traditional methods by generating synthetic features within a hidden layer rather than entirely new inputs. This strategy not only improves accuracy but also reduces computational overhead. The generative model for these features is simpler than the input layer, optimizing computational efficiency. Moreover, placing the synthetic samples in a hidden layer bypasses the initial model segments, saving substantial computational resources. Despite this departure, we still use Generative Adversarial Networks (GANs) [3], including refined and simplified generator and discriminator networks.

One application of OSR is authentication, because, besides recognizing known subjects, the model also needs to reject unknown subjects. The data serving as a base for classification is not necessarily in the form of images. Other types of data can also occur, for example, time series. We are unaware of any OSR model tailored for different data types besides images. Some existing methods appear to be easy to adapt for various kinds of data, but none have been applied to or tested on those. Hence, it is necessary to develop OSR models capable of processing time-series data.

Initially designed for image datasets using convolutional networks, our OSR model is highly adaptable to various data types. This adaptability is due to the model's modular architecture, specifically the feature extraction module located just before the hidden layer, where synthetic samples are generated. Once the necessary features are extracted, the generative and feature-classifier components work seamlessly together.

In this work, we adapt this model to classify multi-channel time-series data, focusing on biometric signals. We aim to accurately identify users based on the vibrational patterns of their hands, which are captured by accelerometer and gyroscope sensors in mobile phones [4]. Our preliminary results, published in an earlier paper, demonstrate promising outcomes using one-dimensional convolutional networks for feature extraction. The model also retains its favorable time complexity, a significant advantage of its original design [5].

The focus of this work is not on advancing biometric methods per se, but rather on the broader machine learning challenge of Open-Set Recognition (OSR)—enabling classifiers to detect and reject inputs from previously unseen classes reliably. The biometric dataset serves as a case study to demonstrate the model's adaptation to time-series data, but the methodology itself is domain-independent.

In this work, we conducted a comprehensive experiment, implementing and evaluating several feature extraction methods. We combined these methods into a single model, where the resulting feature vector is the concatenated output of the individual techniques. This approach enhances the model's performance, albeit at the cost of additional computational resources due to the simultaneous execution of multiple feature-extraction models.

This paper is structured as follows: We commence with an exhaustive literature review, providing a comprehensive background to contextualize our work. Next, we present an overview of the original OSR model. Following this, we discuss in detail how our model was adapted to accommodate the new data type, including data preprocessing and feature extraction methodologies. Finally, we present our results, analyzing the model's performance and cost-efficiency using various combinations of feature extraction methods.

## 2. Literature Review

In this section, we provide a succinct overview of the relevant literature. First, we introduce the concept of Open-Set Recognition, followed by an overview of the most

relevant existing solutions. Then, we introduce our previous solution to the problem, which serves as the basis for the work described in this article. Lastly, we present the dataset utilized to evaluate our model's performance.

### *2.1. Theory of Open-Set Recognition*

Many algorithms have long been used to solve classification tasks where only some samples belong to any known class [6,7] or the model lacks sufficient confidence [8,9] in classifying a sample. These models have addressed problems similar to OSR, but without laying down the theoretical background for it. OSR itself was finally formalized by Scheirer et al. [10]:

Let $O$ denote the open space (i.e., the space far from any known data). The open space risk is defined as follows:

$$R_O(f) = \frac{\int_O f(x)dx}{\int_{S_O} f(x)dx} \tag{1}$$

where $S_O$ denotes the space containing both the positive training examples and the positively labeled open space, and $f$ is the recognition function, with $f(x) = 1$ if the sample $x$ is recognized as a known class, and $f(x) = 0$ otherwise. Intuitively, $R_O(f)$ measures the proportion of the function's support that lies in open space, i.e., the extent to which the classifier incorrectly labels regions far from the training data as known.

**Definition 1.** *Open-Set Recognition Problem: Let $V$ be the set of training samples, $R_O$ the open-space risk, and $R_\epsilon$ the empirical risk (i.e., the closed-set classification risk, associated with misclassifications). Then, Open-Set Recognition is employed to find an $f \in H$, where $H$ is the hypothesis space of measurable recognition functions, such that $f(x) > 0$ indicates assignment to a known class, and $f$ minimizes the open-set risk $R_O$:*

$$\underset{f \in H}{arg\,min}\{R_O(f) + \lambda_r R_\epsilon(f(V))\} \tag{2}$$

*where $\lambda_r$ is a regularization parameter balancing open-space risk and empirical risk.*

**Definition 2.** *The openness of an Open-Set Recognition problem is defined as follows:*

$$O = 1 - \sqrt{\frac{2 \times |C_{TR}|}{|C_{TA}| + |C_{TE}|}} \tag{3}$$

*where $C_{TR}$, $C_{TA}$, and $C_{TE}$ denote the sets of training, target, and test classes, respectively.*

### *2.2. Existing Approaches*

After introducing the concept of Open-Set Recognition, Scheirer et al. [10] proposed the 1-vs-Set Machine as an initial solution. This specialized Support Vector Machine (SVM) model is designed to address open-set challenges. After training, the model incorporates a second hyperplane parallel to the first. Inputs classified between these hyperplanes are labeled as positive. The authors argue that by comparing the volume of a d-dimensional ball to the positively labeled slab inside it, the open-space risk of the model approaches zero as the ball's radius increases. Despite this approach, the positively labeled space remains unbounded.

In a subsequent study, ref. [11] applied the Radial Basis Function (RBF) kernel to the SVM model. They noted that the radial kernel function $K$ satisfies $\lim_{d(x,x')\to\infty} K(x, x') = 0$, where d(x, x′) represents the distance between feature vectors $x$ and $x'$. The study identified a negative bias term as a necessary and sufficient condition for ensuring a bounded

positively labeled open space. This was achieved by adding a regularization term for the bias in the objective function.

Bendale et al. proposed OpenMax [12], a method that adapts a neural network classifier trained initially in a closed-set scenario using a SoftMax layer. OpenMax modifies this setup to allow for the rejection of open-set samples.

Traditional SVMs and SoftMax classifiers are initially tailored for closed-set scenarios, where all classes are known during training. To effectively reject open-set samples, these models must be adapted, which involves rethinking how the input or feature space is divided using hyperplanes or other methods. Such adaptations require fundamentally different approaches to improve the results.

In Open-Set Recognition, the likelihood that a sample belongs to a known class can be estimated through the distribution of its distances to training data in the feature space.

Distance-based classifiers naturally align with the open-set framework. Besides identifying the most similar class for a given input, they yield a similarity score that can be thresholded to decide whether the input should be assigned to that class or rejected as unknown.

Ref. [13] adapted the nearest neighbor method for open-set conditions. To classify a sample $s$, its nearest neighbor $t$ is located, followed by another neighbor $u$ that belongs to a different class than $t$. The ratio $R = d(t,s)/d(u,s)$ is then compared against a threshold $T$: if $R < T$, the label of $t$ is assigned to $s$; otherwise, $s$ is rejected.

Miller et al. [14] proposed an alternative that uses predefined class centroids instead of pairwise sample comparisons. Their method projects inputs into a logit space and evaluates Euclidean distances between the input logit and each class mean for classification.

A central difficulty in open-set learning is the lack of negative (unknown) samples during training, as these appear only at the time of testing. Introducing synthetic data to approximate the distribution of unknowns can help mitigate this limitation.

Generative Adversarial Networks (GANs), introduced by Goodfellow et al. [3], provide a way to produce realistic artificial data. A GAN is composed of two competing models: a generator that maps noise to candidate samples, and a discriminator that attempts to separate real from generated instances.

Kong and Ramanan [15] proposed an OSR-specific GAN framework that conditions the generator on feature vectors to synthesize class-related data, thereby improving rejection of unknowns.

Other works, such as those by Jo et al. and Ge et al. [16,17], also employed GAN-based augmentation strategies for Open-Set Recognition.

Generative modeling has been successfully applied in biometric and physiological signal contexts as well, for tasks such as classification [18] as well as for data augmentation and validation [19,20].

Generating negative samples enhances the performance of an OSR model. However, their substantial additional computational overhead represents a significant drawback.

Most OSR models are primarily designed for processing images, highlighting the need for algorithms working on time-series data. Among the pioneers in this field were Tornai et al. [21], who, following the extraction of statistical features from data collected via smartphones' sensors, applied the $P_I - SVM$ [22] and EVM [23] models to time-series data. Their work demonstrated the feasibility of OSR on time series, albeit with room for improvement in the results.

The latter solution was similar to that employed in this work in that it utilized OSR on biometric data for authentication purposes. As Maiorana et al. discussed, another source of such biometric data can be the user's keystroke dynamics when typing various kinds of texts, e.g., PINs or passwords [24].

Wandji Piugie et al. [25] converted the time-series data of the HAR dataset [26] into images and processed the converted data with the standard convolutional network-based models. We argue that images are not inherently better suited for efficient feature extraction than time series; the reason for having better results on images is that models working on them received significantly more attention. Therefore, developing more time-series-optimized solutions remains a pressing need.

### *2.3. Previous Work*

We previously implemented an OSR method that employs a distance-based approach, diverging from the traditional softmax-based structure to better accommodate open-set scenarios. In this method, training is simplified into a quadratic regression by using fixed class centers. To prepare the model for future unknown inputs, synthetic samples were generated within a hidden feature layer rather than in the input space. The neural network was divided into two segments: the first segment extracts features from samples, with its output serving as the layer where synthetic features are generated. This configuration enabled the training procedure described in Algorithm 1.

Initially, both segments of the model were pre-trained as a unified network. Subsequently, the outputs from the pre-trained first segment were recorded and used as real inputs for training the generative model. The genuine features, along with those produced by the generative model, were then used to train the second segment of the network further. Figure 1 illustrates the overall model architecture.

**Algorithm 1** Procedure for training the model. Here, $N_1$: feature extractor; $N_2$: classifier; $N_G$: generator; $N_D$: discriminator; $Y$: fixed class centers.

**Require:** $X = (x_1, \ldots, x_n)$ training samples, numbers of iterations $n_1, n_2$
**Ensure:** $(N_1, N_2, Y)$ trained networks and fixed class centers
Initialize $N_1, N_2, N_G, N_D$ with random parameters; initialize class centers $Y = (y_1, \ldots, y_k)$
**for** $i = 1$ to $n_1$ **do** ▷ Train feature extractor and classifier
  **for** each batch $x_j \subset X$ **do**
    $out \leftarrow N_2(N_1(x_j))$
    $loss \leftarrow$ quadratic loss$(out, Y)$
    Update $N_1$ and $N_2$ with the gradient of $loss$
  **end for**
**end for**
$f_1(X) = (N_1(x_1), \ldots, N_1(x_n))$ ▷ Extract features
$(N_G, N_D) \leftarrow GAN(N_G, N_D, f_1(X))$ ▷ Train the generative model using $f_1(X)$ as real samples
$z \leftarrow$ random noise
$X_G \leftarrow N_G(z)$ ▷ Generate synthetic features
**for** $i = 1$ to $n_2$ **do** ▷ Refine classifier with generated samples
  **for** each batch $j$ **do**
    $out \leftarrow N_2(f_1(X)_j)$
    $loss \leftarrow$ quadratic loss$(out, Y)$
    $out \leftarrow N_2((X_G)_j)$
    $loss \leftarrow loss +$ quadratic loss$(out, Y)$
    Update $N_2$ with the gradient of $loss$
  **end for**
**end for**
**return** $(N_1, N_2, Y)$

The model outperformed most competing methods on commonly used image datasets. For instance, on CIFAR10, it achieved an open-set detection AUC of 0.839 and a closed-set accuracy of 0.914. Both values were the highest among the evaluated OSR algorithms,

with the closed-set accuracy trailing only behind that of highly optimized closed-set classifiers [2].

Like most OSR approaches, however, the original model was tailored for image data, using convolutional architectures optimized for spatial features. Direct application to time series or other modalities is not feasible due to fundamental differences in data structure. This limitation necessitated adapting the method to accommodate sequential data. Fortunately, the modular design of the model significantly simplifies this process: only the feature extraction component requires modification, while the remaining architecture—including the generative module and the distance-based classification mechanism—remains unchanged. This flexibility allows the core principles of the original method to be preserved across different data types.

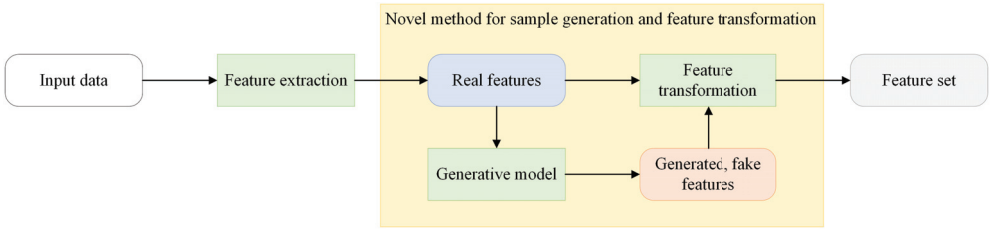



**Figure 1.** Schematic representation of the model. It remains effectively the same across various types of data–time series as well; only the feature extraction has to be modified to an appropriate model.

### *2.4. Dataset*

The primary motivation behind this study is to demonstrate the adaptability of our previously proposed OSR model to time-series data. To this end, we evaluate the method in a user identification setting based on hand-gesture signals, which serves as a biometric case study. A dedicated database tailored to this purpose is currently under development. The software that runs on smartphones and performs the measurements is complete; however, the necessary quantity of data has yet to be collected. Therefore, we conduct our tests on a publicly available dataset, that of Jiokeng et al., who devised a distinct biometric authentication system in a related effort. In their work, the basis of the classification is the subject's heart signal, detected through the vibration of the hand and measured by a phone held in the hand. The data, when collected in this manner, exhibits a poor signal-to-noise ratio, with the meaningful component being very faint. Still, with extensive preprocessing efforts, the authors achieved high-accuracy results, although only in a closed-set scenario with 112 users [4,27].

The dataset contains measurements from 112 participants (93 male, 19 female), aged between 20 and 60 years. Each user contributed 10 measurement sessions of 30 s each, recorded with a Samsung Galaxy S8 smartphone. This resulted in a total of 1120 raw recordings. The dataset's challenging properties—weak signals embedded in substantial noise—make it particularly well-suited for testing the robustness and adaptability of OSR approaches.

## 3. Adapting the Previously Proposed OSR Model to Time-Series Data

As shown in Figure 1 and described above, the model is composed of distinct components. The first part extracts suitable features for training the generative model. The second component classifies these features. Separately, a generative model produces negative inputs.

The latter two components operate on extracted features and are therefore independent of the input data type. As a result, adapting the model to time-series data requires modifying only the feature extraction component.

Three different feature extraction methods were implemented: a convolutional network with 1D convolutional layers reflecting the distinct nature of the data compared to images, predefined statistical features that do not need training, and a lightweight neural network operating in the frequency domain. Their combinations were also tested; multiple of these can be used in the same model, running in parallel. The outputs of the individual methods were then concatenated into the output of the model's first part. This, along with the individual methods, each with different complexities and levels of performance, gives plenty of room to calibrate the model, optimizing it more for either resources or performance.

### *3.1. Convolutional Network*

Convolutional Neural Networks (CNNs) have demonstrated exceptional performance in handling images due to their ability to process two-dimensional spatial data. When dealing with time-series data, which typically exists in a single dimension (not counting the channels of multivariate time series comparable to the channels of colored images), it is logical to utilize convolutional layers designed for one-dimensional data. Our research identified the architecture illustrated in Figure 2 as the most effective among the various configurations tested. This model comprises five stages, each doubling the number of channels from the previous stage. Each stage contains several 1D convolutional layers activated by the ReLU function, followed by a max pooling layer. The design of our network closely parallels the VGG19 network architecture [28]. Ultimately, the network generates a feature vector matching the size of those produced in our earlier image-based projects [2], ensuring consistency and compatibility with our existing methodologies. We have already implemented this method in [5]. In this work, the method is tested with slightly improved hyperparameters and compared to other methods.

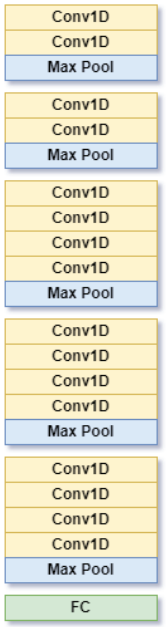



**Figure 2.** The structure of the convolutional network responsible for feature extraction.

### *3.2. Predefined Statistical Features*

Jiokeng and colleagues [4] demonstrated promising results with a Support Vector Machine (SVM) model that leveraged predefined features. Their findings suggest that these features efficiently encapsulate the essential information needed for the task. Consequently, a similar approach has been adopted in this study, with notable advantages. By employing

predefined features, we can bypass the initial phase of model training, requiring only a single extraction of features from each data element.

The selected features include statistical measures such as the mean, median, variance, and standard deviation, alongside other data aggregates like total signal energy, binned entropy, and permutation entropy. Additionally, ref. [4] employed Fourier transform, incorporating all Fourier coefficients and the complete set of auto-correlation and cross-correlation values between signal axes. However, incorporating all these coefficients resulted in their overwhelming dominance in the feature vector, effectively overshadowing the other aggregate features. Our final feature vector excluded these coefficients to maintain a balanced and representative feature set. The exhaustive list of features—covering statistical, entropy-based, and frequency-related descriptors—is provided in the Appendix A.

### *3.3. Neural Network in Frequency Domain*

In the frequency domain, the data can reveal significant insights that might not be immediately apparent. A notable benefit of this representation is that it does not necessitate using complex Convolutional Neural Networks (CNNs). Instead, more straightforward and cost-effective fully connected networks (FCNs) can be utilized. This is because translational invariance—a property where a slight shift in an image or time series does not change the input's essence—does not apply in the frequency domain. Each value in a specific position retains its unique significance regardless of any shift. Consequently, we employed a relatively small, Fully Connected Network to extract features from the Fast Fourier Transform (FFT) values effectively. Specifically, the FCN consists of three hidden layers of 512 neurons each, with ReLU activations, followed by a linear output layer of 256 neurons. The output of this final layer serves as the feature vector that is passed on to the OSR model.

Recurrent networks such as LSTMs [29] could also be considered for feature extraction from sequential data. They are particularly advantageous for modeling long-term dependencies or handling variable-length sequences. However, in our case, each input window is of fixed length and relatively short, where convolutional or fully connected approaches can already capture the relevant temporal or spectral patterns. LSTMs would therefore introduce additional computational cost without a clear performance benefit. For this reason, we reserve their exploration for future work.

#### Preprocessing

Each acquisition session was stored in a single file containing measurements from multiple sensors. For our experiments, only the accelerometer and gyroscope data were retained, as these form the basis of the classification task. The raw signals from these two sensors were sampled at irregular timestamps and with non-uniform sampling rates, which required resampling and synchronization. Following the procedure in [4], the data were resampled to a uniform frequency of 200 Hz, aligning the accelerometer and gyroscope streams. Since both sensors operate on three axes, the result was a synchronized six-channel time series.

To focus on the relevant signal components, a fourth-order Butterworth bandpass filter was applied with cutoff frequencies of 0.5 Hz and 30 Hz. The filtered signals were then segmented into windows of a 1.5 s duration with a stride of 0.05 s, which produces a large number of overlapping segments while ensuring sufficient variability for training. Importantly, the division into training and test sets was performed at the session level: two out of ten sessions per subject (20%) were assigned to the test set, and all segments from those sessions were included there. This ensures that overlapping windows never

span across training and test sets, preventing information leakage and maintaining strict separation between the two.

For each segment, three different representations were produced and stored together for unified access: the raw six-channel signal (for the CNN-based extractor), the Fourier transform (for the frequency-domain FCN), and the predefined statistical and entropy-based features (for the handcrafted extractor). The Fourier transform values were processed as real and imaginary coefficients, while the handcrafted features are detailed in Appendix A. All representations were normalized to zero mean and unit variance—per channel for the raw and FFT data, and per feature element for the predefined features.

## 4. Experimental Results

The model has been extensively tested on the dataset described in Section 2.4. We are unaware of any OSR experiments conducted on this dataset by others. Thus, only the different feature extraction methods and their combinations can be compared regarding the open-set detection performance. The closed-set performance (accuracy) can still be compared to that presented in [4]. The relevant hardware specifications were as follows:

- Intel(R) Core(TM) i7-9800X CPU @ 3.80 GHz;
- NVIDIA(R) GeForce RTX(TM) 2080 Ti;
- 128 GB RAM.

For each experiment, the set of classes was randomly divided into known and unknown groups. The training subset of the known classes was used for model training, while the corresponding test subset was used for evaluation. For the unknown classes, only the test samples were included, serving as negative examples during evaluation. The training subsets of the unknown classes were not used at all. This procedure was repeated independently in each run, ensuring that the unknown classes represented truly unseen categories at the time of testing.

### *4.1. Evaluation Metrics*

A suitable and commonly used metric for evaluating OSR methods is the F1 measure, which is capable of capturing both closed-set and open-set performance to some extent. However, this metric is highly sensitive to the model's calibration—for example, the confidence threshold above which a sample is considered to belong to a known class. Hence, we instead chose the two metrics described below.

**Closed-set accuracy:** This metric considers only the results for the positive (known) samples. It measures the percentage of correctly classified samples. The model's prediction is the class with the highest probability, regardless of whether this probability is above a threshold, or the sample would otherwise be classified as unknown. Since all classes in our dataset contribute an equal number of samples, overall accuracy is an appropriate and unbiased measure of closed-set performance. In the OSR setting, it is equally as important to ensure that known samples are not misclassified into other known classes as it is to detect unknowns. Reporting closed-set accuracy therefore complements the open-set detection metrics by indicating how well the model preserves discrimination among the enrolled classes.

**AUC**: The receiver operating characteristic (ROC) curve is obtained by plotting the true positive rate (sensitivity) against the false positive rate (1—specificity) at each relevant threshold setting. The area under this curve provides a calibration-free measure of open-set detection performance. While calculating it, the known samples are considered positive—regardless of their class—and the unknown samples are considered negative [30].

**EER**: The Equal Error Rate is the point on the ROC curve where the false acceptance rate (FAR) equals the false rejection rate (FRR). Like the AUC, it is derived from the ROC, but unlike the AUC, it corresponds to a specific operating point. While the AUC is the most widely used metric in OSR research, the EER is a standard measure in biometrics. Including it therefore allows us to position our results in the broader context of biometric evaluation.

The AUC provides a threshold-independent assessment of open-set detection capability, making it particularly suitable for comparing methods across different datasets and configurations. In practical biometric deployments, however, operating thresholds must be chosen to balance the rates of false acceptances and false rejections. This calibration can, for example, be performed by cross-class validation, as demonstrated in our previous work [2], where one testing scenario followed a protocol for outlier detection using the F1 measure. In that setup, the threshold was determined via cross-class validation under the assumption that both types of errors—misclassifying unknowns and rejecting known-class samples—carry equal weight.

### *4.2. Results*

The model was evaluated with several known-class configurations ranging from 10 to 60, using all non-empty subsets of the three feature extraction strategies introduced earlier. Table 1 reports the open-set detection performance. Among the single-method setups, the 1D convolutional network consistently delivers the strongest results, closely followed by predefined statistical features. While the frequency-domain model alone is not competitive as a standalone solution, it provides a measurable boost when combined with other methods. Therefore, it remains a valuable option in scenarios where computational resources are plentiful and performance is the top priority, making the performance improvement worth the added cost. In fact, incorporating multiple feature extraction methods generally leads to clear performance improvements. The number of known classes does not noticeably degrade performance, and the low standard deviation values confirm the robustness of the approach with respect to class selection. To complement the AUC results, Table 2 presents the corresponding EER values. Including the EER allows our results to be related more directly to the biometrics literature, where it is a widely used measure. As expected, the EER values are strongly and almost linearly correlated with the AUC, so they convey largely the same information about model performance. Nonetheless, their inclusion facilitates comparison with prior biometrics studies.

The closed-set accuracy results (Table 3) remain high across all configurations. Although marginally below the upper bound reported in the original HandBCG study [4] in a purely closed-set scenario (98.27% to 99%), our approach maintains strong accuracy while introducing the capability to reject unknown classes—an essential open-set property. The ranking of the methods remains consistent with the AUC results, although the differences in accuracy are smaller.

**Table 1.** Open-set detection AUC scores using different feature extraction methods and their combinations, averaged over five runs. Each configuration was evaluated using a varying number of known classes.

| **Known Classes** | | **10** | **20** | **30** | **40** | **50** | **60** |
|---|---|---|---|---|---|---|---|
| Convolutional network | Mean | 0.750 | 0.772 | 0.797 | 0.811 | 0.828 | 0.798 |
| | Std | 0.048 | 0.045 | 0.054 | 0.083 | 0.065 | 0.039 |
| Predefined features | Mean | 0.753 | 0.747 | 0.762 | 0.775 | 0.761 | 0.752 |
| | Std | 0.038 | 0.032 | 0.033 | 0.028 | 0.035 | 0.040 |
| Frequency domain | Mean | 0.702 | 0.617 | 0.701 | 0.732 | 0.652 | 0.688 |
| | Std | 0.072 | 0.103 | 0.065 | 0.073 | 0.054 | 0.059 |
| Conv. + Predef. | Mean | 0.801 | 0.824 | 0.829 | 0.794 | 0.794 | 0.811 |
| | Std | 0.042 | 0.025 | 0.034 | 0.020 | 0.036 | 0.064 |
| Conv. + FFT | Mean | 0.756 | 0.788 | 0.793 | 0.822 | 0.797 | 0.825 |
| | Std | 0.078 | 0.057 | 0.053 | 0.052 | 0.047 | 0.060 |
| Predef. + FFT | Mean | 0.770 | 0.788 | 0.779 | 0.753 | 0.744 | 0.718 |
| | Std | 0.079 | 0.054 | 0.063 | 0.049 | 0.057 | 0.132 |
| All three | Mean | 0.823 | 0.794 | 0.804 | 0.825 | 0.842 | 0.820 |
| | Std | 0.043 | 0.045 | 0.037 | 0.041 | 0.035 | 0.029 |

**Table 2.** Open-set detection performance in terms of EER score, measured from the same runs as in Table 1.

| **Known Classes** | | **10** | **20** | **30** | **40** | **50** | **60** |
|---|---|---|---|---|---|---|---|
| Convolutional network | Mean | 0.302 | 0.293 | 0.278 | 0.252 | 0.217 | 0.245 |
| | Std | 0.040 | 0.033 | 0.047 | 0.041 | 0.036 | 0.052 |
| Predefined features | Mean | 0.300 | 0.310 | 0.249 | 0.292 | 0.277 | 0.295 |
| | Std | 0.020 | 0.047 | 0.026 | 0.026 | 0.045 | 0.023 |
| Frequency domain | Mean | 0.338 | 0.388 | 0.345 | 0.293 | 0.385 | 0.349 |
| | Std | 0.087 | 0.022 | 0.045 | 0.026 | 0.057 | 0.013 |
| Conv. + Predef. | Mean | 0.228 | 0.203 | 0.209 | 0.257 | 0.252 | 0.258 |
| | Std | 0.025 | 0.017 | 0.020 | 0.057 | 0.046 | 0.054 |
| Conv. + FFT | Mean | 0.303 | 0.265 | 0.244 | 0.180 | 0.290 | 0.198 |
| | Std | 0.050 | 0.033 | 0.053 | 0.045 | 0.041 | 0.057 |
| Predef. + FFT | Mean | 0.308 | 0.250 | 0.249 | 0.308 | 0.588 | 0.332 |
| | Std | 0.053 | 0.038 | 0.031 | 0.035 | 0.109 | 0.047 |
| All three | Mean | 0.225 | 0.277 | 0.269 | 0.340 | 0.201 | 0.239 |
| | Std | 0.042 | 0.028 | 0.053 | 0.054 | 0.017 | 0.017 |

**Table 3.** Closed-set classification accuracy using different feature extraction setups. All results correspond to the same runs reported in Table 1.

| **Known Classes** | | **10** | **20** | **30** | **40** | **50** | **60** |
|---|---|---|---|---|---|---|---|
| Convolutional network | Mean | 0.950 | 0.943 | 0.906 | 0.910 | 0.924 | 0.900 |
| | Std | 0.027 | 0.028 | 0.025 | 0.032 | 0.024 | 0.030 |
| Predefined features | Mean | 0.941 | 0.927 | 0.932 | 0.913 | 0.915 | 0.890 |
| | Std | 0.018 | 0.022 | 0.024 | 0.028 | 0.024 | 0.028 |
| Frequency domain | Mean | 0.853 | 0.863 | 0.797 | 0.752 | 0.735 | 0.711 |
| | Std | 0.032 | 0.038 | 0.040 | 0.031 | 0.043 | 0.045 |
| Conv. + Predef. | Mean | 0.958 | 0.941 | 0.912 | 0.914 | 0.910 | 0.919 |
| | Std | 0.025 | 0.033 | 0.030 | 0.030 | 0.038 | 0.035 |
| Conv. + FFT | Mean | 0.959 | 0.927 | 0.909 | 0.918 | 0.890 | 0.900 |
| | Std | 0.041 | 0.046 | 0.035 | 0.043 | 0.050 | 0.037 |
| Predef. + FFT | Mean | 0.931 | 0.919 | 0.927 | 0.923 | 0.883 | 0.895 |
| | Std | 0.033 | 0.036 | 0.036 | 0.042 | 0.040 | 0.040 |
| All three | Mean | 0.955 | 0.950 | 0.922 | 0.894 | 0.922 | 0.915 |
| | Std | 0.029 | 0.041 | 0.038 | 0.029 | 0.029 | 0.036 |

Table 4 highlights the computational trade-offs between the examined methods during the first training phase. As expected, the convolutional network is the most resource-intensive option. However, the additional cost translates to a solid performance advantage. At the other end of the spectrum, predefined features entirely eliminate the first phase of training. Their extraction cost is negligible (performed once), and the subsequent model is extremely lightweight, making this setup ideal for constrained environments such as embedded systems. Frequency-domain features also offer a resource-efficient alternative, requiring significantly less computation than convolutional networks—though this comes at the price of reduced performance. In all cases, the flexibility of combining feature sets allows practitioners to balance performance and efficiency according to deployment constraints.

Overall, these results demonstrate a favorable performance–cost trade-off across all configurations, underlining the scalability of the proposed approach. A lightweight yet effective model can be built with predefined features for low-resource settings, while maximum accuracy can be achieved through convolutional networks or their combinations with other methods.

**Table 4.** Runtime performance using different feature extraction methods. The reported values (in ms) were measured during the first phase of training, except in the case of the predefined features, where the first phase is completely eliminated.

| Method | Feature Extraction (ms) | Whole Model (ms) | Training a Batch (ms) |
|---|---|---|---|
| Convolutional network | 2.1 | 2.5 | 17.0 |
| Predefined features | – | 0.23 | – |
| Frequency domain | 0.36 | 1.2 | 3.1 |
| All three combined | 3.3 | 4.0 | 23.0 |

A major advantage of our design lies in generating samples within the hidden layer. This strategy almost eliminates the overhead of producing additional training data while retaining the performance benefits of augmentation. The hidden layer's simpler structure requires only a lightweight generative model, avoiding the need to propagate generated samples through the entire network. Previous work [2] confirmed this as a key efficiency advantage, and our experiments reinforce this observation. On the time-series dataset, the generative model trains in only **5.2 ms** per batch—comparable to the image-based results due to identical feature vector sizes. Notably, there are no established GAN architectures specifically designed for time-series data, which makes input-level sample generation not only computationally expensive but also practically challenging. Our approach sidesteps this limitation by working in the feature space, eliminating the need to design or train a specialized generative model for raw sequences. Furthermore, skipping the heavy first stage of the model provides substantial runtime savings: for example, when using the convolutional network as a feature extractor, **91%** of the cost of processing a generated batch is eliminated (2.1 ms versus 0.23 ms). This efficiency, combined with strong open-set performance, makes the method not only practical but also uniquely suited for domains where data-level generative models are less feasible.

## 5. Conclusions

Our exploration of Open-Set Recognition (OSR) in time-series data has unveiled several critical insights with significant implications. OSR is a powerful enhancement of traditional classification techniques, proving particularly relevant in real-world applications. One of the most pressing areas of application is authentication, which serves as a valuable use case for our research.

Authentication systems must not only recognize legitimate users but also effectively identify unknown or unauthorized individuals. This is where OSR becomes indispensable, acting as a robust defense against potential security breaches and unauthorized access. Our research highlights the crucial role OSR can play in enhancing the security and reliability of authentication systems, emphasizing the urgency of integrating OSR into practical applications.

Additionally, our study identifies a significant gap in current research: the need for OSR methods specifically designed for time-series data. Our model demonstrates that it is feasible to adapt OSR techniques to time-series datasets, suggesting that this area will likely attract increasing research attention in the future.

In our experiments, we evaluated various feature extraction methods for OSR. The 1D convolutional network emerged as the most effective, although it incurred the highest computational cost. Predefined features offered a more lightweight alternative, delivering performance only slightly inferior to that of the convolutional network. Combining multiple methods further improved performance but also increased computational demands. This adaptability enables our approach to be tailored to diverse scenarios, allowing users to prioritize either accuracy or, in constrained environments such as embedded systems, optimize the model for computational efficiency while maintaining satisfactory performance.

Looking ahead, several promising research avenues emerge. First, further evaluation on additional biometric modalities, such as voice, ECG, or gait, could demonstrate the generality of our OSR framework beyond the current case study. Sequential architectures like LSTMs also represent a natural extension, as they can explicitly model long-term temporal dependencies; while unnecessary for the fixed-length short segments considered here, they could become valuable for other settings. Similarly, multimodal or lightweight sensor-fusion approaches may increase robustness, especially in mobile or embedded applications. An interesting direction is the application of the adapted OSR framework to continuous monitoring scenarios, where long-term non-stationarity of signals may be addressed.

Another direction concerns the variability of real-world conditions. While the present dataset already contains faint signals embedded in significant noise, making it a challenging benchmark, future work could explore robustness across different acquisition environments and sensor qualities. Adaptation strategies such as transfer learning or online learning may also help maintain performance under varying signal conditions or during long-term monitoring. Finally, threshold optimization and calibration strategies tailored to open-set detection should be studied more explicitly to bridge the gap between research evaluation and deployment in real authentication systems.

In parallel with these extensions, we are developing a new biometric database in which participants are classified based on hand gestures recorded while picking up their phones, using measurements from the device's sensors. This aims to support a continuous authentication system that operates passively, without requiring any explicit action from the user. Future research may also study how time-dependent changes in the signal—caused by shifts in user behavior or health conditions—affect classification performance.

In summary, our findings underscore the transformative potential of OSR in enhancing authentication systems, particularly when applied to time-series data. As we continue to refine these techniques, OSR is poised to become an increasingly vital component in developing secure and adaptable systems.

**Author Contributions:** Conceptualization: A.P.H. and K.T.; methodology: A.P.H.; formal analysis and investigation: A.P.H., L.S.D. and N.A.H.; writing—original draft preparation: A.P.H. and K.T.; writing—review and editing: A.P.H., K.T., L.S.D., N.A.H., J.J. and T.Z.; funding acquisition: K.T. and T.Z.; resources: K.T. and T.Z.; supervision: K.T. and T.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the National Research, Development, and Innovation Office through the grant TKP2021-NVA-26.

**Data Availability Statement:** The dataset used for the experiments was created by Jiokeng et al. [27] and is publicly available at https://doi.org/10.5281/zenodo.5187910.

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

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| OSR | Open-Set Recognition |
| FFT | Fast Fourier Transform |
| SVM | Support Vector Machine |
| CNN | Convolutional Neural Network |
| FCN | Fully Connected Network |
| GAN | Generative Adversarial Network |
| RBF | Radial Basis Function |

## Appendix A. Full List of Predefined Features

Unless otherwise noted, each feature is computed independently for all six channels (three-axis accelerometer and three-axis gyroscope). The input is represented as a matrix $X \in \mathbb{R}^{n\times 6}$, where $n$ is the time dimension.

**Table A1.** Predefined statistical and spectral features extracted per channel.

| Feature | Definition or Notes |
|---|---|
| | *Time-Domain Statistics* |
| **Mean** | $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$ |
| **Median** | Median of $\{x_i\}_{i=1}^{n}$ |
| **Standard Deviation** | $\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2}$ |
| **Variance** | $\sigma^2$ |
| **Skewness** | $\frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^3}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2\right)^{3/2}}$ |
| **Kurtosis** | $\frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2\right)^{2}}$ |
| **Minimum** | $\min_i x_i$ |
| **Maximum** | $\max_i x_i$ |
| **Range** | $\max_i x_i - \min_i x_i$ |
| **RMS (Root-Mean Square)** | $\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$ |
| **Std-to-Range Ratio** | $\frac{\sigma}{\max_i x_i - \min_i x_i}$ |
| **Percentage Above Mean** | $\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(x_i > \mu)$ |
| **Percentage Above Median** | $\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(x_i > \text{median})$ |
| **Absolute Sum of Changes** | $\sum_{i=2}^{n}\lvert x_i - x_{i-1}\rvert$ |

**Table A1.** *Cont.*

| Feature | Definition or Notes |
|---|---|
| | *Entropy and Energy Features* |
| **Total Energy** | $\sum_{i=1}^{n} x_i^2$ |
| **Approximate Entropy** | Standard definition applied per channel |
| **Binned Entropy** | $-\sum_j p_j \log p_j$, with $p_j$ = histogram bin probability |
| **Permutation Entropy** | Ordinal-pattern-based entropy measure |
| | *Spectral and Correlation Features* |
| **FFT Statistics** | Computed on magnitude spectrum; mean, median, variance, and standard deviation |
| **Autocorrelation Statistics** | Based on positive-lag coefficients $\mathrm{ACF}(k) = \sum_{i=1}^{n-k} x_i x_{i+k}$; mean, median, variance, and standard deviation |
| **Cross-Correlation Statistics** | Between axis pairs (XY, XZ, YZ): $\mathrm{CCF}_{a,b}(k) = \sum_{i=1}^{n-k} x_{a,i} x_{b,i+k}$; aggregated mean, median, variance, and standard deviation |

## References


1. Wan, L.; Zeiler, M.; Zhang, S.; Cun, Y.L.; Fergus, R. Regularization of Neural Networks using DropConnect. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 17–19 June 2013; Dasgupta, S., McAllester, D., Eds.; Volume 28, pp. 1058–1066.
2. Halász, A.P.; Al Hemeary, N.; Daubner, L.S.; Zsedrovits, T.; Tornai, K. Improving the Performance of Open-Set Recognition with Generated Fake Data. *Electronics* **2023**, *12*, 1311. [CrossRef]
3. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS'14), Montreal, QC, Canada, 8–13 December 2014; Volume 2, pp. 2672–2680.
4. Jiokeng, K.; Jakllari, G.; Beylot, A.L. I Want to Know Your Hand: Authentication on Commodity Mobile Phones Based on Your Hand's Vibrations. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2022**, *6*, 1–27. [CrossRef]
5. Halász, A.; Daubner, L.; Al-Hemeary, N.; Juhász, J.; Zsedrovits, T.; Tornai, K. Adapting Open-Set Recognition Method to Various Time-Series Data. In Proceedings of the 19th International Conference on Web Information Systems and Technologies-DMMLACS, Rome, Italy, 15–17 November 2023; INSTICC, SciTePress: Setúbal, Portugal, 2023; pp. 595–601. [CrossRef]
6. Bodesheim, P.; Freytag, A.; Rodner, E.; Denzler, J. Local Novelty Detection in Multi-class Recognition Problems. In Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 5–9 January 2015. [CrossRef]
7. Nguyen, A.; Yosinski, J.; Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 427–436. [CrossRef]
8. Fumera, G.; Roli, F. Support Vector Machines with Embedded Reject Option. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2002. [CrossRef]
9. Grandvalet, Y.; Rakotomamonjy, A.; Keshet, J.; Canu, S. Support Vector Machines with a Reject Option. *Adv. Neural Inf. Process. Syst.* **2008**, *21*, 537–544.
10. Scheirer, W.J.; Rocha, A.; Sapkota, A.; Boult, T.E. Toward Open Set Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1757–1772. [CrossRef] [PubMed]
11. Júnior, P.; Wainer, J.; Rocha, A. Specialized Support Vector Machines for open-set recognition. *arXiv* **2016**, arXiv:1606.03802. [CrossRef]
12. Bendale, A.; Boult, T. Towards Open Set Deep Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1563–1572. [CrossRef]
13. Júnior, P.; Souza, R.; Werneck, R.; Stein, B.; Pazinato, D.; Almeida, W.; Penatti, O.; Torres, R.; Rocha, A. Nearest neighbors distance ratio open-set classifier. *Mach. Learn.* **2017**, *106*, 1–28. [CrossRef]
14. Miller, D.; Sunderhauf, N.; Milford, M.; Dayoub, F. Class Anchor Clustering: A Loss for Distance-Based Open Set Recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Virtual, 5–9 January 2021; pp. 3570–3578.

15. Kong, S.; Ramanan, D. OpenGAN: Open-Set Recognition via Open Data Generation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 3233–3243. [CrossRef] [PubMed]
16. Jo, I.; Kim, J.; Kang, H.; Kim, Y.D.; Choi, S. Open Set Recognition by Regularising Classifier with Fake Data Generated by Generative Adversarial Networks. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 2686–2690. [CrossRef]
17. Zongyuan Ge, S.D.; Garnavi, R. Generative OpenMax for Multi-Class Open Set Classification. In *Proceedings of the British Machine Vision Conference (BMVC)*; Tae-Kyun, K., Stefanos Zafeiriou, G.B., Mikolajczyk, K., Eds.; BMVA Press: Durham, UK, 2017; pp. 42.1–42.12. [CrossRef]
18. Alzantot, M.; Garcia, L.; Srivastava, M. PhysioGAN: Training High Fidelity Generative Model for Physiological Sensor Readings. *arXiv* **2022**, arXiv.2204.13597. [CrossRef]
19. Li, X.; Metsis, V.; Wang, H.; Ngu, A.H.H. TTS-GAN: A Transformer-based Time-Series Generative Adversarial Network. In Proceedings of the Artificial Intelligence in Medicine: 20th International Conference on Artificial Intelligence in Medicine, AIME 2022, Halifax, NS, Canada, 14–17 June 2022.
20. Laganá, F.; Pellicanò, D.; Arruzzo, M.; Pratticò, D.; Pullano, S.; Fiorillo, A. FEM-Based Modelling and AI-Enhanced Monitoring System for Upper Limb Rehabilitation. *Electronics* **2025**, *14*, 2268. [CrossRef]
21. Tornai, K.; Scheirer, W.J. Gesture-based User Identity Verification as an Open Set Problem for Smartphones. In Proceedings of the IAPR International Conference On Biometrics, Crete, Greece, 4–7 June 2019.
22. Jain, L.P.; Scheirer, W.J.; Boult, T.E. Multi-class Open Set Recognition Using Probability of Inclusion. In Proceedings of the ECCV, Zurich, Switzerland, 6–12 September 2014.
23. Rudd, E.M.; Jain, L.P.; Scheirer, W.J.; Boult, T.E. The Extreme Value Machine. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 762–768. [CrossRef] [PubMed]
24. Maiorana, E.; Kalita, H.; Campisi, P. Mobile keystroke dynamics for biometric recognition: An overview. *IET Biom.* **2020**, *10*, 1–23. [CrossRef]
25. Wandji Piugie, Y.; Charrier, C.; Manno, J.; Rosenberger, C. Deep features fusion for user authentication based on human activity. *IET Biom.* **2023**, *12*, 222–234 [CrossRef]
26. Reyes-Ortiz Jorge, A.D.G.A.O.L.; Parra, X. Human Activity Recognition Using Smartphones. In *UCI Machine Learning Repository*; Springer: Berli/Heildelberg, Germany, 2013. [CrossRef]
27. Jiokeng, K.; Jakllari, G.; Beylot, A.-L. *Hand-BCG & SCG Signals Dataset*; Zenodo: Geneva, Switzerland, 2021. [CrossRef]
28. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556. [CrossRef]
29. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
30. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [CrossRef]

*electronics*

MDPI

*Article*

# Assessing Chatbot Acceptance in Policyholder's Assistance Through the Integration of Explainable Machine Learning and Importance–Performance Map Analysis

**Jaume Gené-Albesa [1] and Jorge de Andrés-Sánchez [2,*]**

[1] Department of Business Administration, Campus de Bellissens, University Rovira i Virgili, 43204 Reus, Spain; jaume.gene@urv.cat
[2] Social and Business Research Laboratory, Campus de Bellissens, University Rovira i Virgili, 43204 Reus, Spain
* Correspondence: jorge.deandres@urv.cat

**Abstract:** Companies are increasingly giving more attention to chatbots as an innovative solution to transform the customer service experience, redefining how they interact with users and optimizing their support processes. This study analyzes the acceptance of conversational robots in customer service within the insurance sector, using a conceptual model based on well-known new information systems adoption groundworks that are implemented with a combination of machine learning techniques based on decision trees and so-called importance–performance map analysis (IPMA). The intention to interact with a chatbot is explained by performance expectancy (PE), effort expectancy (EE), social influence (SI), and trust (TR). For the analysis, three machine learning methods are applied: decision tree regression (DTR), random forest (RF), and extreme gradient boosting (XGBoost). While the architecture of DTR provides a highly visual and intuitive explanation of the intention to use chatbots, its generalization through RF and XGBoost enhances the model's explanatory power. The application of Shapley additive explanations (SHAP) to the best-performing model, RF, reveals a hierarchy of relevance among the explanatory variables. We find that TR is the most influential variable. In contrast, PE appears to be the least relevant factor in the acceptance of chatbots. IPMA suggests that SI, TR, and EE all deserve special attention. While the prioritization of TR and EE may be justified by their higher importance, SI stands out as the variable with the lowest performance, indicating the greatest room for improvement. In contrast, PE not only requires less attention, but it may even be reasonable to reallocate efforts away from improving PE in order to enhance the performance of the more critical variables.

**Keywords:** chatbots; insurance; decision tree regression; random forest; XGBoost; Shapley additive explanations (SHAP)

## 1. Introduction

Industry 4.0 has revolutionized industrial production through the integration of advanced technologies such as artificial intelligence (AI), the Internet of Things (IoT), automation, and real-time data analytics [1]. The transformative power of Industry 4.0 has extended to other sectors, including finance, giving rise to Finance 4.0. Finance 4.0 leverages digitalization to provide innovative solutions, enhancing the efficiency of the global financial system [2]. These solutions are commonly referred to as Fintech and have driven disruptive models such as digital banking, cryptocurrencies, and smart contracts, marking the beginning of a new era [3].

Insurance 4.0 is a specialized branch of Finance 4.0, focusing on the insurance industry. Thus, applications of Industry 4.0 technologies in the insurance field are labelled as Insurtech [4]. Insurtech impacts all areas of the insurance business, enabling automation in risk assessment, policy personalization, and the streamlining of claims processing through smart contracts and connected devices to prevent losses. Insurtech is redefining how insurers operate, making them more efficient, accessible, and customer-centric [5].

Among the most significant Insurtech applications, chatbots stand out as a key tool in customer service, improving both the operational efficiency of financial and insurance organizations and user accessibility [6]. Chatbots can respond to queries in real time, streamline policy management, facilitate claims reporting, and guide customers through complex processes without requiring immediate human intervention. Furthermore, they contribute to delivering a personalized experience by analyzing user data and providing tailored recommendations to meet specific needs [7]. They also help reduce operational costs and response times, ultimately increasing customer satisfaction. Their ability to operate 24/7 makes them an indispensable tool for enhancing customer service in an increasingly digitalized market [8].

Despite their benefits, many customers remain skeptical about chatbots in the insurance sector, perceiving them as providing impersonal and limited assistance. The lack of empathy in interactions, automated responses that sometimes fail to resolve complex queries, and difficulties in quickly escalating issues to a human agent contribute to user frustration [8].

This study analyzes the drivers of chatbot acceptance among policyholders for managing tasks related to their active insurance policies (e.g., filing a claim). The analysis of conversational robot acceptance in customer service is of particular interest in the insurance industry, as the use of an insurance policy—entailing claims notification—always requires communication with the insurer, both during the initial contact and in the subsequent transmission of relevant details [9].

The approach adopted in this study is based on the technology acceptance model (TAM) [10] and the unified theory of acceptance and use of technology (UTAUT) [11]. Specifically, it seeks to explain the intention to use (IU) chatbots for managing active policies through the following constructs: performance expectancy (PE), effort expectancy (EE), social influence (SI), and trust (TR). While the first three constructs are the most relevant in the literature on conversational robot acceptance [12], the inclusion of trust is justified by its key role both in the economic function of insurance [13] and in the adoption of conversational robots [14].

The analytical framework is presented in Figure 1. Concretely, this paper addresses two research questions:

- RQ1: What is the explanatory and predictive power of the proposed model?
- RQ2: What are the constructs that require greater attention for the successful implementation of chatbots?

The first contribution of this study is methodological. Unlike most of the existing literature, which typically addresses RQ1 using structural equation modeling (SEM), this research employs machine learning techniques (MLT). Concretely, it uses decision tree regression (DTR) and its ensemble generalizations—random forest (RF) and extreme gradient boosting (XGBoost). While SEM requires the specification of linear and pre-defined interactions among constructs, DTR offers a data-driven approach that identifies decision thresholds and interaction effects that may not be apparent in linear models. Additionally, it enables decision-makers to intuitively visualize relationships between variables while accounting for nonlinearity [15]. This approach is particularly useful for analyzing behavioral phenomena such as consumer behavior [16,17] and the acceptance of new technologies,

where the literature often lacks consensus on how explanatory variables interact to shape attitudes [18–20].

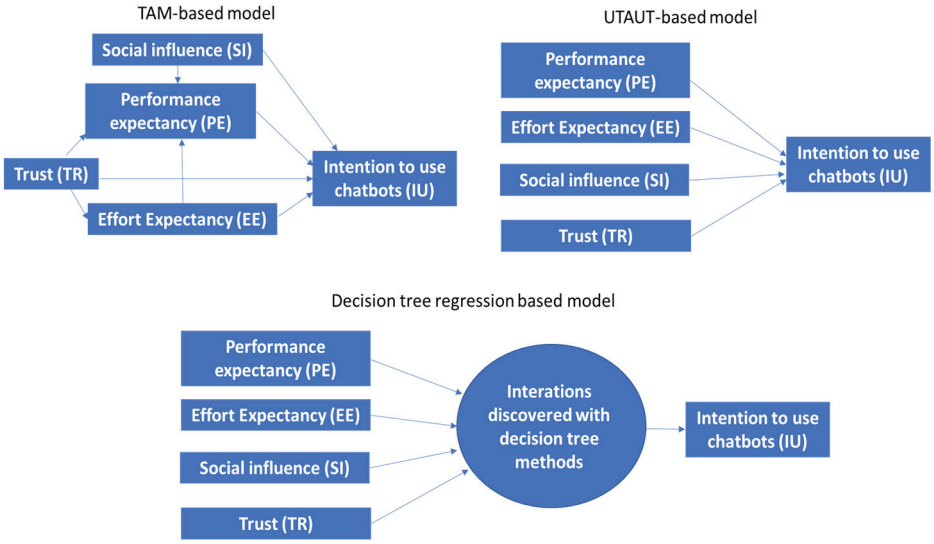



**Figure 1.** Conceptual framework of this paper versus TAM and UTAUT approaches.

In extended versions of the technology acceptance model (TAM), such as TAM2 [21], the influence of effort expectancy (EE) and social influence (SI) on intention to use (IU) is partially mediated by perceived usefulness (PU). On the other hand, the unified theory of acceptance and use of technology (UTAUT) assumes only direct effects. In the more specific context of chatbot adoption, some studies adopt a TAM-based perspective in which trust influences IU indirectly through PE and EE [22,23], while others posit a direct effect of trust on IU [24,25].

This study does not posit specific interaction hypotheses between predictors. Instead, we examine how explanatory variables correlate with IU, allowing the decision tree architecture to uncover the interaction patterns present in the data. Furthermore, combining DTR with ensemble models such as RF [26] or XGBoost [27] enhances the model's predictive performance. In this context, DTR can be interpreted as a representative or average tree within the ensemble of trees generated by RF and XGBoost. Figure 1 presents two conceptual modeling approaches previously used to explain chatbot acceptance and highlights the architecture ultimately adopted in this study.

To address RQ2, the study applies Shapley additive explanations (SHAP) [28], which enable interpretation of the relative importance of each explanatory variable in the predictions generated by the best-performing decision tree-based model—whether DTR, or more likely, RF or XGBoost. Subsequently, an importance–performance map analysis (IPMA) will be used, based on the diagonal partitioning of the IPM [29]. This will be adapted from the structural equation modeling approach proposed by [30] to the use of SHAP as a measure of the relative importance of variables. The value of IPMA lies in recognizing that the variable with the greatest impact does not necessarily warrant the most attention. If the performance of a given variable is already high compared to the other explanatory variables, improving it further may require substantial effort. Thus, it may be more effective to focus on variables with lower importance but where performance improvements are more feasible, potentially generating a greater overall impact on the target variable.

A key innovative contribution of this study lies in the integration of explainable machine learning techniques—specifically, decision tree regression, random forest, and XGBoost—with Shapley additive explanations (SHAP) and importance–performance map analysis (IPMA) within the context of technology acceptance research. While previous

studies on technology adoption have largely relied on structural equation modeling or other traditional statistical approaches, our method enables both high predictive accuracy and an intuitive understanding of the interaction patterns among explanatory variables without needing hypotheses about mediations and moderations. This combined approach not only bridges the gap between interpretability and predictive performance but also offers a replicable framework for future research in other technology adoption domains.

In addition, this research expands the scope of recent work by applying the proposed methodology to the insurance sector—a service industry where trust plays a central role—thereby capturing sector-specific behavioral drivers often overlooked in generalist studies. By comparing our findings with those from the latest empirical research across diverse geographical contexts and chatbot applications, we demonstrate how the explanatory hierarchy of variables can shift depending on cultural, industrial, and service-channel factors. This comparative perspective reinforces the originality of the study, as it provides nuanced insights that are both theoretically relevant and practically actionable for AI-powered service implementation.

## 2. Framework

Performance expectancy (PE) refers to the degree to which users perceive that a system enhances their performance in carrying out a task [11]. There are several reasons to conclude that chatbots are useful in insurance procedures. On the one hand, basic administrative tasks can be completed more quickly than when relying solely on human assistance [31]. Similarly, conversational robots do not substitute conventional interaction ways with the insurer but rather are an additional instrument that allows the improvement of policyholders' assistance [32]. That variety of communication channels is frequently valued and helps build customer satisfaction [33].

PE is likely the most influential construct in the acceptance of chatbots for customer service in both banking [6,34–37] and insurance contexts, where its impact has been observed both directly [8,38,39] and indirectly through mediating mechanisms [7]. Therefore, we propose:

**Hypothesis 1.** *Performance expectancy positively influences the intention to use chatbots for managing active insurance policies.*

Venkatesh et al. [11] define effort expectancy (EE) as the extent to which an individual believes that using a technology requires little effort. In the use of chatbots to provide customer service in insurance companies, EE refers to the absence of drawbacks for policyholders when carrying out procedures related to in-force contracts. In theory, conversational bots have specific benefits compared to other communication methods. They offer round-the-clock support and are more available than human agents [31]. Furthermore, they present fewer usability barriers compared to other digital technologies, as they can be accessed from multiple devices, including smartphones, tablets, computers, and landlines [22].

At present, there is widespread agreement that chatbots have not yet reached a level of sophistication that allows smooth and flawless interaction in every situation. Frequently, chatbots deliver unclear replies to users, which undermines their perceived usability and effectiveness. This, in turn, leads to a decline in user acceptance and satisfaction [40]. Notable issues in this regard include technological anxiety towards robots, which significantly influences usability perceptions and customer attitudes towards chatbots [41], as well as the inability of chatbots to recognize vocal tones and inflections that help determine the direction of a conversation [40].

The relevance of effort expectancy in the acceptance of chatbots has been well documented in financial contexts, particularly in the use of banking and insurance services. In the banking sector, its influence has been reported in various studies [6,35,37,42], while in the insurance domain, it has been shown to play both a direct role [8,38] and a mediated one [7,41]. Therefore:

**Hypothesis 2.** *Effort expectancy positively influences the intention to use chatbots for managing active insurance policies.*

SI refers to the extent to which individuals perceive that important people believe they should use a new technology [11]. It is a well-established fact that peer opinions, such as those of friends or family members, have a significant impact on overall technology acceptance, as individuals tend to seek social approval [11].

The opinion of close insurance advisers is often relevant in policyholders' decision-making [43]. Despite the widespread adoption of chatbots in business practices, most consumers remain skeptical and reluctant to engage with them [44]. In fact, chatbots are primarily used to provide initial help to users and consumers [43]. The relevance of social influence in explaining the acceptance of conversational robots has been demonstrated in the contexts of both banking [37] and insurance procedures [8]. So, we suggest:

**Hypothesis 3.** *Social influence positively influences the intention to use chatbots for managing active insurance policies.*

The importance of TR in policyholders' acceptance of Insurtech solutions, including chatbots, should be analyzed from a dual perspective. It embeds the unique nature of the financial and insurance industry and the interactions between companies and policyholders that are facilitated by robots. Therefore, TR becomes a critical factor in understanding customer attitudes and behavioral intentions [13].

Trust is the foundation of any financial transaction and is even more crucial in the insurance market, where both the insurer and the policyholder must rely on mutual trust in an environment characterized by a high degree of adverse selection and moral hazard [45]. A policyholder's trust in an insurance company can be defined as the perception that its services will offer reliable compensation in the event of a loss and that interactions related to claims will be satisfactory [45]. The relevance of trust in the acceptance of chatbots has been observed in contexts related to insurance, such as banking services [6,42], as well as within the insurance sector itself—both directly [7,39,41] and indirectly, mediated by performance expectancy and effort expectancy [22]. Therefore, we propose:

**Hypothesis 4.** *Trust positively influences the intention to use chatbots for managing active insurance policies.*

## 3. Materials and Methods

### *3.1. Sample and Sampling*

The paper analyzed an online survey distributed via social media platforms (LinkedIn, Facebook, Telegram) and moderated mailing lists, conducted between 20 December 2022 and 12 March 2023.

Respondents were encouraged to share the survey hyperlink with others, meaning the sampling methodology used was mixed, combining convenience sampling and snowball sampling. The estimated time to complete the questionnaire was 10–15 min.

Considering the duration of data collection, it was adequate for a cross-sectional study such as ours. Such studies require a certain window of time to obtain an adequate number

of responses, but they are ultimately a snapshot at a specific point in time; so the survey must be anchored within a defined moment. According to the literature reviewed on cross-sectional studies focused on chatbot acceptance—among those authors who actually reported how long it took them to collect data, which not all did—the time frame ranged from half a month [22] to three months [38,41].

Regarding the focus on a specific cultural context, this was also common in social sciences and human behavior studies. Such research sought responses within a defined geographical area, either to inform action or to gain insights without "contamination" from other contexts. For example, in Asia, Ref. [22] focused on Korea, Ref. [24] on India, Ref. [35] on China, and [42] on Bangladesh. In Europe, Ref. [23] was set in Spain, Ref. [25] in the United Kingdom, Ref. [34] in Romania, and Ref. [39] in Germany.

As we sought opinions from genuinely informed consumers, only responses from individuals who held at least two insurance policies were accepted. Given that the survey targeted a very specific population segment, convenience sampling could be considered appropriate [46]. Moreover, respondents were not compensated, making it reasonable to assume they were genuinely motivated to answer the questions and paid attention to their responses.

The initial number of observations was 252. We subsequently discarded incomplete responses, resulting in a final sample of 226 responses, which was considered statistically adequate according to the heuristic "ten times rule" [47], which suggests that the minimum required sample size should be 40, given that there were only four explanatory variables for behavioral intention. Additionally, using the G*Power 3.1 software [48], we verified that this sample size provided a statistical power of 80% for a linear regression with four variables, assuming a significance level of 5% and an effect size of at least 0.05, which corresponded to a minimum coefficient of determination of 4.76%. The profile of the individuals in the sample is shown in Table 1.

**Table 1.** Sociodemographic profile of the sample ($n$ = 226).

| Variable | Responses |
| --- | --- |
| Gender | 53.10% of responses came from men and 44.69% from women, and 2.21% provided other responses. |
| Age | 14.16% of responses came from individuals under 40 years old, 53.98% from those aged between 40 and 55, 30.09% from individuals over 55 years old and 1.77% did not answer. |
| Academic background | 87.17% of respondents reported having completed a university degree and 12.83% reported being undergraduate. |
| Income level | 30.09% reported an income not exceeding EUR 1750, 38.94% reported an income level between EUR 1750 and EUR 3000, a 30.09% reported earning over EUR 3000 and 0.88% did not answer. |
| Number of insurance policies | 47.79% of respondents reported holding between 2 and 4 policies, while 52.21% held more than 4 policies. |

### 3.2. Measurement Model

The survey was conducted using a structured questionnaire written in Spanish, with the items presented in Table A1 of the Appendix A. Initially, the questionnaire was dis-

tributed among six professionals from the insurance industry in Spain. After receiving their feedback and incorporating it into a revised version, it was assessed by an additional twelve voluntaries that were not professionally linked with the insurance industry.

Regarding the scales used, the IU, PE, EE, and SI measures were based on the proposals of [11], adapted to the use of chatbots in the policyholder–insurer relationship. The trust scale was based on [49].

Responses were collected using an eleven-point Likert scale (ranging from 0, indicating strong disagreement with the statement, to 10, representing strong agreement), where 5 was the neutral value.

### *3.3. Data Analysis*

First step: Since the study dealt with latent variables, the first step involved assessing the internal reliability and the discriminant validity of scales [47]. This includes calculating Cronbach's alpha, the composite reliability index, and the average variance extracted (AVE) and conducting factor extraction through exploratory factor analysis. Additionally, the correlation matrix of the constructs was analyzed to provide a first assessment of the consistency of the hypotheses regarding the direction of the relationships between the model variables. This step was performed using the psych package in R.

Second step: The final scores for each construct were determined by calculating the weighted average of the items based on the factor extraction, which was then rescaled to a 100-point reference system. This approach follows [30] for evaluating the performance of latent variables.

If construct $X$ is composed of $I$ items $x_i$, where $i = 1, 2, \cdots, I$, and we denote $w_i$ as the percentage of variance extracted for the $i$th item, then the value of the construct for the $j$th observation, $X_j$, was calculated as the weighted average of the scores for each item of that observation ($x_{ij}$), weighted by the factor extraction of that item ($w_i$). Since xi takes values between 0 and 10, and $X_j$ is, following [30], referenced on a 100-point scale, we calculate:

$$X_j = \frac{\sum_{i=1}^{I} x_{i,j} w_i}{\sum_{i=1}^{I} w_i} \cdot 10.$$

This step was carried out using the *psych* and *dplyr* packages.

Third step: Subsequently, a decision tree regression (DTR) model was fitted with all explanatory constructs. Since the variables were already measured on a 100-point scale, rather than using standardized factor extractions, this facilitates the interpretation of the tree and the cut-off values at the nodes. The sign of the relationship between an explanatory variable X and IU was inferred from how observations are distributed across the nodes in which it participates. If a threshold X < Xa is required to reach terminal nodes associated with lower acceptance, then the relationship is positive. Conversely, if reaching these nodes requires X > Xa, a negative relationship can be inferred. It is important to note that, in assessing the sign of the relationship, we considered not only the primary splits but also the surrogate splits—that is, the alternative splits that would be used if the observation for the variable responsible for the primary split were missing. This step was conducted using the *rpart* and *rpart.plot* packages in R.

Fourth step: When the objective extends beyond model explanation to achieve more accurate fits and predictions, ensemble methods such as RF and XGBoost generalize decision trees in a way that enhances predictive performance—albeit at the cost of interpretability, which is a key strength of single decision trees.

For RF and XGBoost, all explanatory variables were also included, and a hyperparameter tuning process was performed [50,51]. The dataset was randomly split into 80% for training and 20% for testing. Within the training set, 70% of the full dataset was used

for actual model training, and hyperparameter tuning was performed via 10-fold cross-validation applied to this 70%. This internal cross-validation step acted as the validation phase, replacing the need for a fixed 10% hold-out.

Hyperparameter tuning for RF focused on three parameters [51,52]: the number of variables randomly selected at each split (*mtry*), the number of trees in the forest (*ntree*), and the minimum number of observations in a terminal node (*nodesize*).

For XGBoost, we tuned the learning rate (eta), the maximum tree depth (*max_depth*), the minimum child weight (*min_child_weight*), and the number of boosting rounds (*nrounds*) [51,52].

Notice that, in contrast, DTR was fitted directly using the rpart package on the entire dataset without additional tuning, as the goal was to preserve interpretability.

This step is executed using the *caret* package in R, in combination with the *randomForest* and *xgboost* packages to implement the respective algorithms.

Fifth step: Once the RF and XGBoost models had their hyperparameters tuned (Step 4), and the DTR was fitted as described above, all models were evaluated using the coefficient of determination ($R^2$), root mean squared error (RMSE), and mean absolute error (MAE) to assess in-sample fit.

Out-of-sample predictive performance was assessed using Monte Carlo cross-validation with repeated random subsampling (80/20 split, 5000 repetitions). In each repetition, 80% of the data was randomly selected for training and 20% for testing. Predictive performance was quantified using Stone–Geisser's $Q^2$ statistic, along with RMSE and MAE, averaged across all repetitions to ensure robustness and mitigate the variability associated with a single train–test split. This step is performed using the *caret*, *rsample*, *randomForest*, and *xgboost* packages.

This evaluation addressed research question 1, enabling a robust assessment of the models' predictive and explanatory capabilities, as well as the visualization of interaction patterns that drive chatbot acceptance, thereby facilitating the evaluation of Hypotheses H1–H4. An overall overview about fourth and fifth step in provided in Table 2.

**Table 2.** Data partitioning and cross-validation methods used in Steps 4 and 5.

| Method | Training Set | Validation Set | Testing Set | Cross-Validation Method (Step 5) |
|---|---|---|---|---|
| DTR | 100% | — | 20% | Monte Carlo CV, 5000 reps |
| RF | 70% | 10% CV on training | 20% | Monte Carlo CV, 5000 reps |
| XGBoost | 70% | 10% CV on training | 20% | Monte Carlo CV, 5000 reps |

Note: The first three columns refer to the data partitioning used for model training and hyperparameter tuning (Step 4). The last column indicates the cross-validation method applied for final evaluation (Step 5).

Sixth step: To test whether the differences in predictive performance between models were statistically significant, we conducted paired-sample *t*-tests and ANOVA on the prediction metrics. This analysis provided an evidence-based comparison of the models and helped identify whether certain decision tree-based methods consistently outperformed others. This step was implemented using the *rstatix* R package.

Seventh step: To address research question 2, we first computed SHAP values for each variable across all observations. These values allowed us to calculate the mean absolute SHAP values, which represent the average contribution of each variable to the model predictions. This enabled the construction of a hierarchy of relevance among the explanatory variables, offering insights into their relative importance in explaining IU chatbots. This step was carried out using the *iml* package in R.

Eighth step: Finally, to complete the analysis of research question 2, an importance–performance map analysis (IPMA) was conducted. The performance of the constructs was stated simply as the sample mean of the items, that were rescaled in 100 [30]. On the other hand, the importance of variables was their average absolute value of SHAP. The interpretation of the importance–performance map followed the diagonal partitioning approach proposed by [29] and is illustrated in Figure 2.

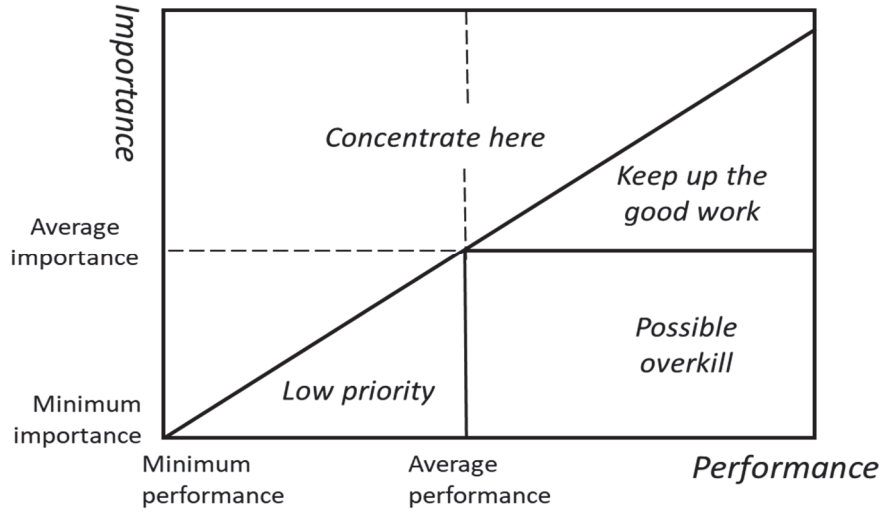



**Figure 2.** Importance–performance map interpretation used in this study. Note: Adapted from Abalo et al. [29].

## 4. Results

### *4.1. Analysis of Research Question 1*

Table 3 shows the descriptive statistics of the items that make up the scales and the measures of their internal validity. It can be observed that every item received a score considerably less than 5, indicating a very low evaluation among users. The scales exhibit internal consistency (Cronbach's alpha and the composite reliability index > 0.7) and convergent validity, as the factor extraction for the constructs is >0.7, and the average variance extracted is, in all cases, >0.5. Table 4 shows that the constructs have discriminant validity according to the Fornell–Larcker criterion, as the correlations between the constructs never exceed the square root of the average variances extracted.

In Table 4, it can be also checked that the hypotheses regarding the positive relationships between PE, EE, SI, and TR, with IU, are supported by the Pearson correlations, which are consistently positive and significantly different from zero. This is further confirmed by Figure 3, where all variables contribute to the splitting of at least one node. In each case, values below the threshold lead to a lower level of chatbot acceptance. Indeed, Figure 3 and Table 5 show that TR, EE, and SI each serve as the primary split into two partitions, while PE serves as the primary split into one. Table 5 presents not only the primary splits but also the surrogate splits. It can also be observed that when the explanatory variables act as surrogate splits, their direction of influence suggests a positive relationship of all variables with IU. Observations with values below the threshold are classified into nodes associated with lower levels of acceptance. So, from Tables 4 and 5 we can conclude that Hypothesis H1, H2, H3, and H4 can be accepted.

**Table 3.** Descriptive statistics and measures of internal reliability of scales.

| Item | Mean | SD | Factor Loading | CA | CR | AVE |
|---|---|---|---|---|---|---|
| | | | | 0.891 | 0.894 | 0.822 |
| IU1 | 1.27 | 1.87 | 0.921 | | | |
| IU2 | 2.24 | 2.7 | 0.862 | | | |
| IU3 | 1.38 | 2.06 | 0.935 | | | |
| | | | | 0.92 | 0.932 | 0.76 |
| PE1 | 2.44 | 2.63 | 0.877 | | | |
| PE2 | 2.71 | 2.66 | 0.91 | | | |
| PE3 | 2.57 | 2.58 | 0.904 | | | |
| PE4 | 2.46 | 2.61 | 0.914 | | | |
| PE5 | 3.29 | 2.86 | 0.742 | | | |
| | | | | 0.885 | 0.893 | 0.813 |
| EE1 | 2.88 | 2.82 | 0.864 | | | |
| EE3 | 2.16 | 2.27 | 0.922 | | | |
| EE4 | 2.64 | 2.64 | 0.917 | | | |
| | | | | 0.927 | 0.929 | 0.872 |
| SI1 | 1.75 | 1.94 | 0.922 | | | |
| SI2 | 1.61 | 2.05 | 0.953 | | | |
| SI3 | 2.03 | 2.15 | 0.927 | | | |
| | | | | 0.83 | 0.865 | 0.745 |
| TR1 | 2.07 | 2.5 | 0.912 | | | |
| TR2 | 3.46 | 3.04 | 0.836 | | | |
| TR3 | 2.08 | 2.18 | 0.839 | | | |

Note: CA = Cronbach's alpha, CR = composite reliability measure, AVE = average variance extracted.

**Table 4.** Matrix with Pearson correlations and the square root of the average variance extracted.

| | IU | PE | EE | SI | TR |
|---|---|---|---|---|---|
| IU | 0.906 | | | | |
| PE | 0.719 | 0.872 | | | |
| EE | 0.732 | 0.816 | 0.901 | | |
| SI | 0.713 | 0.697 | 0.648 | 0.934 | |
| TR | 0.734 | 0.861 | 0.796 | 0.690 | 0.863 |

Note: (a) The square root of AVE is on the main diagonal. (b) All Pearson correlations are significant with $p < 0.001$.

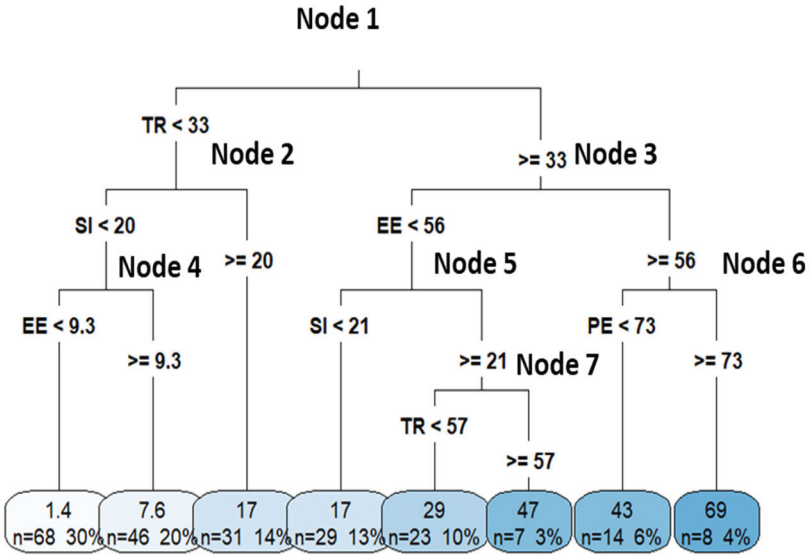



**Figure 3.** Results of decision tree regression. Note: $R^2$ = 69.23%, RMSE = 11.010, and MAE = 7.830.

**Table 5.** Principal splits (first row) and subrogate splits in the decision tree nodes (Figure 3).

| Node 1 | Node 2 | Node 3 | Node 4 | Node 5 | Node 6 | Node 7 |
|---|---|---|---|---|---|---|
| TR < 32.92 | SI < 19.81 | EE < 56.46 | EE < 9.27 | SI < 21.29 | PE < 72.76 | TR < 56.88 |
| PE < 35.81 | PE < 29.63 | PE < 58.76 | PE < 10.56 | PE < 25.65 | TR < 63.50 | PE < 55.96 |
| EE < 33.09 | TR < 29.56 | TR < 64.70 | TR < 1.62 | EE < 30.01 | EE < 73.30 | SI < 63.35 |
| SI < 31.71 | EE < 50.09 | SI < 50.06 | SI < 6.59 | TR < 46.49 | SI < 51.67 | EE < 50.09 |

Moreover, Figure 3 shows that the coefficient of determination indicates that the DTR explains nearly 70% of the variability in the response variable, which can be considered substantial. However, both the explanatory and predictive capabilities of the DTR can be enhanced by applying RF and XGBoost.

Following the fourth step described in Section 3.3, the best-performing RF model was obtained by tuning the number of variables randomly selected at each split (*mtry* = 1), the total number of trees in the ensemble (*ntree* = 100), and the minimum number of observations required in a terminal node (*nodesize* = 1). Figure 4 shows how the error decreases as the key parameter *ntree* increases, and that beyond the value of 100, it stabilizes.

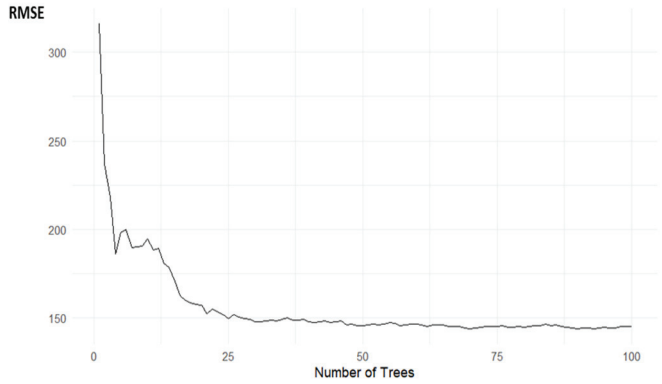



**Figure 4.** RMSE evolution of the optimal number of trees in the tuning of the random forest model.

Similarly, the optimal XGBoost model was achieved by tuning the learning rate (*eta* = 0.1), the maximum depth of the trees (*max_depth* = 4), the minimum child weight (*min_child_weight* = 5), and the number of boosting rounds (*nrounds* = 42). In both cases, fine-tuning was performed using 10-fold cross-validation, selecting the configuration that minimized the root mean squared error (RMSE). Figure 5 illustrates how the XGBoost error behaves as a function of the hyperparameters *eta* and *nrounds*.

Table 6 shows that RF, followed by XGBoost, achieves the highest $R^2$ values and substantially lower error metrics compared to DTR. Furthermore, the results of the Monte Carlo cross-validation presented in Table 7 indicate that RF exhibits the best generalization performance, followed by XGBoost and, lastly, DTR. It is also worth noting that in all cases the $Q^2$ values exceed 50%, indicating that all decision tree-based methods demonstrate a high level of generalizability [47].

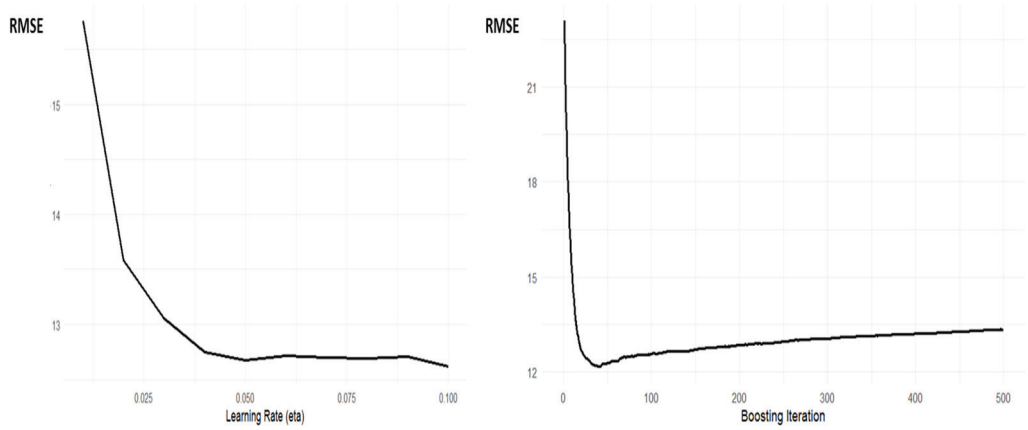



**Figure 5.** RMSE evolution of the learning rate and boosting iteration in the tuning of the optimal XGBoost model.

**Table 6.** Adjustment capability metrics of the three machine learning methods used.

| | $R^2$ | RMSE | MAE |
|---|---|---|---|
| Decision tree | 69.23% | 11.010 | 7.830 |
| Random Forest | 95.57% | 4.692 | 3.406 |
| XGBoost | 80.95% | 9.862 | 7.004 |

**Table 7.** Mean of the performance of predictive metrics of the three machine learning methods used.

| | $Q^2$ | RMSE | MAE |
|---|---|---|---|
| Decision tree | 51.70% | 14.04 | 9.91 |
| Random Forest | 63.40% | 12.23 | 8.55 |
| XGBoost | 59.80% | 12.9 | 8.98 |
| ANOVA | F = 27.70 ($p < 0.001$) | F = 33.96 ($p < 0.001$) | F = 33.17 ($p < 0.001$) |

*4.2. Analysis of Research Question 2*

Although Table 7 suggests that the method with the highest predictive performance is RF, we conducted a more in-depth analysis by performing pairwise comparisons of the significance of differences in the prediction metrics, which consistently favored RF. The mean difference analysis presented in Table 8 indicates that, regardless of the metric used, the superior predictive performance of RF compared to the other methods is statistically significant, with a $p$-value < 0.001. Conversely, the method with the poorest predictive performance is DTR. Therefore, the SHAP analysis is based on the random forest fit.

**Table 8.** Paired-sample *t*-tests for mean differences in predictive metrics.

| | $Q^2$ | | | RMSE | | | MAE | | |
|---|---|---|---|---|---|---|---|---|---|
| | diff | t-Ratio | $p$-Value | diff | t-Ratio | $p$-Value | diff | t-Ratio | $p$-Value |
| DTR vs. RF | −11.70% | −26.40 | <0.001 | 1.81 | 27.24 | <0.001 | 1.36 | 24.32 | <0.001 |
| DTR vs. XGBoost | −8.10% | −18.13 | <0.001 | 1.14 | 14.44 | <0.001 | 0.93 | 16.39 | <0.001 |
| RF vs. XGBoost | 3.60% | 16.56 | <0.001 | −0.67 | −20.83 | <0.001 | −0.43 | −13.40 | <0.001 |

Note: diff stands for the difference between mean performance metrics in Table 7.

The analysis of SHAP values in the beeswarm plot in Figure 6 reveals that TR is the most influential predictor of chatbot acceptance, showing consistently high contributions to the model's output, especially when its original values are high. This indicates that as

users' trust in the chatbot increases, so does the predicted intention to use it—highlighting a strong positive relationship. In contrast, PE exhibits the lowest SHAP values overall, with limited growth in explanatory power even at higher levels, suggesting that perceived usefulness plays a comparatively minor role in shaping user intention. EE and SI occupy an intermediate position. For both variables, SHAP values tend to increase moderately with higher original values, indicating a positive, but less dominant, impact.


**Figure 6.** Beeswarm plot of the RF adjustment of the conceptual framework in Figure 1.

Table 9 presents the mean absolute SHAP values for the four explanatory variables, along with the results of paired-sample comparisons. While TR displays the highest average absolute SHAP value, the differences between TR, EE, and SI are not statistically significant. However, PE is found to be significantly less relevant than the other three variables. Therefore, although TR appears to be the most important predictor, its contribution is not significantly greater than that of EE and SI. In contrast, the lower relevance of PE is statistically significant when compared to all other predictors.

**Table 9.** Mean absolute SHAP values for each predictor variable.

| Var 1 | Var 2 | Mean Absolute SHAP (Var 1) | Mean Absolute SHAP (Var 2) | Difference | t-Ratio | *p*-Value |
|---|---|---|---|---|---|---|
| PE | EE | 3.126 | 3.941 | −0.815 | −4.591 | <0.001 |
| PE | SI | 3.126 | 3.846 | −0.719 | −3.634 | <0.001 |
| PE | TR | 3.126 | 3.985 | −0.859 | −6.025 | <0.001 |
| EE | SI | 3.941 | 3.846 | 0.096 | 0.384 | 0.702 |
| EE | TR | 3.941 | 3.985 | −0.044 | −0.197 | 0.844 |
| SI | TR | 3.846 | 3.985 | −0.140 | −0.715 | 0.476 |

Figure 7 presents the importance–performance map constructed based on the eighth step of Section 3 and the SHAP values shown in Table 9. While TR and EE have slightly higher importance than SI, they also exhibit higher performance levels, making their improvement considerably more challenging. In contrast, PE shows the lowest importance combined with the highest performance, placing it in the potential overkill zone. In other words, it clearly does not require immediate attention, and it may even be justified to reallocate efforts away from improving this item in order to focus on enhancing the three more relevant constructs.

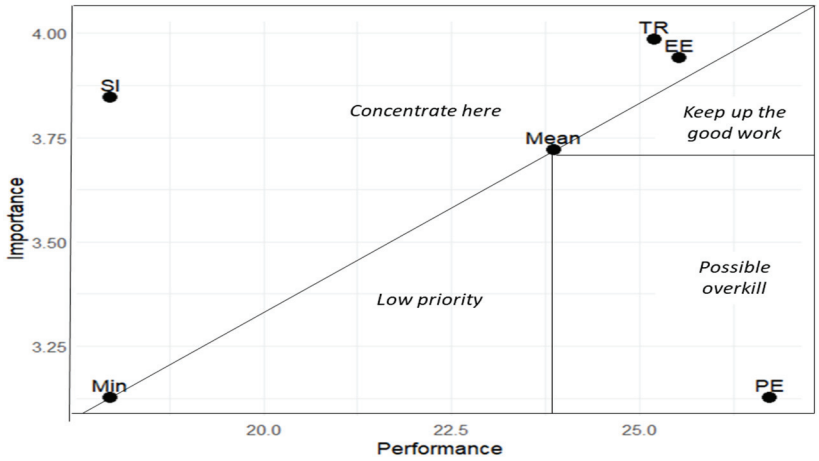



**Figure 7.** Importance–performance map of the assessed variables to produce acceptance of chatbots.

## 5. Discussion

### *5.1. General Considerations*

Regarding the first research question (RQ1), we found that the model fit the data well across all machine learning methods, providing a detailed understanding of how the explanatory factors contributed to both acceptance and rejection. While the ensemble decision tree methods yielded better model fit and predictive performance, the simple decision tree regression allowed us to assess the extent to which the hypotheses proposed in Section 2 were supported, as well as to visualize how the explanatory variables interacted to segment the sample into seven distinct user types, ordered by their level of chatbot acceptance. The hypothesized positive relationship between the explanatory variables and acceptance was reflected in the fact that all variables contributed to the partitioning of at least one node of the decision tree, and the resulting partitions suggested a positive association with IU. The results obtained with the decision tree regression were consistent with the fact that all explanatory variables showed a significant correlation with IU in the expected (positive) direction.

In RQ2, we investigated which explanatory variables were most relevant in explaining the intensity of chatbot acceptance. This analysis was conducted using the Shapley additive explanations (SHAP) measure. The results showed that the most influential variables, in order of importance, were TR, EE, SI, and, lastly, PE. It is also worth noting that while the mean absolute SHAP values for the first three variables did not differ significantly from each other, the SHAP value for PE was significantly lower than those of the top three.

The use of DTR enabled a deep understanding of how variables interacted to classify potential chatbot users into different levels of usage—eight, in this case, corresponding to the number of terminal nodes shown in Figure 3. We observed that while TR was used in the initial splits and was therefore decisive in differentiating among all user typologies, SI and EE acted as discriminating factors at low and intermediate levels of acceptance. In contrast, PE only contributed to refining the classification of users who exhibited the highest levels of acceptance.

Trust had a positive and significant influence on behavioral intention. It was, in fact, the variable with the greatest influence on the acceptance of conversational robots. This outcome could be attributed to two key elements: first, the inherent characteristics of the insurance industry, which relies heavily on trust [45], and second, the importance of this concept in the adoption of robotic technologies, making trust a highly important factor in AI-powered Insurtech [13]. Our findings aligned with those from previous studies on

conversational robots in various fields and in countries such as Korea [22], Lebanon [53], and Germany [54].

The positive relationship between EE and IU in the context of the insurer–insured relationship was expected, as convenience could be a relevant factor in the acceptance of chatbots in this area, as reported in Sweden [7]. This result was also in line with studies on customer interactions with chatbots in several countries across Asia, Europe, and North America [53,55–59].

The relevance of SI in influencing IU was consistent with the acceptance of conversational robots by consumers in many cultural contexts and customer service settings, including Korea [22], Lebanon [53], India [56], China [60], and Romania [61].

However, it should be noted that the relevance of PE was secondary compared with the rest of the variables. Regarding PE, this finding could be explained by the fact that the relevance of utilitarian motivations in the use of technologies is paramount when the use of the information system is mandatory [21]. However, chatbot services for customer service should be understood in a multichannel interaction context [43], where their use is optional. In fact, Refs. [54,62] in two different European countries did not observe a significant influence of PE on IU.

The IPMA provided a deeper understanding of the key variables to increase the acceptance of chatbots. The results indicated that social influence, as well as trust and effort expectancy, were not only the most important variables, but also those that, based on their current performance levels, offered the greatest scope for improvement. Therefore, these variables should be prioritized in implementation and improvement strategies. In contrast, performance expectancy, although relevant, was more consolidated among users, requiring secondary attention.

### *5.2. Theoretical Implications of the Findings in This Paper*

We showed that a UTAUT model with four explanatory variables (PE, EE, SI, and TR) explained 70% of the variability in IU using decision tree regression, and this explanatory power increased up to 95% when using random forest. Furthermore, the predictive capacity of the model remained high regardless of the machine learning method applied. This should not be interpreted as decision tree regression being inferior to random forests or XGBoost; rather, these methods belong to the same family and can be used in a complementary manner.

One of the greatest strengths of DTR was its interpretability. Through its structure of nodes and branches, it was possible to understand how an average decision-maker assigned a particular rating or evaluated a level of acceptance, as in the case of adopting technologies such as robots. Although this method did not provide $p$-values, its interpretation was intuitive. The distribution of observations within the nodes allowed the inference of the direction of the relationship between factors [15]. Moreover, the predictive performance of DTR could be enhanced through decision tree-based techniques such as random forests or XGBoost [63]. To the best of our knowledge, explanatory approaches in consumer behavior analysis were scarce, and within the literature on chatbot acceptance in B2C interactions, virtually non-existent. Therefore, from a methodological standpoint, this study offered a novel perspective based on the use of explainable machine learning techniques.

The use of Shapley additive explanations (SHAP) enabled the quantification of each explanatory variable's importance. This importance was combined with the performance level of each variable to conduct an importance–performance matrix analysis (IPMA), similar to the approach proposed by Ringle and Sarstedt [30], thereby enhancing the explanatory power of partial least squares structural equation modeling. In summary, this study demonstrated that combining machine learning decision tree methods with IPMA

could be highly beneficial for economic and business analyses. This constituted the second methodological contribution of the study, as—although the use of IPMA was common in business analysis—the application of SHAP to quantify variable importance, to the best of our knowledge, represented an original methodological focus.

### 5.3. Practical Implications of the Findings in This Paper

This has significant implications for the insurance industry. The IPMA results indicate that the variables requiring focused attention for a successful chatbot implementation in the insurer–insured relationship are social influence, trust, and effort expectancy, which are summarized in Table 10 and elaborated on in the following paragraphs. So, regarding improving social perception, effective measures could include:

**Table 10.** Practical recommendations for enhancing chatbot implementation in insurance.

| Focus Area | Recommendation |
| --- | --- |
| Social Influence | • Humanize the chatbot through empathetic language, a name, and visual identity.<br>• Launch educational campaigns showing benefits and use cases via videos and testimonials. |
| Trust | • Ensure seamless escalation to human agents when needed, with transparent handovers.<br>• Clearly inform users they are interacting with a chatbot and explain its limitations. |
| Effort Expectancy | • Design a simple, intuitive interface with natural language processing and mobile compatibility.<br>• Provide onboarding support (tutorials, tooltips, step-by-step guidance).<br>• Incorporate accessibility features (e.g., voice commands, screen–reader support).<br>• Enable memory of past interactions and reduce redundant questions.<br>• Train human agents to avoid repeating information during handovers.<br>• Offer personalization (pre-filled data, smart suggestions) and communicate convenience benefits. |

- Humanizing the chatbot [58], giving it natural, empathetic language, assigning it a name and visual identity to make it more recognizable and friendly, and programming responses that reflect understanding and empathy, especially in sensitive situations like claims management.
- Educating and familiarizing customers with chatbot use. This can be achieved through informative campaigns that share details on how to use the chatbot and its benefits via the insurer's channels. Videos or interactive guides showing how the chatbot can assist in various processes, along with testimonials from policyholders who have had positive experiences with the system, could also be useful.

Some measures to increase trust in the chatbot include:

- Emphasizing the need for the chatbot to handle complex cases and errors appropriately. It is crucial to implement systems that automatically detect when the chatbot cannot resolve a request and must seamlessly refer the case to a human agent. Furthermore, it is important to clearly explain to users the transition from bot to human to avoid frustrations.
- Ensuring transparency and clear communication between the chatbot and the policyholder. This involves informing users from the start that they are interacting with a bot, clarifying when they will be transferred to a human agent, and ensuring the client understands the chatbot's capabilities and limitations from the outset.

To improve effort expectancy, the following can be suggested:

- First, simplifying the user interface is essential. The chatbot should offer a clear and intuitive design that guides users through tasks with minimal effort. Leveraging natural language processing (NLP) allows users to interact using everyday language, eliminating the need to learn specific commands. Ensuring compatibility across devices—particularly smartphones—is also key to promoting ease of access.
- In addition, providing onboarding support can greatly reduce perceived effort. Interactive tutorials, embedded tooltips, and step-by-step instructions for common procedures (e.g., filing a claim) help users feel confident from the start. Offering multilingual support and using clear, jargon-free language ensure that a broader range of users can engage effectively with the chatbot. Accessibility features, such as voice commands and screen–reader compatibility, should also be incorporated to accommodate users with diverse needs.
- Moreover, the chatbot's functionality should be reliable and consistent. This includes avoiding repetitive requests for the same information, enabling memory of previous interactions, and offering seamless handovers to human agents when needed. In this case the company has to provide training and procedures to the agents to whom the policyholder will be transferred, with the aim of avoiding repetition of information already given and preventing users from feeling that their time is wasted when assisted by a chatbot. Personalization features—such as pre-filled data and smart suggestions—can further reduce user effort. Finally, communicating the benefits of using chatbots, including time savings and convenience, and sharing testimonials from satisfied users, can positively shape expectations and reduce perceived difficulty.

## 6. Conclusions

### *6.1. Principal Takeaways*

This study offers several insights into the drivers of chatbot acceptance in the insurance sector. It builds on the well-known TAM and UTAUT frameworks, enriched with the construct of trust and analyzed through machine learning techniques based on decision trees.

The first key takeaway is that the proposed model—incorporating performance expectancy (PE), effort expectancy (EE), social influence (SI), and trust (TR)—is both theoretically robust and empirically sound. Using decision tree regression methods, the model explains about 70% of the variability in the intention to use (IU) chatbots. When ensemble methods such as random forest and XGBoost are applied, predictive performance rises sharply, with $R^2$ values up to 95%. These findings show the relevance of the selected constructs and the effectiveness of decision tree-based methods for capturing complex interaction patterns and improving prediction accuracy in technology adoption research.

The second major insight concerns the relative importance of the explanatory variables. SHAP analysis shows that TR, EE, and SI are the most influential predictors of chatbot acceptance, with no statistically significant differences among them. In contrast, PE—though traditionally a central driver in technology acceptance models—has significantly lower importance, despite a high-performance rating among users. This challenges assumptions inherited from earlier models like TAM and UTAUT, especially in contexts where chatbot use is optional and part of a multichannel service environment.

A third takeaway comes from the importance–performance map analysis (IPMA). TR and EE rank highest in importance, but their strong performance levels suggest that further improvements could be costly and bring diminishing returns. SI, however, shows high importance and lower performance, making it a priority for strategic action. PE falls into the "overkill" zone—low importance but high performance—suggesting that further enhancement may be unnecessary or inefficient at this stage.

From an analytical perspective, we found that the use of DTR provides an empirical view of how the explanatory variables of a phenomenon—in this case, PE, EE, SI, and TR—interact to produce acceptance or rejection of chatbots. In our hypothesis development, we did not specify mediated or moderated relationships; instead, these are empirically uncovered through the DTR model. Moreover, the consideration of surrogate splits offers a deeper understanding of the interaction between explanatory variables and the direction of their influence on the outcome. To the best of our knowledge, this application of DTR has not yet been leveraged in consumer behavior studies focused on the acceptance of new communication channels with firms.

Finally, the study confirms that combining machine learning techniques with explainability tools such as SHAP and managerial instruments like IPMA provides a powerful and integrated approach to understanding user behavior in technology adoption contexts. This methodological synergy not only improves predictive accuracy, but also bridges the gap between complex algorithmic outputs and practical and intuitive managerial interpretation. By translating model results into interpretable and strategically relevant insights, the approach facilitates more informed decision-making regarding technology design, communication strategies, and resource allocation. Furthermore, it demonstrates the potential of hybrid methodologies to move beyond mere prediction and toward prescriptive guidance for implementation, especially in settings where user acceptance is critical to success. The integration of SHAP and IPMA proves especially valuable in dynamic business environments where decision-making must rely on real-time diagnostics rather than historical patterns, which may hold limited relevance. This is particularly true in the context of emerging technologies—such as AI-powered chatbots—where rapid innovation outpaces the applicability of past data and requires timely, actionable insights to guide effective implementation.

In the case of chatbot deployment in insurance services, and in the moment of the study, efforts should be directed primarily toward improving users' trust, perceived ease of use, and social perceptions, rather than focusing solely on performance-related expectations. This nuanced approach is essential for increasing user acceptance and optimizing the integration of artificially driven tools in customer service.

### *6.2. Limitations and Future Research Directions*

We recognize the constraints of this empirical research. This study was carried out in a specific territory, Spain, with the majority of answers gathered from platforms like LinkedIn. Users of these platforms tend to have higher education and professional experience, typically ranging from mid-level management to executive roles. As a result, the educational and economic backgrounds of the participants may influence our findings on the behavioral intent to use chatbots. The study of other social groups and cultures may yield completely different results. A study on the acceptance of ChatGPT by Zoomers for general uses in Croatia shows, unlike our study, that performance expectancy is the most relevant variable for chatbot acceptance [59].

Therefore, caution is advised when generalizing our results to policyholders from other cultures or those with professional and educational profiles that differ from the sample group. To draw broader conclusions, it would be essential to include a more diverse range of countries and socio-economic profiles of the respondents.

It should also be noted that the study focuses on a specific economic sector and a very particular type of customer service: the management of in-force insurance policies. Extrapolating the findings of this study to other sectors (e.g., non-financial industries) or even to different contexts within the insurance sector—such as providing advice on future contracts—should be approached with caution. Within the insurance domain, this study

could be extended to other potential services offered to customers, such as suggesting new products to existing policyholders or assisting individuals interested in initiating new contracts with the company.

While the RF model demonstrated a high explanatory capacity ($R^2$), it is important to acknowledge the potential risk of model overfitting, particularly in complex, non-parametric algorithms. Although the Monte Carlo cross-validation procedure mitigates this concern by evaluating predictive performance on multiple holdout samples, the possibility of overly optimistic estimates cannot be entirely ruled out. It should also be noted that RF achieved a high predictive capacity ($Q^2 > 50\%$)—substantially higher than that of DTR—yet still notably lower than its own explanatory capacity. Future research could further address this issue by testing alternative regularization techniques, tuning strategies, or simpler model specifications to confirm the robustness of the findings.

The analysis in this paper is based on a cross-sectional survey, meaning the conclusions cannot be extended to long-term trends and are limited to a specific geographical area (Spain). However, the data collection period is consistent with other cross-sectional studies, which range from half a month in Croatia [59], one month in Canada [38], two months in Romania [34,61], to three months [38] in Canada. The findings therefore represent a snapshot tied to a specific moment in the introduction and development of chatbot technology within a particular industry, namely the insurance sector. These results are particularly useful for informing decisions in contexts similar to the one studied, as earlier stages of chatbot development and market penetration are not directly comparable to the current landscape.

The fields of artificial intelligence and Insurtech are evolving at a rapid pace, and public perceptions of emerging technologies are highly dynamic. A more comprehensive understanding would require comparable studies conducted at different stages of chatbot evolution. While longitudinal approaches—spanning, for example, a decade—may be well-suited for cultural or ethnographic research, they may be less relevant for managerial decision-making in fast-moving environments where the technology is still undergoing rapid growth and has not yet reached full consolidation.

**Author Contributions:** Conceptualization: J.d.A.-S. and J.G.-A.; methodology: J.d.A.-S.; validation: J.G.-A.; formal analysis: J.d.A.-S.; investigation: J.d.A.-S. and J.G.-A.; resources: J.d.A.-S.; data curation: J.G.-A.; writing—original draft preparation: J.d.A.-S.; writing—review and editing: J.G.-A.; visualization: J.G.-A.; supervision: J.G.-A.; project administration: J.G.-A.; funding acquisition: J.d.A.-S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by Telefonica and the Telefonica Chair on Smart Cities of the Universitat Rovira i Virgili and Universitat de Barcelona (project number 42.DB.00.18.00).

**Institutional Review Board Statement:** (1) All participants received detailed written information about the study and procedure; (2) no data directly or indirectly related to the health of the subjects were collected, and therefore the Declaration of Helsinki was not mentioned when informing the subjects; (3) the anonymity of the collected data was ensured at all times; (4) the research received a favorable evaluation from the Ethics Committee of the researchers' institution (CEIPSA-2022-PR-0005).

**Informed Consent Statement:** All respondents gave permission for the processing of their responses for the content of this publication.

**Data Availability Statement:** The data supporting the analysis is available at https://doi.org/10.7910/DVN/LK4LAT (accessed on 10 August 2025).

**Conflicts of Interest:** The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| DTR | Decision Tree Regression |
| EE | Effort Expectancy |
| IPMA | Importance–Performance Map Analysis |
| PE | Performance Expectancy |
| RF | Random Forest |
| SHAP | Shapley Additive Explanations |
| SI | Social Influence |
| TAM | Technology Acceptance Model |
| TR | Trust |
| UTAUT | Unified Theory of Acceptance and Use of Technology |
| XGBoost | Extreme Gradient Boosting |

## Appendix A

**Table A1.** Items of latent variables assessed in this study.

| **Items** |
|---|
| Intention to Use (IU) |
| IU1. I intend to be assisted by chatbots. |
| IU2. I predict that I will use a service managed by chatbots. |
| IU3. I will opt for management carried out by chatbots. |
| Performance Expectancy (PE) |
| PE1. The use of chatbots can be useful for managing my claims. |
| PE2. Using chatbots will make it easier for me to report my claims. |
| PE3. Using chatbots is useful and will allow me to receive compensations I am entitled to more quickly. |
| PE4. Using chatbots is useful and will allow me to manage my claims with less effort and fewer undesired effects (such as errors made by the insurance company's agent). |
| PE5. Using chatbots allows the insurance company to offer better service to customers at lower costs. |
| Effort Expectancy (EE) |
| EE1. It will be easy for me to adapt to using chatbots in my dealings with my insurer. |
| EE2. It will be easier to manage my claims with the existence of chatbots. |
| EE3. It will be easy for me to use the channels provided by the insurer for communica-tion if they are managed by chatbots. |
| Social influence (SI) |
| SI1. The people who are important to me believe that using chatbots facilitates the claims process. |
| SI2. The people who influence me believe that, if I could choose a claims channel, I should opt for one that uses chatbots. |
| SI3. The people whose opinions I value believe that using chatbots in insurance man-agement by the insured is an advance. |
| Trust (TR) |
| TR1. The use of chatbots in my relationship with the insurer gives me trust. |
| TR2. The use of chatbots makes it easier for the insurer to fulfil its commitments and obligations. |
| TR3. In managing claims through chatbots, the interests of the insured are taken into account. |

## References


1. Tamvada, J.P.; Narula, S.; Audretsch, D.; Puppala, H.; Kumar, A. Adopting New Technology Is a Distant Dream? The Risks of Implementing Industry 4.0 in Emerging Economy SMEs. *Technol. Forecast. Soc. Change* **2022**, *185*, 122088. [CrossRef]
2. He, M. Fintech 4.0 and Financial Systems. In *Innovation, Sustainability, and Technological Megatrends in the Face of Uncertainties: Core Developments and Solutions*; Turi Abeba, N., Lekhi, P., Eds.; Springer Nature: Cham, Switzerland, 2024; pp. 55–72. [CrossRef]
3. Mhlanga, D. Industry 4.0 in Finance: The Impact of Artificial Intelligence (AI) on Digital Financial Inclusion. *Int. J. Financ. Stud.* **2020**, *8*, 45. [CrossRef]
4. Nicoletti, B. Industry 4.0 and Insurance 4.0. In *Insurance 4.0: Benefits and Challenges of Digital Transformation*; Springer International Publishing: Cham, Switzerland, 2021; pp. 11–40. [CrossRef]
5. Sosa, I.; Montes, Ó. Understanding the InsurTech Dynamics in the Transformation of the Insurance Sector. *Risk Manag. Insur. Rev.* **2022**, *25*, 35–68. [CrossRef]
6. Nguyen, D.M.; Chiu, Y.-T.H.; Le, H.D. Determinants of Continuance Intention toward Banks' Chatbot Services in Vietnam: A Necessity for Sustainable Development. *Sustainability* **2021**, *13*, 7625. [CrossRef]
7. Gebert-Persson, S.; Gidhagen, M.; Sallis, J.E.; Lundberg, H. Online Insurance Claims: When More than Trust Matters. *Int. J. Bank Mark.* **2019**, *37*, 579–594. [CrossRef]
8. Andrés-Sánchez, J.; Gené-Albesa, J. Explaining Policyholders' Chatbot Acceptance with an Unified Technology Acceptance and Use of Technology-Based Model. *J. Theor. Appl. Electron. Commer. Res.* **2023**, *18*, 1217–1237. [CrossRef]
9. Eckert, C.; Neunsinger, C.; Osterrieder, K. Managing Customer Satisfaction: Digital Applications for Insurance Companies. *Geneva Pap. Risk Insur. Issues Pract.* **2022**, *47*, 569–602. [CrossRef]
10. Davis, F.D. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Q.* **1989**, *13*, 319–340. [CrossRef]
11. Venkatesh, V.; Morris, M.G.; Davis, G.B.; Davis, F.D. User Acceptance of Information Technology: Toward a Unified View. *MIS Q.* **2003**, *27*, 425–478. [CrossRef]
12. Alsharhan, A.; Al-Emran, M.; Shaalan, K. Chatbot Adoption: A Multiperspective Systematic Review and Future Research Agenda. *IEEE Trans. Eng. Manag.* **2023**, *71*, 10232–10244. [CrossRef]
13. Zarifis, A.; Cheng, X. A Model of Trust in Fintech and Trust in Insurtech: How Artificial Intelligence and the Context Influence It. *J. Behav. Exp. Financ.* **2022**, *36*, 100739. [CrossRef]
14. Gatzioufa, P.; Saprikis, V. A Literature Review on Users' Behavioral Intention toward Chatbots' Adoption. *Appl. Comput. Inform.* **2022**. [CrossRef]
15. Loh, W.-Y. Classification and Regression Trees. *WIREs Data Min. Knowl. Discov.* **2011**, *1*, 14–23. [CrossRef]
16. Imani, M.; Arabnia, H.R. Hyperparameter Optimization and Combined Data Sampling Techniques in Machine Learning for Customer Churn Prediction: A Comparative Analysis. *Technologies* **2023**, *11*, 167. [CrossRef]
17. Imani, M.; Beikmohammadi, A.; Arabnia, H.R. Comprehensive Analysis of Random Forest and XGBoost Performance with SMOTE, ADASYN, and GNUS under Varying Imbalance Levels. *Technologies* **2025**, *13*, 88. [CrossRef]
18. Chung, D.; Jeong, P.; Kwon, D.; Han, H. Technology Acceptance Prediction of Robo-Advisors by Machine Learning. *Intell. Syst. Appl.* **2023**, *18*, 200197. [CrossRef]
19. Richter, N.F.; Tudoran, A.A. Elevating Theoretical Insight and Predictive Accuracy in Business Research: Combining PLS-SEM and Selected Machine Learning Algorithms. *J. Bus. Res.* **2024**, *173*, 114453. [CrossRef]
20. Cuc, L.D.; Rad, D.; Cilan, T.F.; Gomoi, B.C.; Nicolaescu, C.; Almași, R.; Pandelica, I. From AI Knowledge to AI Usage Intention in the Managerial Accounting Profession and the Role of Personality Traits—A Decision Tree Regression Approach. *Electronics* **2025**, *14*, 1107. [CrossRef]
21. Venkatesh, V.; Davis, F.D. A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies. *Manag. Sci.* **2000**, *46*, 186–204. [CrossRef]
22. Han, J.; Conti, D. The Use of UTAUT and Post Acceptance Models to Investigate the Attitude towards a Telepresence Robot in an Educational Setting. *Robotics* **2020**, *9*, 34. [CrossRef]
23. de Andrés-Sánchez, J.; Gené-Albesa, J. Not with the Bot! The Relevance of Trust to Explain the Acceptance of Chatbots by Insurance Customers. *Humanit. Soc. Sci. Commun.* **2024**, *11*, 110. [CrossRef]
24. Kasilingam, D.L. Understanding the Attitude and Intention to Use Smartphone Chatbots for Shopping. *Technol. Soc.* **2020**, *62*, 101280. [CrossRef]
25. Pitardi, V.; Marriott, H.R. Alexa, She's Not Human But... Unveiling the Drivers of Consumers' Trust in Voice-Based Artificial Intelligence. *Psychol. Mark.* **2021**, *38*, 626–642. [CrossRef]
26. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
27. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [CrossRef]

28. Lundberg, S.M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4765–4774.
29. Abalo, J.; Varela, J.; Manzano, V. Importance Values for Importance–Performance Analysis: A Formula for Spreading out Values Derived from Preference Rankings. *J. Bus. Res.* **2007**, *60*, 115–121. [CrossRef]
30. Ringle, C.M.; Sarstedt, M. Gain More Insight from Your PLS-SEM Results. *Ind. Manag. Data Syst.* **2016**, *116*, 1865–1886. [CrossRef]
31. DeAndrade, I.M.; Tumelero, C. Increasing Customer Service Efficiency through Artificial Intelligence Chatbot. *Rev. Gest.* **2022**, *29*, 238–251. [CrossRef]
32. Standaert, W.; Muylle, S. Framework for Open Insurance Strategy: Insights from a European Study. *Geneva Pap. Risk Insur. Issues Pract.* **2022**, *47*, 643–668. [CrossRef] [PubMed]
33. Gené-Albesa, J. Interaction Channel Choice in a Multichannel Environment, An Empirical Study. *Int. J. Bank Mark.* **2007**, *25*, 490–506. [CrossRef]
34. Alt, M.A.; Vizeli, I.; Săplăcan, Z. Banking with a Chatbot–A Study on Technology Acceptance. *Stud. Univ. Babes Bolyai Oeconomica* **2021**, *66*, 13–35. [CrossRef]
35. Huang, S.Y.; Lee, C.-J.; Lee, S.-C. Toward a Unified Theory of Customer Continuance Model for Financial Technology Chatbots. *Sensors* **2021**, *21*, 5687. [CrossRef] [PubMed]
36. Shaikh, I.A.K.; Khan, S.; Faisal, S. Determinants Affecting Customer Intention to Use Chatbots in the Banking Sector. *Innov. Mark.* **2023**, *19*, 257–268. [CrossRef]
37. Toh, T.-J.; Tay, L.-Y. Banking Chatbots: A Study on Technology Acceptance among Millennials in Malaysia. *J. Logist. Inform. Serv. Sci.* **2022**, *9*, 1–15. [CrossRef]
38. PromTep, S.; Arcand, M.; Rajaobelina, L.; Ricard, L. From What Is Promised to What Is Experienced with Intelligent Bots. In *Advances Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC)*; Springer International Publishing: Cham, Switzerland, 2021; Volume 1, pp. 560–565. [CrossRef]
39. Rodríguez-Cardona, D.; Janssen, A.; Guhr, N.; Breitner, M.H.; Milde, J. A Matter of Trust? Examination of Chatbot Usage in Insurance Business. In Proceedings of the 54th Hawaii International Conference on System Sciences, Honolulu, HI, USA, 5–8 January 2021; pp. 556–565. [CrossRef]
40. Vassilakopoulou, P.; Haug, A.; Salvesen, L.M.; Pappas, I.O. Developing Human/AI Interactions for Chat-Based Customer Services: Lessons Learned from the Norwegian Government. *Eur. J. Inf. Syst.* **2023**, *32*, 10–22. [CrossRef]
41. Rajaobelina, L.; PromTep, S.; Arcand, M.; Ricard, L. Creepiness: Its Antecedents and Impact on Loyalty When Interacting with a Chatbot. *Psychol. Mark.* **2021**, *38*, 2339–2356. [CrossRef]
42. Hasan, S.; Godhuli, E.R.; Rahman, S.; Mamun, A.A. The Adoption of Conversational Assistants in the Banking Industry: Is the Perceived Risk a Moderator? *Heliyon* **2023**, *9*, e20220. [CrossRef]
43. Andrés-Sánchez, J.; Gené-Albesa, J. Assessing Attitude and Behavioral Intention toward Chatbots in an Insurance Setting: A Mixed Method Approach. *Int. J. Hum. Comput. Interact.* **2023**, *40*, 4918–4933. [CrossRef]
44. Van Pinxteren, M.M.E.; Pluymaekers, M.; Lemmink, J.G.A.M. Human-like Communication in Conversational Agents: A Literature Review and Research Agenda. *J. Serv. Manag.* **2020**, *31*, 203–225. [CrossRef]
45. Guiso, L. Trust and Insurance. *Geneva Pap. Risk Insur. Issues Pract.* **2021**, *46*, 509–512. [CrossRef]
46. Andrade, C. The Inconvenient Truth About Convenience and Purposive Samples. *Indian J. Psychol. Med.* **2020**, *43*, 86–88. [CrossRef] [PubMed]
47. Hair, J.F.; Risher, J.J.; Sarstedt, M.; Ringle, C.M. When to Use and How to Report the Results of PLS-SEM. *Eur. Bus. Rev.* **2019**, *31*, 2–24. [CrossRef]
48. Faul, F.; Erdfelder, E.; Buchner, A.; Lang, A.-G. Statistical Power Analyses Using G*Power 3.1: Tests for Correlation and Regression Analyses. *Behav. Res. Methods* **2009**, *41*, 1149–1160. [CrossRef]
49. Morgan, R.M.; Hunt, S.D. The Commitment-Trust Theory of Relationship Marketing. *J. Mark.* **1994**, *58*, 20–38. [CrossRef]
50. Probst, P.; Wright, M.N.; Boulesteix, A.-L. Hyperparameters and Tuning Strategies for Random Forest. *WIREs Data Min. Knowl. Discov.* **2019**, *9*, e1301. [CrossRef]
51. Kuhn, M.; Johnson, K. Over-Fitting and Model Tuning. In *Applied Predictive Modeling*; Springer: New York, NY, USA, 2013; pp. 61–92. [CrossRef]
52. Alhazeem, E.; Alsobeh, A.; Al-Ahmad, B. Enhancing Software Engineering Education through AI: An Empirical Study of Tree-Based Machine Learning for Defect Prediction. In *SIGITE '24: Proceedings of the 25th Annual Conference on Information Technology Education*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 153–156. [CrossRef]
53. Mostafa, R.B.; Kasamani, T. Antecedents and Consequences of Chatbot Initial Trust. *Eur. J. Mark.* **2022**, *56*, 1748–1771. [CrossRef]
54. Gansser, O.A.; Reich, C.S. A New Acceptance Model for Artificial Intelligence with Extensions to UTAUT2: An Empirical Study in Three Segments of Application. *Technol. Soc.* **2021**, *65*, 101535. [CrossRef]
55. Joshi, H. Integrating Trust and Satisfaction into the UTAUT Model to Predict Chatbot Adoption—A Comparison between Gen-Z and Millennials. *Int. J. Inf. Manag. Data Insights* **2025**, *5*, 100332. [CrossRef]

56. Goli, M.; Sahu, A.K.; Bag, S.; Dhamija, P. Users' Acceptance of Artificial Intelligence-Based Chatbots: An Empirical Study. *Int. J. Technol. Hum. Interact.* **2023**, *19*, 18. [CrossRef]
57. Liu, M.; Yang, Y.; Ren, Y.; Jia, Y.; Ma, H.; Luo, J.; Fang, S.; Qi, M.; Zhang, L. What Influences Consumer AI Chatbot Use Intention? An Application of the Extended Technology Acceptance Model. *J. Hosp. Tour. Technol.* **2024**, *15*, 667–689. [CrossRef]
58. Akram, S.; Buono, P.; Lanzilotti, R. Recruitment Chatbot Acceptance in a Company: A Mixed Method Study on Human-Centered Technology Acceptance Model. *Pers. Ubiquitous Comput.* **2024**, *28*, 961–984. [CrossRef]
59. Biloš, A.; Budimir, B. Understanding the Adoption Dynamics of ChatGPT among Generation Z: Insights from a Modified UTAUT2 Model. *J. Theor. Appl. Electron. Commer. Res.* **2024**, *19*, 863–879. [CrossRef]
60. Xie, C.; Wang, Y.; Cheng, Y. Does Artificial Intelligence Satisfy You? A Meta-Analysis of User Gratification and User Satisfaction with AI-Powered Chatbots. *Int. J. Hum. Comput. Interact.* **2024**, *40*, 613–623. [CrossRef]
61. Iancu, I.; Iancu, B. Interacting with Chatbots Later in Life: A Technology Acceptance Perspective in COVID-19 Pandemic Situation. *Front. Psychol.* **2023**, *13*, 1111003. [CrossRef] [PubMed]
62. de Andrés-Sánchez, J.; Gené-Albesa, J. Drivers and Necessary Conditions for Chatbot Acceptance in the Insurance Industry. Analysis of Policyholders' and Professionals' Perspectives. *J. Organ. Comput. Electron. Commer.* **2024**, 1–28. [CrossRef]
63. Ngai, E.W.T.; Wu, Y. Machine Learning in Marketing: A Literature Review, Conceptual Framework, and Research Agenda. *J. Bus. Res.* **2022**, *145*, 35–48. [CrossRef]

*electronics*

MDPI

*Article*

# From AI Knowledge to AI Usage Intention in the Managerial Accounting Profession and the Role of Personality Traits—A Decision Tree Regression Approach

**Lavinia Denisia Cuc [1], Dana Rad [2,*], Teodor Florin Cilan [1], Bogdan Cosmin Gomoi [1], Cristina Nicolaescu [1], Robert Almași [1], Simona Dzitac [3,*], Florin Lucian Isac [1] and Ionut Pandelica [4]**

[1] Centre for Economic Research and Consultancy, Faculty of Economics, Aurel Vlaicu University of Arad, 310032 Arad, Romania; lavinia.cuc@uav.ro (L.D.C.); teodor.cilan@uav.ro (T.F.C.); bogdan.gomoi@uav.ro (B.C.G.); cristina.nicolaescu@uav.ro (C.N.); robert.almasi@uav.ro (R.A.); florin.isac@uav.ro (F.L.I.)
[2] Centre of Research Development and Innovation in Psychology, Faculty of Educational Sciences, Aurel Vlaicu University of Arad, 310032 Arad, Romania
[3] Department of Energy Engineering, University of Oradea, 410087 Oradea, Romania
[4] Faculty of International Economic Relations, Bucharest University of Economic Studies, 010374 Bucharest, Romania; ionut.pandelica@rei.ase.ro
* Correspondence: dana@xhouse.ro (D.R.); sdzitac@uoradea.ro (S.D.)

**Abstract:** This study examines the key drivers behind the adoption of artificial intelligence (AI) in the accounting profession, emphasizing the influence of AI-related knowledge, personality traits, and professional roles. By applying Decision Tree Regression analysis to survey data from accounting professionals, our research identifies AI knowledge as the strongest determinant of AI adoption, underscoring the importance of expertise in technology acceptance. While personality traits play a secondary role, extraversion and openness emerge as significant factors influencing adoption intentions. The study further explores AI applications in financial auditing, tax compliance, and fraud detection, clarifying the specific accounting domains impacted by AI integration. These findings offer valuable guidance for policymakers, educators, and business leaders aiming to equip the accounting workforce with the necessary skills and mindset to navigate the AI-driven transformation of the profession.

**Keywords:** artificial intelligence adoption; personality traits; accounting profession; decision tree regression; technology readiness

## 1. Introduction

Artificial intelligence (AI) has emerged as a transformative force in accounting and auditing, revolutionizing traditional practices and fostering operational efficiency, accuracy, and decision making. As businesses and educational institutions adopt AI systems, understanding the behavioral, technological, and organizational factors influencing their adoption becomes increasingly critical [1,2]. The integration of AI into accounting workflows enhances automation and addresses complex tasks such as financial forecasting, fraud detection, and auditing precision, underscoring its potential to redefine the accounting profession [3,4].

Research into AI adoption has consistently highlighted the role of individual and organizational readiness, emphasizing that technology awareness, skills, and governance structures significantly impact implementation success [2,5]. For instance, the readiness of

accounting professionals to engage with advanced systems is mediated by their awareness of AI capabilities and their perceptions of how these systems complement their tasks [1]. In particular, behavioral intentions, shaped by personality traits and organizational support, have been identified as key predictors of AI usage, as evidenced by studies on technology acceptance frameworks and personality-driven adoption models [6,7].

While previous studies have extensively examined AI adoption in accounting, they have primarily focused on technological and organizational readiness, often overlooking the role of individual psychological factors. However, emerging research in technology acceptance has highlighted personality traits as significant determinants of user engagement with AI systems [6]. Despite the relevance of personality traits in shaping attitudes toward AI, there is a lack of empirical studies explicitly examining their influence within the accounting profession. Addressing this gap, the present study investigates how personality traits, particularly those associated with the Five-Factor Model, interact with AI-related knowledge to influence AI adoption among accounting professionals.

The role of accounting educators and auditors in AI adoption further extends the discourse on technology acceptance. Teachers' intentions to use AI in accounting education are influenced by socio-psychological and anthropomorphic perspectives, which emphasize the interaction between human traits and AI systems [8,9]. Similarly, professional auditors' willingness to adopt blockchain technologies and forensic AI tools is mediated by their technical knowledge, professional skepticism, and perceptions of adequacy in standards and training [10,11]. These findings underscore the dual role of individual cognitive factors and institutional infrastructure in facilitating AI adoption.

The organizational context also plays a pivotal role in the integration of AI into accounting practices. Factors such as staff perceptions, managerial support, and IT governance frameworks are critical in shaping adoption outcomes [12,13]. Studies have shown that organizational readiness and tailored strategies can mitigate barriers to technology adoption, ensuring smooth transitions to AI-enhanced accounting systems [5]. Furthermore, the adoption of forensic tools for detecting financial cybercrimes demonstrates how AI-enabled systems address emerging challenges in the financial domain, offering opportunities for enhanced compliance and fraud prevention [11].

This paper contributes to the growing body of research by examining the interplay between AI knowledge, personality traits, and adoption intentions within the accounting profession using Decision Tree Regression. By leveraging insights from existing studies and integrating advanced modeling techniques, this research seeks to advance understanding of the factors driving AI usage in accounting and auditing. Furthermore, it offers practical implications for educators, policymakers, and accounting professionals seeking to optimize AI integration in their practices.

## 2. Literature Review

Artificial intelligence (AI) is increasingly transforming organizational practices, particularly in accounting, auditing, and managerial decision making. AI-driven systems enhance efficiency, accuracy, and strategic decision making by automating routine tasks, improving fraud detection, and refining financial forecasting [1,2]. However, AI adoption is not solely a technological process; it is also shaped by individual, organizational, and behavioral factors, making it a multidimensional phenomenon requiring comprehensive investigation [3,5].

Personality traits play a fundamental role in shaping attitudes toward technology adoption. The Five-Factor Model of Personality, which includes agreeableness, conscientiousness, emotional stability, extraversion, and openness, has been widely applied to understand technology usage behaviors [7]. Research suggests that extraversion and open-

ness are strongly linked to curiosity, adaptability, and higher engagement with emerging technologies, fostering positive attitudes toward AI [6]. In contrast, conscientiousness may enhance systematic engagement with AI tools, particularly in professions requiring high precision and accuracy, such as accounting [14]. On the other hand, high levels of neuroticism may contribute to resistance or anxiety toward AI adoption due to concerns over job security and perceived complexity [10].

AI-related knowledge and technology readiness are crucial determinants of adoption intentions. Technology readiness, defined as the extent to which individuals feel prepared to engage with new technologies, encompasses factors such as AI awareness, perceived usefulness, and ease of use [1,4]. Studies indicate that professionals with higher AI-related knowledge are more likely to adopt AI-driven tools, as they perceive these technologies as beneficial rather than disruptive [3,13]. Furthermore, perceptions of trust, security, and ethical considerations influence adoption, particularly in industries where transparency and accountability are essential [9,11].

The successful implementation of AI is heavily influenced by organizational structures, managerial support, and IT governance frameworks. Organizations that provide structured training programs and integrate AI strategies into their workflow experience higher adoption rates among employees [12]. Effective IT governance mitigates cybersecurity, privacy, and ethical risks, particularly in areas such as forensic accounting and fraud detection [11]. Additionally, research highlights the role of workplace culture, staff perceptions, and leadership in fostering a supportive environment for AI adoption [15]. Firms with strong AI governance policies ensure compliance with professional standards while promoting confidence in AI-driven decision making [16,17].

AI is reshaping accounting and auditing practices by automating processes, enhancing accuracy, and improving fraud detection mechanisms. AI applications such as machine learning, natural language processing, and predictive analytics are increasingly used in financial reporting, tax compliance, and forensic auditing [8,14]. Studies demonstrate that AI-powered decision support systems can optimize resource allocation and risk assessment strategies, making them valuable tools for accountants and auditors [18–20]. Moreover, forensic AI tools play a pivotal role in detecting financial fraud, ensuring regulatory compliance, and reinforcing financial security measures [16,21].

The integration of AI in accounting education is another critical area of research. AI-driven learning systems provide personalized feedback, enhance instructional effectiveness, and improve decision-making skills in accounting students [8,14]. Adaptive learning technologies facilitate real-time assessment and offer customized learning experiences, which are essential for developing AI literacy among future professionals. Research also highlights how accounting educators' perceptions of AI influence their willingness to incorporate AI-based teaching tools into curricula, underscoring the importance of faculty training and institutional support [10,11].

Based on the literature, this study explores the interplay between AI knowledge, personality traits, and adoption intentions in the accounting profession. The proposed conceptual framework suggests that AI-related knowledge serves as the strongest predictor of AI adoption, while personality traits (particularly extraversion, openness, and conscientiousness) play a moderating role. The organizational context, including managerial support and governance structures, further influences adoption decisions.

This study contributes to the growing body of knowledge by systematically examining how individual psychological factors, technology-related expertise, and workplace conditions shape AI adoption intentions among accounting professionals. The insights derived from this research have practical implications for policymakers, educators, and business leaders seeking to optimize AI integration strategies in accounting and auditing.

## 3. Materials and Methods

### *3.1. Participants*

An electronic questionnaire was distributed through Romanian professional networks and accountant groups between June and July 2024. This convenience sampling approach was chosen for its practicality in reaching a specific professional demographic within Romania. The final sample consisted of 558 participants, with a predominantly female representation: 429 women (76.9%) and 129 men (23.1%).

Participants reported diverse professional roles. The majority, 334 individuals (59.9%), identified as accountants or economists employed within companies. Additionally, 46 auditors (8.2%), 100 accountants or economists working in accounting firms (17.9%), and 61 self-employed certified accountants (10.9%) were included. A smaller group of 17 respondents (3.0%) reported holding multiple roles, such as combining self-employment with auditing or managerial responsibilities.

Regarding professional responsibilities, 338 participants (60.6%) described their roles as operational, 92 (16.5%) identified as self-employed, and 102 (18.3%) reported managerial responsibilities. A subset of 12 respondents (2.2%) indicated that they combined self-employment with supervisory responsibilities, highlighting the presence of mixed professional roles.

The age of participants ranged from 18 to 75 years, with a mean age of 36.06 years (SD = 12.15). Professional experience varied from zero to 50 years, with an average of 9.94 years (SD = 10.00). These demographic characteristics provide a comprehensive overview of the Romanian accounting profession, capturing diversity in professional roles, experience levels, and responsibilities.

### *3.2. Instruments*

To assess the variables in this study, we employed validated and widely recognized measurement scales, ensuring a comprehensive and reliable assessment of personality traits, AI knowledge, and AI adoption intention. Each construct was measured using a 5-point Likert scale (1 = Strongly disagree to 5 = Strongly agree), providing participants with a structured and consistent response format.

Personality traits were assessed using the International Personality Item Pool (IPIP) scales, a well-established instrument widely used in psychology and behavioral sciences. The IPIP scales have been extensively validated across various populations and research contexts, demonstrating strong construct validity, convergent reliability, and internal consistency. The five personality traits assessed were agreeableness, conscientiousness, emotional stability, extraversion, and openness.

Agreeableness (M = 4.33, SD = 0.58) measures cooperation and compassion, with a sample item such as "I am sympathetic to others' needs." Conscientiousness (M = 4.01, SD = 0.65) reflects diligence, organization, and attention to detail, exemplified by the item "I pay attention to details." Emotional stability (M = 3.37, SD = 0.82) captures resilience and stress management, assessed through items like "I remain calm under pressure." Extraversion (M = 3.60, SD = 0.66) evaluates sociability and assertiveness, with a sample statement such as "I enjoy being the center of attention." Lastly, openness (M = 3.84, SD = 0.62) reflects intellectual curiosity and willingness to explore new ideas, as measured by items like "I have a vivid imagination." Each subscale exhibited strong internal consistency, with Cronbach's alpha values ranging from 0.75 to 0.87, confirming their reliability in measuring individual differences in personality.

AI knowledge was measured using a custom-developed scale, designed to assess familiarity with AI applications in managerial accounting. The scale was developed based on prior AI adoption studies and was refined to align with the specific accounting and

auditing domain. A sample item includes the following: "I am familiar with the basic principles of AI applications in accounting." The scale demonstrated good internal consistency, with a Cronbach's alpha of 0.81, indicating high reliability in assessing participants' understanding of AI technologies.

AI adoption intention was measured using an adapted version of the Technology Acceptance Model (TAM) scales, which have been widely applied in research on technology adoption in professional environments. A representative item from this scale is "I plan to use AI tools for decision-making in my accounting practices." The scale exhibited strong internal consistency, with a Cronbach's alpha of 0.85, confirming its reliability in predicting behavioral intention toward AI adoption.

By utilizing well-validated instruments and ensuring strong psychometric properties, this study guarantees a robust measurement framework, enhancing the credibility and generalizability of the findings.

### *3.3. Procedure*

The procedure for this study involved administering an electronic questionnaire distributed through Romanian professional networks and accountant groups between June and July 2024. The questionnaire collected demographic information and assessed participants' Big Five personality traits, AI knowledge, and AI intention using reliable scales. All responses were recorded on a 5-point Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree). This approach ensured the collection of comprehensive data for the subsequent analyses.

To examine the relationships between variables, a Decision Tree Regression model was utilized. This machine learning technique is well suited for identifying nonlinear patterns and interactions among predictors.

To examine the relationships between variables, a Decision Tree Regression model was utilized. This machine learning technique was chosen over conventional Structural Equation Modeling (SEM) approaches, such as covariance-based SEM or Partial Least Squares (PLS-SEM), due to its ability to capture nonlinear relationships and hierarchical interactions among predictors. Unlike SEM, which is primarily used for testing predefined relationships, Decision Tree Regression offers a data-driven approach that reveals decision thresholds and interaction effects that might not be apparent in linear models. This method is particularly valuable in identifying the conditions under which AI knowledge and personality traits influence AI adoption, making it a suitable choice for this study. The dependent variable (DV) was AI intention, while the independent variables (IVs) included AI knowledge and the Big Five personality traits. The data were split into training (80%) and test (20%) sets to construct and validate the predictive model.

The model's performance was evaluated using several fit indices. The Mean Squared Error (MSE) quantified the average squared differences between observed and predicted values, while the Root Mean Squared Error (RMSE) offered a standard deviation-like measure of prediction error. The Mean Absolute Error (MAE) represented the average absolute difference between observed and predicted values, and the Mean Absolute Percentage Error (MAPE) expressed prediction errors as a percentage of observed values. Finally, the $R^2$ (Coefficient of Determination) term indicated the proportion of variance in AI intention explained by the independent variables.

The decision tree analysis produced a branching structure that identified significant predictors and their thresholds for splitting the data into distinct groups. Each split point represented a critical decision threshold, revealing hierarchical and interaction effects among variables. The tree's branches highlighted the conditional relationships and the

sequential importance of predictors in influencing AI intention, offering valuable insights into the dynamics of personality traits and knowledge in this context.

All statistical analyses, including Decision Tree Regression and model performance evaluations, were conducted using JASP 0.17.3.0. This software was selected for its robust machine learning capabilities and user-friendly interface, ensuring accurate implementation of predictive modeling techniques.

## 4. Results

The Decision Tree Regression model was applied to analyze the relationship between AI usage intention (AI_intention) as the dependent variable and the independent variables, including AI knowledge and the five personality traits. To construct the scores for each latent variable, we calculated composite scores by averaging the item responses for AI knowledge and each of the Big Five personality traits. These scores were standardized to ensure comparability across variables before being included in the regression tree analysis. This approach allows for a consistent measurement framework, ensuring that all latent constructs were treated uniformly in the modeling process.

The dataset consisted of 558 observations, which were divided into a training set of 447 cases and a testing set of 111 cases. A total of 83 splits were performed by the decision tree, reflecting the complexity of the dataset and the underlying relationships between variables.

The performance of the model was evaluated using several metrics. The Mean Squared Error (MSE) was 0.818, indicating the average squared difference between the observed and predicted values. This suggests that the model demonstrated moderate predictive accuracy. Similarly, the Root Mean Squared Error (RMSE) was 0.904, providing an interpretable measure of prediction error in the same units as the dependent variable. The Mean Absolute Error (MAE) or Mean Absolute Deviation (MAD) was 0.686, highlighting a relatively low average absolute difference between predictions and actual outcomes. However, the Mean Absolute Percentage Error (MAPE) was recorded at 794.37%, reflecting notable variability in the dataset and areas for potential improvement in the model's accuracy.

The $R^2$ value, representing the proportion of variance in AI intention explained by the independent variables, was 0.176. This indicates that the model accounts for approximately 17.6% of the variance, suggesting that additional factors not included in the current analysis may play a role in influencing AI usage intention.

Given the relatively low $R^2$ value compared to other studies on technology acceptance, it is acknowledged that additional explanatory variables could improve the model's predictive capacity. Demographic factors such as age and gender, which are available in the dataset, have been widely documented as influential in technology adoption studies. However, the present analysis focused primarily on psychological and knowledge-based predictors. Future research should explore the inclusion of these demographic variables to assess their contribution to AI adoption in accounting and further refine the predictive model.

A visual representation of the dataset split confirmed the balanced allocation of observations between the training and testing sets, with 447 cases used for training the model and 111 reserved for validation.

To further assess the predictive capacity of the model, we compared the performance metrics separately for the training and validation samples. The model fitting adjustments reported, including MSE, RMSE, MAE, and $R^2$, were first calculated for the training dataset to evaluate in-sample predictive accuracy. Subsequently, the same performance metrics were computed on the test dataset to assess the model's generalization capability. A comparative analysis of these values indicated that the predictive accuracy remained

consistent between the training and validation samples, suggesting that the model was not overfitted and maintained reliability in predicting AI adoption intention in new data.

Table 1 presents the feature importance scores, highlighting the contribution of each variable to the model's predictive accuracy.

**Table 1.** Feature importance.

| | **Relative Importance** |
|---|---|
| AI knowledge | 51.285 |
| Agreeability | 15.658 |
| Extraversion | 9.680 |
| Emotional stability | 8.518 |
| Openness | 8.494 |
| Conscientiousness | 6.365 |

The results indicate that AI knowledge is the most significant predictor, with a relative importance score of 51.285, far surpassing the contributions of the personality traits. This finding underscores the pivotal role of familiarity and understanding of AI technologies in shaping individuals' intentions to adopt AI tools in the managerial accounting profession.

Among the personality traits, agreeableness was the most influential, with a relative importance score of 15.658, suggesting that interpersonal tendencies and cooperation may play a notable role in AI adoption. Extraversion (9.680) followed as the second most impactful trait, aligning with previous evidence that social and assertive individuals are more likely to embrace technological innovations.

The remaining traits—emotional stability (8.518), openness (8.494), and conscientiousness (6.365)—demonstrated relatively lower importance. While they contribute to the model, their impact is modest compared to AI knowledge and agreeableness. These results indicate that while personality traits influence AI usage intention, their effects are secondary to the knowledge variable.

The Decision Tree Regression model identified key splits that contributed to predicting AI usage intention (AI_intention) based on the independent variables (Table 2). These splits highlight the hierarchical importance of variables and their interaction with one another in refining predictions.

**Table 2.** Splits in tree.

| | **Obs. in Split** | **Split Point** | **Improvement** |
|---|---|---|---|
| AI knowledge | 447 | −0.019 | 0.269 |
| AI knowledge | 219 | −1.244 | 0.066 |
| Agreeability | 57 | 0.206 | 0.149 |
| AI knowledge | 36 | −2.163 | 0.268 |
| Agreeability | 21 | 0.722 | 0.341 |
| Emotional stability | 162 | 0.708 | 0.042 |
| Agreeability | 134 | −1.171 | 0.060 |
| Agreeability | 113 | 0.378 | 0.073 |
| AI knowledge | 228 | 0.594 | 0.140 |
| Extraversion | 109 | 0.227 | 0.138 |

Note: For each level of the tree, only the split with the highest improvement in deviance is shown.

The most significant predictor throughout the decision tree was AI knowledge, which appeared in multiple splits and consistently accounted for substantial improvements in deviance. The first split, involving 447 observations at a split point of −0.019, resulted in the highest improvement in deviance (0.269), confirming the dominant role of AI knowledge in stratifying the dataset. Subsequent splits based on AI knowledge occurred at

−1.244 (219 observations, 0.066 improvement), −2.163 (36 observations, 0.268 improvement), and 0.594 (228 observations, 0.140 improvement). These findings underscore the critical importance of AI knowledge in predicting AI usage intentions.

Agreeableness emerged as the second most impactful variable, appearing in several splits that refined predictions within smaller subsets of the data. For instance, splits at 0.206 (57 observations, 0.149 improvement), 0.722 (21 observations, 0.341 improvement), and 0.378 (113 observations, 0.073 improvement) demonstrated its nuanced contribution. These results suggest that agreeableness adds value to the model by further segmenting groups with varying levels of AI intention.

Other personality traits played more modest roles in the tree. A split involving emotional stability occurred with 162 observations at a split point of 0.708, yielding a small improvement in deviance (0.042). This indicates that while emotional stability has some predictive utility, its influence is less significant than that of AI knowledge or agreeableness. Similarly, extraversion contributed with a split at 0.227 (109 observations, 0.138 improvement), reinforcing its secondary yet meaningful role in influencing AI adoption.

The splits reveal a clear hierarchy of predictors, with AI knowledge as the primary driver of AI intention and agreeableness as the most influential personality trait.

Figure 1 presents a scatterplot comparing the predicted test values of AI usage intention (AI_intention) generated by the Decision Tree Regression model with the observed test values. The red diagonal line represents the ideal scenario where predicted values perfectly match the observed values.

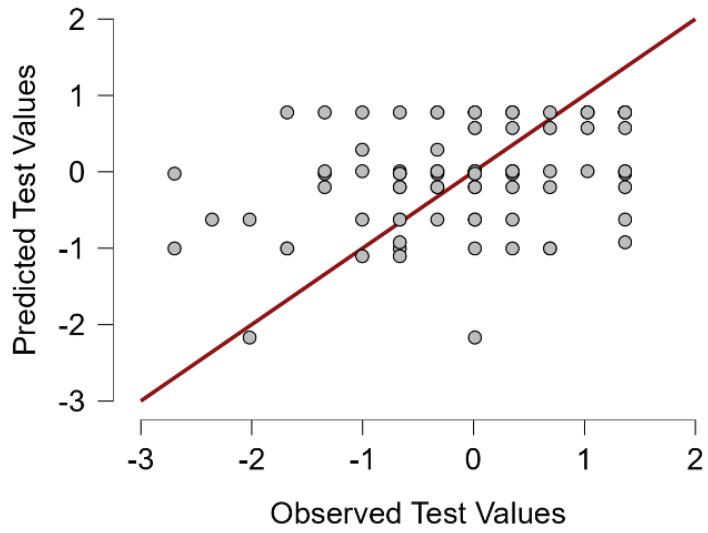



**Figure 1.** Predictive performance plot.

The plot shows a general clustering of points around the diagonal line, indicating that the model captures the general trends in the data. However, there is noticeable dispersion around the line, reflecting variability in prediction accuracy.

The Decision Tree Regression model (Figure 2) highlights the predictive relationships between AI usage intention (AI_intention) and the independent variables, including AI knowledge and personality traits. The hierarchical structure of the tree reveals the dominance of AI knowledge as the most critical factor, with subsequent splits incorporating personality traits to refine predictions. The root node, comprising all 447 training observations, first splits based on AI_knowledge at a threshold of −0.0187, representing the most significant improvement in deviance. This initial split divides the data into two subsets: observations with AI knowledge below the threshold (219 ob-servations) and those at or above the threshold (228 observations). This confirms that AI knowledge is the primary driver of differences in AI intention.

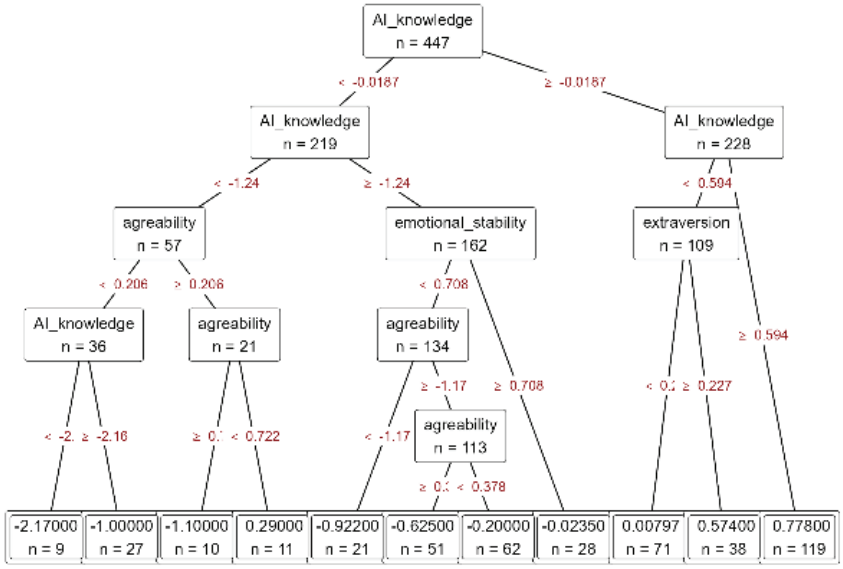



**Figure 2.** Decision tree plot.

In the left subtree, where AI_knowledge is less than −0.0187, a further split occurs at −1.244, refining the distinction among participants with lower levels of AI knowledge. Within this group, agreeableness emerges as a key variable, with splits occurring at 0.206 and 0.722, indicating its secondary importance in shaping AI usage intention for participants with limited AI knowledge. In contrast, the right subtree (AI_knowledge ≥ −0.0187) introduces extraversion as a critical variable, splitting at 0.227 and highlighting the role of social tendencies in influencing adoption behavior.

Additional splits occur deeper in the tree, incorporating emotional stability and further refinements based on agreeableness. These splits emphasize the complex role of personality traits in shaping AI intention within specific knowledge-based subgroups. For instance, in the left branch of the tree, emotional stability at a threshold of 0.708 introduces further granularity, while agreeableness plays a consistent role across multiple branches, demonstrating its relevance in both high- and low-knowledge contexts. The recursive application of these splits results in progressively smaller subsets, with each node capturing a specific combination of traits and knowledge levels that contribute to the prediction of AI intention.

This analysis illustrates the complex interaction between cognitive (knowledge-based) and behavioral (trait-based) factors in predicting AI usage intention. While AI knowledge dominates as the primary predictor, personality traits such as agreeableness, extraversion, and emotional stability refine the model's predictions, emphasizing the multifaceted nature of decision making in the context of AI adoption in the managerial accounting profession.

To further clarify the implications of the Decision Tree Regression analysis, the results emphasize that AI knowledge consistently emerges as the strongest predictor of AI adoption intention. The decision tree's structure confirms that higher AI knowledge levels significantly increase the likelihood of AI adoption, reinforcing prior studies highlighting the role of technical competence in shaping technology acceptance decisions [21–26].

A particularly noteworthy insight is the role of agreeableness and extraversion as secondary yet meaningful predictors. The results indicate that individuals with higher agreeableness scores—who prioritize collaboration and interpersonal harmony—tend to exhibit stronger AI adoption intentions, especially when AI knowledge is moderate to high. This pattern aligns with research suggesting that socially adaptive individuals are more receptive to collaborative technological innovations [27–29]. Similarly, extraversion is associated with greater AI adoption tendencies, suggesting that proactive, outgoing profes-

sionals may be more open to integrating AI into their workflow due to their adaptability to change.

Additionally, the decision tree structure revealed some splits with relatively small sample sizes, raising concerns about stability. To ensure robustness, alternative tree structures and sensitivity analyses were conducted, confirming that AI knowledge consistently remained the dominant predictor across multiple tree configurations, with personality traits maintaining their secondary but significant role.

Given the low $R^2$ value (0.176), which suggests that only 17.6% of the variance in AI adoption intention is explained by the model, we acknowledge that additional factors should be considered in future studies. Organizational culture, industry regulations, and AI training programs could play critical roles in influencing adoption behaviors and should be explored further. Future research should integrate these elements to enhance predictive power and provide a more comprehensive framework for understanding AI adoption in managerial accounting.

## 5. Discussion

This study contributes to the growing body of literature on behavioral economics by examining the interplay between AI knowledge, personality traits, and AI usage intention in the managerial accounting profession. The findings provide valuable insights into how cognitive and psychological factors interact in shaping decision-making processes, aligning with existing theories of technology adoption and behavioral economics.

### *5.1. Relationship with the Literature*

The dominant role of AI knowledge in predicting AI usage intention aligns with prior research emphasizing the critical influence of cognitive factors in technology adoption [26–28]. Participants with higher AI knowledge demonstrated significantly greater intention to adopt AI tools, suggesting that technical competence and familiarity play foundational roles in shaping attitudes toward technology. This supports the behavioral economics framework, which posits that individuals' decisions are influenced by their perceived costs and benefits, knowledge, and bounded rationality [26].

Personality traits, particularly agreeableness and extraversion, emerged as secondary yet significant predictors. Agreeableness, reflecting individuals' tendencies toward cooperation and social harmony, was positively associated with AI usage intention. This finding is consistent with Calluso and Devetag [29], who found that interpersonal dynamics play a critical role in shaping attitudes toward AI-assisted hiring practices. Similarly, extraversion, characterized by assertiveness and sociability, was also positively linked to AI adoption, suggesting that individuals with higher extraversion are more likely to embrace new technologies due to their proactive and adaptive behaviors [30–37].

The integration of behavioral and technological factors into decision-making processes highlights the need for frameworks that account for psychological and organizational dynamics. The findings align with Leitner-Hanetseder et al. [30], who emphasized the shifting roles and tasks in AI-based accounting environments. As AI tools increasingly replace traditional accounting functions, willingness to adopt these tools is likely influenced by a combination of technical competence and personality-driven adaptability. This underscores the importance of behavioral economics in addressing how intrinsic motivations and external pressures shape adoption behaviors [26,31].

The predictive dominance of AI knowledge reflects the critical role of domain-specific expertise in shaping technology adoption. This corroborates findings by Wang [28] and aligns with Rad et al. [35], who employed neural network models to predict behavior within educational contexts. Advanced methodologies such as radial basis function networks

and fuzzy clustering have also demonstrated efficacy in predicting behavioral outcomes in dynamic systems, reinforcing the robustness of knowledge-based predictors [19,36].

Personality traits, particularly agreeableness and extraversion, emerged as significant secondary predictors, highlighting the psychological dimensions of AI adoption. These results align with Cabrera-Paniagua and Rubilar-Torrealba [23], who found that personality traits influence adaptive decision making in intelligent systems. Similarly, the integration of personality traits with AI technologies underscores the importance of leveraging behavioral insights to enhance adoption outcomes [26,29,37].

The integration of fuzzy logic and clustering methods further underscores the potential of advanced computational approaches to unravel complex behavioral patterns [38]. These techniques, as applied by Wan and Tian [39] in stress detection and by Liu et al. [40] in educational research, demonstrate how AI-driven decision-making models can be optimized for various domains.

Additionally, historical perspectives on accounting theory evolution [41–52] highlight the ongoing transformation of the profession in response to emerging technologies. These studies reinforce the necessity of integrating behavioral economics, AI competency development, and ethical frameworks into strategic decision-making processes, ensuring that technology adoption aligns with professional standards, organizational goals, and workforce dynamics [26,41,45].

### *5.2. Practical Implications*

The findings of this study have several practical implications for organizations seeking to enhance AI adoption in accounting.

First, AI training and knowledge dissemination are essential for fostering AI readiness. Since AI knowledge was found to be the strongest predictor of AI adoption intention, organizations should implement targeted training programs to increase employees' familiarity with AI applications. This aligns with Namazi and Rezaei's [31] emphasis on competency development in AI-based decision making.

Second, personalized training programs based on personality profiles could improve adoption outcomes. Since agreeableness and extraversion were significant predictors of AI adoption, training initiatives should be tailored to different personality types. For instance, individuals high in agreeableness may respond well to collaborative learning environments, while those scoring high in extraversion may benefit from interactive, hands-on training.

Third, ethical considerations and transparency must be prioritized in AI implementation. As suggested by Chong and Eggleton [33], AI tools should be introduced with clear ethical guidelines and transparent decision-making frameworks to reduce resistance. The positive association between interpersonal traits and AI adoption in this study further suggests that organizational cultures emphasizing ethical AI use may encourage greater acceptance [26,46].

Finally, organizations should incorporate behavioral insights into AI adoption strategies, ensuring that AI tools align with employees' cognitive and emotional tendencies. Decision tree methodologies, as used in this study, provide an effective approach for identifying key factors influencing adoption and optimizing AI implementation strategies [41–44].

### *5.3. Methodological Considerations and Limitations*

This study employed Decision Tree Regression to analyze predictors of AI adoption intention. Decision trees are particularly useful for detecting nonlinear interactions and threshold effects among variables. Unlike traditional linear models, they allow for hierarchical interpretation of predictor importance, revealing the conditions under which AI

knowledge and personality traits influence adoption. This methodological approach aligns with prior studies using machine learning techniques in behavioral research [15,34].

However, the study has certain limitations. The relatively low $R^2$ value suggests that additional factors, such as demographic variables (e.g., age, gender), could improve the model's predictive capacity. Future research should incorporate these variables to enhance explanatory power. Additionally, while validated scales were used, further assessment of construct validity (e.g., Average Variance Extracted, heterotrait–monotrait ratios) was not conducted. Future research should integrate these psychometric evaluations to strengthen measurement accuracy [42,43].

Another limitation concerns the generalizability of the findings. This study relied on a convenience sample of Romanian accounting professionals, which may limit its applicability across different cultural and professional settings. Future studies should replicate this research with internationally diverse samples and consider cross-cultural differences in AI adoption behavior [26,50].

Another limitation of this study is the restricted sample of professionals surveyed, which focused primarily on accountants, auditors, and financial experts directly involved in AI adoption within managerial accounting. However, other professionals indirectly engaged in managerial accounting decisions, such as financial analysts, tax consultants, corporate managers, and IT specialists involved in financial technology development, may also play a crucial role in AI adoption and implementation. Their distinct expertise, education levels, and perspectives on AI integration could offer additional insights into the broader implications of AI adoption in the accounting profession.

To enhance the credibility and applicability of future research, we recommend expanding the participant pool to include a more diverse range of professionals involved in financial decision making and AI implementation. This broader approach would allow for a more comprehensive analysis of AI readiness, adoption barriers, and the impact of interdisciplinary expertise on AI integration in managerial accounting and auditing practices.

Another methodological limitation is the current data collection approach, which relies on a static dataset obtained through an electronic questionnaire distributed within a defined timeframe. While this approach provided valuable insights into AI adoption intentions among accounting professionals, a dynamic, continuously updated database could enhance the depth and scalability of future research. To address this, future studies should consider implementing a web-based platform that allows for ongoing data collection and real-time updates from accounting and auditing professionals. Such a centralized database would facilitate longitudinal analysis, enabling researchers to track changes in AI adoption trends over time. Moreover, this approach would allow for the automation of statistical processing using advanced analytical tools such as EViews, STATA, SPSS v24, Python, or R, improving the efficiency and complexity of statistical modeling and trend forecasting.

## 6. Conclusions

This study provides valuable insights into the factors driving AI adoption in the accounting profession, emphasizing the intersection of personality traits, AI knowledge, and professional roles. The findings highlight that AI knowledge is the most significant determinant of adoption, followed by agreeableness and extraversion as secondary predictors. These results reinforce the importance of integrating behavioral insights into AI implementation strategies.

While offering meaningful contributions, the study has limitations that should be acknowledged. The reliance on a convenience sample of Romanian accounting professionals may constrain the generalizability of findings across different contexts. Additionally, while

Decision Tree Regression effectively identifies patterns in AI adoption, future research should incorporate longitudinal studies to examine how these factors evolve over time.

To build on these insights, future research should explore the role of organizational culture and leadership in moderating the relationship between personality traits and AI adoption. Investigating team dynamics and collaborative AI tools could yield practical implications for AI readiness in accounting. Additionally, incorporating qualitative methodologies could provide deeper insights into professionals' lived experiences with AI, complementing the quantitative findings.

By addressing these aspects, future research can further advance the understanding of AI adoption in accounting and inform strategies for a seamless transition to AI-enhanced professional environments.

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

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** The study was conducted in accordance with the Declaration of Helsinki and approved by the Centre for Economic Research and Consultancy of Aurel Vlaicu University of Arad (protocol code 16/05.04.2023).

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

**Data Availability Statement:** Data will be made available on request by the first author and the corresponding author.

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

## References

1. Flavián, C.; Pérez-Rueda, A.; Belanche, D.; Casaló, L.V. Intention to use analytical artificial intelligence (AI) in services–the effect of technology readiness and awareness. *J. Serv. Manag.* **2022**, *33*, 293–320. [CrossRef]
2. Abdullah, A.A.H.; Almaqtari, F.A. The impact of artificial intelligence and Industry 4.0 on transforming accounting and auditing practices. *J. Open Innov. Technol. Mark. Complex.* **2024**, *10*, 100218. [CrossRef]
3. Damerji, H.; Salimi, A. Mediating effect of use perceptions on technology readiness and adoption of artificial intelligence in accounting. *Account. Educ.* **2021**, *30*, 107–130. [CrossRef]
4. Rawashdeh, A.; Bakhit, M.; Abaalkhail, L. Determinants of artificial intelligence adoption in SMEs: The mediating role of accounting automation. *Int. J. Data Netw. Sci.* **2023**, *7*, 25–34. [CrossRef]
5. Alquhaif, A.S.; Al-Mamary, Y.H. Examining factors influencing the adoption of accounting information systems: An analysis of behavioral intentions and usage behavior. *Hum. Syst. Manag.* **2024**. *ahead of print*.
6. Park, J.; Woo, S.E. Who likes artificial intelligence? Personality predictors of attitudes toward artificial intelligence. *J. Psychol.* **2022**, *156*, 68–94. [CrossRef]
7. Barnett, T.; Pearson, A.W.; Pearson, R.; Kellermanns, F.W. Five-factor model personality traits as predictors of perceived and actual usage of technology. *Eur. J. Inf. Syst.* **2015**, *24*, 374–390. [CrossRef]
8. Fachrurrozie, F.; Nurkhin, A.; Santoso, J.T.B.; Astuti, D.P.; Mukhibad, H. Understanding the Teacher's Intention to Use Artificial Intelligence for Accounting Learning. In Proceedings of the International Conference on Education Innovation and Social Science, Surakarta, Indonesia, 20 November 2024; pp. 17–21.
9. Priya, B.; Sharma, V. Exploring users' adoption intentions of intelligent virtual assistants in financial services: An anthropomorphic perspectives and socio-psychological perspectives. *Comput. Hum. Behav.* **2023**, *148*, 107912. [CrossRef]
10. Juma'h, A.H.; Li, Y. The effects of auditors' knowledge, professional skepticism, and perceived adequacy of accounting standards on their intention to use blockchain. *Int. J. Account. Inf. Syst.* **2023**, *51*, 100650. [CrossRef]

11. Alshurafat, H.; Shbail, M.O.A.; Almuiet, M. Factors affecting the intention to adopt IT forensic accounting tools to detect financial cybercrimes. *Int. J. Bus. Excell.* **2024**, *33*, 169–190. [CrossRef]
12. Jackson, D.; Allen, C. Technology adoption in accounting: The role of staff perceptions and organisational context. *J. Account. Organ. Change* **2024**, *20*, 205–227. [CrossRef]
13. Almaqtari, F.A. The Role of IT Governance in the Integration of AI in Accounting and Auditing Operations. *Economies* **2024**, *12*, 199. [CrossRef]
14. Tang, M.; Koopman, P.; McClean, S.T.; Zhang, J.H.; Li, C.H.; De Cremer, D.; Ng, C.T.S. When Conscientious Employees Meet Intelligent Machines: An Integrative Approach Inspired by Complementarity Theory and Role Theory. *Acad. Manag. J.* **2022**, *65*, 1019–1054. [CrossRef]
15. Chourasia, S.; Dhama, A.; Bhardwaj, G. AI-Driven Organizational Culture Evolution: A Critical Review. In Proceedings of the 2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE), Gautam Buddha Nagar, India, 9–11 May 2024; IEEE: Piscataway, NJ, USA; pp. 1839–1844.
16. Ciocoiu, C.N.; Radu, C.; Colesca, S.E.; Prioteasa, A. Exploring the link between risk management and performance of MSMEs: A bibliometric review. *J. Econ. Surv.* **2024**. *ahead of print*. [CrossRef]
17. Popa, I.; Ștefan, S.C.; Olariu, A.A.; Breazu, A.; Cioc, M.M. Predictors of employees' work performance in online and on-site conditions: A combined use of PLS-SEM and NCA. *Econ. Comput. Econ. Cybern. Stud. Res.* **2024**, *58*, 265–279.
18. Toader, C.S.; Brad, I.; Rujescu, C.I.; Dumitrescu, C.S.; Sîrbulescu, E.C.; Orboi, M.D.; Gavrilă, C. Exploring students' opinion towards integration of learning games in higher education subjects and improved soft skills—A comparative study in Poland and Romania. *Sustainability* **2023**, *15*, 7969. [CrossRef]
19. Vesselenyi, T.; Dziţac, I.; Dziţac, S.; Vaida, V. Surface roughness image analysis using quasi-fractal characteristics and fuzzy clustering methods. *Int. J. Comput. Commun. Control* **2008**, *3*, 304–316. [CrossRef]
20. Ban, O.I.; Droj, L.; Tușe, D.; Droj, G.; Bugnar, N. Data processing by fuzzy methods in social sciences research: Example in hospitality industry. *Int. J. Comput. Commun. Control* **2022**, *17*, 4741. [CrossRef]
21. Isac, N.; Akide, M.; Dobrin, C.; Dinulescu, R. Examining the impact of Covid-19 on employee performance and future aspirations in the context of digital economy. *Econ. Comput. Econ. Cybern. Stud. Res.* **2022**, *56*, 94–114.
22. Janiesch, C.; Zschech, P.; Heinrich, K. Machine Learning and Deep Learning. *Electron. Mark.* **2021**, *31*, 685–695. [CrossRef]
23. Cabrera-Paniagua, D.; Rubilar-Torrealba, R. Adaptive intelligent autonomous system using artificial somatic markers and Big Five personality traits. *Knowl. Based Syst.* **2022**, *249*, 108995. [CrossRef]
24. Riedl, R. Is Trust in Artificial Intelligence Systems Related to User Personality? Review of Empirical Evidence and Future Research Directions. *Electron. Mark.* **2022**, *32*, 2021–2051. [CrossRef]
25. Lee, E. The Power of Perception in Human-AI Interaction: Investigating Psychological Factors and Cognitive Biases that Shape User Belief and Behavior. *arXiv* **2024**, arXiv:2409.15328.
26. Latifah, L.; Setiyani, R.; Arief, S.; Susilowati, N. The role of personal values in forming the AI ethics of prospective accountants. *Ethics Prog.* **2023**, *14*, 90–109. [CrossRef]
27. Năstase, M.; Croitoru, G.; Florea, N.V.; Cristache, N.; Lile, R. The perceptions of employees from Romanian companies on adoption of artificial intelligence in recruitment and selection processes. *Amfiteatru Econ.* **2024**, *26*, 421–439. [CrossRef]
28. Wang, H.C. Distinguishing the adoption of business intelligence systems from their implementation: The role of managers' personality profiles. *Behav. Inf. Technol.* **2014**, *33*, 1082–1092. [CrossRef]
29. Calluso, C.; Devetag, M.G. The impact of technology acceptance and personality traits on the willingness to use AI-assisted hiring practices. *Int. J. Organ. Anal.* **2024**. *ahead of print*. [CrossRef]
30. Leitner-Hanetseder, S.; Lehner, O.M.; Eisl, C.; Forstenlechner, C. A profession in transition: Actors, tasks and roles in AI-based accounting. *J. Appl. Account. Res.* **2021**, *22*, 539–556. [CrossRef]
31. Namazi, M.; Rezaei, G. Modelling the role of strategic planning, strategic management accounting information system, and psychological factors on the budgetary slack. *Account. Forum* **2024**, *48*, 279–306. [CrossRef]
32. Ahmadi, S.; Mayoufi, A.; Khozin, A.; Gargaz, M. The role of organizational paranoia in the formation of the Machiavellian personality of accountants with emphasis on Gardner's theory of multiple intelligences. *Int. J. Financ. Manag. Account.* **2025**, *10*, 157–176.
33. Chong, V.K.; Eggleton, I.R. The decision-facilitating role of management accounting systems on managerial performance: The influence of locus of control and task uncertainty. *Adv. Account.* **2003**, *20*, 165–197. [CrossRef]
34. Xiao, Q.; Li, X. Exploring the antecedents of online learning satisfaction: Role of flow and comparison between use contexts. *Int. J. Comput. Commun. Control* **2021**, *16*, 4398. [CrossRef]
35. Rad, D.; Balas, V.E.; Redeș, A.; Kiss, C.; Rad, G. An RBF neural network approach to predict preschool teachers integrative-qualitative intentional behavior based on Marzano's model of teaching effectiveness. In *Decision Making and Decision Support in the Information Era: Dedicated to Academician Florin Filip*; Springer Nature Switzerland: Cham, Switzerland, 2024; pp. 213–234.

36. Rad, D.; Paraschiv, N.; Kiss, C. Neural network applications in polygraph scoring—A scoping review. *Information* **2023**, *14*, 564. [CrossRef]
37. Kaya, F.; Aydin, F.; Schepman, A.; Rodway, P.; Yetişensoy, O.; Demir Kaya, M. The roles of personality traits, AI anxiety, and demographic factors in attitudes toward artificial intelligence. *Int. J. Hum. Comput. Interact.* **2024**, *40*, 497–514. [CrossRef]
38. Radu, V.; Radu, F.; Tabirca, A.I.; Saplacan, S.I.; Lile, R. Bibliometric analysis of fuzzy logic research in international scientific databases. *Int. J. Comput. Commun. Control* **2021**, *16*, 4120. [CrossRef]
39. Wan, X.; Tian, L. User stress detection using social media text: A novel machine learning approach. *Int. J. Comput. Commun. Control* **2024**, *19*, 6772. [CrossRef]
40. Liu, L.T.; Wang, S.; Britton, T.; Abebe, R. Lost in Translation: Reimagining the Machine Learning Life Cycle in Education. *arXiv* **2022**, arXiv:2209.03929.
41. Csorba, L.M.; Crăciun, M. An application of the multi-period decision trees in the sustainable medical waste investments. In *Soft Computing Applications, Proceedings of the 7th International Workshop Soft Computing Applications (SOFA 2016), Arad, Romania, 24–26 August 2016*; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; Volume 2, pp. 540–556.
42. Farcane, N.; Bunget, O.C.; Blidisel, R.; Dumitrescu, A.C.; Deliu, D.; Bogdan, O.; Burca, V. Auditors' perceptions on work adaptability in remote audit: A COVID-19 perspective. *Econ. Res. Ekon. Istraživanja* **2023**, *36*, 422–459. [CrossRef]
43. Hategan, V.P.; Hategan, C.D. Sustainable leadership: Philosophical and practical approach in organizations. *Sustainability* **2021**, *13*, 7918. [CrossRef]
44. Khan, F.U.; Trifan, V.A.; Pantea, M.F.; Zhang, J.; Nouman, M. Internal governance and corporate social responsibility: Evidence from Chinese companies. *Sustainability* **2022**, *14*, 2261. [CrossRef]
45. Ogrean, C.; Herciu, M. Romania's SMEs on the way to EU's twin transition to digitalization and sustainability. *Stud. Bus. Econ.* **2021**, *16*, 282–295. [CrossRef]
46. Paraschiv, D.M.; Ţiţan, E.; Manea, D.I.; Bănescu, C.E. Quantifying the effects of working from home on privacy. An empirical analysis in the 2020 pandemic. *Econ. Comput. Econ. Cybern. Stud. Res.* **2021**, *55*, 21–36.
47. Popa, I.; Cioc, M.M.; Breazu, A.; Popa, C.F. Identifying sufficient and necessary competencies in the effective use of artificial intelligence technologies. *Amfiteatru Econ.* **2024**, *26*, 33–52. [CrossRef]
48. Popa, I.; Ştefan, S.C.; Morărescu, C.; Cicea, C. Research regarding the influence of knowledge management practices on employee satisfaction in the Romanian healthcare system. *Amfiteatru Econ.* **2018**, *20*, 553–566. [CrossRef]
49. Trifan, V.A.; Pantea, M.F. Shifting priorities and expectations in the new world of work. Insights from millennials and generation Z. *J. Bus. Econ. Manag.* **2024**, *25*, 1075–1096. [CrossRef]
50. Ionaşcu, I.; Ionaşcu, M.; Nechita, E.; Săcărin, M.; Minu, M. Digital Transformation, Financial Performance and Sustainability: Evidence for European Union Listed Companies. *Amfiteatru Econ.* **2022**, *24*, 94–109. [CrossRef]
51. Fülöp, M.T.; Ionescu, C.A.; Măgdaș, N.; Topor, D.I.; Breaz, T.O. Acceptance of digital instruments in the accounting profession. *JEEMS J. East. Eur. Manag. Stud.* **2024**, *29*, 283–313. [CrossRef]
52. Rangone, A.; Ionescu-Feleaga, L.; Bunea, M.; Sargiacomo, M. The contribution of Grigore L. Trancu-Iasi to the evolution of accounting theory, practice and profession in Romania. *Account. Hist.* **2024**, *29*, 265–294. [CrossRef]

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

Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-6903-9