

Article

Not peer-reviewed version

A Personalized Mining Algorithm for Grassroots Network Data Based on Deep Learning

Connor Phillips and [Ank Agarwal](#)*

Posted Date: 9 January 2026

doi: 10.20944/preprints202601.0728.v1

Keywords: deep learning; neural networks; grassroots network data; data mining; pruning rules; feature selection



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

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

Article

A Personalized Mining Algorithm for Grassroots Network Data Based on Deep Learning

Connor Phillips and Ank Agarwal *

Brown University, Providence, Rhode Island, USA

* Correspondence: ankagarwal1@outlook.com

Abstract

To enhance the accuracy and efficiency of mining grassroots network data and to better support practical applications, this study proposes a personalized mining algorithm for grassroots network data based on deep learning. A multi-module neural network architecture is designed to process, filter, and transform raw grassroots network data. Through data preprocessing and hierarchical refinement, the algorithm generates high-precision, structured datasets suitable for personalized mining tasks. A five-layer neural network-comprising an input layer, convolutional input layer, hidden layer, convolutional output layer, and prediction output layer-is constructed to support an integrated training and testing workflow. A redundancy-elimination rule is introduced to prune unnecessary neural network parameters, followed by a maximum weight extraction rule to guide the personalized mining of grassroots network data. Experimental results demonstrate that the proposed algorithm achieves high convergence speed during training and offers superior performance in testing accuracy, enabling precise and reliable mining of high-dimensional and heavily interconnected data. This work lays a technical foundation for the effective utilization and intelligent analysis of grassroots network data.

Keywords: deep learning; neural networks; grassroots network data; data mining; pruning rules; feature selection

I. INTRODUCTION

With the continuous advancement of database technologies and computer network systems, various grassroots network platforms have emerged and have become an indispensable part of people's daily life and work [1]. During the operation of these platforms, diverse categories of user data are aggregated within grassroots networks. To enhance the utilization of cyberspace resources and broaden the application scope of grassroots network data, it is essential to conduct efficient data mining. Consequently, extracting the required information from massive and heterogeneous grassroots network data has become a key research challenge [2,3].

Traditional data mining approaches typically rely on inspecting data resources, categorizing them into encapsulated types, and then applying mining algorithms to each classified region. However, analysis of multiple grassroots network mining studies shows that earlier mining methods often suffer from poor logical ordering. When mining sequences contain disordered or weakly correlated data, these approaches may lead to misjudgment, reduced mining efficiency, data interruption, and information loss, ultimately decreasing the accuracy and effectiveness of mining results [4]. To address these issues, researchers have proposed various improved techniques. Decision-tree-based mining methods optimize the construction of hierarchical rules and achieve multi-level mining, resulting in improved accuracy but relatively low efficiency [5]. Machine-learning-based mining methods employ intelligent models to enhance data mining performance; however, the extracted results may still lack stability [6].

Deep learning represents a class of multi-layer learning approaches, with neural networks being one of its most representative forms. Neural networks simulate the structural and functional

mechanisms of the human brain, providing strong abilities in pattern modeling, feature extraction, and autonomous learning. Due to these advantages, neural networks demonstrate strong performance in complex data mining tasks [7,8]. The fuzzy neural network (FNN), which integrates neural computation with fuzzy logic, further enhances the capability to process uncertain information and improves learning stability and scalability in large-scale applications [9].

Based on the above analysis, this study proposes a deep-learning-based personalized mining algorithm for grassroots network data. A five-layer fuzzy neural network is constructed and trained through self-supervised learning, followed by pruning operations to extract the corresponding fuzzy neural network rules. These rules are then applied to personalized data mining tasks, enabling effective utilization of grassroots network data and supporting the expansion of its application scope.

II. RELATED WORK

The evolution of deep learning and its integration with data mining tasks have led to significant advancements in personalized information processing, particularly in network-based data systems. One of the most transformative directions in this domain involves the application of graph neural networks (GNNs) and transformer-based models. These models excel in capturing the complex relationships inherent in structured and unstructured data.

Comprehensive surveys have outlined how GNNs and graph transformers are leveraged across vision and language tasks [10,11]. Their adaptability to spatial and temporal dependencies makes them suitable for grassroots network data environments, where structural correlations are prevalent. Graph attention mechanisms have further enhanced this capacity, supporting tasks like anomaly detection and social representation learning [12–14].

The importance of pruning and quantization techniques in deep learning models, particularly GNNs, has been highlighted in recent research [15], aligning with our method's pruning phase to eliminate redundant neural connections. Foundational techniques in dropout regularization and wavelet shrinkage also contribute significantly to model optimization and generalization strategies, resonating with the denoising and redundancy-removal mechanisms in our work [16,17].

Transformer-based models have demonstrated strong performance in heterogeneous and temporal data scenarios [18,19]. Applications in risk monitoring, healthcare, and microservice routing illustrate the flexibility of transformer architectures and their synergy with graph-based structures [20–22]. The integration of retrieval-augmented generation frameworks and cloud-native observability systems reflects broader advances in deep learning's interpretability and robustness [23,24].

Graph-based methods have also proven effective across various domains, including EEG anomaly detection, credit fraud detection, and fake news classification [13,25,26]. These studies demonstrate that methodologies centered around graph structures and attention mechanisms are broadly applicable, enhancing pattern recognition and feature extraction across heterogeneous datasets.

Other foundational methods, including inductive representation learning and semi-supervised graph convolution, provide essential building blocks for learning from irregular and large-scale network data [27,28]. Early methods like DeepWalk have contributed to representation learning techniques that remain relevant in user-centric data environments [14].

Additional advances in machine learning—including matrix factorization, Bayesian inference, and causal representation learning—complement the inference and rule extraction capabilities emphasized in this work [29–31]. Techniques for symbolic-to-numerical data transformation, as well as pattern recognition, provide essential theoretical underpinnings for deep learning model design [32,33].

Finally, classic algorithms and systems such as PageRank, as well as progress in educational data mining and scalable learning models, have laid a foundation for contemporary deep learning applications [34–36]. Recent innovations in summarization, anomaly detection, multi-task learning,

and temporal modeling continue to expand the capabilities of personalized mining algorithms in high-dimensional, complex data environments [37–40].

III. RESEARCH ON THE PERSONALIZED MINING ALGORITHM FOR GRASSROOTS NETWORK DATA BASED ON DEEP LEARNING

A. Overall Design of the Mining Process

The personalized mining algorithm for grassroots network data based on a fuzzy neural network consists of three major stages: data preparation, fuzzy neural network construction and training, and pruning and rule extraction. The complete workflow is illustrated in Figure 1.

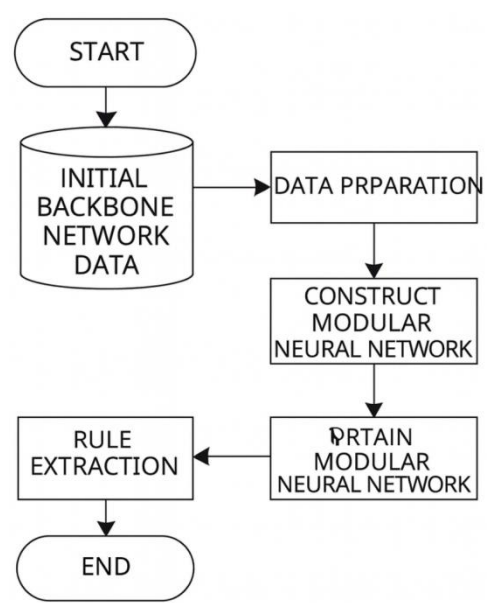


Figure 1. Process of the Personalized Grassroots Network Data Mining Algorithm Based on the Fuzzy Neural Network.

B. Data Preparation Stage

The data preparation stage forms the foundation of the personalized mining algorithm for grassroots network data. Its purpose is to prepare suitable data for the subsequent mining process. This stage is composed of three components: data cleaning, data selection, and data representation. The main objective is to define, process, and represent the data to be mined so that it can be effectively used by the mining algorithm. The functional structure of the data preparation stage is shown in Figure 2.

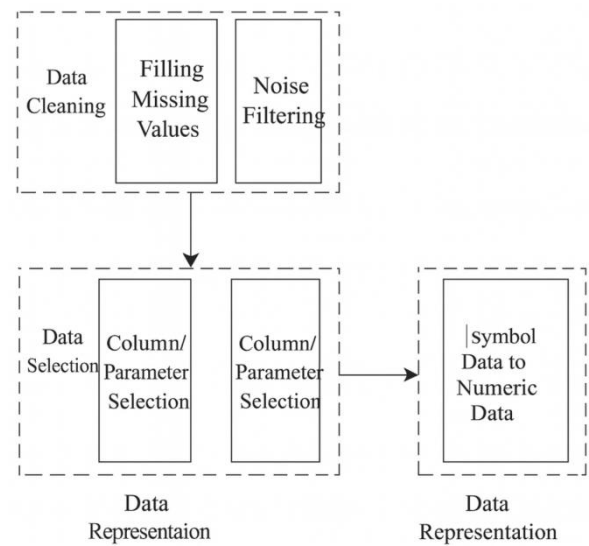


Figure 2. Functional Structure Diagram of the Data Preparation Stage.

The specific functions of each component are described as follows:

(1) Data Cleaning

Due to the diversity of data sources within grassroots networks, the collected data inevitably contain inaccuracies, incompleteness, redundancy, or inconsistency. Therefore, data cleaning is required to fill missing data, correct inconsistencies, and perform denoising operations [10]. Among these, denoising can be implemented using nonlinear wavelet filtering methods.

(2) Data Selection

Data selection is conducted along two dimensions to extract relevant data from grassroots network datasets. Depending on the mining task, selection may be performed along parameter dimensions, rows, or records to obtain subsets suitable for mining.

(3) Data Representation

The main task of this component is to transform the cleaned and selected data into a form compatible with the fuzzy neural network mining algorithm. Since the fuzzy neural network processes numerical data, symbolic data must be converted into numerical representations. A suitable Hash function can be used to map a given character string into a unique numerical form.

Although grassroots networks contain many data types, these can generally be categorized into three groups:

- continuous numerical data,
- discrete numerical data, and
- symbolic data.

The conversion relationships among these types are shown in Figure 3.

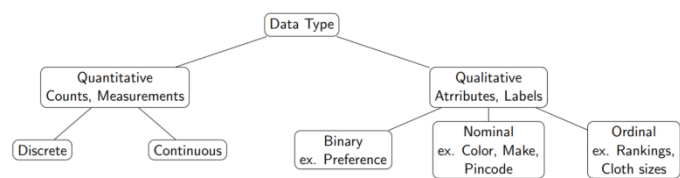


Figure 3. Data Representation and Transformation.

For example, the symbolic value “Apple” can be converted into a corresponding discrete numerical value using a Hash function. The resulting discrete numerical value can then be encoded or further quantified as a continuous numerical value.

C. Construction and Training of the Fuzzy Neural Network

1). Construction of the Fuzzy Neural Network

A five-layer fuzzy neural network is constructed, where the first layer is the input layer; the second layer is the fuzzification layer. This layer transforms each input attribute into fuzzified membership values corresponding to the linguistic variables small, medium, and large, producing three membership degrees. The maximum membership value is set to 1 and the others to 0, generating the output of the third layer.

If the second layer contains N nodes, then $N = 3n$, where n denotes the number of input variables. The third layer is the hidden layer, fully connected to the second layer and consisting of H units. The fourth layer is the fuzzy rule layer, also fully connected to the third layer, and contains Q units, where $Q = 3m$, and m denotes the number of outputs. The fifth layer is the expected output layer, corresponding to the fuzzified output of the input after defuzzification.

For a data sequence of a given attribute in the grassroots network, statistical techniques are applied to preprocess this sequence and obtain the fuzzification membership functions, thereby achieving the fuzzification process [11]. The membership functions are defined as:

$$S = e^{-k_1 \left(\frac{x - \delta_1}{2\sigma^2} \right)^2}, \quad M = e^{-k_2 \left(\frac{x - \delta_2}{2\sigma^2} \right)^2}, \quad B = e^{-k_3 \left(\frac{x - \delta_3}{2\sigma^2} \right)^2}$$

In (1), S, M, B denote the membership degrees for *small*, *medium*, and *large*, respectively. The slope control parameters of the three membership functions are represented as $e^{-k_1}, e^{-k_2}, e^{-k_3}$. The centers of the membership functions $\delta_1, \delta_2, \delta_3$ are determined by the mean μ and variance σ of the data sequence, computed as:

$$\mu = \frac{1}{k} \sum_{i=1}^k \log x_i, \quad \sigma = \sqrt{\frac{1}{k-1} \sum_{i=1}^k (\log x_i - \mu)^2}$$

$$\delta_1 = e^{\mu - \sigma}, \quad \delta_2 = e^{\mu}, \quad \delta_3 = e^{\mu + \sigma}$$

In (2), x_i represents the i -th sample of the attribute x , where $i = 1, 2, \dots, k$.

2). Training of the Fuzzy Neural Network

To enhance the accuracy of the fuzzy neural network, parameter optimization is performed through a supervised training process. The preprocessed grassroots network data are divided into training and testing subsets, and the training samples consist of multiple pairs of input attributes and their corresponding expected outputs.

The training process begins by feeding the input variables into the first layer, where they are directly transmitted to the fuzzification layer. In the second layer, each input attribute is transformed into three membership degrees-representing the linguistic categories small, medium, and large-based on the predefined membership functions. These fuzzified values serve as the inputs to the hidden layer.

In the hidden layer, each neuron aggregates the fuzzified inputs using weighted connections. The aggregated values are then passed through an activation function, generating the internal fuzzy feature representations of the network. These features are subsequently transmitted to the fourth layer, where fuzzy rules are applied. Each rule combines the outputs of the hidden layer and produces a corresponding rule activation strength.

The fifth layer performs defuzzification, transforming the fuzzy rule outputs into the final numerical outputs of the network. The difference between these outputs and the expected values is then used to compute the training error, which guides the adjustment of network parameters. Through iterative updates, the connection weights are modified in the direction that reduces the error, allowing the network to gradually learn the mapping relationship between inputs and outputs.

This training process is repeated across all samples until the network achieves stable convergence. Training terminates when either the predefined maximum number of iterations is reached or the error falls below the required threshold. Through this iterative learning procedure, the fuzzy neural network progressively improves its ability to perform accurate personalized mining of grassroots network data.

D. Network Pruning and Rule Extraction Stage

To minimize the number of weights and nodes in the fuzzy neural network, a pruning mechanism is applied after training. The objective is to remove redundant or insignificant weights so that the trained network becomes more compact while improving its overall accuracy [12]. Pruning begins by defining two thresholds: one that limits the allowable network error and another that determines the minimum weight magnitude required for retention. These thresholds are jointly constrained to ensure that pruning does not destabilize the network or lead to excessive loss of information.

During pruning, the weights within each layer are evaluated. If a weight exceeds the upper bound defined by the pruning criteria, it is removed. Likewise, if a weight contributes less than the minimum required magnitude, it is eliminated as well. When none of the weights satisfy the predefined conditions for removal, the weight with the smallest magnitude in the network is deleted to guarantee continuous refinement of the network structure. After each pruning operation, the network undergoes retraining to restore or improve its accuracy. If the retrained network maintains or surpasses the required accuracy level, pruning proceeds; if the accuracy deteriorates, the pruning process is halted and the current network structure is retained.

Once the pruning phase is completed, the remaining network weights serve as the basis for extracting fuzzy rules. Rules associated with weight values that are zero or approach zero are regarded as invalid and removed, while rules with significant weights are preserved. This ensures that all extracted rules reflect meaningful contributions to the network's decision-making process. Based on these retained rules, the trained fuzzy neural network is applied to the testing dataset to perform personalized mining of grassroots network data.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

To evaluate the effectiveness of the proposed algorithm, two standard grassroots network datasets, Dataset A and Dataset B, were selected for experimentation. Dataset A consists of network-related text data with a total of 127 categories and 97,874 web pages, whereas Dataset B contains common textual data with 20,051 documents. Both datasets were divided into training and testing subsets. The comparisons were performed using the traditional PageRank-based mining algorithm [5], a machine-learning-based mining algorithm [6], and the proposed method as the main evaluation target. Each algorithm was applied to the same training and testing sets, and the experimental outcomes were analyzed to compare their performance.

A. Comparison of Overall Mining Efficiency

The mining efficiency of the three algorithms was compared by measuring their processing speed in both the training and testing phases. The comparison results are shown in Figure 4.

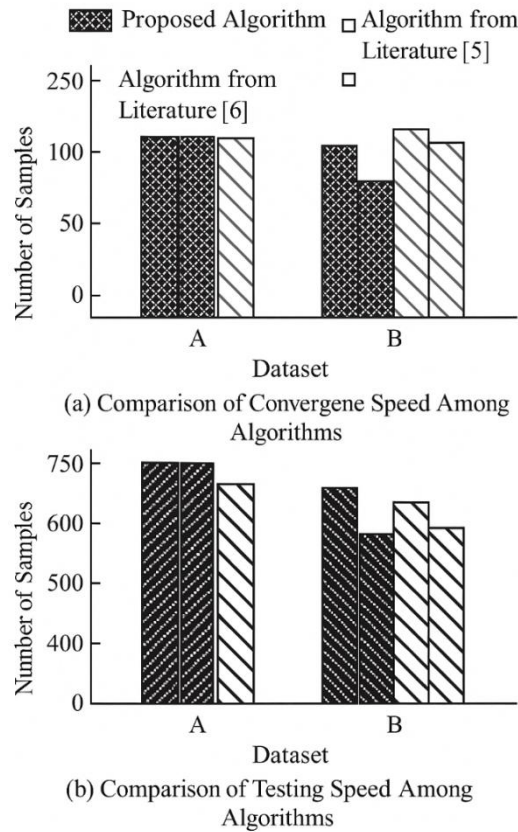


Figure 4. Comparison of acquisition speed and training/testing speed among algorithms.

The results in Figure 4 indicate that the processing speed of the proposed algorithm is significantly higher than that of the other two methods, both in terms of unit-time data handling and training/test throughput. This demonstrates that the proposed algorithm achieves notable improvements in mining efficiency and ensures faster convergence during the training process.

B. Comparison of Mining Accuracy

To ensure the reliability of the accuracy evaluation, both datasets A and B were used to conduct 15 rounds of training and testing for each algorithm. The average performance over the 15 runs was used as the final accuracy comparison metric. Mining accuracy was assessed using three indicators: precision P , recall R , and similarity D . The corresponding definitions are as follows.

Precision P represents the proportion of correctly mined data relative to the total mined results. It is computed as:

$$P = \frac{|\varphi \cap r|}{|\varphi|}$$

where φ denotes the total amount of mined data, and r denotes the actual data that should be mined.

Recall R reflects the proportion of correctly mined data relative to the total amount of actual data to be mined. It is defined as:

$$R = \frac{|\varphi \cap r|}{|r|}$$

Similarity D measures the degree of resemblance between the algorithm's mined output and the actual required data. Higher values indicate better mining performance. It is calculated as:

$$D(\theta, \gamma) = \frac{|\theta \cap \gamma|}{|\theta|}$$

where γ represents the mined data obtained by the algorithm, and θ denotes the actual required data.

The mining accuracy comparison among the three algorithms is summarized in Table 1.

Table 1. Comparison of Mining Accuracy Among Different Algorithms.

Algorithm	Dataset	Precision P (%)	Recall R (%)	Similarity D (%)
Proposed Algorithm	Dataset A	98.5	97.86	95.47
Proposed Algorithm	Dataset B	94.83	92.11	90.03
Method in [5]	Dataset A	93.46	91.27	89.06
Method in [5]	Dataset B	84.8	80.44	79.33
Method in [6]	Dataset A	87.48	82.7	84.33
Method in [6]	Dataset B	82.77	73.73	76.28

Analysis of Table 1 shows that, when mining Dataset A, the proposed algorithm achieves the highest precision, recall, and similarity compared with the other two algorithms. For Dataset B, although the performance of the proposed method is slightly lower than that on Dataset A, it still outperforms the other two algorithms. These results indicate that the proposed algorithm consistently provides high mining accuracy and maintains stable mining performance across different datasets.

References

1. M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in Proc. RecSys, 2010, pp. 135-142.
2. G. Palla, A. L. Barabási, and T. Vicsek, "Quantifying social group evolution," Nature, vol. 446, no. 7136, pp. 664-667, 2007.
3. W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in Proc. NIPS, 2017, pp. 1024-1034.
4. D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of Things: Vision, applications and research challenges," Ad Hoc Networks, vol. 10, no. 7, pp. 1497-1516, 2012.
5. D. Heckerman, D. Geiger and D. M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," Machine Learning, vol. 20, no. 3, pp. 197-243, 1995.
6. J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques. Morgan Kaufmann, 2011.
7. R. S. Baker and P. S. Inventado, "Educational data mining and learning analytics," in Learning Analytics. New York, NY: Springer, 2014, pp. 61-75.
8. D. O. Hebb, The Organization of Behavior: A Neuropsychological Theory. Wiley, 1949.
9. A. I. Moustapha and R. R. Selmic, "Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection," IEEE Transactions on Instrumentation and Measurement, vol. 57, no. 5, pp. 981-988, 2008.
10. Chen C., Wu Y., Dai Q., Zhou H. Y., Xu M., Yang S., ... and Yu Y., "A survey on graph neural networks and graph transformers in computer vision: A task-oriented perspective," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.
11. Peng S., Zhang X., Zhou L., and Wang P., "YOLO-CBD: Classroom Behavior Detection Method Based on Behavior Feature Extraction and Aggregation," Sensors, vol. 25, no. 10, p. 3073, 2025.
12. Velickovic P., Cucurull G., Casanova A., Romero A., Liò P., and Bengio Y., "Graph attention networks," stat, vol. 1050, no. 20, pp. 10-48550, 2017.
13. Zhang X. and Wang Q., "EEG anomaly detection using temporal graph attention for clinical applications," Journal of Computer Technology and Software, vol. 4, no. 7, 2025.

14. Sehgal U., Kaur K., and Kumar P., "Notice of violation of IEEE publication principles: The anatomy of a large-scale hyper textual web search engine," Proc. 2009 Second Int. Conf. Computer and Electrical Engineering, vol. 2, pp. 491-495, Dec. 2009.
15. Perozzi B., Al-Rfou R., and Skiena S., "Deepwalk: Online learning of social representations," Proc. 20th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, pp. 701-710, Aug. 2014.
16. Khedri K., Rawassizadeh R., Wen Q., and Hosseinzadeh M., "Pruning and quantization impact on graph neural networks," arXiv preprint arXiv:2510.22058, 2025.
17. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., and Salakhutdinov R., "Dropout: A simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929-1958, 2014.
18. Donoho D. L. and Johnstone I. M., "Ideal spatial adaptation by wavelet shrinkage," Biometrika, vol. 81, no. 3, pp. 425-455, 1994.
19. Xie A. and Chang W. C., "Deep learning approach for clinical risk identification using transformer modeling of heterogeneous EHR data," arXiv preprint arXiv:2511.04158, 2025.
20. Wu Y., Qin Y., Su X., and Lin Y., "Transformer-based risk monitoring for anti-money laundering with transaction graph integration," Proc. 2025 2nd Int. Conf. Digital Economy, Blockchain and Artificial Intelligence, pp. 388-393, June 2025.
21. Hu C., Cheng Z., Wu D., Wang Y., Liu F., and Qiu Z., "Structural generalization for microservice routing using graph neural networks," arXiv preprint arXiv:2510.15210, 2025.
22. Pan S. and Wu D., "Trustworthy summarization via uncertainty quantification and risk awareness in large language models," arXiv preprint arXiv:2510.01231, 2025.
23. Guan T., Sun S., and Chen B., "Faithfulness-aware multi-objective context ranking for retrieval-augmented generation," 2025 (preprint).
24. Wang C., Yuan T., Hua C., Chang L., Yang X., and Qiu Z., "Integrating large language models with cloud-native observability for automated root cause analysis and remediation," 2025 (preprint).
25. Cao K., Zhao Y., Chen H., Liang X., Zheng Y., and Huang S., "Multi-hop relational modeling for credit fraud detection via graph neural networks," 2025 (preprint).
26. Rama Moorthy H., Avinash N. J., Krishnaraj Rao N. S., Raghunandan K. R., Dodmane R., Blum J. J., and Gabralla L. A., "Dual stream graph augmented transformer model integrating BERT and GNNs for context aware fake news detection," Scientific Reports, vol. 15, no. 1, p. 25436, 2025.
27. Hamilton W., Ying Z., and Leskovec J., "Inductive representation learning on large graphs," Advances in Neural Information Processing Systems, vol. 30, 2017.
28. Kipf T. N., "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:1609.02907, 2016.
29. Jamali M. and Ester M., "A matrix factorization technique with trust propagation for recommendation in social networks," Proc. Fourth ACM Conf. Recommender Systems, pp. 135-142, Sept. 2010.
30. Huang J., Li S. Z., and Zhang Y., "Learning Bayesian networks from data: An efficient approach," IEEE Trans. Syst., Man, Cybern., vol. 30, no. 6, pp. 263-274, 2000.
31. Li J., Gan Q., Wu R., Chen C., Fang R., and Lai J., "Causal representation learning for robust and interpretable audit risk identification in financial systems," 2025 (preprint).
32. Li Z., Liu F., Yang W., Peng S., and Zhou J., "A survey of convolutional neural networks: Analysis, applications, and prospects," IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 12, pp. 6999-7019, 2021.
33. Bishop C. M. and Nasrabadi N. M., Pattern Recognition and Machine Learning, Springer, 2006.
34. Brin S. and Page L., "The anatomy of a large-scale hypertextual web search engine," Computer Networks and ISDN Systems, vol. 30, no. 1-7, pp. 107-117, 1998.
35. Baker R. S. and Inventado P. S., "Educational data mining and learning analytics," in Learning Analytics, pp. 61-75, Springer, 2014.
36. Kim Y., "Convolutional neural networks for sentence classification," arXiv preprint arXiv:1408.5882, 2014.
37. Lyu N., Wang Y., Chen F., and Zhang Q., "Advancing text classification with large language models and neural attention mechanisms," arXiv preprint arXiv:2512.09444, 2025.

38. A self-supervised learning framework for robust anomaly detection in imbalanced and heterogeneous time-series data, 2025 (preprint).
39. Hu X., Kang Y., Yao G., Kang T., Wang M., and Liu H., "Dynamic prompt fusion for multi-task and cross-domain adaptation in LLMs," arXiv preprint arXiv:2509.18113, 2025.
40. Lyu N., Chen F., Zhang C., Shao C., and Jiang J., "Deep temporal convolutional neural networks with attention mechanisms for resource contention classification in cloud computing," 2025.

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