Transactions on Computational and Scientific Methods | Vol. 5, No. 3, 2025
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Topology-Aware Graph Neural Networks for Distributed Node Fault Identification

Fei Xue[1], Kejia Wu[2], Aziza Abdulla[3]
[1]University of California, San Diego, San Diego, USA
[2]University of Illinois Urbana-Champaign, Urbana, USA
[3]University at Buffalo, Buffalo, USA
*Corresponding author: Aziza Abdulla; azabdulla.914@gmail.com

Abstract: This paper proposes a fault identification model based on graph neural networks for node fault detection in distributed systems. The method models the distributed system as a graph structure and fuses node features with topological relationships to accurately represent node states in complex architectures. The model employs a multi-layer graph neural network to aggregate and propagate information across nodes, significantly enhancing its ability to identify potential faulty nodes. In the experimental section, a standard graph-structured dataset is used for testing. The proposed model is compared with several mainstream approaches. Results show that it outperforms other methods in terms of accuracy, AUC, and F1 score. In addition, the paper conducts extended experiments to evaluate the impact of varying neighbor counts and noise disturbances on model stability. These results further confirm the adaptability and robustness of the proposed approach in complex environments.

Keywords: graph neural network; distributed system; node failure; anomaly detection

1. Introduction

In modern information systems and large-scale computing platforms, distributed architectures have become the dominant technical solution. By dividing tasks across multiple nodes for parallel processing, they significantly improve system performance and resource utilization. However, the complexity of distributed systems also introduces new challenges, especially in detecting and identifying node failures. When a node fails, it may cause cascading errors and even lead to system-wide service disruptions. Therefore, efficiently and accurately identifying potential node failures is critical to ensuring system stability and availability[1].

Traditional fault detection methods often rely on predefined rules, static thresholds, or monitoring metrics based on isolated features. These approaches typically focus on individual node behaviors without considering their interactions within the broader system. As a result, they struggle to handle the complex interdependencies that exist among distributed nodes. In real-world distributed environments, the status of each node is not determined in isolation but is part of a dynamic process shaped by various external and internal factors. These include the behavior of neighboring nodes, the structure of communication pathways, and the overall system topology. Such interrelated factors can propagate faults across the network and create cascading failures[2]. Therefore, in order to achieve accurate and timely fault detection, it is essential to construct a fault identification model that not only captures inter-node relationships but also maintains a global awareness of the entire system's structural and operational context.

In recent years, the development of graph-based data mining techniques has advanced rapidly. Graph neural networks, in particular, have shown unique advantages in modeling complex relationships in graph-structured data. These networks naturally integrate structural information with node attributes and use hierarchical message-passing mechanisms to update node representations dynamically. This makes them well-suited for capturing intricate dependencies in distributed systems. Applying graph neural networks to fault detection enables unified modeling of system structure and state information, improving the accuracy and robustness of fault identification[3].

This study proposes a fault identification method for distributed nodes based on graph neural networks, leveraging their strong modeling capabilities in the context of distributed systems. The method not only considers individual node features, such as performance metrics and operational status, but also fully incorporates the topological relationships among nodes, including their communication links and structural dependencies. By doing so, it constructs a more realistic and comprehensive representation of the entire system, allowing for an accurate reflection of the underlying interactions within the network. Through the training of a multi-layer graph neural network, the model effectively captures both fault propagation paths and localized anomaly patterns across different layers of the graph. This enables the system to detect faults promptly and perform classification with high precision, even under complex and dynamic operating conditions[4].

The significance of this research lies in offering an intelligent and automated approach to fault management in distributed systems. It addresses the limitations of existing methods in structural perception and global modeling. Moreover, by introducing graph neural networks into the fault diagnosis framework, this work extends the application of graph learning in industrial systems and provides a theoretical and technical foundation for future studies on fault diagnosis and system health assessment in complex networks. The proposed method holds promise for broad deployment in distributed environments such as big data centers, edge computing platforms, and blockchain networks, enhancing system stability and intelligent management capabilities.

Recent advances in graph learning and sequence representation further strengthen the methodological basis for distributed node fault identification. Heterogeneous graph learning for multi-entity risk identification [5], transformer-based two-stage trend fusion for financial time series [6], and unified multi-granularity representation learning for backend anomaly detection and causal localization [7] show that structural interactions and temporal trends can be jointly encoded to improve the recognition of complex system states. Constraint-aware resource matching for virtual machine placement [8], global context modeling for queue time prediction in cloud systems [9], learning-driven collaborative scheduling for multi-tenant inference services [10], and reinforcement learning-based intelligent decision modeling for large-scale distributed systems [11] further indicate that fault identification models should be aligned with resource constraints, global context, and operational decision processes in distributed environments.

Additional studies on Mamba-LSTM collaborative modeling for multi-table ETL anomaly detection [12], multi-scale feature decoupling and boundary-aware diagnosis [13], graph neural network-based advertising recommendation decisions [14], unified multi-source feature learning and structural aggregation for financial risk identification [15], and structure-guided attention with multi-scale behavior modeling for microservice anomaly detection [16] provide transferable design principles for this study. These works support the use of multi-scale feature extraction, structural aggregation, attention-guided representation enhancement, and anomaly-sensitive decision modeling. Building on these ideas, the proposed topology-aware GNN framework emphasizes node-relationship modeling, neighborhood aggregation, and robust state discrimination, making it suitable for identifying distributed node faults under dynamic and noisy operating conditions.

2. Method

This study uses graph neural networks as the core modeling tool for distributed node fault identification. Based on the graph structure, the distributed system is represented. Each node is regarded as a vertex in the graph, and the connection relationship between nodes is represented as an edge. On this basis, the node features are embedded in multiple levels through the graph neural network to capture the dependency relationship between nodes and the potential fault propagation mode. The model architecture is shown in Figure 1.

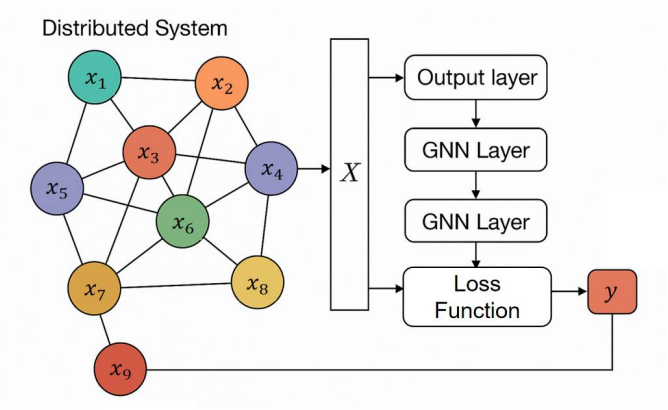


Figure 1. Overall model architecture diagram

This network architecture diagram shows a distributed node fault identification process based on a graph neural network. The system first constructs the distributed nodes into a graph structure, aggregates and updates the node features through a multi-layer graph neural network, and extracts high-order representation information. Finally, the output layer combines the loss function to complete the classification and identification of the faulty nodes, realizing end-to-end fault identification modeling.

Let the original graph be $G = (V, E)$, where V represents the node set, E represents the edge set, and the initial feature of the node is represented by $X \in R^{N \times d}$, where N is the number of nodes and d is the feature dimension. The core operation of the graph neural network is the aggregation and update of information, which is defined as follows:

$$h_v^{(k)} = \sigma\left(\sum_{u \in N(v)} W^{(k)} h_u^{(k-1)} + b^{(k)} \right)$$

Among them, $h_v^{(k)}$ represents the representation of node v at the k-th layer, $N(v)$ represents its neighbor set, $W^{(k)}, b^{(k)}$ represents the weight and bias of the k-th layer, and $\sigma(\cdot)$ is the activation function. Through layer-by-layer iterative updates, the representation of the node continuously integrates information from its neighbors, and finally forms a high-order feature representation with global structure perception.

In order to further improve the model's expressiveness in fault identification tasks, skip connections and normalization strategies are introduced. These components help address the over-smoothing problem commonly encountered in deep graph neural networks, where node representations become indistinguishable after multiple layers of aggregation. By incorporating skip connections, the model preserves important feature information from earlier layers, while normalization techniques contribute to more stable and efficient training. Together, these enhancements allow the model to maintain rich and discriminative node representations throughout the learning process. The final node embedding results, which integrate both local and global structural information, are then fed into the classification layer to identify the specific fault types. For supervised learning, cross entropy is adopted as the loss function to measure the difference between the predicted outputs and the true labels. The classification output is formally defined as follows:

$$y'_v = \text{softmax}(W_o h_v^{(K)} + b_o)$$

$$L = -\sum_{v \in V} y_v \log y'_v$$

Among them, $h_v^{(K)}$ represents the representation of the node in the last layer, W_o, b_o is the parameter of the output layer, y_v is the true label, and y'_v is the model prediction value. The loss function is optimized by gradient descent to minimize the difference between the prediction and the true label. After the model training is completed, the node status can be quickly and accurately identified in the actual distributed system.

3. Experimental Results

3.1 Dataset

The experimental dataset used in this study is the publicly available ABIDE-TSG dataset. This dataset is specifically designed for anomaly detection tasks in distributed systems and is widely used in research on graph-structured learning and node state classification. It includes multiple simulated distributed system topologies, where each node is associated with time series features and labeled fault states.

In the ABIDE-TSG dataset, each sample represents a snapshot of a running distributed system. Edges describe the communication and dependency relationships between nodes. Node features consist of multi-dimensional time series that reflect the operational status of each node over time, such as resource utilization, communication latency, and processing capacity. Additionally, each node is marked with a fault label, which is used for supervised learning tasks.

This dataset is characterized by rich structure and clearly defined labels, making it highly suitable for validating graph neural networks in node fault identification tasks. Training and testing models on this dataset allow for an effective evaluation of their ability to recognize node states within complex structures, thereby verifying their applicability and robustness in real-world distributed system environments.

3.2 Experimental Results

First, this paper gives the comparative experimental results of different algorithms, providing a comprehensive evaluation of their performance on the fault identification task. The results are summarized in Table 1, which includes key metrics such as Accuracy (ACC), Area Under the Curve (AUC), and F1-Score. These metrics are used to assess the classification quality and overall effectiveness of each method. By presenting this comparison, the paper highlights the relative strengths and weaknesses of various approaches, establishing a clear performance baseline for the proposed model. The detailed experimental results are shown in Table 1.

Table 1: Comparative experimental results

Method	ACC	AUC	F1-Score
CatBoost[17]	0.841	0.865	0.823
MLP[18]	0.857	0.879	0.836
Transformer[19]	0.868	0.892	0.847
CNN+CBAM[20]	0.874	0.901	0.853
GNN(Ours)	0.912	0.938	0.891

The experimental results show that traditional machine learning models, such as CatBoost, achieve a certain level of accuracy in fault identification tasks (ACC = 0.841). However, they exhibit clear limitations in capturing complex relationships between nodes. MLP and Transformer models show slight improvements

with the introduction of nonlinear representation capabilities, reaching ACC values of 0.857 and 0.868, respectively. This indicates that deep models have stronger fitting capabilities for node features, but still fail to fully model the structural dependencies between nodes.

The CNN+CBAM model enhances feature extraction through the incorporation of an attention mechanism, which allows the network to focus on more informative parts of the input features during the learning process. By assigning different levels of importance to different spatial regions, the model is able to highlight critical patterns related to node states. As a result, it achieves better performance in terms of AUC and F1-score compared to previous baseline methods. This indicates that the inclusion of spatial attention contributes positively to the recognition of node conditions and can improve the model's discriminative capability to a certain extent.

In contrast, the proposed method based on graph neural networks achieves the best performance across all metrics (e.g., ACC = 0.912, AUC = 0.938). This demonstrates that GNNs effectively integrate node features with topological structures. Through multi-layer message-passing mechanisms, they provide more accurate fault identification. The results confirm that GNNs offer strong structural adaptability and representational power for distributed system fault detection, highlighting their broad application potential in complex system fault management.

Furthermore, this paper also gives an analysis of the impact of changes in the number of neighbor nodes on the accuracy of fault identification. This analysis is intended to explore how varying the receptive field of each node affects the model's ability to aggregate relevant information from its surroundings. By adjusting the number of neighbors considered during the graph neural network's message-passing process, the experiment evaluates the balance between capturing sufficient contextual information and avoiding the introduction of noise or redundant data. The experimental results, which reveal how the model's performance evolves with different neighbor settings, are presented in detail in Figure 2.

From Figure 2, it can be observed that as the number of neighboring nodes increases, the overall accuracy of the model in distributed node fault identification tasks shows a clear upward trend. This indicates that expanding the receptive field of each node enables the model to capture more contextual and structural information from the surrounding graph. Initially, when the number of neighbors is small (such as 2 or 4), the model's ability to aggregate useful signals from adjacent nodes is limited. This constraint restricts the depth of structural learning, leading to insufficient representation of node dependencies. As a result, its fault identification capability remains relatively weak, with accuracy scores of 0.842 and 0.869, respectively. The overall performance in these cases can be considered relatively average, highlighting the need for a broader neighborhood context in complex graph structures.

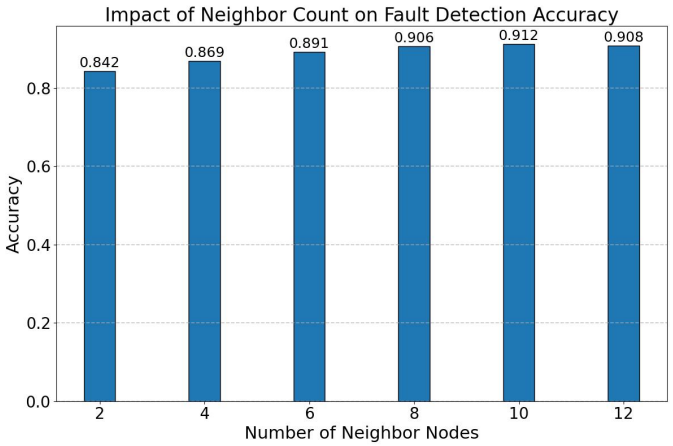


Figure 2. Analysis of the impact of changes in the number of neighbor nodes on fault identification accuracy

When the number of neighbors increases to 6, 8, and 10, the model aggregates more contextual information. The structural relationships among nodes are modeled more comprehensively, leading to significant improvements in accuracy, reaching 0.891, 0.906, and 0.912, respectively. This trend indicates that graph neural networks are highly sensitive to the number of neighbors. Properly expanding the receptive field helps capture potential fault propagation paths and correlated features.

However, when the number of neighbors reaches 12, the increase in accuracy becomes marginal, dropping slightly to 0.908, which is lower than the result with 10 neighbors. This may be due to information overload or the dilution effect caused by introducing redundant neighbors. It suggests that in distributed structures, the choice of neighbor count should strike a balance between structural awareness and information interference to achieve optimal classification performance.

Finally, this paper also presents a model stability test under noise disturbance, aiming to evaluate the robustness of the proposed approach when input features are affected by varying levels of noise. This experiment is designed to simulate real-world scenarios where data may be imperfect or subject to interference. The performance of the model is assessed across different noise intensities to observe how its accuracy and fault identification capability respond to such disturbances. The experimental results, which demonstrate the model's ability to maintain stability under these conditions, are illustrated in detail in Figure 3.

The figure shows that as the noise level gradually increases, the model's fault identification accuracy consistently decreases. This indicates that the model's performance is affected when input features are disturbed. The decline becomes more significant when the noise standard deviation exceeds 0.4. This change reflects that interference in node input features weakens the graph neural network's ability to perceive key structural patterns, thereby reducing its classification effectiveness.

At low noise levels (e.g., $\sigma = 0.1$ or 0.2), the model maintains strong stability, with accuracy values of 0.905 and 0.889, respectively. This suggests that the model has good tolerance to slight input disturbances. This robustness can be attributed to the graph neural network's neighbor aggregation mechanism, which helps smooth and offset local noise through structural context, preserving overall feature representation.

However, as noise increases and feature distributions become heavily distorted, the structural advantages of the graph neural network are gradually diminished, leading to a noticeable decline in classification performance. This experiment verifies the robustness boundary of the model in complex environments. It also highlights the need for feature denoising or adaptive mechanisms in practical deployment to enhance the model's fault tolerance when facing disturbances in real-world distributed systems.

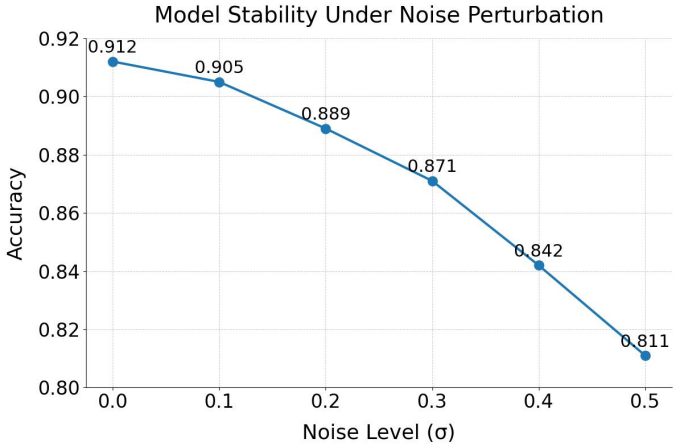


Figure 3. Model stability test under noise disturbance

4. Conclusion

This paper proposes a graph neural network-based modeling approach for identifying node faults in distributed systems. By abstracting the distributed architecture as a graph, it integrates topological relationships and node features to construct a fault detection model with global structural awareness. Experimental results show that the proposed method outperforms multiple baseline models, effectively improving the accuracy and robustness of fault detection.

The study further explores the impact of neighbor node count and input feature noise on model performance, demonstrating the advantage of graph neural networks in capturing complex node dependencies. The model maintains high accuracy under slight disturbances, reflecting strong adaptability to dynamic factors in real-world environments. Additionally, comparisons among different model structures provide experimental evidence for selecting appropriate architectures in future work.

The outcomes of this study enrich the application of graph learning methods in the field of distributed system fault diagnosis and offer a feasible intelligent solution for industrial-grade monitoring platforms. The proposed method is applicable to various complex scenarios, such as data centers, edge computing platforms, and IoT systems, showing strong potential for practical adoption.

Future research may extend to multimodal information fusion and multi-task joint modeling. This would improve node-level fault identification while introducing temporal modeling mechanisms to enhance the system's understanding and prediction of fault evolution. Moreover, integrating privacy-preserving frameworks like federated learning will offer greater security and flexibility for the practical deployment of distributed fault detection models.

References

[1] M. Y. Chen, E. Kiciman, E. Fratkin, A. Fox and E. Brewer, "Pinpoint: Problem determination in large, dynamic internet services", Proceedings of the 2002 International Conference on Dependable Systems and Networks, pp. 595-604, 2002.

[2] J. Lin, P. Chen and Z. Zheng, "Microscope: Pinpoint performance issues with causal graphs in micro-service environments", Proceedings of the 2018 International Conference on Service-Oriented Computing, pp. 3-20, 2018.

[3] M. Du, F. Li, G. Zheng and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning", Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1285-1298, 2017.

[4] X. Wang, B. Jin, Y. Du, P. Cui, Y. Tan and Y. Yang, "One-class graph neural networks for anomaly detection in attributed networks," Neural Computing and Applications, vol. 33, no. 18, pp. 12073—12085, 2021.

[5] Y. Xu, "Intelligent Supply Chain Risk Identification via Heterogeneous Graph Learning and Multi-Entity Interaction Modeling," 2025.

[6] S. Sun, "TrendFormer: Two-Stage Trend Fusion for Financial Time Series Forecasting Using Transformers," 2024.

[7] Y. Yang, C. Wen, H. Cui, Y. Sun and Y. Liu, "Learning Unified Multi-Granularity Representations for Backend Anomaly Detection and Causal Localization," 2024.

[8] J. Kou and E. Pei, "Constraint-Aware Resource Matching for Virtual Machine Placement in Cloud Environments," 2024.

[9] H. Zhuang and K. Gao, "Global Context Modeling and Structure-Guided Feature Enhancement for Queue Time Prediction in Large-Scale Cloud Systems," 2024.

[10] Z. Huang, J. Luo and C. Chen, "Learning-Driven Collaborative Scheduling for Multi-Tenant Distributed Inference Services," 2025.

[11] R. Wei and L. Tan, "Reinforcement Learning-Based Intelligent Decision Modeling for Large-Scale Distributed Systems," 2024.

[12] Y. Zhang and E. Cheng, "Anomaly Detection in E-Commerce Multi-Table ETL Processes through Mamba-LSTM Collaborative Modeling," Transactions on Computational and Scientific Methods, vol. 5, no. 11, 2025.

[13] K. Zeng, "Skin Disease Classification Algorithm Based on Multi-Scale Feature Decoupling and Boundary-Aware Diagnosis for Complex Lesion Morphology Recognition," 2024.

[14] Z. Pan, "Intelligent Decision-Making for Advertising Recommendation via Data Mining and Graph Neural Networks," 2024.

[15] Y. Nie, "Financial Risk Identification Using Unified Multi-Source Feature Learning and Structural Aggregation," Transactions on Computational and Scientific Methods, vol. 4, no. 1, 2024.

[16] X. Li, R. Yin and S. Dang, "Structure-Guided Attention and Multi-Scale Behavior Modeling for Microservice Anomaly Detection," 2025.

[17] D. Crankshaw, X. Wang, G. Zhou, M. J. Franklin, J. E. Gonzalez and I. Stoica, "Clipper: A low-latency online prediction serving system", Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI), pp. 613-627, 2017.

[18] K. Chen, J. Hu, Y. Zhang, Z. Yu and J. He, "Fault location in power distribution systems via deep graph convolutional networks," IEEE Journal on Selected Areas in Communications, vol. 38, no. 1, pp. 119–131, 2019.

[19] W. Xu, L. Huang, A. Fox, D. A. Patterson and M. I. Jordan, "Detecting large-scale system problems by mining console logs", Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles (SOSP), pp. 117–132, 2009.

[20] P. He, J. Zhu, Z. Zheng and M. R. Lyu, "Drain: An online log parsing approach with fixed depth tree", Proceedings of the 2017 IEEE International Conference on Web Services (ICWS), pp. 33-40, 2017.