Journal of Computer Technology and Software

ISSN: 2998-2383

Vol. 3, No. 4, 2024

Graph Learning Framework for Precise Anomaly Localization in Distributed Microservice Environments

Zhihao Xue
Rose-Hulman Institute of Technology, Terre Haute, USA
xuezhihao@outlook.com

Abstract: This paper proposes an automatic anomaly call chain localization method based on graph neural networks to address the challenge of anomaly detection in modern microservice systems. The method constructs a directed graph using the invocation relationships between microservices. The operational features of service nodes are encoded as node attributes. A graph neural network is then used to model the structure of the call chain in depth, enabling precise identification of potential anomaly nodes. Unlike traditional methods that rely on static rules or single-point metric analysis, the proposed model incorporates structural awareness and multi-hop information aggregation. This allows it to effectively capture the structural patterns of anomaly propagation within service call paths. To evaluate the model's performance, a series of experiments were conducted. These assess the model under different conditions, including feature combination strategies, system load intensity, anomaly type diversity, and propagation path depth. The experimental results show that the proposed method achieves significantly better performance than representative public methods in recent literature, across key metrics such as F1 Score, Precision, and Recall. The model demonstrates higher accuracy and stability. Furthermore, the method maintains strong identification capability in handling complex anomaly propagation patterns, high-concurrency loads, and multi-level service chains. This highlights its practical applicability in real engineering environments. The proposed method provides an effective modeling framework for intelligent anomaly localization in large-scale microservice systems. It also advances the application of graph learning techniques in system operations and maintenance.

Keywords: Microservice anomaly location, graph neural network, call chain modeling, intelligent system operation and maintenance, distributed system architectures

1. Introduction

In modern distributed system architectures, microservices have become a widely adopted design paradigm. They offer modularity, flexibility, and scalability for various Internet applications and enterprise systems. As business logic becomes increasingly complex, the number of microservices grows exponentially. The interactions among services are becoming more intricate, resulting in a highly dynamic and heterogeneous system[1,2]. While this change improves maintainability and development efficiency, it also introduces new challenges in system stability and fault diagnosis. When performance degradation, request failures, or response delays occur, quickly and accurately locating the faulty service from a large and dynamic call chain becomes a critical issue in operations and development[3].

Traditional anomaly localization methods mainly rely on rule engines, static configurations, or manual expertise. These approaches often fail under complex scenarios such as concurrent multi-service calls, asynchronous message passing, and long-span invocation chains. On the one hand, the call chain may vary based on user request paths, service versions, or deployment environments, making it hard for static rules to provide full coverage. On the other hand, anomaly propagation can be nonlinear or non-local. Minor fluctuations in upstream services may cause severe failures downstream in specific conditions. These factors increase the difficulty of fault tracing. Therefore, a more generalized and intelligent localization method is urgently needed to handle the complex behavior patterns of microservices in real-world settings[4].

In recent years, observability infrastructures have advanced significantly. Techniques such as distributed tracing, metrics monitoring, and log collection have become mature. These enable systems to produce large volumes of structured data related to service behavior. Among them, trace data is particularly valuable. It presents the dependency and call paths between services in the form of directed graphs, making it highly useful for failure diagnosis. This inherent graph structure makes it feasible to apply graph neural networks. As a deep learning method for graph-structured data, graph neural networks provide powerful representation capabilities. They can capture dependencies, patterns, and trends among nodes, offering a new direction for anomaly localization in microservices[5].

Using graph neural networks for anomalous call chain localization enables deep mining of both topological structures and dynamic behavior patterns. It avoids the limitation of relying solely on single-node information. By modeling and learning from the call graph, it becomes possible to identify abnormal propagation paths, detect bottleneck services, and uncover latent faults. Graph neural networks also exhibit a degree of transferability. They can adapt to call graph changes across time and environments, maintaining robustness and real-

time performance during system operation. Compared to traditional methods, this approach aligns better with the operating logic of modern microservices and offers greater engineering value[6].

In summary, with the widespread adoption of microservice architectures and increasing system complexity, traditional anomaly localization methods face challenges in both efficiency and accuracy. In this context, the automatic localization of anomalous call chains based on graph neural networks addresses the urgent need for intelligent operations in distributed systems. It provides theoretical and technical support for improving system stability, accelerating anomaly response, and optimizing resource allocation. This study aims to explore key issues and core mechanisms in this direction and promote the integration and practical application of intelligent operation technologies in microservice environments.

2. Related work

In the field of microservice anomaly localization, existing research mainly focuses on static rule-based methods, statistical learning models, and graph-based analysis. Traditional static rule methods usually rely on predefined monitoring thresholds and manually crafted anomaly rules. These methods evaluate system performance metrics such as CPU usage, memory consumption, and request latency[7]. While effective in monolithic systems, they face serious limitations in microservice architectures. Complex inter-service relationships and dynamic invocation paths often lead to anomalies that involve multiple nodes or spread across nodes. As a result, static methods struggle to accurately identify root causes. Moreover, these approaches heavily depend on expert knowledge, lack adaptability, and incur high maintenance costs. They become increasingly inadequate as the system scales or the business changes frequently[8].

With the growing application of machine learning in system operations, some studies have introduced supervised or unsupervised learning methods. These approaches model and predict system metrics using learning-based techniques. Typically, they score anomalies for individual services based on time series modeling. This helps support anomaly detection and localization. However, the invocation behavior in microservice systems is inherently relational and structured. Relying solely on time series or single-node metrics fails to capture causal links and dependencies between services. This modeling approach often overlooks complex propagation patterns in collaborative anomalies, leading to higher false positive and false negative rates. It performs especially poorly under scenarios involving multi-point resonance or cascading failures across upstream and downstream services[9].

To address the graph-structured nature of microservice systems, some studies have explored anomaly localization based on invocation chain graphs. Invocation chains represent the execution paths of service requests and record the propagation process across services. They contain rich temporal and topological information. In these graphs, services are represented as nodes and invocation relationships as edges. Anomalies often form specific substructures or path patterns in the graph[10]. Therefore, graph-based detection methods are better suited to describe inter-service interaction behavior. Some research has employed graph mining or path-matching techniques to identify abnormal paths. Compared to traditional models, these methods show improved precision and better identification of anomaly propagation across services. However, they still suffer from challenges in manual feature design and limited model generalization.

In recent years, the emergence of graph neural networks has introduced a new modeling paradigm for microservice anomaly localization. Graph neural networks perform feature aggregation and propagation over nodes and their neighbors. This enables the automatic learning of structural features and behavioral patterns in service invocation chains. Compared to traditional graph analysis methods, graph neural networks support end-to-end modeling. They do not require manual feature engineering and offer stronger representation and generalization capabilities. This approach is particularly suitable for capturing dynamic service dependencies and anomaly propagation paths. It provides a new technical foundation for efficient and accurate anomaly localization. As a result, graph neural network-based research is becoming a key direction in intelligent microservice operations. It holds significant theoretical value and practical potential.

3. Method

This research method is based on the call chain data of the microservice system, constructs a directed graph to represent the call relationship between services, and uses graph neural networks to model and reason about the graph structure, thereby realizing the automatic location of abnormal services. Specifically, each call chain in the system is regarded as a subgraph containing multiple service nodes and their call edges. The nodes represent microservice instances, and the edges represent the call directions. Each node contains not only the basic characteristics of the service (such as latency, error code, call frequency, etc.) but also its contextual information in the link. The model architecture is shown in Figure 1.

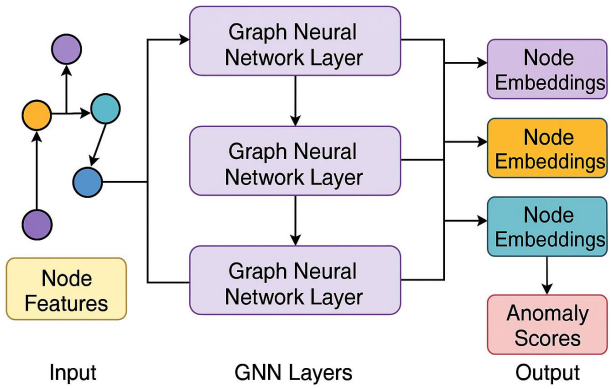

Figure 1. Overall model architecture diagram

We represent the raw features of each service node as a vector $x_i \in R^d$, where d represents the feature dimension.

In order to capture the dependency and exception propagation patterns between services, we construct a service

call graph $G = (V, E)$, where V represents the set of service nodes and E represents the call edges between services. In the graph neural network, the representation of each layer of nodes is updated by the features of neighboring nodes. We use a propagation mechanism based on graph convolution, and the representation of node i in layer l is:

$$h_i^{(l)} = \sigma(\sum_{j \in N(i)} \frac{1}{c_{ij}} W^{(l)} h_j^{(l-1)})$$

Where $N(i)$ represents the neighbor set of node i, c_{ij} is the normalization coefficient, $W^{(l)}$ is the weight matrix of the lth layer, σ is the activation function, and $h_j^{(l-1)}$ is the representation of the neighbor nodes in the l-1th layer.

To enhance the model's sensitivity to abnormal propagation paths, we introduce an attention mechanism to perform weighted aggregation on neighbor nodes, that is, when updating node representations, we consider the different importance of neighbors to the abnormal state of the current node. The weighted information received by node i from neighbor j is:

$$h_i^{(l)} = \sigma(\sum_{j \in N(i)} a_{ij} W h_j^{(l-1)})$$

This mechanism improves the model's ability to identify key propagation paths and effectively strengthens the focus on potential abnormal service nodes.

Finally, after updating the multi-layer graph neural network, we map the representation of each node to the abnormal score space to predict whether the service is in an abnormal state. The prediction process is implemented through a simple feedforward network, defined as:

$$y_i = sigmoid(w^T h_i^{(L)} + b)$$

Where w and b are learnable parameters, and $y_i \in (0,1)$ represents the abnormal probability of node i. The entire model is trained by minimizing the binary cross entropy loss function, and the objective function is:

$$L = -\sum_{i \in V} (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i))$$

Where $y_i \in \{0,1\}$ is the real label of the service node. This method can effectively capture abnormal patterns in complex call chains and realize automatic identification and location of abnormal services.

4. Experimental Results

4.1 Dataset

This study uses the Alibaba Cluster Trace 2018 as the primary dataset to construct microservice call graphs and node features. The dataset originates from a real large-scale distributed computing cluster. It includes extensive task scheduling, resource usage, and instance-level execution information. The data covers millions of containers and nodes. Organized by timestamps, it records fine-grained interactions between different service instances. This provides a rich and realistic foundation for studying anomaly detection in complex microservice architectures.

Key fields in the dataset include task ID, container ID, resource usage such as CPU and memory, task type, scheduling status, and time intervals. Through appropriate preprocessing and correlation analysis, it is possible to construct a call graph containing both node features and edge relationships. This graph serves as input for graph neural network training. Each node represents a specific microservice instance. Each edge represents a service request call. Time series data can be mapped to dynamic attributes or feature sequences in the graph structure.

This dataset is chosen due to its strong real-world business background and complex system behavior patterns. It captures both normal and abnormal operational states. It also reflects interaction patterns among microservices under different loads and failure scenarios. The authenticity and diversity of this data make it an ideal choice for anomaly call chain localization tasks. It helps enhance the generalization ability and practical value of the model in real-world applications.

4.2 Experimental Results

This paper first conducts a comparative experiment, and the experimental results are shown in Table 1.

Table1: Comparative experimental results

Method	F1-Score	Precision	Recall
Ours	91.4	89.7	93.2
GCN-Micro[11]	86.3	84.1	88.6
TraceAnomaly[12]	82.7	80.5	85.1
DeepTrace[13]	78.9	76.3	81.7

The experimental results show that the proposed method for automatic anomaly call chain localization based on graph neural networks significantly outperforms baseline models in terms of F1-Score, Precision, and Recall. The F1-Score reaches 91.4%, indicating a well-balanced performance between accuracy and completeness. This result highlights the strength of graph neural networks in capturing complex structural dependencies among microservices. In scenarios with multi-hop service calls and unclear anomaly propagation paths, graph-based modeling significantly improves the identification of abnormal nodes.

In terms of Precision, the proposed method achieves 89.7%, higher than all baseline models. This indicates fewer false positives in anomaly node detection. Due to long and dynamic call chains, traditional methods often mistake short-term fluctuations as anomalies. By aggregating multi-layer contextual information of each node, the proposed model reduces such misclassifications. Moreover, the attention mechanism assigns different weights to neighboring nodes, allowing the model to focus more on key points along the anomaly path. This improves decision precision.

For Recall, the method achieves 93.2%, noticeably higher than public methods such as GCN-Micro, TraceAnomaly, and DeepTrace. This shows a stronger ability to identify actual anomalous nodes. High recall is especially important in microservice environments where rapid identification of the root cause is critical. It supports system stability and shortens recovery time. Traditional models often rely on local indicators or sequential features. They tend to miss cross-node anomaly patterns. Graph neural networks can integrate features across nodes, addressing this limitation effectively.

Overall, the strong performance across all metrics is attributed to the structural modeling strategy based on call graphs. Unlike time-series models that rely only on metric fluctuations or traditional models that analyze nodes in isolation, graph neural networks integrate global information. They can learn potential anomaly propagation paths and inter-node influences within the call chain. This aligns well with the nature of structural anomaly diffusion in microservice systems. It also enhances the adaptability and practical value of the method in real-world deployments.

This paper also gives an experiment on the influence of node feature combinations on positioning accuracy, and the experimental results are shown in Figure 2.

The results in the figure show that the combination of node features has a clear impact on anomaly localization performance. When multiple features are integrated, all evaluation metrics improve significantly. Although single features (F1, F2, F3) can support anomaly detection to some extent, their performance varies. This indicates that different feature types have different abilities in capturing anomaly propagation patterns. Among them, F2 performs slightly better when used alone, suggesting higher discriminative power. However, its combination with other features still leaves room for further improvement.

When combining features as F1+F2 or F1+F3, both F1-Score and Precision increase. This indicates that multi-source features can complement each other during graph neural network propagation, enhancing the completeness of node representations. Notably, the integration of F1+F2+F3 achieves the best results across all metrics. This suggests that comprehensive feature representation helps the model better capture abnormal patterns between service nodes. The findings also confirm the importance of introducing multi-dimensional features into graph structures, as they provide richer contextual information for the model.

In terms of Precision, the trend shows a clear decline in false positives as more features are combined. This proves that high-quality feature combinations effectively reduce the misclassification of normal nodes. In microservice systems, false positives often trigger unnecessary diagnostic procedures. Therefore, improving Precision has practical value for operational efficiency. The model enhances its ability to locate anomaly paths through deep neighbor propagation and multi-feature aggregation mechanisms.

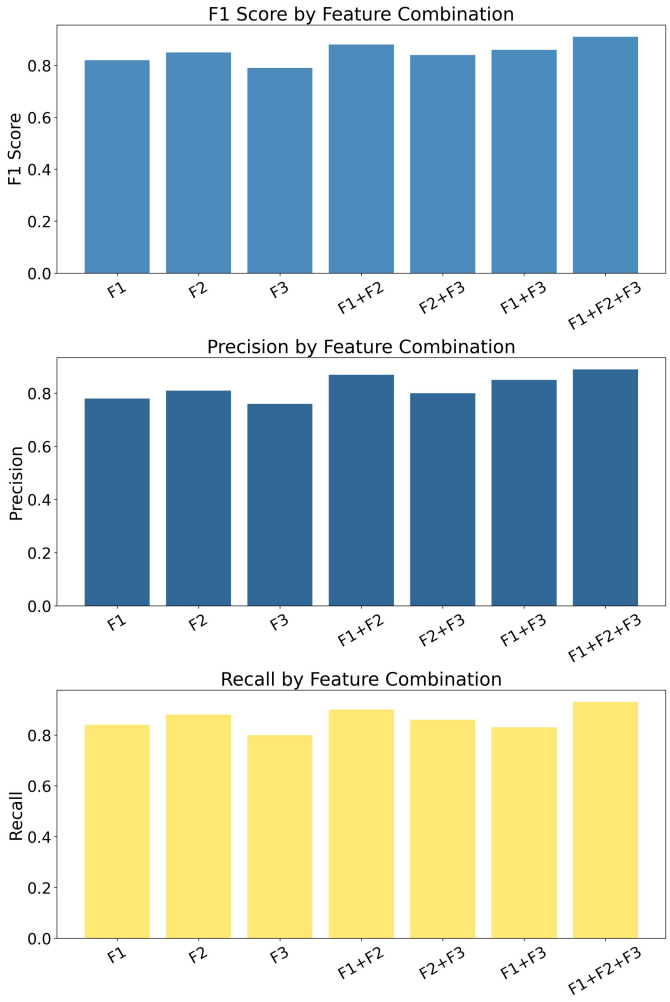

Figure 2. Experiment with the influence of node feature combination on positioning accuracy

The consistent improvement in Recall further demonstrates the benefit of multi-feature fusion in identifying abnormal nodes. Under the F1+F2+F3 configuration, the model can identify nearly all anomalous service nodes. This is particularly valuable for handling complex scenarios in microservice systems, such as chained propagation and multi-hop anomaly paths. Overall, the analysis shows that node feature selection and combination are key factors influencing the effectiveness of graph neural network-based anomaly detection. Designing appropriate feature fusion strategies is essential for improving model accuracy.

This paper also presents a comparative experiment on the robustness of the model under different load scenarios, and the experimental results are shown in Figure 3.

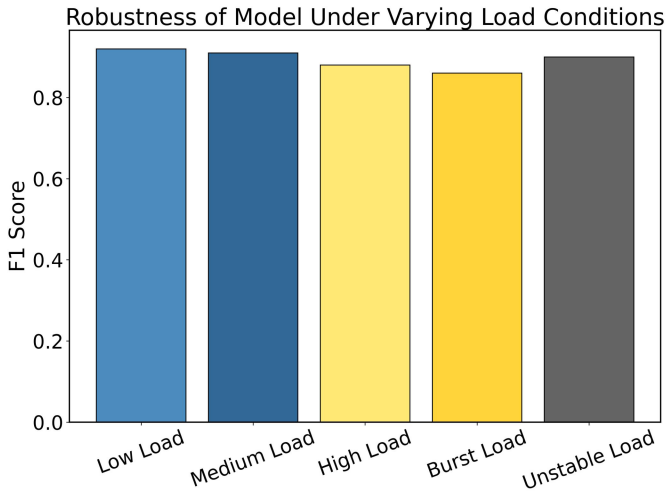

Figure 3. Comparative experiment on model robustness under different load scenarios

The results in the figure indicate that the proposed method demonstrates strong stability under different system load scenarios. The F1 Score consistently remains at a high level. This suggests that the graph neural network model has good robustness. It can adapt to the complex environments of real-world microservice systems, where load frequently fluctuates and call chain structures change dynamically. The model performs most stably under low and medium loads. This implies that with less resource pressure, the structure of the call chain is clearer, and the anomaly propagation path is easier to capture.

Under high load and burst load conditions, the model's performance shows a slight decline but remains within an acceptable range. This performance drop is mainly due to the increased overlap of service requests and signal interference caused by a surge in traffic between services. These factors make anomaly node identification more challenging. However, the model integrates both upstream and downstream structural information and supports multi-hop information propagation. As a result, it maintains a certain level of accuracy in complex paths and avoids the sharp performance degradation seen in traditional methods under such conditions.

The performance recovery under unstable load conditions is noteworthy. This scenario simulates a system with frequent and irregular fluctuations. In such highly dynamic call graphs, the structural modeling capability of the graph neural network becomes more apparent. Through continuous feature aggregation among nodes, the model can detect structural patterns of anomaly propagation under non-stationary conditions. This further demonstrates the model's adaptability to changes in graph structure. It is suitable for long-term deployment in real production environments.

In summary, the experiments confirm that the proposed method maintains reliable detection performance across various load conditions. It demonstrates practical engineering value and strong environmental adaptability. Compared with traditional algorithms that rely heavily on fixed data distributions, the graph neural network model is better suited to handle dynamic dependencies and high variability in microservice systems. This highlights the broad applicability and real-world relevance of the proposed approach for intelligent anomaly localization.

This paper also presents an experiment on the impact of anomaly-type diversity on the generalization ability of the model, and the experimental results are shown in Figure 4.

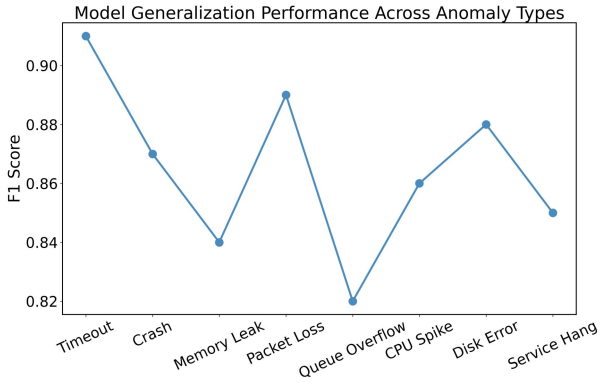

Figure 4. Experiment on the impact of anomaly-type diversity on model generalization ability

The experimental results in the figure show that the proposed model maintains a high overall F1 Score when handling various types of anomalies. This demonstrates strong generalization ability. However, the model's detection performance varies across anomaly types. This suggests that the complexity of different anomaly categories still affects the model's performance. The best results are observed for Timeout and Packet Loss anomalies. These types have clear temporal characteristics or strong upstream-downstream propagation patterns. This aligns well with the graph neural network's mechanism, which emphasizes structural propagation features.

In contrast, the model performs slightly worse on hidden anomalies such as Memory Leak and Queue Overflow. These anomalies are usually non-instantaneous. Their signals spread slowly or are masked by multi-level services in the call chain. This makes the modeling task more difficult. Even so, the model still performs within an acceptable range. This indicates that the graph neural network has a certain capacity to represent weak propagation anomalies. Future work could enhance temporal modeling or causal chain representation between upstream and downstream services.

For anomalies such as CPU Spikes, Disk Error, and Service Hang, the model shows moderately stable performance. These types often involve resource-level disruptions and can trigger chain reactions across nodes. The graph-based model leverages multi-hop aggregation to sense the impact range. This contributes to better anomaly identification. In particular, the improved results on Disk Error suggest that the model can capture low-frequency but high-impact anomalies. This is especially important for ensuring stability in complex microservice systems.

In summary, the experimental results demonstrate that the model has strong generalization across different anomaly types. It maintains stable performance under various failure mechanisms. This capability is crucial in real-world

microservice operations, where anomaly patterns are diverse and cannot be fully covered by training data. The deep exploitation of structural information by the graph neural network enhances the model's tolerance to unseen anomalies and broadens its recognition scope. This confirms the practicality and scalability of the proposed method for intelligent anomaly localization tasks.

This paper also presents a detailed experiment designed to investigate the effect of abnormal propagation path length on the model's recognition capability. The focus of this experiment is to examine how the number of hops, or the distance that an anomaly signal must travel through the service call chain, influences the model's ability to accurately identify anomalous nodes within the system. The experimental setup considers varying propagation path lengths, allowing for an in-depth analysis of how structural depth in the call graph impacts the overall detection performance of the proposed graph neural network model. By evaluating the model under different path length conditions, this experiment aims to better understand the relationship between anomaly diffusion patterns and the structural modeling capacity of the system. The corresponding results and observations are visually represented and summarized in Figure 5.The experimental results show that the length of the anomaly propagation path has a significant impact on the model's detection capability. As the path becomes longer, the F1 Score exhibits a clear downward trend. This indicates that when anomaly signals must pass through more hops from the source node to the target node, the model's overall judgment becomes less effective. Since the message-passing mechanism in graph neural networks relies on feature aggregation from neighboring nodes, longer paths may lead to signal dilution or interference from structural noise. This degrades the final decision accuracy.

The downward trend in the Precision metric further confirms that the model is more prone to false positives in the presence of long propagation paths. This issue is particularly evident in microservice systems. When the call chain becomes deep, some normal service nodes located near the anomaly path may be incorrectly identified due to their structural proximity. Although the proposed method incorporates some structural awareness, it still faces the problem of feature attenuation when dealing with deep anomaly chains. This suggests that future research should introduce stronger modeling capabilities for long-range dependencies to reduce false positives.

The changes in Recall indicate that the model becomes less capable of detecting true anomalous nodes as the path length increases. In particular, Recall drops significantly under the 5-hop and 6-hop conditions. This shows that anomaly signals tend to be lost or mixed with background noise during long-path propagation in complex call structures. The result highlights the importance of building deeper graph neural architectures or incorporating global structural awareness. These enhancements are necessary to ensure effective tracking of anomalies that propagate across multiple service layers in large-scale microservice systems. Overall, the experiments confirm that path depth is a critical factor. It significantly influences both the robustness and boundary of the model's detection performance.

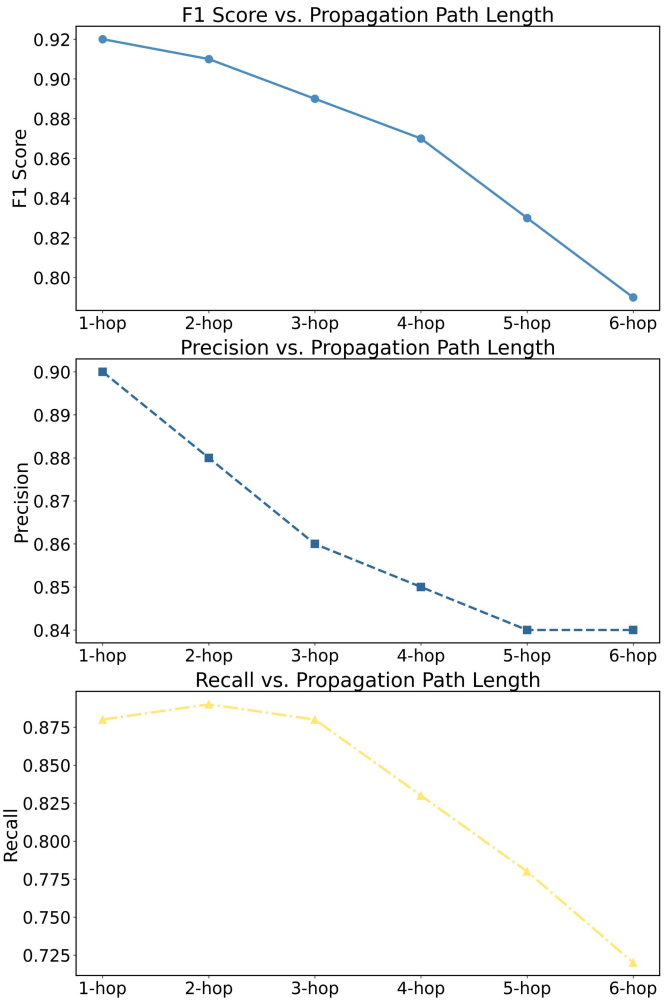

Figure 5. Experiment on the influence of abnormal propagation path length on model recognition ability

5. Conclusion

This study addresses the problem of automatic anomaly call chain localization in microservice systems. A graph neural network-based method is proposed, which effectively combines the graph representation of system structure with node behavior features. By performing multi-layer perception and aggregation on the call chain graph, the method improves the accuracy of identifying anomaly propagation paths. It overcomes the limitations of traditional rule-based or single-metric analysis approaches. It is capable of handling complex scenarios such as multi-hop dependencies and upstream collaborative failures. The method also demonstrates strong generalization and robustness. Experimental results show that the proposed approach consistently achieves stable and high performance across various anomaly types and operational conditions, indicating its practical value.

From multiple experimental dimensions, the proposed model shows both stable and responsive performance when faced with challenges such as anomaly diversity, system load variations, and extended propagation paths. The ability of graph neural networks to model structural information in

microservice environments provides a new perspective for anomaly localization tasks. This enables a shift from experience-based operations to data-driven intelligent management. The method offers scalable technical support for microservice operations, helping to improve fault response efficiency, reduce manual intervention, and ensure the stable operation of complex business systems.

This work not only makes a practical contribution to the field of intelligent operations but also provides a reference paradigm for applying graph neural networks in engineering systems. Similar problems of anomaly propagation exist in systems such as edge computing, cloud-native architectures, and distributed databases. The proposed modeling approach has strong transferability and adaptability. It can be applied to larger and more complex networked systems. It also opens new paths for integrating graph learning with observability technologies, promoting the advancement of intelligent monitoring and autonomous operations.

Future research may introduce time-aware mechanisms to model the evolution of call chains. This would enhance the model's ability to perceive the timing and duration of anomalies. Additionally, improving model interpretability could allow it to generate traceable anomaly paths and causal analysis outputs, supporting human decision-making. Incorporating advanced techniques such as federated learning and incremental graph learning into the anomaly detection framework may further enhance adaptability in cross-cluster, heterogeneous, and dynamic environments. This would support the development of more intelligent, efficient, and sustainable operational systems.

References

[1] J. Chen, F. Liu, J. Jiang, G. Zhong, D. Xu, Z. Tan and S. Shi, "TraceGra: A trace-based anomaly detection for microservice using graph deep learning", Computer Communications, vol. 204, pp. 109-117, 2023.

[2] J. Huang, Y. Yang, H. Yu, J. Li and X. Zheng, "Twin graph-based anomaly detection via attentive multi-modal learning for microservice system", Proceedings of the 2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 66-78, 2023.

[3] Chen J, Liu F, Jiang J, et al. TraceGra: A trace-based anomaly detection for microservice using graph deep learning[J]. Computer Communications, 2023, 204: 109-117.

[4] Liang X, Li L, Peng H. Unsupervised Microservice Log Anomaly Detection Method Based on Graph Neural Network[C]//International Conference on Swarm Intelligence. Singapore: Springer Nature Singapore, 2024: 197-208.

[5] Shi K, Li J, Liu Y, et al. BSDG: Anomaly Detection of Microservice Trace Based on Dual Graph Convolutional Neural Network[C]//International Conference on Service-Oriented Computing. Cham: Springer Nature Switzerland, 2022: 171-185.

[6] Somashekar G, Dutt A, Adak M, et al. GAMMA: Graph Neural Network-Based Multi-Bottleneck Localization for Microservices Applications[C]//Proceedings of the ACM Web Conference 2024. 2024: 3085-3095.

[7] Hussain H, Abuoda G, Litoiu M. Exploring Approaches to Integrate Performance Prediction and Anomaly Detection in Microservices Systems[C]//2024 34th International Conference on Collaborative Advances in Software and COmputiNg (CASCON). IEEE, 2024: 1-4.

[8] Golovkina A, Mogilnikov D, Ruzhnikov V. Graph Neural Networks for Metrics Prediction in Microservice Architecture[C]//International Conference on Computational Science and Its Applications. Cham: Springer Nature Switzerland, 2024: 343-357.

[9] Raeiszadeh M, Ebrahimzadeh A, Saleem A, et al. Real-time anomaly detection using distributed tracing in microservice cloud applications[C]//2023 IEEE 12th International Conference on Cloud Networking (CloudNet). IEEE, 2023: 36-44.

[10] Liu X, Liu Y, Wei M, et al. LMGD: Log-Metric Combined Microservice Anomaly Detection through Graph-based Deep Learning[J]. IEEE Access, 2024.

[11] Nguyen H X, Zhu S, Liu M. A survey on graph neural networks for microservice-based cloud applications[J]. Sensors, 2022, 22(23): 9492.

[12] Zhang K, Zhang C, Peng X, et al. Putracead: Trace anomaly detection with partial labels based on gnn and pu learning[C]//2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE). IEEE, 2022: 239-250.

[13] Tsimpourlas F, Rooijackers G, Rajan A, et al. Embedding and classifying test execution traces using neural networks[J]. IET Software, 2022, 16(3): 301-316.