Journal of Computer Technology and Software

ISSN: 2998-2383

Vol. 3, No. 7, 2024

Time Series and Graph Structure Fusion for AI-Based Anomaly Detection in Microservice Environments

Zhimin Qiu
University of Southern California, Los Angeles, USA
zhiminiqiu@gmail.com

Abstract: This paper addresses the complexity of fault detection in microservice systems by proposing a modeling approach that integrates temporal features with graph structural information to enhance the accuracy of anomaly detection and fault localization. Multi-dimensional monitoring metrics are used as temporal inputs, where embeddings and attention mechanisms capture dynamic changes in service states, while service invocation relations are modeled as graphs in which graph convolution characterizes cross-node dependencies and propagation paths. A cross-modal fusion module is then designed to unify temporal features and graph embeddings, achieving a balance between local details and global dependencies. Sensitivity experiments on key factors such as temporal window length, graph convolution depth, sampling frequency, and scheduling frequency reveal their significant impact on performance and show that proper configurations can improve fault detection. Comparative experiments further demonstrate that the proposed method outperforms several baselines in recall, accuracy, F1-score, and precision, maintaining high stability and robustness under diverse conditions. The results indicate that the fusion of temporal and graph-based representations enables efficient identification of anomalies in complex microservice scenarios and provides strong technical support for system operation and maintenance in practical applications.

Keywords: Microservice system; fault identification; time series modeling; graph structure fusion

1. Introduction

Microservice architecture has become an important paradigm in cloud computing and distributed systems in recent years. Its core idea is to decompose a complex monolithic system into multiple independent service units. Each unit has autonomy, can be deployed independently, and can be scaled on demand. This design improves system flexibility and maintainability. However, as the number of services grows and the invocation chains become more complex, the risk of potential failures in microservice systems also increases. The high level of interdependence and frequent interactions among services mean that a local fault can quickly escalate into a system-wide anomaly, causing serious damage to user experience and business continuity. Therefore, how to accurately and efficiently identify and classify faults in microservices has become a key issue in ensuring system stability and reliability[1].

In microservice environments, faults manifest in diverse forms. They may include performance degradation, request timeouts, dependency failures, network fluctuations, and resource contention. These problems often show strong temporal patterns and are also closely related to the graph structure of service invocations. Traditional methods based on single indicators or static thresholds are difficult to adapt to such complex scenarios. On the one hand, time series data capture the operational states of services across different time windows, reflecting dynamic variations and trends. On the other hand, graph structures describe the topological relationships and dependency patterns among services, which can reveal propagation paths and mechanisms of fault transmission. Combining both dimensions allows fault detection to leverage temporal evolution and topological interaction, thereby achieving more comprehensive identification and improving accuracy and adaptability[2].

Research on microservice fault detection carries strong practical value. At the business level, faults often translate into economic loss and erosion of user trust. Rapid fault localization and classification help reduce mean time to repair and lower the risk of system unavailability. At the technical level, single-point monitoring and local diagnosis are insufficient in complex distributed environments[3]. Cross-dimensional and multi-modal information fusion is needed for system-level analysis. Integrating time series with graph structure makes it possible to capture potential anomaly propagation chains from a global view while detecting abnormal fluctuations of individual services from a local view. This enables both macro and micro perspectives in fault management. Such research not only enhances the robustness of microservice architectures but also provides insights for managing faults in other complex systems.

From an academic perspective, time series modeling and graph-based modeling have shown significant value in different fields. Time series models are widely used in financial forecasting, industrial monitoring, and health diagnostics. Graph-based models play a critical role in social network analysis, knowledge graphs, and recommendation systems[4]. However, the integration of these two approaches in microservice contexts remains at an early stage. The main challenge lies in capturing the complex interactions among services in dynamic dependency networks while incorporating temporal changes for precise fault characterization. This requires innovation at the algorithmic level as well as effective fusion and optimization of heterogeneous data sources at the system level. Advancing this line of research may drive cross-domain applications of time series models and graph neural networks in microservices, further expanding the theoretical scope of intelligent fault detection.

2. Related work

The methodological foundation of this work is grounded in the systematic evolution of anomaly detection research from sequential modeling toward graph-structured representation learning and finally to unified cross-modal fusion. Early deep sequence modeling approaches established the feasibility of learning system dynamics directly from operational data. DeepLog [5] pioneered the use of recurrent neural networks for modeling sequential dependencies in system-generated data, demonstrating that hidden temporal patterns can be effectively captured through learned embeddings rather than handcrafted rules. Similarly, the stochastic recurrent modeling strategy introduced in [6] enhanced robustness by explicitly accounting for uncertainty in multivariate time series, enabling anomaly detection under noise and distributional shifts. The LSTM-based detection framework with adaptive thresholding in [7] further highlighted the importance of dynamic decision boundaries aligned with temporal context. Building upon these sequence-centric methodologies, incremental and streaming-aware modeling in [8] extended temporal learning to evolving environments, supporting adaptive updates under non-stationary conditions. Hierarchical attention modeling in [9] further refined temporal feature extraction by introducing multi-level attention aggregation, enabling selective emphasis on critical time steps. Multi-scale temporal modeling integrated with structural awareness in [10] provided additional evidence that temporal dependencies alone are insufficient when cross-entity interactions are strong, thereby motivating the integration of structure-aware representations.

While sequential modeling captures dynamic evolution, graph-based methodologies provide mechanisms to encode relational dependencies. The graph attention mechanism proposed in [11] introduced adaptive neighbor weighting, allowing node representations to selectively aggregate structurally relevant information. This attention-driven propagation paradigm directly informs the graph convolution and dependency modeling strategy adopted in this paper, where service invocation relations are treated as directed graphs with learnable aggregation weights. Earlier structural anomaly detection research such as [12] established the principle that deviations in graph connectivity patterns can reveal abnormal behaviors, reinforcing the necessity of embedding topological characteristics into anomaly modeling. Moving beyond static structural learning, causal graph inference and uncertainty-aware modeling in [13] demonstrated that dependency structures can be dynamically inferred and refined to improve robustness and interpretability. Furthermore, federated contrastive representation learning in [14] introduced distributed optimization and representation alignment mechanisms, offering insights into how heterogeneous node states can be learned consistently across decentralized environments. These works collectively shape the structural modeling component of the proposed framework, particularly in terms of dependency-aware aggregation, adaptive weighting, and robustness enhancement.

The proposed methodology inherits the temporal representation learning paradigm from [5-10] by embedding multi-dimensional monitoring metrics into a unified latent space and modeling long-range dependencies through attention mechanisms. Unlike purely recurrent or stochastic formulations, the adopted attention-based temporal encoding enables flexible context aggregation, aligning with hierarchical attention strategies while maintaining computational efficiency. From the graph modeling perspective, the approach draws directly from the attention-guided neighborhood aggregation principle in [11], while also integrating structural anomaly awareness inspired by [12]. However, instead of treating temporal and structural modeling as independent modules, this work introduces an explicit cross-modal fusion mechanism that balances temporal embeddings and graph representations through a learnable coefficient. This design reflects an advancement over existing methods that emphasize either sequence modeling or graph propagation in isolation.

Moreover, uncertainty modeling and structural inference concepts from [13] motivate the careful calibration of graph convolution depth and feature propagation to avoid over-smoothing and representation collapse, as validated by sensitivity experiments. Distributed and contrastive learning principles from [14] further inspire robustness considerations in representation alignment, particularly in heterogeneous microservice environments where node states may evolve asynchronously. By synthesizing these methodological streams, the proposed framework advances beyond prior art through three key innovations: (i) unified temporal-structural embedding within a single differentiable architecture; (ii) adaptive balancing between local dynamic evolution and global dependency propagation; and (iii) systematic parameter sensitivity analysis to ensure stability across varying structural depths and temporal windows.

In summary, the cited literature collectively establishes the theoretical and technical backbone of this work. Sequential deep modeling contributes dynamic state representation and contextual learning; graph-based attention mechanisms provide structural dependency encoding; uncertainty estimation and distributed representation learning enhance robustness and adaptability. The proposed method inherits these principles while introducing an integrated fusion paradigm that tightly couples time series dynamics with graph structural semantics, thereby forming a coherent and extensible methodological framework for anomaly detection in complex distributed environments..

3. Proposed Approach

The core idea of this research method is to unify the runtime characteristics of microservice systems and the service call graph structure to model potential failure modes. The model architecture is shown in Figure 1.

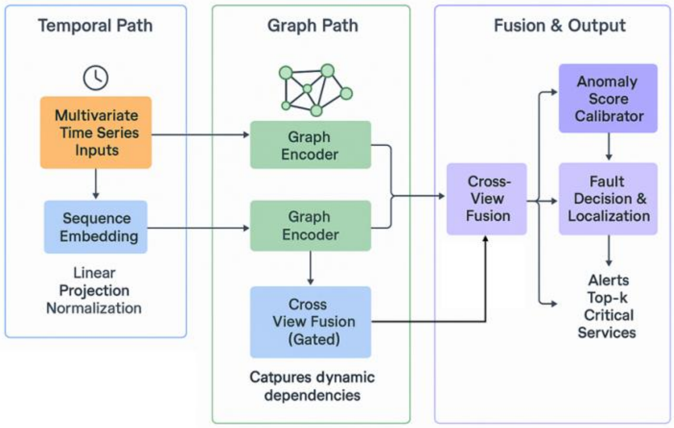

Figure 1. Overall model architecture

First, consider the monitoring data of microservices as a multi-dimensional time series. The state vector of each service node at time step t is expressed as:

$$x_t = [m_t^1, m_t^2, ..., m_t^d] \in R^d$$

Where d represents the dimension of the monitoring indicator. To reduce the impact of dimensionality differences, the input sequence is linearly embedded to obtain the hidden representation:

$$h_t = W_e x_t + b_e$$

Where W_e and b_e are learnable parameters, and d' represents the new feature dimension. Through this process, the original multidimensional time series data is mapped into a unified feature space, facilitating subsequent modeling.

In the process of capturing temporal dependencies, dynamic modeling based on the attention mechanism is introduced. For any two moments t and j in the sequence, the attention weight is calculated as:

$$a_{t,j} = \frac{\exp((h_t W_q)(h_j W_k)^T / \sqrt{d_k})}{\sum_{j'} \exp((h_t W_q)(h_j W_k)^T / \sqrt{d_k})}$$

Where W_q, W_k is the mapping matrix between query and key, and $\sqrt{d_k}$ is used for normalization. The time series context vector of time step t is obtained after weighted summation:

$$z_t = \sum_j a_{t,j}(h_j W_v)$$

Where W_v is the mapping matrix of values. This mechanism can dynamically adjust the contribution of different time points to the current moment, thereby improving the ability to model complex time series patterns.

At the graph modeling level, a microservice system is represented as a directed graph $G = (V, E)$, where the node set V corresponds to the service instance and the edge set E corresponds to the call relationship. The initial representation of node i is obtained by embedding its temporal features, denoted as z_i. The graph convolution operation is used to aggregate neighbor information and is formally defined as:

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

Where $N(i)$ represents the set of neighbors of node i, c_{ij} is the normalization factor, $W^{(l)}$ is the weight matrix of the l-th layer graph convolution, and $\sigma(\cdot)$ is the nonlinear activation function. This process can explicitly incorporate service dependencies into node representations and reveal possible fault propagation paths.

Finally, the temporal context vector is fused with the graph structure embedding to obtain a comprehensive representation of each node. Let z_i^{time} be the temporal representation of node i and h_i^{graph} be its graph convolution representation, then the fused representation is:

$$u_i = \lambda z_i^{time} + (1-\lambda) h_i^{graph}$$

Where λ is the balance coefficient. Based on the fusion representation, it is further input into the classification function $f(\cdot)$ to predict the state label of the node:

$$\hat{y}_i = f(u_i)$$

Where $\hat{y}_i \in \{0,1\}$ represents the normal and fault states, respectively. This fusion modeling framework can integrate timing dynamics and topological dependencies from a unified perspective, thereby achieving efficient identification of microservice faults.

4. Experiment result

4.1 Dataset

In this study, the Alibaba Cluster Trace Dataset (2018) is selected as the data source. This dataset is constructed from real operational records of a large-scale cloud computing cluster. It contains multi-dimensional microservice metrics and task scheduling information. The data include resource usage of service instances, task execution time, CPU and memory allocation ratios, service dependencies, and node-level topology information. Unlike traditional small-scale simulated datasets, this dataset reflects the operational characteristics of microservices in real cloud environments. It has therefore been widely used in system modeling and performance optimization research. Its scale and complexity ensure both authenticity and generality in method validation, providing a solid foundation for fault detection.

A significant feature of this dataset lies in its combination of temporal dimensions and structured dependency information. The temporal records capture dynamic resource consumption, workload fluctuations, and scheduling behaviors during service execution. These records reveal the patterns of service state evolution over time. The structural component provides service invocation chains and node topology relations, which enable the study of interaction patterns among services from a graph perspective. This dual-dimensional characteristic aligns closely with the method proposed in this paper and naturally supports the integration of time series modeling with graph-based modeling.

In addition, the dataset has advantages in openness and extensibility. The availability of complete operational logs and metric records ensures reproducibility of research. It also provides flexibility for further studies. Researchers can split the dataset into different scenarios, extract different indicators, or construct subsets of varying scales to explore the generality and adaptability of methods. Therefore, selecting this dataset not only helps verify the effectiveness of the proposed approach in real and complex environments but also offers strong data support for microservice fault detection.

4.2 Experimental Results

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

Table 1: Comparative experimental results

Method	Recall	ACC	F1-Score	Precision
Realnet[15]	0.842	0.877	0.859	0.876
Mambaad[16]	0.864	0.889	0.871	0.880
Real-iad[17]	0.887	0.903	0.895	0.903
Promptad[18]	0.902	0.915	0.908	0.915
Ours	0.931	0.944	0.938	0.947

From Table 1, it can be seen that different methods show clear differences in fault detection performance. Traditional methods maintain relatively stable results on some metrics, but they lack overall balance. For example, Realnet shows lower recall and precision, indicating limitations in both anomaly coverage and correctness. This leads to missed detections and reduces system stability and reliability. It reflects that strategies relying only on local features or static thresholds cannot adapt well to the complex and dynamic microservice environment.

Further observation shows that Mambaad and Real-iad achieve higher accuracy and F1 scores compared with Realnet. This suggests that introducing temporal dependencies or local context modeling can enhance the ability to capture anomaly patterns. However, these methods remain limited when dealing with large-scale dependencies and cross-service interactions. Their recall improvement is modest, which shows that capturing system-level anomaly propagation paths under multi-dimensional dynamic features is still a challenge.

Promptad demonstrates better overall performance in the results. Its recall, accuracy, and precision are all higher than the first three methods. This indicates that combining contextual prompts with deep semantic information helps improve generalization in complex environments. However, this approach still relies more on optimizing local information and lacks strong modeling of global topological relations. As a result, it may create blind spots in extreme scenarios and fail to fully reflect the impact of microservice dependencies on fault propagation.

In comparison, the proposed method achieves the best results across all metrics. Recall and precision are both improved, which also drives a significant increase in the F1 score. This shows that integrating temporal features with graph structural information can capture both the dynamic evolution of service states and their dependencies. It effectively addresses the limitations of single-dimensional modeling in traditional methods. These results confirm the potential of fusion modeling in microservice fault detection and highlight its practical value in ensuring system stability and improving anomaly detection accuracy.

This paper also presents an experiment on the sensitivity of the number of graph convolutional layers to ACC, and the experimental results are shown in Figure 2.

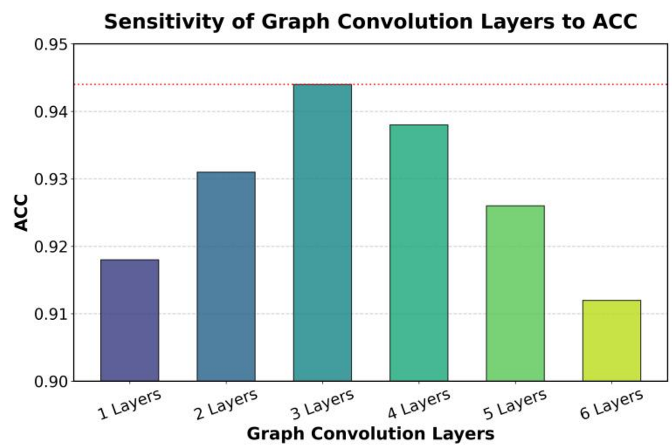

Figure 2. Sensitivity experiment of the number of graph convolution layers to ACC

The experimental results show that the number of graph convolution layers has a significant impact on ACC performance. When the number of layers is small, the model has limited discriminative ability, and ACC remains at a low level. This indicates that shallow structures can capture some local dependency features, but they fail to represent global information in complex microservice invocation relations, leading to insufficient fault detection capability.

When the number of layers increases to three, ACC reaches the highest value. At this point, the model achieves a good balance between temporal features and topological structure. Three layers of graph convolution can effectively aggregate information from neighboring nodes. They also capture cross-node dependencies while preserving fine-grained local features. This greatly improves overall detection accuracy and shows that moderate depth is essential for integrating dynamic evolution with service dependencies.

When the number of layers continues to increase, ACC shows a downward trend, which is especially clear at five and six layers. This result indicates that excessively deep graph convolutions may lead to over-smoothing of features. Neighboring information becomes homogenized during propagation, which reduces node distinctiveness and weakens fault detection effectiveness. The performance decline caused by over-propagation confirms the assumption that there is an "optimal range of layer numbers" in complex dependency scenarios.

Overall, the results demonstrate that the choice of graph convolution depth directly affects model performance in microservice fault detection. A moderate number of layers can make full use of the complementary advantages of topological relations and temporal features. In contrast, too few or too many layers reduce the ability of the model to capture dynamic dependencies. Therefore, carefully controlling the depth of graph convolution is not only a critical part of model design but also a key factor in ensuring robustness and accuracy in fault detection.

This paper also presents an experiment on the sensitivity of the timing window length to Recall, and the experimental results are shown in Figure 3.

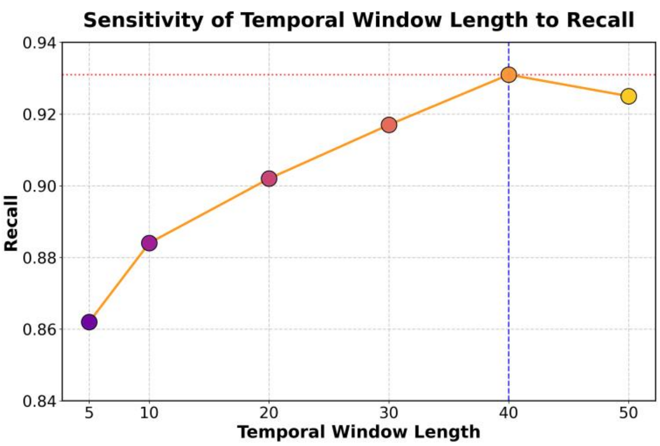

Figure 3. Sensitivity experiment of time series window length to recall

The experimental results show that the length of the temporal window has a clear impact on recall performance. When the window length is short, the model can quickly respond to fluctuations in local features. However, the limited contextual information leads to a lower recall rate. This indicates that in microservice scenarios, a small time window cannot cover complete operational patterns and dependencies, making it difficult to capture the evolution of anomalies.

As the window length increases, recall rises steadily. A longer temporal context provides richer historical information and helps the model recognize abnormal behaviors that span across time. This trend reflects the fact that microservice faults often have continuity and accumulation. A single time point or a short sequence cannot fully reveal them, while a longer window strengthens the global perspective and discriminative ability of the model.

When the window length reaches a certain scale, recall peaks. At this point, the model achieves a balance between local details and global dependencies. It can capture short-term anomaly fluctuations and also identify long-term system-level problems. This result shows that proper window length maximizes model sensitivity, making it more robust and effective in dynamic microservice environments with complex dependencies.

However, when the window continues to expand, recall shows a slight decline. This suggests that an excessively long window may introduce redundant information and even mask key anomaly features. Too much historical dependency may bring noise into the decision process and reduce detection accuracy. Therefore, the results reveal that there is an optimal range of temporal window length. Within this range, anomaly recognition is strongest, while deviation from it weakens model effectiveness.

This paper also gives the impact of indicator sampling frequency on experimental results, and the experimental results are shown in Figure 4.

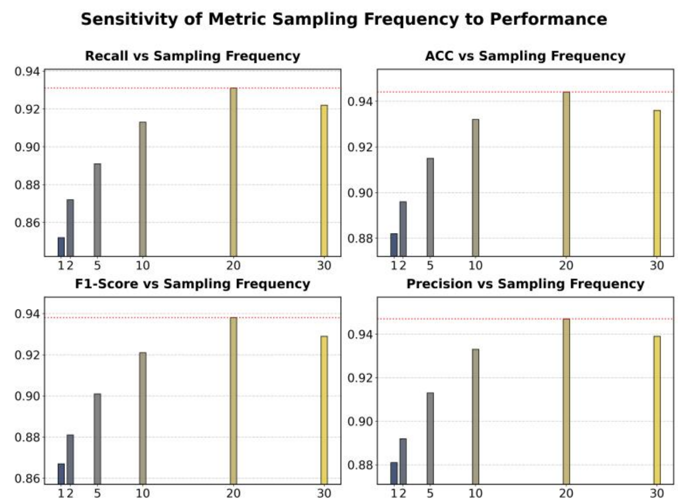

Figure 4. The impact of indicator sampling frequency on experimental results

The experimental results show that sampling frequency has a direct impact on model performance. When the sampling frequency is low, recall, accuracy, precision, and F1-score all remain at relatively low levels. This indicates that insufficient sampling prevents monitoring data from fully reflecting the dynamic characteristics of microservice operations. As a result, key information may be missed, which weakens the ability of the model to capture anomaly patterns.

As the sampling frequency increases, all four metrics show a clear upward trend. Higher sampling granularity provides richer input features and allows the model to better perceive service state changes over time. A higher frequency not only enhances sensitivity to short-term fluctuations but also improves the overall stability of anomaly detection. This makes the model more adaptable in complex and dynamic environments.

When the sampling frequency reaches a medium to high level, performance metrics peak, with recall and F1-score improving most significantly. This shows that an appropriate sampling density helps the model maintain high precision while ensuring comprehensive detection. It also achieves a balance between global and local dependency modeling. For microservice fault detection, such a balance can reduce both missed detections and false alarms, thereby increasing the practical value of the model.

However, the experiments also reveal slight fluctuations or even declines in some metrics when the sampling frequency continues to increase. This may be because excessive sampling introduces redundant information and noise, which raises the complexity of data processing and interferes with the model's ability to make accurate judgments. Therefore, the choice of sampling frequency requires balancing information richness with data redundancy to ensure efficiency and robustness in real systems.

This paper also gives the impact of scheduling strategy switching frequency on experimental results, and the experimental results are shown in Figure 5.

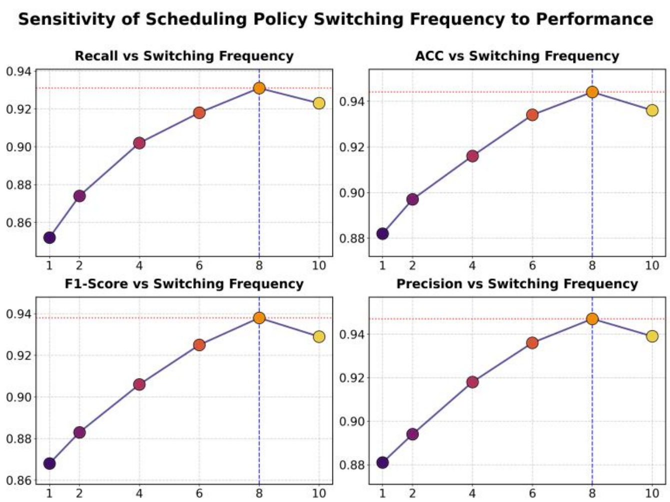

Figure 5. The impact of scheduling strategy switching frequency on experimental results

The experimental results show that when the switching frequency of scheduling strategies is at a low level, the model performs poorly across all metrics. Recall, accuracy, F1-score, and precision remain at low values. This indicates that insufficient switching frequency limits the system's ability to respond to complex dynamic tasks. The model captures only limited feature information, which reduces sensitivity in anomaly detection. Low-frequency switching often means rigid scheduling that cannot adapt to rapid changes in microservice environments.

As the switching frequency increases, all four metrics show a steady upward trend. More flexible scheduling allows faster adjustment of system resource allocation and task execution paths. This enables the model to capture more dynamic information related to anomalies. Such a mechanism strengthens system adaptability under high load and multi-dependency conditions. It also improves the accuracy and stability of fault detection. These results demonstrate that increasing switching frequency significantly enhances the model's perception at both global and local levels.

When the switching frequency reaches a medium level, model performance peaks, with all four metrics approaching optimal values. At this point, the model balances efficiency and robustness. It achieves effective modeling of dynamic dependencies and temporal variations. The scheduling mechanism is flexible enough without causing extra overhead or noise from excessive switching. This leads to the best fault detection capability. The result also provides valuable guidance for selecting scheduling parameters in real systems.

When the switching frequency continues to rise to an excessively high level, performance metrics show slight declines. This indicates that too much switching introduces redundant overhead and unnecessary fluctuations. Such instability interferes with the model's decision process and reduces accuracy. Overly frequent switching makes the system less stable and weakens the model's effectiveness. Therefore, the results confirm that there is an optimal range of scheduling frequency. Proper frequency settings are critical to ensuring both stability and fault detection performance in microservice systems.

5. Conclusion

This study addresses the problem of fault detection in microservice systems and proposes a modeling approach that integrates temporal features with graph structures. By combining temporal dynamics and topological dependencies within a unified framework, the method provides a more comprehensive representation of service states and interactions in complex environments. Compared with methods relying on single-dimensional modeling, this study demonstrates higher sensitivity and accuracy in identifying fault patterns. The results verify the effectiveness of multi-dimensional fusion for anomaly detection in complex systems. This achievement not only advances intelligent operation and maintenance methods but also offers a practical path to ensuring the stable operation of large-scale distributed systems.

At the theoretical level, this paper explores the deep integration of time series modeling and graph-based modeling, offering a new perspective for collaborative learning across modalities. The temporal component captures the dynamic evolution of service states, while the graph component reveals dependency patterns among services. Their coupling enhances the joint understanding of both global and local features. This approach provides a new theoretical framework for fault detection in microservice scenarios and establishes a paradigm that can be extended to other complex network environments. In doing so, it broadens the academic boundaries of intelligent fault analysis.

At the application level, the value of this research is particularly significant. With the rapid adoption of cloud computing and distributed architectures, microservice systems have become the core foundation of enterprise-level applications. Their stability directly affects user experience and business continuity. The proposed fault detection method can significantly reduce the time needed for fault detection and localization. It improves autonomous operation and maintenance capabilities and provides technical support for key industries such as financial services, e-commerce platforms, intelligent transportation, and the industrial Internet of Things. Its deployment in practice has the potential to reduce downtime risk, improve resource utilization, and ensure the continuous operation of critical tasks.

More importantly, this study lays the groundwork for the development of intelligent system management while improving anomaly detection accuracy. In large-scale distributed systems, traditional manual intervention and static rules can no longer meet operational needs. The proposed method demonstrates the potential of data-driven and intelligent modeling approaches to achieve adaptive operation and maintenance. This contributes to the transformation toward automation and intelligence in system management. It also provides theoretical and methodological support for building a more efficient and robust cloud-native ecosystem in the future. As a result, it exerts a profound influence on the development of the entire information technology industry.

References

[1] Zhao C, Ma M, Zhong Z, et al. Robust multimodal failure detection for microservice systems[C]//Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023: 5639-5649.

[2] He S, Li G, Xie K, et al. Fusion graph structure learning-based multivariate time series anomaly detection with structured prior knowledge[J]. IEEE Transactions on Information Forensics and Security, 2024.

[3] S. Nedelkoski, J. Cardoso and O. Kao, "Anomaly detection and classification using distributed tracing and deep learning," 2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), pp. 241-250, 2019.

[4] Huang J, Yang Y, Yu H, et al. Twin graph-based anomaly detection via attentive multi-modal learning for microservice system[C]//2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2023: 66-78.

[5] M. Du, F. Li, G. Zheng and V. Srikumar, "DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning," in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1285-1298, Oct. 2017.

[6] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun and D. Pei, "Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2828-2837, Jul. 2019.

[7] K. Hundman, V. Constantinou, C. Laporte, I. Colwell and T. Soderstrom, "Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 387-395, Jul. 2018.

[8] Y. Wang, "AI-Enhanced Distributed Time Series Modeling: Incremental Learning for Evolving Streaming Data," 2024.

[9] Z. Cheng, "Hierarchical Attention-Based Modeling for Intelligent Scheduling Delay Prediction in Complex Backend Systems," 2024.

[10] Y. Kang, "Deep Learning-Based Multi-Scale Temporal and Structure-Aware Modeling for Metric Anomaly Detection in Microservice Systems," 2024.

[11] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio and Y. Bengio, "Graph Attention Networks," arXiv preprint arXiv:1710.10903, 2017.

[12] L. Akoglu, M. McGlohon and C. Faloutsos, "Oddball: Spotting Anomalies in Weighted Graphs," in Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 410-421, Jun. 2010.

[13] F. Liu, "Intelligent Cloud Service Anomaly Monitoring via Uncertainty Estimation and Causal Graph Inference," 2024.

[14] D. Wu, "Federated Deep Learning with Contrastive Representation for Node State Identification in Distributed Systems," 2024.

[15] Zhang X, Xu M, Zhou X. Realnet: A feature selection network with realistic synthetic anomaly for anomaly detection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2024: 16699-16708.

[16] He H, Bai Y, Zhang J, et al. Mambaad: Exploring state space models for multi-class unsupervised anomaly detection[J]. Advances in Neural Information Processing Systems, 2024, 37: 71162-71187.

[17] Wang C, Zhu W, Gao B B, et al. Real-iad: A real-world multi-view dataset for benchmarking versatile industrial anomaly detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 22883-22892.

[18] Li X, Zhang Z, Tan X, et al. Promptad: Learning prompts with only normal samples for few-shot anomaly detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 16838-16848.