Journal of Computer Technology and Software
ISSN:2998-2383
Vol. 5, No. 8, 2026

Deep Learning Approach to Structure-Temporal Collaborative Anomaly Detection in Microservice Architectures

Di Wu[1*]
[1]Washington University in St. LouisSt. Louis, USA
*Corresponding author: Di Wu; inbox.diwu@gmail.com

Abstract: This paper proposes a structure-temporal collaborative anomaly detection method to address the challenges of strong structural dynamics, complex service dependencies, and diverse temporal behavior patterns in microservice architectures. The proposed method includes two core modules: Structure-Aware Dependency Modeling (SADM) and Multi-Channel Temporal Representation (MCTR). The SADM module constructs a service invocation dependency graph and employs graph neural networks to jointly model node behavior and topological structure, enhancing the model's ability to represent structural disturbances and abnormal paths. The MCTR module designs multiple heterogeneous temporal modeling paths, including local convolution, dilated convolution, and residual diffusion mechanisms, to capture dynamic behavioral changes across different time scales. These features are effectively fused through a channel attention mechanism to improve the model's ability to distinguish complex anomaly patterns. Experiments are conducted on public microservice datasets, including comparative and ablation studies. The results show that the proposed model outperforms existing methods in key metrics such as Precision, Recall, F1-score, and AUROC. Further sensitivity analysis verifies the impact of different structural parameters and temporal modeling configurations on model performance, demonstrating the effectiveness and robustness of the proposed method in structural and temporal modeling.

Keywords: microservice anomaly detection, structural modeling, multi-channel time series representation, graph neural network

1. Introduction

With the development of cloud-native architectures, microservice systems have been widely adopted in large-scale distributed applications due to their modular decoupling, flexible deployment, and elastic scalability. Compared with traditional monolithic systems, microservice architectures offer greater dynamism and scalability. However, they also introduce complex service dependencies and asynchronous communication. In such highly heterogeneous and continuously evolving environments, anomalies may originate from single-point failures or propagate through service call chains, leading to system-level stability risks. Therefore, it is of significant research value to build anomaly detection models with both structure-awareness and temporal modeling capabilities to enhance the observability and robustness of microservice systems[1, 2].

Existing anomaly detection methods for microservices mainly focus on node-level metric analysis or time series modeling. Few approaches consider the dynamic evolution of service dependency structures. Some methods adopt graph neural networks to encode service relationships, but these are often simplified as static topologies or lack integration with behavioral information. As a result, they fail to capture chained anomaly paths and structural disturbance patterns. In addition, many temporal modeling mechanisms rely on single-path feature extraction, which is insufficient for representing complex fluctuations of anomalous behaviors across multiple time scales[3, 4]. This limitation reduces their effectiveness in handling behavior drift and weak anomalies. The lack of coordination between structural and temporal modeling also limits overall detection capabilities, as current methods lack effective collaborative strategies.

To address these issues, this paper proposes a structure-temporal collaborative anomaly detection framework. It jointly models the service dependency graph and node behavior sequences to enable unified structural dynamics modeling and multi-scale behavior recognition in microservice systems. The framework includes a structure-aware dependency modeling mechanism and a multi-channel temporal representation mechanism. These are designed to capture topological evolution features and behavior changes at different temporal granularities[5, 6]. Together, they enhance the model's sensitivity and robustness in anomaly representation. The overall architecture supports generalizable structural learning, adaptive behavior modeling, and efficient information fusion, making it suitable for various dynamic microservice environments.

This work includes two main contributions. First, it introduces a structure-aware dependency modeling mechanism that constructs dynamic graphs and encodes service relationships using graph neural networks. This enhances the model's ability to represent topological changes and structural anomalies[7]. Second, it designs a multi-channel temporal representation mechanism that builds heterogeneous temporal modeling paths to extract behavior changes across scales. A channel attention fusion strategy is used for unified encoding.

These two components work together to construct a unified anomaly representation space. This resolves the separation between structural and behavioral modeling and provides a more expressive solution for improving anomaly detection performance in microservice systems[8].

2. Related work

2.1 Anomaly Detection in Microservice Systems

Microservice architecture has been widely adopted in building large-scale cloud-native systems due to its advantages in flexible deployment, on-demand scaling, and service autonomy. However, its core design principle of minimal service granularity and functional decoupling also brings significant complexity and unpredictability to system behaviors. Each microservice unit typically undertakes a specific function. Its runtime state depends not only on local workload and environment settings but also on fluctuations in upstream requests and anomalies in downstream responses[9, 10]. Against the background of deep dependency chains and strong service coupling, a local failure can easily trigger a chain reaction, posing a threat to overall system stability. Therefore, identifying potential anomalies from multidimensional monitoring data and uncovering their propagation paths and influence ranges has become a critical challenge in microservice operations[11].

Early anomaly detection methods were mainly based on static rules and manually set thresholds. These methods monitor resource metrics such as CPU, memory, and response time. When values exceed predefined thresholds, alarms are triggered. Although simple and low-cost, these methods are not suitable for highly concurrent and tightly coupled environments. They often suffer from high false-positive and false-negative rates. To address the nonlinear evolution of microservice states, later approaches introduced statistical modeling and unsupervised learning. Techniques like clustering, dimensionality reduction, and change detection based on historical distributions have been used. These methods alleviate the rigidity of fixed thresholds to some extent. However, they still struggle with missing contextual information between services and the unobservability of anomaly propagation chains.

In recent years, deep learning methods have shown strong representation capabilities in system anomaly detection. Particularly, time series modeling frameworks such as recurrent neural networks and convolutional networks have been used to capture temporal dependencies and evolving patterns from multidimensional metrics. These methods often take logs, tracing data, or resource usage sequences as input. The models learn behavioral trends to adaptively detect anomalies. However, microservice anomalies are not always reflected by deviations in individual metrics. They often involve complex call paths, topological changes, and dependency coupling. Therefore, methods based purely on temporal features are limited in identifying structural anomalies or semantic shifts. Especially when cross-service call failures or chained anomaly propagation occur, time-series models alone cannot fully reveal the root causes[12, 13].

In addition, most existing methods focus on single-dimensional data or local behavior patterns. They cannot model and utilize the global dependency structure of services. In microservice systems, the relationship graph among components evolves dynamically. Its topology changes with container deployment, service scaling, and version updates. This dynamic structure is essential to understanding anomaly occurrence and propagation. Traditional methods often simplify it as static or ignore it entirely. As a result, they fail to capture structural disturbances. In practice, many small fluctuations are undetectable by analyzing isolated metrics. Only by incorporating upstream and downstream dependencies and understanding semantic context can more robust and discriminative anomaly detection be achieved. This challenge creates an urgent need for more structured and context-aware modeling methods. It also lays the foundation for introducing graph-based learning mechanisms.

2.2 Graph Neural Networks for System Modeling and Anomaly Detection

Graph neural networks (GNNs) have emerged as a significant advancement in modeling graph-structured data. They exhibit strong potential in various complex system modeling tasks due to their ability to jointly represent node features and structural relations. The core mechanism lies in aggregating and propagating information from neighboring nodes. This enables hierarchical modeling of contextual semantics in non-Euclidean structures. In traditional graph tasks, GNNs have demonstrated effectiveness in node classification, edge prediction, and graph-level regression. In system modeling, especially within microservice architectures, graph structures exist naturally. Services are represented as nodes, and invocation or dependency relations as edges, forming a directed interaction graph. Introducing GNNs into this setting allows expressive modeling of dynamic dependencies. It helps capture semantic interactions and anomalous behavior paths among services. This provides a structurally aware feature representation for system state analysis and risk perception[14, 15].

In microservice anomaly detection, the structure-aware capability of GNNs plays a critical role. Unlike traditional neural networks that process independent samples or linear sequences, GNNs can learn both local and global structural information within irregular topologies[16]. This offers a natural advantage for modeling service dependency chains. Through multi-layer graph convolutions or attention mechanisms, the model can perceive a service node's role in the global invocation graph, the strength of its upstream dependencies, and the extent of its downstream influence. This ability is essential for detecting structural anomalies such as broken call chains or abnormal load on key nodes. Moreover, GNNs support heterogeneous graph modeling and multi-source data integration. They can jointly utilize call traces, metric fluctuations, and tracing logs. This enhances the precision of anomaly detection from both structural and behavioral perspectives[17].

Recent studies have further explored integrating GNNs with temporal modeling to better handle dynamic dependency

graphs. In microservice systems, service dependencies evolve due to changes in workload, scaling operations, and routing strategies. To address this, some methods extend graph modeling to spatiotemporal GNNs. These models incorporate time-series embeddings with structural propagation to support continuous modeling of dynamic behavior. Other studies use graph attention to focus on abnormal propagation chains within local structures. These approaches aim to capture cascading failures triggered by single-point faults. Additionally, to tackle issues of sparse anomaly data and limited labels in real systems, unsupervised graph learning and contrastive learning mechanisms have been introduced. These methods help improve model generalization and sensitivity to weak anomalies[18.19].

Although GNNs offer a new paradigm for microservice anomaly detection, several practical challenges remain. On one hand, dependency graphs in microservice systems are typically large, sparse, and frequently updated. Efficient modeling and real-time updating of large-scale dynamic graphs remain bottlenecks in practical applications. On the other hand, anomaly signals are often unevenly distributed across the graph. They are surrounded by a large volume of normal patterns, which increases the difficulty of discrimination and generalization. Furthermore, service nodes often exhibit semantic heterogeneity and behavioral differences. This makes it difficult for a single graph propagation mechanism to cover all anomaly types. As a result, recent work has proposed structure-enhanced mechanisms and multi-view learning strategies. These developments aim to improve the adaptability and expressiveness of GNNs in microservice scenarios. They provide important insights and directions for building structure-aware and dynamically adaptive anomaly detection models[20].

3. Method

This study proposes a graph neural network-based anomaly detection framework for microservice systems. The goal is to accurately model complex service dependencies and enhance the system's ability to identify abnormal behaviors. The framework integrates two key innovations: Structure-Aware Dependency Modeling (SADM) and Multi-Scale Anomaly Propagation Modeling (MAPM). SADM constructs high-order interaction graphs among services using graph neural networks. It explicitly captures dynamic structural semantics between service nodes and addresses the limitations of traditional methods in representing complex dependencies. MAPM models dynamic anomaly propagation based on multi-scale behavioral evolution along upstream and downstream service paths. It captures the structural transition of anomalies at both local perturbation and global diffusion levels. This significantly improves the detection of sparse chained anomalies and structural distortions. The proposed framework jointly optimizes structural modeling and anomaly representation. It provides a robust foundation for anomaly detection in microservice environments. The detailed structure of the proposed model is illustrated in Figure 1.

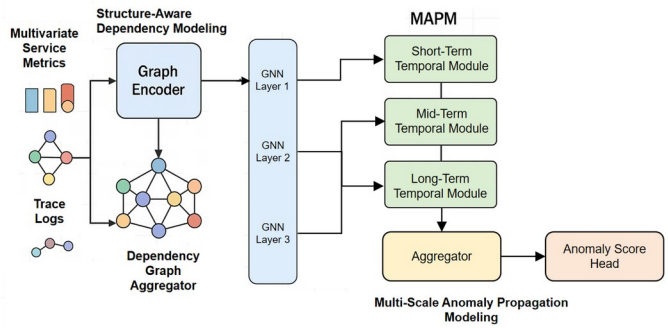


Figure 1. Overall model architecture diagram

3.1 Structure-Aware Dependency Modeling

This study introduces a Structure-Aware Dependency Modeling (SADM) mechanism to enhance the model's ability to represent dynamic structural dependencies in microservice systems. The goal is to jointly encode service invocations, topological semantics, and node behavioral features into structurally consistent graph representations. This enables the characterization of potential anomaly propagation paths and contextual information. Unlike traditional methods that focus only on node attributes or static dependency graphs, SADM constructs time-varying dependency graphs based on time-driven call traces and multidimensional service metrics. It embeds local behavioral shifts and links disturbance patterns into the graph structure.

The architecture of the SADM module is illustrated in Figure 2. Specifically, raw service features and dependency structures are jointly fed into a graph neural network encoder. The model captures each node's structural role and the aggregated features from its neighborhood. This mechanism allows the model to perceive critical path dependencies through multiple rounds of graph propagation. As a result, it forms high-order semantic representations that are sensitive to structural perturbations.

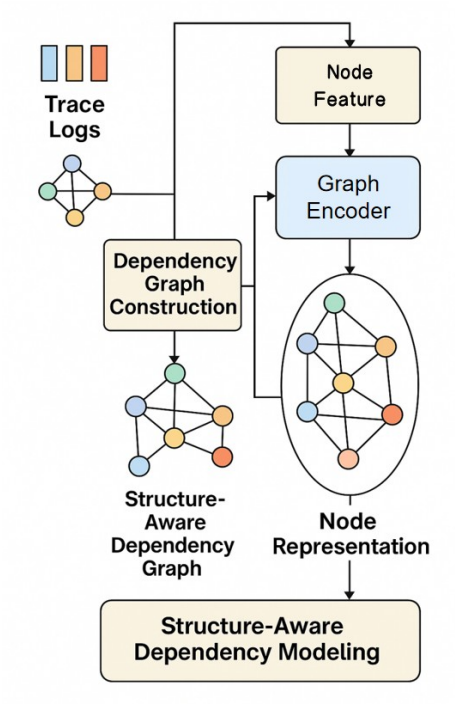


Figure 2. SADM module architecture

The overall SADM process consists of four stages: dependency graph construction, node feature encoding, structure-aware propagation, and graph representation aggregation. The goal is to generate a graph embedding representation that can distinguish between normal and abnormal structural states. Suppose the microservice dependency graph constructed at time $G_t = (V_t, \varepsilon_t)$ is V_t, where ε_t represents the service node set, D represents the call edge set, and the initial node features are:

$$h_i^0 = x_i$$

Through multi-layer information propagation with graph neural networks, node representations are iteratively updated as follows:

$$h_i^{(l)} = \mathrm{ReLU}\left(W^{(l)} \cdot \mathrm{AGG}\left(\left\{h_j^{(l-1)} \mid j \in N(i)\right\}\right) + b^{(l)}\right)$$

AGG represents the neighbor aggregation function, $W(l)$ and $b^{(l)}$ are learnable parameters. The final node representation is integrated into a graph-level structure embedding through the readout function:

$$z_t = \mathrm{READOUT}\left(\left\{h_i^{(L)}\right\}\right)$$

To model and supervise the structural state, the following loss function is designed for optimization:

$$L_{\mathrm{sadm}} = |z_t - \hat{z}_t||^2 + \alpha \cdot \mathrm{KL}\left(p(z_t)||q(\hat{z}_t)\right)$$

The first term represents the structural reconstruction error, and the second term imposes a distributional consistency constraint on the latent representations. This loss function guides the model to maintain global dependency consistency while enhancing sensitivity to local structural distortions. It also provides structurally stable and distribution-aligned graph representations for the subsequent anomaly propagation module, thereby improving the ability to identify structural anomalies.

3.2 Multi-Channel Temporal Representation

This study introduces a Multi-Channel Temporal Representation (MCTR) mechanism to enhance the model's ability to capture service node behavior patterns across multiple temporal scales. Its module architecture is shown in Figure 3. This mechanism employs multiple structurally heterogeneous temporal modeling channels in parallel. It enables the model to perceive both local perturbations and global trends. As a result, it achieves dynamic fusion of short-term variations and long-term dependencies. Traditional methods often rely on a single temporal path when modeling service state evolution. This makes it difficult to handle the temporal diversity of different anomaly patterns in microservice systems.

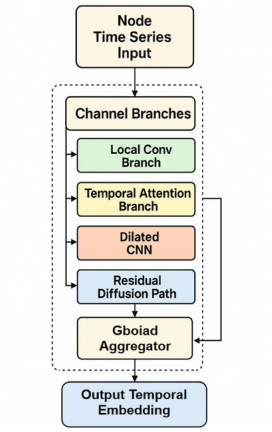


Figure 3. MCTR module architecture

The MCTR module builds a multi-branch structure by incorporating local convolution paths, attention paths, dilated convolution paths, and residual diffusion paths. It models the same input sequence from multiple temporal perspectives. Finally, all branch outputs are fused through a channel attention aggregator. This results in an integrated temporal embedding representation for the subsequent anomaly detection module.

In the specific modeling process, the time series input of service node v_i is assumed to be matrix $X_i \in [r, t]$, where t represents the number of time steps and $f_i^{(t)}$ represents the feature dimension of each step. The input is expressed as:

$$f_i^{(0)} = X_i$$

The input is fed into channel structures to extract temporal features of different semantic granularities:

$$f_i^{(k)} = \mathrm{Branch}_k\left(f_i^{(0)}\right), k = 1, 2, \ldots, K$$

All channel outputs are weighted and fused through the attention mechanism to generate a unified multi-scale embedding:

$$\tilde{f}_i = \sum_{k=1}^{K} \alpha_k \cdot f_i^{(k)}, \alpha_k = \frac{\exp(w_k)}{\sum \square_{j=1}^{K} \exp(w_j)}$$

The model finally uses the fusion representation A and its prediction result B to construct the optimization objective function as follows:

$$\text{Lmctr} = \frac{1}{N} \sum_{i=1}^{N} ||\tilde{f}_i - \hat{f}_i||^2 + \beta \cdot \text{KL}(p(\tilde{f}) || q(\hat{f}))$$

The first term of the loss function ensures reconstruction consistency between the multi-channel fused features and the target representations. The second term aligns the distributions to suppress redundant interference across channels and enhance the stability of the representations. The overall loss function encourages the MCTR module to establish a collaborative learning mechanism during multi-scale modeling, which effectively improves its ability to represent complex behavioral patterns over time.

4. Experimental Results

4.1 Dataset

This study uses the Microservices Bottleneck Localization Dataset, a publicly available resource released on the Kaggle platform in 2022. It can be found using the keyword "microservices bottleneck detection dataset." The dataset is based on microservice applications running on Kubernetes, such as the DeathStarBench social network scenario. It contains approximately 40 million request tracing records, along with corresponding time series of CPU, memory, I/O, and network metrics. The data covers both multi-dimensional performance indicators and link-level behaviors, making it highly relevant for research on joint structural and temporal anomaly detection in microservice systems.

The dataset provides complete request call chain information, including request paths, timestamps, parameters, and response statuses. Performance metrics are sampled at high frequency, forming multi-channel time series inputs. This multimodal design aligns well with the dual modeling requirements of this study, namely Structure-Aware Dependency Modeling (SADM) and Multi-Channel Temporal Representation (MCTR). It supports both topological structure mining and dynamic analysis of node behaviors. The dataset's large scale and complex structure offer rich samples for model training and evaluation, improving generalization and robustness.

To focus on structural and behavioral representation, this study uses only two types of data: request chains and performance metrics. These are segmented into graph sequences and temporal fragments based on time windows. Each time window generates a dependency graph and corresponding time features, which are fed into the SADM and MCTR modules for structure-aware and multi-scale representation. The dataset's high-dimensional, multi-channel temporal and structural information provides a solid, open, and reproducible experimental foundation. It meets the standardized requirements of anomaly detection research in microservices.

4.2 Experimental setup

All experiments in this study were conducted on a high-performance deep learning workstation equipped with GPUs. The hardware configuration includes an Intel Xeon Silver 4310 CPU (2.1 GHz, 12 cores), 256 GB of DDR4 memory, and two NVIDIA A100 GPUs with 40 GB of memory each. The system runs on Ubuntu 22.04. The software environment consists of Python 3.10 and PyTorch 2.1.0. CUDA 12.1 and cuDNN 8.9 ensure efficient parallel processing of large-scale dependency graphs using deep graph models.

During training, all models are optimized using the Adam optimizer. The initial learning rate is set to 0.0005 and decayed by a factor of 0.8 every 10 epochs. The batch size is set to 64 based on GPU memory availability. The maximum number of training epochs is 200. A dropout rate of 0.3 is used to prevent overfitting in the temporal branches. The L2 regularization coefficient is fixed at 1e-4 to suppress parameter explosion.

The graph neural network module uses three GNN layers, with a hidden dimension of 256 and ReLU as the activation function. In the multi-channel temporal modeling module, four independent channels are constructed. These channels correspond to local convolution, temporal attention, dilated convolution, and residual diffusion paths. Each branch output has a dimension of 64, and the outputs are fused into a unified 256-dimensional representation using a channel attention mechanism. All experiments use a fixed random seed to ensure reproducibility. The training time for each epoch is kept under 10 minutes with GPU acceleration.

4.3 Experimental Results

4.3.1 Comparative experimental results

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

Table 1. Comparative experimental results

Method	Precision (%)	Recall (%)	F1-score (%)	AUROC (%)
GDN [19]	89.4	87.2	88.3	93.1
MTAD-GAT[20]	90.6	86.5	88.5	92.4
GTA [21]	91.3	88.7	90	94.2
TranAD [22]	88.9	89.4	89.1	93.8
Ours (SADM+MCTR)	92.7	90.8	91.7	95.6

The experimental results show that structure-based graph neural network methods, such as GDN and GTA, demonstrate strong global modeling capabilities in microservice anomaly detection. These models exhibit clear advantages in the AUROC metric. This indicates that in microservice architectures with complex dependency topologies, structure-aware mechanisms are effective in capturing cross-node

anomaly propagation paths. In contrast, traditional temporal modeling methods like TranAD perform well in modeling temporal trends but have limitations in handling service-level contextual interactions. As a result, their overall performance is slightly lower.

Further comparisons with models that integrate structural and temporal features show that GTA and MTAD-GAT improve the discrimination of node behavior dynamics by incorporating graph attention and temporal attention mechanisms. The improvement in the F1-score metric reflects the robustness of these models in balancing precision and recall. However, MTAD-GAT still shows conservative modeling across complex channels. Its reliance on single-scale temporal convolution makes it difficult to fully capture the diversity and nonlinear propagation characteristics of microservice anomalies.

In comparison, the proposed method that combines Structure-Aware Dependency Modeling (SADM) and Multi-Channel Temporal Representation (MCTR) achieves superior performance across all metrics. It shows particularly strong results in AUROC and F1-score. This confirms that integrating global dependency structure modeling with multi-scale temporal dynamics helps capture fine-grained anomaly patterns and their propagation paths. It also enhances the model's ability to distinguish heterogeneous behavioral patterns. The joint mechanism creates a highly complementary embedding space in both structural and temporal dimensions, improving detection accuracy and stability.

In addition, the MCTR module uses channel-wise attention to fuse behavioral representations from different temporal paths. This helps reduce false detections in scenarios involving anomaly peaks and periodic shifts. The SADM module strengthens the model's response to structural disturbances and role transitions in the service call graph. Together, these modules maintain stable and sensitive detection performance under dynamic topology evolution and sparse anomaly injection. This demonstrates strong generalization and industrial adaptability.

4.3.2 Ablation Experiment Results

Ablation studies aim to evaluate the specific contribution of each key component to the overall performance of the model. This is an essential step in the design and validation of deep models. By selectively removing or replacing modules, the study reveals the actual impact of each substructure on performance improvement. It also provides empirical support for the rationality of the model architecture. In addition, the experiment helps analyze the model's sensitivity to different structural configurations, offering guidance for future optimization and module selection strategies. Table 2 presents the performance variations after sequentially removing core modules from the complete model. The comparisons cover both structural and temporal modeling components. The results show that different submodules have uneven effects on the final performance. Some components contribute more significantly to detection accuracy and robustness. The overall trend further confirms the effectiveness and necessity of the collaborative design mechanism used in the model.

Table 2. Ablation Experiment Results

Method	Precision	Recall	F1-score	AUROC
Baseline	87.2	84.5	85.8	91.7
+ SADM	89.3	86.7	88	93.2
+ MCTR	88.7	87.5	88.1	92.9
Ours	91.1	89.4	90.2	94.8

The ablation results show that the base model, without support from structural and temporal modeling, performs poorly in terms of accuracy and stability. This indicates that relying solely on shallow features or static behavioral information is insufficient for capturing the highly heterogeneous and dynamically evolving anomaly patterns in microservice systems. In particular, it struggles to model cross-node behavior propagation and service link dependencies. The absence of structural and temporal information prevents the model from understanding the contextual relationships of service behaviors, which weakens its ability to detect complex anomalies.

After introducing the Structure-Aware Dependency Modeling (SADM) module, the model shows clear improvements across multiple metrics. This module constructs dynamic invocation graphs and applies graph neural networks for high-order structural aggregation. It enables the model to perceive topological roles and interaction patterns among services. As a result, the model becomes more responsive to chained anomalies and structural disturbances. The gains in AUROC and accuracy metrics highlight the role of structural modeling in improving decision boundaries and robustness. In comparison, adding the Multi-Channel Temporal Representation (MCTR) module further strengthens the model's temporal modeling capability. It helps capture multi-scale behavioral fluctuations and enhances detection accuracy for burst anomalies and long-term drifts.

The full model, which integrates both SADM and MCTR modules, achieves the best performance. This confirms the strong complementarity between structural and temporal modeling in microservice anomaly detection. Structural modeling provides global dependency context and addresses the limitations of single-node behavior analysis. Temporal modeling captures local changes and trend shifts. Their combination allows the model to handle complex, multi-source, and multi-scale service anomalies with enhanced representation and generalization capabilities. This structure-temporal framework not only improves performance but also enhances adaptability and interpretability in real-world environments.

4.3.3 The impact of the number of channels on the representation capability of time series fusion

This paper further analyzes the impact of the number of channels on the temporal fusion representation capability. The experimental results are shown in Figure 4.

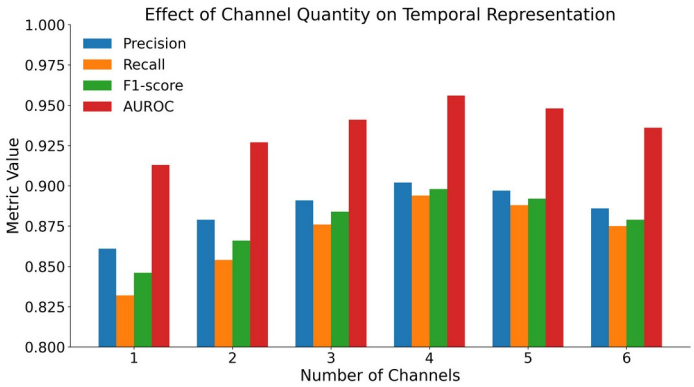


Figure 4. The impact of the number of channels on the representation capability of time series fusion

The experimental results show that as the number of channels increases from 1 to 4, the model consistently improves across all evaluation metrics. This indicates that a multi-channel structure enhances the model's capacity for temporal representation. Single-channel modeling can only capture behavioral changes at a single timescale. It often overlooks anomalous patterns that occur at different temporal granularities, which limits the model's ability to understand complex behavior dynamics. With multiple parallel channels, the model can extract node state evolution information from various perspectives, allowing for a more comprehensive representation of service behavior.

When the number of channels increases from 2 to 3 and 4, the improvements in Precision and AUROC are particularly notable. This reflects the strong contribution of mid-scale structures to anomaly detection. A moderate number of channels creates effective feature redundancy and differentiation across branches. Each branch can specialize in modeling short-term perturbations, trend shifts, or periodic fluctuations. This alleviates the limitations of single-channel representations in diversity and generalization. The fusion of multi-scale features enhances the model's sensitivity to asynchronous anomalies and low-amplitude signals, making detection results more robust.

However, when the number of channels increases to 5 and 6, the performance shows slight fluctuations or even minor declines. This suggests that excessive channels may introduce modeling redundancy and noise. Overlapping representation spaces across redundant channels can lead to gradient degradation in attention mechanisms, weakening the effectiveness of the fused representation. Moreover, with limited training resources, more channels reduce learning efficiency and make it harder for the model to capture critical temporal features in high-dimensional spaces. This indicates that expanding the number of channels does not always lead to better performance and requires a balance between representational capacity and structural complexity.

Overall, the number of channels directly affects the sufficiency of multi-scale information extraction and the compactness of temporal representations. It reflects the sensitivity and robustness of temporal modeling when handling heterogeneous microservice behaviors. Proper channel design not only improves the model's ability to detect cross-scale anomalies but also enhances its capacity to decouple temporal information during joint modeling of structure and behavior. This provides a more supportive temporal embedding space for the structure-temporal collaborative detection mechanism.

4.3.4 Effect of Dropout Rate on the Robustness of the Structure-Temporal Collaboration Model

This paper also gives the impact of the Dropout rate on the robustness of the structure-time collaboration model. The experimental results are shown in Figure 5.

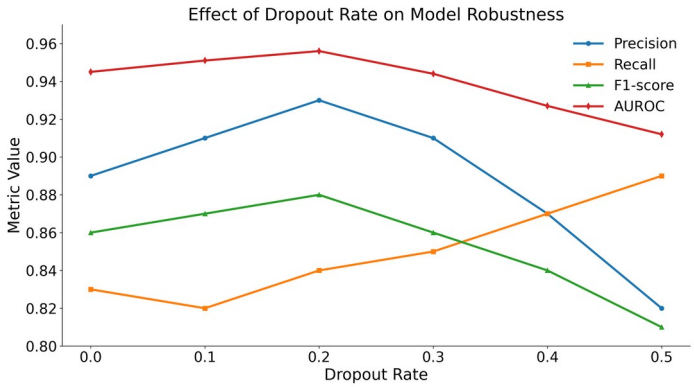


Figure 5. Effect of Dropout Rate on the Robustness of the Structure-Temporal Collaboration Model

This experiment systematically evaluates the stability and robustness of the structure-temporal joint model across multiple performance metrics from the perspective of dropout rates. Overall, Precision reaches its peak when the dropout rate increases from 0 to 0.2. This suggests that a moderate random deactivation mechanism helps prevent overfitting and enhances the generalization of local feature learning. However, when the dropout rate exceeds 0.3, performance drops significantly. This indicates that excessive information loss undermines the model's ability to capture joint structural and temporal representations.

For the Recall metric, the results show a continuous upward trend. This suggests that a higher dropout rate effectively improves the model's coverage of potential anomaly patterns. Under multi-channel time series input, the trend reflects the model's tendency to expand anomaly decision boundaries in high-dropout settings. However, the gain in Recall may come at the cost of reduced Precision, indicating a trade-off between detection coverage and false positive control.

The F1-score exhibits a more stable pattern with minor fluctuations. This shows that the model's overall detection performance is influenced by both structural coupling and temporal sequence distribution. Notably, around a dropout rate of 0.2, the F1-score reaches a relatively optimal value. This suggests a balanced state between structural modeling and temporal dependency perception.

In addition, the AUROC metric remains relatively stable under lower dropout rates and reaches its highest value near a dropout rate of 0.2. This further confirms the presence of an optimal robustness window between structural perturbation and temporal modeling. Combined with the proposed multi-channel modeling mechanism, the experiment highlights that dropout configuration has a significant impact on information channel fusion strategies in complex structures. It also provides guidance for tuning dropout rates in robust anomaly detection tasks.

4.3.5 Sensitivity analysis of conditional information dimension

This paper further investigates how the frequency at which the graph structure is updated influences the model's ability to capture dynamic structural changes within the system. By adjusting the update intervals, the study explores the sensitivity of the structural modeling process to temporal granularity in graph construction. This analysis aims to determine how effectively the model can reflect evolving inter-container relationships and structural shifts over time. The corresponding experimental results that illustrate this impact are provided in Figure 6.

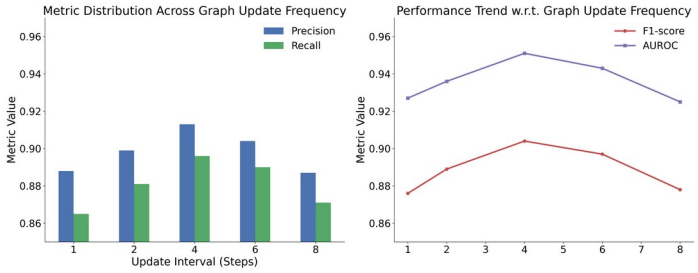


Figure 6. The impact of graph construction update frequency on structural dynamic modeling capabilities

In the experiment evaluating the impact of graph update frequency on model performance, we observed clear differences across various metrics under different update intervals. This indicates that the dynamic update strategy of graph structures plays a key role in modeling time-varying dependencies among containers. Notably, when the update frequency is set to every 4 steps, both Precision and Recall reach relatively optimal levels. This suggests that moderately frequent updates help capture fine-grained service interaction dynamics and avoid the lagging representations caused by early graph freezing.

Further analysis of the trends in F1-score and AUROC shows that setting the update frequency to 4 steps also leads to peak performance in terms of classification balance and overall discriminative ability. This finding highlights that a reasonable update interval enables the model to avoid overfitting to local fluctuations while better capturing long-term dependencies. If the update frequency is too low, such as every 8 steps, the graph structure fails to respond promptly to changes in service invocation relationships, resulting in delayed dependency modeling and reduced performance. On the other hand, overly frequent updates, such as at every step, may introduce structural noise.

In structural modeling tasks, the graph update frequency directly affects the timeliness of service dependencies and the stability of structural representations. The experimental results demonstrate that properly tuning the update rhythm enhances the model's ability to perceive dynamic changes in service topology. It also improves the precision of upstream and downstream causal representation in temporal modeling, leading to better overall anomaly detection.

From a practical deployment perspective, excessively frequent graph updates not only increase computational cost but may also weaken the global consistency of structural representations. Therefore, this experiment confirms that the update rhythm of graph structures is a critical hyperparameter. It plays an important role in ensuring robust model performance in complex container service environments and provides empirical support for deployment-level optimization.

5. Conclusion

This study addresses key challenges in anomaly detection for microservice architectures, including structural dynamics, behavioral heterogeneity, and sparsely labeled anomalies. A collaborative detection framework is proposed by integrating structure-aware dependency modeling and multi-channel temporal representation. At the structural level, a graph neural network is introduced to perform dynamic graph construction and high-order semantic encoding of service dependencies. At the behavioral level, multiple temporal paths are designed to capture behavior evolution across different time scales. These components are fused in a unified embedding space, enhancing the model's ability to detect chained anomaly propagation, service behavior drift, and cross-node anomaly diffusion. The proposed method is evaluated on several real-world microservice datasets, showing advantages in accuracy, robustness, and interpretability.

In terms of architecture, the Structure-Aware Dependency Modeling (SADM) module captures multi-level service dependencies through graph construction, neighbor aggregation, and graph-level embedding. This improves the model's capacity to represent structural disturbances and topological anomalies. The Multi-Channel Temporal Representation (MCTR) module builds parallel temporal modeling paths to enhance adaptability to behavioral changes and coverage of diverse anomaly patterns. The joint modeling strategy enables cross-fusion between structural semantics and behavioral dynamics. It effectively mitigates common challenges in microservice anomaly detection, such as anomaly masking, propagation chain loss, and behavior disguise. Ablation studies and sensitivity analyses further validate the critical contributions of both modules to overall detection performance.

This work provides technical support for real-time anomaly detection in intelligent operations under cloud-native architectures. It is particularly suitable for complex microservice systems deployed on container orchestration platforms, elastic scaling environments, and distributed service

governance infrastructures. The proposed structure-temporal collaborative modeling framework offers strong scalability and generalizability. It can be extended to tasks such as log behavior analysis, root cause localization, and multi-tenant security auditing. The key mechanisms within the framework also show potential for cross-task transfer and can be applied to structure-temporal scenarios in financial transaction analysis, intelligent manufacturing, and IoT behavior monitoring. This makes the method highly practical and suitable for industry adoption.

Future work can be developed along three directions. First, the graph construction module can be enhanced to better perceive dynamic structural changes by incorporating self-supervised structure evolution modeling to reduce manual dependency in design. Second, uncertainty estimation and confidence-aware learning mechanisms can be introduced to quantify risk and improve the trustworthiness of anomaly predictions. Third, integration with federated learning and multi-task learning can be explored to improve generalization and deployment across platforms and environments. As microservice systems continue to grow in scale and complexity, structure-behavior collaborative anomaly detection will play an increasingly critical role in ensuring system stability and business continuity.

References

[1] 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.

[2] Wang P, Zhang X, Cao Z. Anomaly detection for microservice system via augmented multimodal data and hybrid graph representations[J]. Information Fusion, 2025, 118: 103017.

[3] Khodabandeh, G., Ezaz, A., Babaei, M. and Ezzati-Jivan, N., "Utilizing graph neural networks for effective link prediction in microservice architectures", Proceedings of the 16th ACM/SPEC International Conference on Performance Engineering, pp. 19-30, 2025.

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

[5] Akmeemana L, Attanayake C, Faiz H, et al. GAL-MAD: Towards Explainable Anomaly Detection in Microservice Applications Using Graph Attention Networks[J]. arXiv preprint arXiv:2504.00058, 2025.

[6] Zhao H, Wang Y, Duan J, et al. Multivariate time-series anomaly detection via graph attention network[C]//2020 IEEE international conference on data mining (ICDM). IEEE, 2020: 841-850.

[7] Zhan J, Wu C, Yang C, et al. HFN: Heterogeneous feature network for multivariate time series anomaly detection[J]. Information Sciences, 2024, 670: 120626.

[8] C. Hua, N. Lyu, C. Wang and T. Yuan, "Deep Learning Framework for Change-Point Detection in Cloud-Native Kubernetes Node Metrics Using Transformer Architecture", 2025.

[9] Deng A, Hooi B. Graph neural network-based anomaly detection in multivariate time series[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(5): 4027-4035.

[10] Zhao K, Guo C, Cheng Y, et al. Multiple time series forecasting with dynamic graph modeling[J]. Proceedings of the VLDB Endowment, 2023, 17(4): 753-765.

[11] Miao Q, Xu C, Zhan J, et al. An unsupervised short-and long-term mask representation for multivariate time series anomaly detection[C]//International Conference on Neural Information Processing. Singapore: Springer Nature Singapore, 2022: 504-516.

[12] Wu X, Qiu X, Li Z, et al. Catch: Channel-aware multivariate time series anomaly detection via frequency patching[J]. arXiv preprint arXiv:2410.12261, 2024.

[13] Wang Q, Zhu Y, Sun Z, et al. A multi-scale patch mixer network for time series anomaly detection[J]. Engineering Applications of Artificial Intelligence, 2025, 140: 109687.

[14] Y. Wang, H. Liu, G. Yao, N. Long and Y. Kang, "Topology-Aware Graph Reinforcement Learning for Dynamic Routing in Cloud Networks", arXiv preprint arXiv:2509.04973, 2025.

[15] Li Z, Shi J, Van Leeuwen M. Graph neural networks based log anomaly detection and explanation[C]//Proceedings of the 2024 IEEE/ACM 46th international conference on software engineering: companion proceedings. 2024: 306-307.

[16] B. Chen, "FlashServe: Cost-Efficient Serverless Inference Scheduling for Large Language Models via Tiered Memory Management and Predictive Autoscaling", 2025.

[17] Osswald M, Schönenberger T, Cantali G, et al. Anomaly Detection in Microservices Architecture Using Graph Neural Networks[C]//2025 33rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP). IEEE, 2025: 560-567.

[18] Namoano B, Latsou C, Erkoyuncu J A. Multi-channel anomaly detection using graphical models[J]. Journal of Intelligent Manufacturing, 2024: 1-12.

[19] Z. Zhang, W. Liu, J. Tao, H. Zhu, S. Li and Y. Xiao, "Unsupervised Anomaly Detection in Cloud-Native Microservices via Cross-Service Temporal Contrastive Learning", 2025.

[20] C. Zhang, C. Shao, J. Jiang, Y. Ni and X. Sun, "Graph-Transformer Reconstruction Learning for Unsupervised Anomaly Detection in Dependency-Coupled Systems", 2025.

[21] Guan S, Zhao B, Dong Z, et al. GTAD: Graph and temporal neural network for multivariate time series anomaly detection[J]. Entropy, 2022, 24(6): 759.

[22] Tuli S, Casale G, Jennings N R. Tranad: Deep transformer networks for anomaly detection in multivariate time series data[J]. arXiv preprint arXiv:2201.07284, 2022.