**Journal of Computer Technology and Software**

**ISSN: 2998-2383**

**Vol. 3, No. 5, 2024**

# Deep Learning for Root Cause Detection in Distributed Systems with Structural Encoding and Multi-modal Attention

**Yaokun Ren**
Northeastern University, Seattle, USA
renyaokun0907@gmail.com

**Abstract:** This paper addresses key challenges in root cause identification within microservice architectures, focusing on limited structural modeling capabilities and insufficient integration of multi-dimensional features. A Transformer-based model for multi-dimensional fusion root cause identification is proposed. The method includes two core modules: Structure-Aware Trace Encoding (SATE) and Multi-dimensional Attention Fusion (MAF). The SATE module aggregates upstream and downstream dependency information for each node in the service trace. It enhances node representation through structural embedding, thereby preserving the topological dependency features between services. The Transformer module then models the entire structured sequence to capture long-range temporal dependencies within the trace. On this basis, the MAF module introduces separate attention channels from three dimensions: latency, status codes, and dependency paths. This further integrates multi-source monitoring data and improves the model's ability to represent anomaly propagation paths. A series of comparative and ablation experiments validate the effectiveness of the proposed method in terms of accuracy, robustness, and structural generalization. The model maintains high discriminative power even under service topology perturbations and increased trace complexity. These results demonstrate the strong practical potential of the proposed method in complex distributed systems.

**Keywords:** Microservice system; root cause identification; structure-aware coding; multi-dimensional attention mechanism

## 1. Introduction

In today's highly distributed system architecture, microservices have been widely adopted due to their flexibility, scalability, and modularity[1,2]. Compared to traditional monolithic applications, microservices divide applications into multiple independent service units that cooperate through remote procedure calls (RPC) or message queues. This architecture improves development efficiency and deployment flexibility. However, it also introduces unprecedented runtime complexity[3]. When failures occur, the intricate interdependencies between services can lead to fault propagation and difficulty in pinpointing responsibility. Especially under high-concurrency scenarios, the massive volume of call chain data makes manual analysis and traditional monitoring methods almost infeasible. Accurately locating the root cause of anomalies across multi-dimensional, cross-service, and cross-node call chains has become a critical challenge in microservice operations[4].

With the rise of AIOps, an increasing number of studies are exploring the use of artificial intelligence to assist in fault localization. In such tasks, call chain data, as behavioral sequences reflecting system runtime states, provides highly informative features for root cause analysis. A single call chain typically contains key indicators such as service invocation order, latency, and return status. These features are essential for observing the system's causal pathways. However, due to the strong temporal nature, variable length, and complex nested structure of call chains, traditional models relying on feature engineering often struggle to generalize. They fail to handle the diverse anomaly scenarios and dynamic service topologies effectively. Therefore, building an algorithmic framework capable of automatically extracting semantic features and modeling inter-service causal relationships from raw call chains is of significant value for root cause classification[5].

In recent years, the Transformer architecture has shown remarkable advantages in many complex structured data analysis tasks due to its powerful sequence modeling capability and self-attention mechanism. Compared to traditional recurrent neural networks, Transformers can effectively capture long-range dependencies and avoid information loss over long sequences. This makes them particularly suitable for inputs like call chains that contain hierarchical structures and complex dependencies. In microservice environments, the impact of anomalies varies across services, and the propagation paths of anomalies are nonlinear and context-dependent. These characteristics align well with the design of the Transformer. Applying Transformers to call chain modeling allows for deeper exploration of inter-service interaction semantics. It enables global analysis to identify which nodes or paths are likely fault sources, thereby improving the accuracy and interpretability of root cause identification[6].

In real-world scenarios, microservice systems generate billions of call chains daily. Anomalies often exhibit the "low-frequency but high-impact" pattern. Root cause services may be buried in massive volumes of normal calls as noise. Unlike passive alerts based solely on threshold metrics, Transformer-based root cause identification methods can actively extract key features from entire call sequences. They model semantic

relationships along causal chains to precisely detect service nodes responsible for global anomalies[7]. This approach enhances the intelligence level of fault localization. It also provides a practical and generalizable technical path for enterprises, helping to alleviate the operational bottlenecks of "slow detection, delayed localization, and late recovery." Especially in environments where service topologies frequently change and call paths evolve dynamically, this method demonstrates strong adaptability and robustness.

In summary, building a Transformer-based framework for call chain modeling and root cause identification represents a deep integration and extension of current AIOps technologies. It is also a natural response to the high complexity of modern microservice systems. This research not only poses theoretical challenges but also offers broad engineering application prospects. By extracting hidden temporal features and service dependency semantics from call chains, it is possible to construct a root cause analysis system that is more interpretable, real-time, and generalizable. This can provide a solid foundation for the stable operation and intelligent maintenance of large-scale distributed systems.

## 2. Related work

### 2.1 Microservice fault identification

As a mainstream architecture in modern cloud computing and enterprise application development, microservices aim to divide monolithic applications into a set of small, independently deployable and runnable services. Each service focuses on a specific business function[8,9]. This architectural style significantly enhances system modularity. It makes development, testing, deployment, and operations more flexible and efficient. Microservices are usually deployed in containers. They rely on infrastructure such as service discovery, load balancing, and configuration centers to achieve high availability and elastic scaling. Despite the performance and engineering efficiency benefits, microservices introduce new challenges in service interaction complexity, state consistency management, and distributed transaction control. These issues affect system stability and observability. In high-concurrency and complex business scenarios, the intricate service interactions can easily cause fault propagation and make diagnosis difficult[10,11].

As microservice systems scale up, traditional monitoring methods based on node logs or single metrics can no longer provide a comprehensive view of system health. To improve observability, many microservice systems have introduced distributed tracing mechanisms. These mechanisms inject unique identifiers into service calls to record the complete invocation path[12]. Call chain data includes not only the order and hierarchy of service calls, but also key runtime metrics such as latency, return status, and error information. This makes it a vital source for observing system behavior[13]. However, such data is typically complex in structure, high in dimensionality, and strongly time-dependent with contextual relevance. Extracting meaningful signals from massive and heterogeneous call chains to support fault localization has become a key research direction in microservice operations[14].

Against this backdrop, research on intelligent operations for microservices is evolving rapidly. Multiple technical approaches have been proposed for root cause analysis, anomaly detection, and behavior modeling[15]. Some methods use graph structures to model service dependencies, or construct fault propagation paths through rule engines. Others apply clustering and pattern recognition to categorize anomalous behaviors. However, these traditional methods often depend heavily on feature engineering and lack generalization ability. They struggle to cope with real-world conditions such as frequent service changes, diverse anomaly patterns, and unstable call chain structures. There is an urgent need for an algorithmic framework with strong sequence modeling and global dependency capture capabilities. Such a framework can better understand the behavioral semantics within call chains and advance the intelligent development of fault diagnosis in microservice systems[16].

### 2.2 Transformer

The Transformer architecture was originally proposed for natural language processing. It achieved significant breakthroughs in handling sequential data through its unique self-attention mechanism[17]. Unlike traditional recurrent or convolutional neural networks, the Transformer does not rely on the temporal order of sequences to transmit information. Instead, it uses a global self-attention mechanism to establish direct connections between any positions in the input sequence. This design improves the model's ability to capture long-range dependencies. It also greatly accelerates training and enhances parallel computation efficiency[18]. As the architecture has evolved, Transformer models have expanded from natural language processing to fields such as computer vision, graph learning, and time series analysis. They have set new performance benchmarks in many tasks and become one of the most widely used architectures in deep learning research[19].

In scenarios such as intelligent system operations and behavior modeling, the Transformer shows distinct advantages. Its multi-head attention mechanism captures potential dependencies between different positions in a sequence across multiple dimensions. This helps reveal contextual relationships in complex system behavior. For example, in microservice call chains, the anomaly at a service node may not be caused by the node itself[20]. It may result from the combined influence of multiple upstream services. Such complex dependencies are often difficult for traditional models to capture. The Transformer, without relying on fixed windows or structural assumptions, can model the dynamic evolution of service call chains from a global perspective. This provides strong modeling power for anomaly path identification and root cause localization. Its scalability and ability to handle variable-length input sequences make it naturally suited for processing call chains, which are often long, hierarchical, and structurally diverse.

In recent years, with the increasing modularization of model design, the Transformer's application potential in industrial intelligent operations has gained attention. By incorporating components such as positional encoding, residual connections, and feed-forward networks, the Transformer can process complex and sparse input features more stably. It also

demonstrates good generalization and interpretability. In the context of microservice call chains, the behavioral features of service nodes and their contextual semantics can be embedded as inputs. This guides the model in learning the patterns of anomaly propagation in the call paths. It provides a theoretical foundation for building high-accuracy models for root cause identification. It also supports data representation learning needed for interpretability, security, and self-healing capabilities in future systems. Therefore, applying the Transformer to root cause identification in microservice call chains is a meaningful extension of deep sequence modeling techniques. It also offers new approaches and methods for intelligent system diagnostics[21].

## 3. Method

This study proposes a Transformer-based model for root cause identification in microservice call chains. It aims to address the limitations of existing methods in modeling complex service dependencies for anomaly localization. The proposed method introduces two key innovations. First, a Structure-Aware Trace Encoding (SATE) mechanism is designed. It integrates service hierarchy and invocation order information. This enhances the model's ability to perceive structural semantics in the call chain. Second, a Multi-dimensional Attention Fusion (MAF) module is introduced. It models multiple feature channels in parallel, including latency, error code distribution, and upstream-downstream dependencies. This enables the model to capture anomaly propagation patterns from multiple perspectives. As a result, the robustness and generalization ability of root cause identification are improved. The method operates within an end-to-end framework. It does not rely on handcrafted rules or static configurations. It offers strong scalability and adaptability, making it suitable for dynamic and evolving microservice environments. The architecture of the overall model is illustrated in Figure 1.

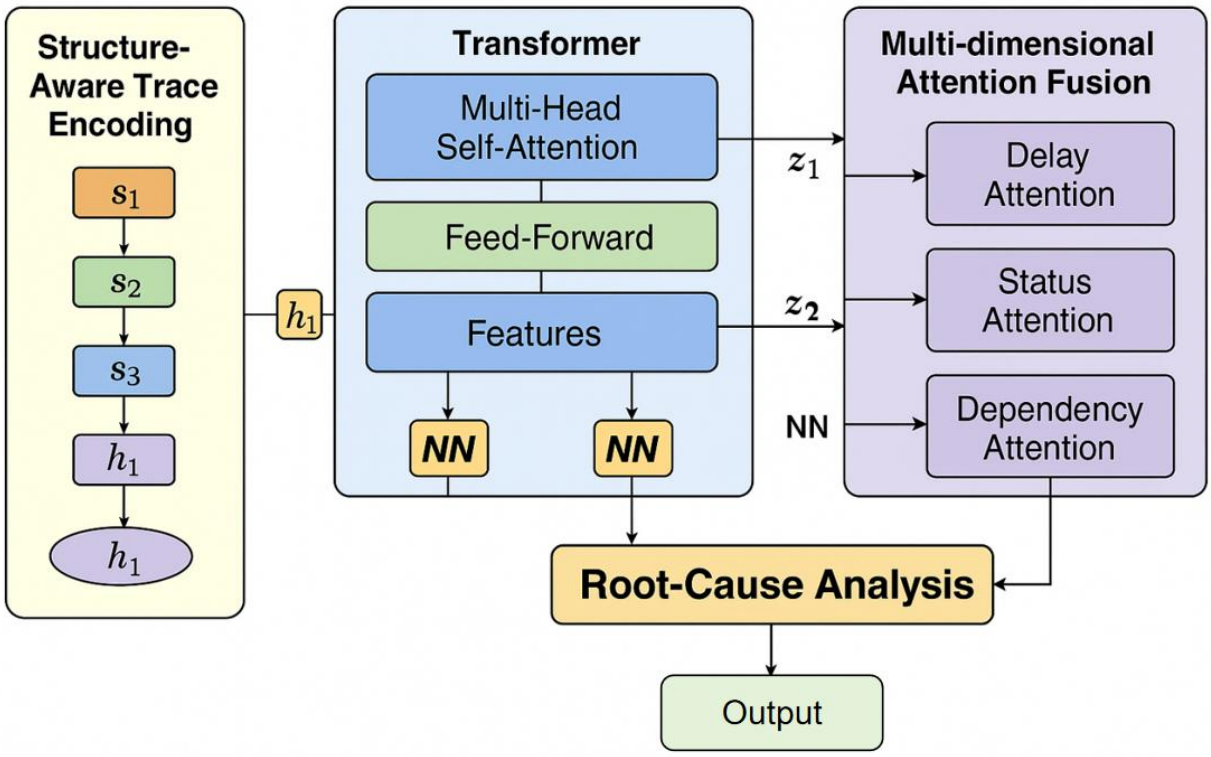


**Figure 1.** Overall model architecture diagram

### 3.1 Structure-Aware Trace Encoding

In a microservice system, the call chain typically takes the form of a directed acyclic graph (DAG), which reflects both the invocation path and the dependency relationships among services. To effectively encode the structural characteristics and semantic information embedded in this graph, we developed a structure-aware call chain encoding mechanism. This mechanism is designed to transform the original call chain into a high-dimensional vector representation that retains the topological and contextual dependencies between services. The encoded representation is optimized for input into the Transformer model, enabling the downstream architecture to capture long-range interactions and complex relationships within the service graph. The overall architecture of this encoding module is illustrated in Figure 2.

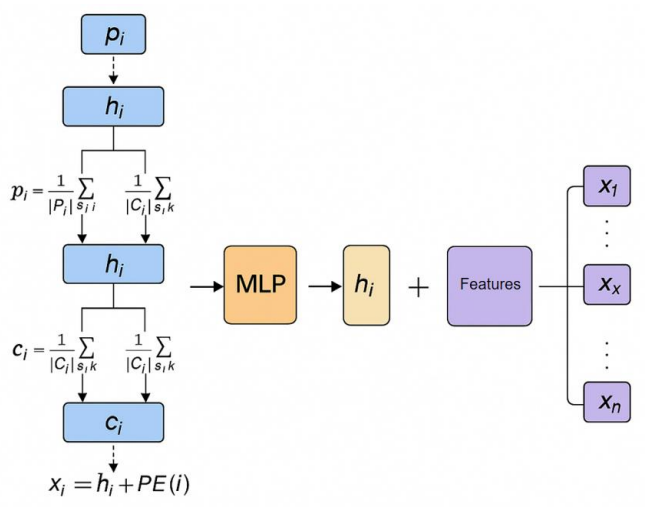


**Figure 2.** SATE module architecture

Assume that a call chain consists of several service nodes, denoted as $T = \{s_1, s_2, ..., s_n\}$ , where each node $s_i$ corresponds to a service instance and its context call information. We encode the original features of each node into a vector $h_i \in R^d$ , which consists of the following types of information: embedded representation of the service identifier, the node's response time, status code, exception mark and other multi-dimensional operation indicators.

To enhance the ability to model structural dependencies between nodes, we introduce an upstream and downstream structure-aware mechanism to encode the service call hierarchy information into the node representation. Specifically, we aggregate information about the direct upstream service set $P(s_i)$ and downstream service set $C(s_i)$ of each node $s_i$ . The aggregation function is defined as follows:

$$P_i = \frac{1}{|P(S_i)|}\sum_{S_j \in P(S_i)} h_j \ , \quad c_i = \frac{1}{|C(S_i)|}\sum_{S_k \in C(S_i)} h_k$$

Next, we concatenate the upstream and downstream representations with the original node representation as structure-aware input:

$$h'_i = MLP(h_i \,//\, p_i \,//\, c_i)$$

$//$ represents the vector concatenation operation, and MLP represents the multi-layer perceptron, which is used to perform dimension regularization and nonlinear transformation on the concatenated representation. This structure enables the representation of each node to contain both its own running status information and explicitly encode its contextual dependency in the entire call graph, thereby improving the model's ability to model abnormal propagation paths.

In addition, to adapt to the input requirements of Transformer, we perform position embedding and path normalization on the encoding of all nodes. We introduce the position encoding function $PE(i)$ to convert the relative position of the service in the call chain into a continuous vector. The resulting sequence input is represented as:

$$x_i = h'_i + PE(i)$$

All node vectors form the input sequence $X = \{x_1, x_2, ..., x_n\}$ in the order of calling, which serves as the input of the subsequent Transformer module. This structure-aware encoding method ensures that the model can not only capture the individual characteristics of the service nodes, but also model the causal structure and calling semantics between nodes, providing stronger representation capabilities for root cause identification.

### 3.2 Multi-dimensional Attention Fusion

In order to further improve the modeling capability of the root cause identification task for multi-dimensional operating states, we designed a multi-dimensional attention fusion module (MAF) to extract key semantic information from multiple perspectives such as delay features, state code distribution, and service dependency structure based on the Transformer encoding results. Its module architecture is shown in Figure 3.

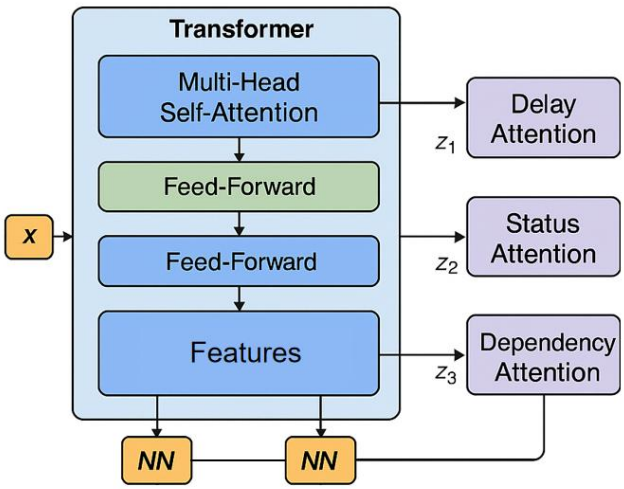


**Figure 3.** MAF module architecture

Suppose the node representation sequence obtained after Transformer encoding is $\{z_1, z_2, ..., z_n\}$ . We will define an independent attention head for each attention dimension and weightedly integrate information from different dimensions to enhance the model's sensitivity and interpretability to abnormal features.

For the delay feature, we designed an attention mechanism based on delay differences to capture the propagation pattern of service response time in the call chain. The delay attention scoring function is defined as:

$$a_{i,j}^{(delay)} = \frac{\exp(-|d_i - d_j|)}{\sum_k \exp(-|d_i - d_k|)}$$

Where $d_i$ represents the response time of service node i, and $\alpha_{i,j}^{(delay)}$ represents the delay correlation of node j to node i. The context after combining the attention weight is expressed as:

$$c_i^{(delay)} = \sum_j \alpha_{i,j}^{(delay)} \cdot z_j$$

The state code attention is used to highlight the state information that indicates abnormal behavior. The state code embedding is defined as $s_i$ , and the state-related attention weight is calculated as:

$$\alpha_{i,j}^{(status)} = \frac{\exp(z_j^T W_s s_j)}{\sum_k \exp(z_j^T W_s s_k)}$$

The fusion is represented as $c_j^{(status)}$ , where $W_s$ is the learnable projection matrix.

In the dimension of structural dependency, we introduce the adjacency relationship $N(i)$ of the service call graph to perform structural attention modeling, so that the model can perceive the propagation path of the node in the topology. Its attention aggregation form is:

$$c_i^{(dep)} = \sum_{j \in N(i)} \beta_{i,j} \cdot z_j$$

Where $\beta_{i,j}$ is the attention weight calculated based on the structural association between nodes, for example, through a GNN-inspired approach or structural position encoding.

Finally, we fuse the context vectors of the three dimensions to generate a multi-view representation for subsequent discrimination decisions. The fusion method can be expressed as:

$$u_i = MLP(c_i^{(delay)} // c_i^{(status)} // c_i^{(dep)})$$

This mechanism explicitly encodes multi-dimensional system signals into the model's attention space, effectively improving the fine-grained perception capability of the representation and providing richer decision-making basis for the identification of root cause service nodes.

## 4. Experimental Results

### 4.1 Dataset

This study uses the Alibaba Trace Open Dataset as the primary data source. The dataset is collected from a real large-scale e-commerce microservice system. It features a highly complex service call structure and rich anomaly samples. It contains billions of trace records and covers the runtime behavior of hundreds of microservice components. It includes multi-dimensional indicators such as call latency, status codes, service identifiers, and service dependencies. These features reflect the operational characteristics and potential anomaly patterns of microservice systems in production environments.

In this dataset, each trace is stored in JSON format. It includes the call sequence between services, timestamps, latency, return codes, and anomaly indicators. The complete service topology is also preserved. The data has an obvious hierarchical structure and strong temporal characteristics. It is an ideal source for structure modeling and root cause analysis tasks. In addition, the dataset labels part of the abnormal traces. This facilitates supervised learning and model validation.

The Alibaba Trace Open Dataset is collected from an industrial-grade microservice system. It has typical features such as high concurrency, large scale, and multi-source heterogeneity. It is well-suited for research on intelligent operations in large-scale systems. The real call chain topology and detailed runtime records provide sufficient training data and validation resources for the structure-aware modeling and multi-dimensional attention mechanisms proposed in this study.

### 4.2 Experimental setup

The experiments in this study were conducted on an offline platform that simulates a real microservice runtime environment. Training and testing samples were constructed based on trace logs from the Alibaba Trace Open Dataset. To validate the effectiveness of the model, we performed supervised partitioning of the trace data according to service anomaly labels. All data were normalized and structurally encoded in a unified manner.

The model was trained using the Adam optimizer. The learning rate was determined through grid search. An early stopping mechanism was applied on the validation set to prevent overfitting. All experiments were carried out on a server equipped with a 32GB GPU. This ensured efficient processing of large-scale trace samples and stable model training. Its detailed configuration is shown in Table 1.

**Table 1:** Specific parameter diagram

| Parameter name | Setting Value |
|---|---|
| Input Dimensions | 128 |
| Number of Transformer Layers | 4 |
| Number of attention heads | 8 |
| Learning Rate | 0.001 |
| Batch size | 64 |
| Number of training epoch | 50 |
| Optimizer | AdamW |
| Hardware Environment | NVIDIA RTX 3090, 32GB GPU |

### 4.3 Experimental Results

*1) Comparative experimental results*

This paper first gives the comparative experimental results, as shown in Table 2.

**Table2:** Comparative experimental results

| Method | Accuracy | AUC | F1-Score |
|---|---|---|---|
| LogAnomaly[22] | 82.1 | 75.3 | 78.5 |
| GTrace[23] | 86.4 | 79.6 | 82.8 |
| MicroRCA[24] | 88.7 | 83.2 | 85.4 |

| | | | |
|---|---|---|---|
| Tadl[25] | 90.5 | 85.9 | 87.1 |
| Ours | 93.6 | 89.4 | 91.0 |

As shown in Table 2, the model proposed in this paper achieves significantly better performance than existing methods in the root cause identification task for microservice systems. Compared to traditional log sequence modeling methods such as LogAnomaly, which lack inherent capabilities for structural modeling and semantic anomaly detection, performance is limited. Its Accuracy, AUC, and F1-score reach only 82.1 percent, 75.3 percent, and 78.5 percent, respectively. This indicates that it struggles to handle the complex service dependencies and cross-node propagation paths in trace data, often resulting in false positives and missed detections.

GTrace and MicroRCA introduce modeling of service topology and show improved performance, especially in AUC and F1-score, which reach 79.6 percent and 83.2 percent, and 82.8 percent and 85.4 percent, respectively. These results highlight the importance of structural information in root cause identification. However, these methods still rely primarily on static service graphs or rule-based approaches. They lack the ability to integrate dynamic invocation behavior and multi-dimensional monitoring signals, which leads to unstable performance under complex call chains. In particular, their identification accuracy is limited when dealing with cross-level anomaly propagation.

Tadl, a recent Transformer-based method, incorporates deep feature modeling and achieves more stable results across all metrics. Its F1-score improves to 87.1 percent. However, its modeling approach focuses on global feature aggregation and lacks fine-grained modeling of structural hierarchy and key semantics along anomaly paths between services. As a result, its robustness and interpretability remain limited in scenarios involving service heterogeneity or topological perturbations.

In contrast, the model proposed in this paper introduces Structure-Aware Trace Encoding (SATE) and Multi-dimensional Attention Fusion (MAF) mechanisms. These components allow simultaneous capture of topological dependencies, anomaly propagation paths, and multi-dimensional metric signals in the call chain. The model achieves unified modeling from both structural and semantic perspectives. It maintains strong discriminative power in complex microservice environments and ultimately reaches 93.6 percent Accuracy, 89.4 percent AUC, and 91.0 percent F1-score. This demonstrates superior adaptability, stability, and practical potential for real-world deployment.

*2) Ablation Experiment Results*

This paper also further gives the results of the ablation experiment, and the experimental results are shown in Table 3.

**Table 3:** Ablation Experiment Results

| Method | Accuracy | AUC | F1-Score |
|---|---|---|---|
| BaseLine | 87.2 | 82.5 | 84.1 |
| +SATE | 90.1 | 85.9 | 87.3 |
| +MAF | 91.4 | 87.2 | 88.5 |
| Ours | 93.6 | 89.4 | 91.0 |

As shown in the ablation results in Table 3, the two core modules in the proposed model, SATE (Structure-Aware Trace Encoding) and MAF (Multi-dimensional Attention Fusion), both play critical roles in improving root cause identification performance. Although the baseline model already uses a Transformer to model trace data, it lacks the ability to represent structural and multi-dimensional semantic information. As a result, its performance on Accuracy, AUC, and F1-score is significantly lower, reaching only 87.2 percent, 82.5 percent, and 84.1 percent, respectively.

After integrating the SATE module into the baseline, the model shows a clear performance improvement. Accuracy rises to 90.1 percent, indicating that the structure-aware encoding mechanism effectively enhances the model's understanding of service dependency context. By introducing topological and upstream-downstream information between services, SATE enables the model to capture potential causal relationships along the anomaly propagation path. This leads to more precise localization of the root cause. Meanwhile, AUC and F1-score improve to 85.9 percent and 87.3 percent, respectively, further validating the positive impact of structural modeling on classification performance.

When the MAF module is added to the baseline, improvements in AUC and F1-score become more pronounced. These metrics reach 87.2 percent and 88.5 percent, respectively. This suggests that the multi-dimensional attention mechanism can effectively focus on key feature channels during anomaly propagation, such as latency fluctuations, status code changes, and structural dependencies. By modeling from multiple perspectives, MAF strengthens anomaly signal representations and enhances the model's ability to detect real anomaly paths. Compared to single-view or static approaches, MAF significantly improves the model's robustness in handling anomalous samples.

Finally, when both modules are incorporated, the full model (Ours) achieves the best results across all metrics. Accuracy reaches 93.6 percent, showing that the model can accurately locate the root cause services responsible for global anomalies in complex microservice traces. These results confirm the synergistic effect of structural modeling and multi-dimensional semantic fusion in root cause analysis. They also demonstrate the strong adaptability and superior effectiveness of our method in real-world scenarios.

*3) Hyperparameter sensitivity experiments*

Furthermore, this paper gives the experimental results of hyperparameter sensitivity. First, the experimental results of learning rate are given, as shown in Table 4.

**Table 4:** Hyperparameter sensitivity experiment results (learning rate)

| Learning Rate | Accuracy | AUC | F1-Score |
|---|---|---|---|
| 0.004 | 88.2 | 83.1 | 85.0 |
| 0.003 | 90.7 | 86.0 | 87.9 |
| 0.002 | 92.4 | 88.1 | 89.6 |
| 0.001 | 93.6 | 89.4 | 91.0 |

As shown in the learning rate sensitivity results in Table 4, the choice of learning rate has a significant impact on model performance. Under a larger learning rate, such as 0.004, the model performs poorly in Accuracy, AUC, and F1-score, reaching only 88.2 percent, 83.1 percent, and 85.0 percent, respectively. This suggests that a larger step size may cause large oscillations during optimization. As a result, the model struggles to converge stably and fails to learn the structural and anomaly features effectively from the trace data.

As the learning rate decreases gradually, the model shows a consistent improvement in performance. When the learning rate is set to 0.003, Accuracy increases to 90.7 percent. This indicates that the model can better fit high-dimensional features from the structure-aware encoding and multi-dimensional attention mechanisms under the current optimization state. In root cause identification tasks, which are highly sensitive to recall, training stability is crucial for capturing anomaly propagation paths. A proper learning rate can help the model more effectively extract key node representations from complex traces.

When the learning rate is further reduced to 0.002, all evaluation metrics continue to improve. AUC reaches 88.1 percent, indicating enhanced discriminative ability between positive and negative samples. At this point, the Transformer encoder and the multi-dimensional attention module can better learn structural dependencies and status signals among services. This leads to finer-grained identification of abnormal behaviors. The gradual refinement of the learning rate also helps the model maintain focus on critical propagation paths and reduces the risk of misidentifying non-root cause nodes.

At a learning rate of 0.001, the model achieves its best performance, with the F1-score reaching 91.0 percent. This shows that in highly complex tasks like root cause identification, lowering the learning rate supports deeper structures in modeling the semantic behaviors of microservices more effectively. A well-chosen learning rate not only accelerates model convergence but also improves the responsiveness of the multi-dimensional attention fusion module to anomaly signals. This provides stronger support for system-level root cause localization.

Furthermore, the experimental results of different optimizers are given, as shown in Table 5.

**Table 5:** Hyperparameter sensitivity experiment results (Optimizer)

| **Optimizer** | **Accuracy** | **AUC** | **F1-Score** |
|---|---|---|---|
| AdaGrad | 88.9 | 83.8 | 85.7 |
| SGD | 90.3 | 85.2 | 87.1 |
| Adam | 92.1 | 87.5 | 89.0 |
| AdamW | 93.6 | 89.4 | 91.0 |

As shown in the optimizer sensitivity results in Table 5, the choice of optimizer has a significant impact on model performance in the root cause identification task. When using AdaGrad, the model performs relatively poorly across all three key metrics. Accuracy reaches 88.9 percent, AUC is 83.8 percent, and F1-score is 85.7 percent. This is likely due to AdaGrad's aggressive penalty on accumulated gradients during training. It often causes the learning rate to decay too quickly, making it difficult to fully train complex structure-aware models. As a result, the model struggles to capture fine-grained, multi-dimensional features in microservice traces.

In contrast, SGD, as a classical first-order optimizer, shows improvement in both accuracy and discriminative ability. It achieves an Accuracy of 90.3 percent. However, due to its reliance on single-step gradients and lack of adaptive mechanisms, it can easily get stuck in local minima, especially in high-dimensional models involving multi-head attention and structure fusion. For anomaly propagation modeling, the instability of SGD may reduce generalization on anomalous samples and lead to a higher miss rate.

When switching to the Adam optimizer, the model achieves notable gains across all metrics. AUC improves to 87.5 percent, and F1-score reaches 89.0 percent. Adam's adaptive learning rate mechanism and first- and second-order moment estimation provide better optimization capability for heterogeneous and sparse trace data. This improves convergence on features along abnormal propagation paths. It also allows faster learning of key weights in the multi-dimensional attention mechanism, resulting in more stable performance in multi-perspective information fusion.

Finally, the model achieves the best performance when using the AdamW optimizer. The F1-score reaches 91.0 percent. AdamW retains the benefits of Adam while introducing a weight decay regularization mechanism. This helps suppress overfitting and improves the stability of parameter learning. In root cause identification tasks that require modeling of complex structural dependencies and abnormal behavior propagation, AdamW better balances the optimization of both structural modeling and semantic aggregation. This leads to higher accuracy and greater robustness.

*4) Evaluation of the generalization ability of the model under different topological complexities*

This paper also gives an evaluation of the generalization ability of the model under different topological complexities, and the experimental results are shown in Figure 4.

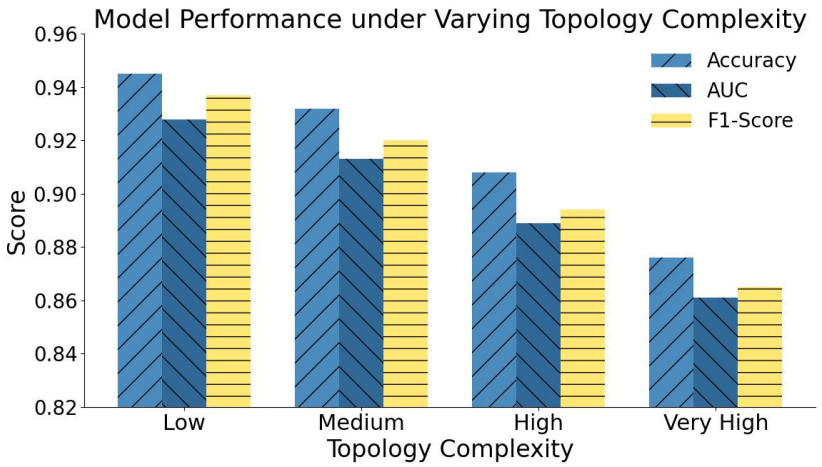


**Figure 4.** The generalization ability of the model under different topological complexities

As shown in Figure 4, the model exhibits significant performance differences under varying levels of topological complexity. As the structure of the trace becomes more complex, overall model performance shows a downward trend. Specifically, as complexity increases from Low to Very High, Accuracy drops from 94.5 percent to 87.6 percent. This indicates that in environments with deeply nested structures or highly coupled service dependencies, the root cause identification task becomes more challenging. The results suggest that extracting structural information and modeling semantics becomes increasingly difficult under such conditions.

The declining trend in the AUC curve further confirms that the model's discriminative ability is affected by high topological complexity. In particular, AUC decreases from 92.8 percent to 86.1 percent. This implies a reduced ability to distinguish root cause nodes from non-root nodes. In complex structures, anomaly propagation paths are more dispersed and the anomaly signals are less obvious. To maintain high discriminative performance, the model needs stronger capabilities for capturing global dependencies and integrating multi-dimensional anomaly information.

The trend of the F1-score, which considers both precision and recall, further supports these findings. At the "Very High" complexity level, the F1-score drops to 86.5 percent. This reflects increased false negatives and false positives in high-complexity topologies. It highlights the importance of the multi-dimensional attention mechanism in handling irregular anomaly propagation and structural heterogeneity. It also suggests that there is still room for improvement when facing extremely complex environments. Overall, the experimental results effectively validate the model's generalization ability under different structural complexities. They demonstrate that the proposed structure-aware encoding and multi-dimensional attention fusion mechanisms offer strong stability and robustness. Even in real-world microservice scenarios with dynamic topologies and highly coupled service interactions, the method maintains strong performance. This shows promising engineering adaptability and valuable potential for further research.

*5) Analysis of the impact of service dependency disturbance on discrimination results*

This paper further gives an analysis of the impact of service dependency disturbance on the discrimination results, and the experimental results are shown in Figure 5.

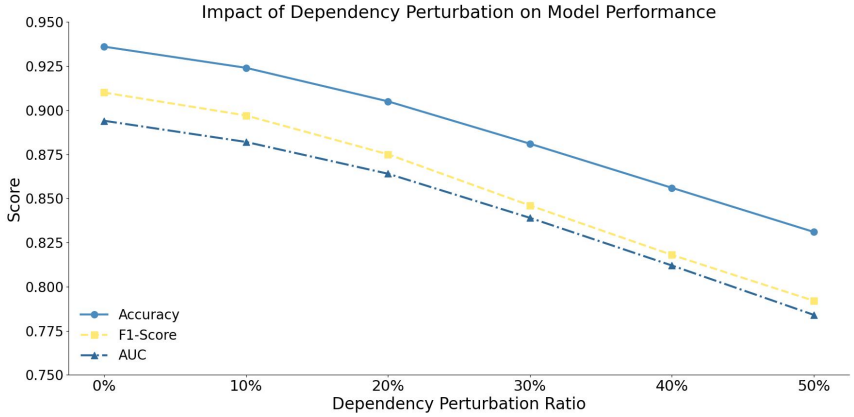


**Figure 5.** Analysis of the impact of service dependency disturbance on discrimination results

As shown in Figure 5, model performance in the root cause identification task consistently declines as the level of service dependency perturbation increases. This indicates that the model is highly sensitive to the completeness of structural information. When the perturbation ratio is 0 percent, the model achieves high performance across all metrics. Accuracy reaches 93.6 percent, F1-score is 91.0 percent, and AUC is 89.4 percent. These results suggest that with a complete dependency structure, the model can effectively capture causal relationships between services and accurately identify root causes.

However, when the perturbation ratio increases to 20 percent or more, model performance begins to degrade significantly. Accuracy drops to 90.5 percent, while F1-score and AUC decrease to 87.5 percent and 86.4 percent, respectively. This trend shows that missing or incorrect dependency links interfere with the model's ability to perform semantic modeling of the trace. In particular, it weakens the model's capacity to reconstruct anomaly propagation paths. The disturbance of dependency information disrupts the actual interaction graph between services, making it difficult for the model to correctly identify which nodes are the true sources of system-wide anomalies.

As the perturbation ratio reaches 50 percent, the model's performance declines more rapidly. The F1-score falls to 79.2 percent. This result reveals that in highly disrupted structural environments, even models equipped with structure-aware encoding and multi-dimensional attention mechanisms are severely impacted. It becomes especially difficult to identify cross-level anomaly paths or long anomaly propagation chains. This demonstrates that structural completeness plays a fundamental role in root cause analysis. Perturbations weaken the model's ability to capture fine-grained anomaly features. In summary, this experiment clearly validates the critical role of service dependency structures in supporting root cause identification models. The proposed structure-aware mechanism performs well under relatively complete dependency conditions. However, it also highlights the need for high-quality dependency tracing mechanisms in real-world deployments. Ensuring the accuracy and completeness of trace data is essential for reliable anomaly diagnosis and effective system self-healing.

## 5. Conclusion

This paper addresses the challenges of root cause identification in microservice systems and proposes a Transformer-based model that integrates structure-aware encoding with a multi-dimensional attention mechanism. The method uses trace data as the core modeling target and fully considers structural dependencies among microservices along with runtime multi-dimensional metrics. This significantly enhances the model's ability to detect and locate anomalies in complex system environments. Through the Structure-Aware Trace Encoding (SATE) module, the model incorporates topological context information of service nodes. The Multi-dimensional Attention Fusion (MAF) module further strengthens the recognition of anomaly propagation paths by modeling from multiple perspectives, including latency, status codes, and dependency structures. Together, these components provide detailed and comprehensive feature support for the root cause localization task.

Experimental results confirm the effectiveness of the proposed method across multiple evaluation metrics. The model demonstrates strong performance in overall accuracy, robustness, and generalization, especially under conditions involving topological disturbances or complex dependency structures. These results show that deep integration of structural semantics and multi-dimensional monitoring data can significantly improve the automation and responsiveness of intelligent operations systems. Compared to traditional rule-based or shallow feature extraction methods, the proposed approach achieves notable advances in depth, flexibility, and adaptability in root cause analysis.

This study not only provides technical support for intelligent operations in microservice systems but also offers methodological insights for anomaly detection and self-healing design in broader distributed systems. In practical industrial environments, service dependencies are often large-scale, and system conditions change dynamically. This places higher demands on the model's structural understanding and multimodal data integration capabilities. The model framework presented in this work offers a unified and high-precision solution for root cause tracing and fault identification under high-complexity conditions. It holds practical value for large-scale online systems, edge computing platforms, and cloud-native architectures.

## 6. Future work

Looking ahead, several directions remain worthy of further exploration. First, improving inference efficiency and model lightweighting without sacrificing modeling power is critical for real-world deployment. Second, given that trace data may be partially missing or sampled incompletely in real systems, developing more robust hybrid models that combine graph and sequence learning is a promising direction. In addition, future work can explore integrating this approach with emerging techniques such as federated learning and self-supervised representation learning. This would enhance the model's transferability and adaptability while preserving data privacy, enabling better support for intelligent system operations and automated fault diagnosis in key application scenarios.

## References


[1] Silva, Francisco, et al. "Towards a fault taxonomy for microservices-based applications." Proceedings of the XXXVI Brazilian Symposium on Software Engineering. 2022.

[2] Zhou, Xiang, et al. "Latent error prediction and fault localization for microservice applications by learning from system trace logs." Proceedings of the 2019 27th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering. 2019.

[3] Ikram, Azam, et al. "Root cause analysis of failures in microservices through causal discovery." Advances in Neural Information Processing Systems 35 (2022): 31158-31170.

[4] Wang, Tao, et al. "Workflow-aware automatic fault diagnosis for microservice-based applications with statistics." IEEE Transactions on Network and Service Management 17.4 (2020): 2350-2363.

[5] Guan, Shuai-Peng, et al. "Research on Fault Detection for Microservices Based on Log Information and Social Network Mechanism Using

BiLSTM-DCNN Model." International Journal of Computational Intelligence and Applications (2023): 2342002.

[6] Zhang, Shenglin, et al. "Fault Diagnosis for Test Alarms in Microservices through Multi-source Data." Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering. 2024.

[7] Jing, Ning, Han Li, and Zhuofeng Zhao. "A microservice fault identification method based on LightGBM." 2022 IEEE 8th International Conference on Cloud Computing and Intelligent Systems (CCIS). IEEE, 2022.

[8] Zhang, Qixun, et al. "Fault localization for microservice applications with system logs and monitoring metrics." 2022 7th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA). IEEE, 2022.

[9] Li, Shanshan, et al. "Understanding and addressing quality attributes of microservices architecture: A Systematic literature review." Information and software technology 131 (2021): 106449.

[10] Flora, José, et al. "A study on the aging and fault tolerance of microservices in kubernetes." IEEE Access 10 (2022): 132786-132799.

[11] Xin, R., Chen, P., & Zhao, Z. (2023). Causalrca: Causal inference based precise fine-grained root cause localization for microservice applications. Journal of Systems and Software, 203, 111724.

[12] Rios, Jesus, Saurabh Jha, and Laura Shwartz. "Localizing and explaining faults in microservices using distributed tracing." 2022 IEEE 15th International Conference on Cloud Computing (CLOUD). IEEE, 2022.

[13] Li, Xi, et al. "An effective parallel convolutional anomaly multi-classification model for fault diagnosis in microservice system." Software Quality Journal 32.3 (2024): 921-938.

[14] Power, Alexander, and Gerald Kotonya. "A microservices architecture for reactive and proactive fault tolerance in iot systems." 2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM). IEEE, 2018.

[15] Li, Zeyan, et al. "Practical root cause localization for microservice systems via trace analysis." 2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS). IEEE, 2021.

[16] Ma, Meng, et al. "Servicerank: Root cause identification of anomaly in large-scale microservice architectures." IEEE Transactions on Dependable and Secure Computing 19.5 (2021): 3087-3100.

[17] Cai, Yang, et al. "Modelcoder: A fault model based automatic root cause localization framework for microservice systems." 2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS). IEEE, 2021.

[18] Ikram, A., Chakraborty, S., Mitra, S., Saini, S., Bagchi, S., & Kocaoglu, M. (2022). Root cause analysis of failures in microservices through causal discovery. Advances in Neural Information Processing Systems, 35, 31158-31170.

[19] Zheng, W., Zhang, G., Zhao, C., & Zhu, Q. (2024). Multichannel consecutive data cross-extraction with 1DCNN-attention for diagnosis of power transformer. International Journal of Electrical Power & Energy Systems, 158, 109951.

[20] Liu, D., He, C., Peng, X., Lin, F., Zhang, C., Gong, S., ... & Wu, Z. (2021, May). Microhecl: High-efficient root cause localization in large-scale microservice systems. In 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP) (pp. 338-347). IEEE.

[21] Zhang, Shenglin, et al. "Failure diagnosis in microservice systems: A comprehensive survey and analysis." ACM Transactions on Software Engineering and Methodology (2024).

[22] Meng, Weibin, et al. "Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs." IJCAI. Vol. 19. No. 7. 2019.

[23] Xie, Zhe, et al. "From point-wise to group-wise: A fast and accurate microservice trace anomaly detection approach." Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2023.

[24] Wu, Li, et al. "Microrca: Root cause localization of performance issues in microservices." NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2020.

[25] Li, Yuewei, et al. "Tadl: Fault localization with transformer-based anomaly detection for dynamic microservice systems." 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). IEEE, 2023.