**Artificial Intelligence and Computing Innovations**

Intelligence Science Publishing

https://www.ispublishing.com

ISSN: 3070-4944

Vo.4, No.1, 2024

# *Memory-Enhanced Transformer for Backend Microservice Anomaly Detection*

**Yun Yang[1]***

*[1]Northeastern University, Boston, USA*

**Corresponding author: jtsydb1185@gmail.com*

***Abstract:*** With the widespread adoption of cloud-native architecture and service-oriented deployment, backend microservice systems, while enhancing business flexibility, also exhibit characteristics such as cross-service propagation, enhanced contextual coupling, and complex temporal evolution of abnormal behavior, posing higher demands on anomaly detection. Addressing the shortcomings of existing methods, such as insufficient utilization of historical states, inadequate characterization of long-distance dependencies, and limited ability to fuse multi-source runtime information, this paper proposes a backend microservice anomaly detection method based on memory-enhanced Transformers. This method, based on logs, monitoring metrics, and link observation data, first constructs continuous temporal input through a unified representation. Then, an external memory unit is introduced to store and retrieve key historical patterns, enabling the model to continuously incorporate anomaly-related contextual clues during current state discrimination. Subsequently, a gating fusion mechanism and Transformer encoding structure are combined to collaboratively model current observation features and historical memory information, thereby enhancing the ability to express complex anomaly semantics and long-term correlations. To achieve anomaly detection, this paper further utilizes reconstruction bias to construct an anomaly scoring method, enabling the model to characterize normal behavioral structures and anomaly deviations from the latent representation space. A comparative study was conducted on anomaly data of open-source microservices. The results show that the proposed method has good recognition ability and stable performance in backend microservice anomaly detection tasks. It can more effectively characterize the anomaly features in complex operating environments and provide a feasible methodological support for anomaly monitoring in intelligent operation and maintenance scenarios.

***Keywords:*** Microservice anomaly detection; memory retrieval mechanism; temporal representation learning; anomaly scoring

## 1. Introduction

With the rapid development of cloud computing, container orchestration, and continuous delivery technologies, backend systems are evolving from traditional monolithic architectures to microservice architectures composed of numerous loosely coupled services. This type of architecture improves system scalability, maintainability, and iteration efficiency through service decomposition, interface decoupling, and elastic scaling, and has become an important implementation method for modern internet platforms, enterprise-level information systems, and intelligent application infrastructure. However, while microservices bring flexible deployment and high concurrency capabilities, they also significantly increase the complexity of the system's operating environment. The dynamic changes in the number of service instances, the lengthy cross-service call chains, and the widespread existence of asynchronous communication mechanisms make the system state more dynamic, interconnected, and uncertain. Traditional anomaly detection methods relying on manual rules or shallow statistical features are insufficient to meet the accurate identification needs in complex scenarios [1, 2].

Backend microservice anomalies typically exhibit characteristics such as multi-source heterogeneity, rapid propagation, and hidden manifestations. Their causes may involve multiple levels, including resource contention, network jitter, service degradation, dependency failure, container drift, and configuration deviations. Anomalies rarely occur in isolation; rather, they accumulate and spread gradually within call chains, time series, and service dependencies, ultimately leading to increased response latency, decreased throughput, and even cascading service failures [3]. Because monitoring logs, call chains, and operational metrics in microservice systems are highly coupled over time, anomaly patterns often exhibit characteristics of long-distance dependencies, local mutations, and global correlations. Relying solely on short-term window modeling or static feature extraction is insufficient to fully capture the crucial contextual information during anomaly evolution. Therefore, developing anomaly detection methods with both time-series modeling and context-aware capabilities for backend microservice scenarios has become an important research direction for ensuring stable system operation and improving intelligent operation and maintenance [4].

In recent years, deep learning-based sequence modeling methods have demonstrated strong representational capabilities in complex time-series analysis tasks. Among them, the Transformer structure, capable of parallel modeling of global dependencies, provides a new technical path for microservice anomaly detection. Compared to traditional time-series models, the Transformer can more effectively capture the correlation information between different time points, providing a stronger expressive foundation for identifying complex anomaly patterns [5, 6]. However, in backend microservice scenarios, anomaly signals often exhibit sparsity, suddenness, and phased occurrences. Simply relying on attention mechanisms for global information aggregation may still face challenges such as insufficient utilization of key historical states, weakened long-term evolutionary memory, and inadequate preservation of anomaly context. Especially in long-term monitoring data, anomaly formation is often closely related to early-stage potential disturbances. Without a stable and effective historical memory mechanism, the model's ability to identify latent and progressive anomalies will be significantly limited.

Based on this background, research on anomaly detection in backend microservices using memory-enhanced Transformers has significant theoretical and practical value. On one hand, this direction helps to further strengthen the model's ability to retain historical key states, anomaly context clues, and long-term dependencies based on sequence global modeling, thereby improving the completeness and robustness of anomaly identification in complex dynamic environments. On the other hand, this research can provide more efficient, accurate, and continuously perceptive anomaly detection support for large-scale distributed systems, cloud-native platforms, and intelligent operation and maintenance systems, reducing the risk of fault propagation and operation and maintenance response costs, and enhancing the system's service quality assurance capabilities. Furthermore, exploring the deep integration of memory enhancement mechanisms and Transformer architecture in complex microservice scenarios can provide valuable methodological references

for time-series anomaly analysis, intelligent monitoring modeling, and secure operation and maintenance of complex software systems.

## 2. BackGround

Backend microservice systems typically consist of multiple independently deployable service units. These services collaborate via lightweight communication protocols and form a complete operational system based on components such as containers, service registration, configuration centers, message queues, and gateways. In this architecture, system business capabilities are broken down into several fine-grained modules. Different modules handle functions such as data processing, request forwarding, access control, cache access, and state maintenance, thereby improving system development efficiency and horizontal scalability. However, the call relationships formed by frequent interactions between services exhibit clear hierarchical, dependent, and dynamic characteristics. Any local state fluctuation can propagate outward along the call path, affecting overall service stability. Since the system continuously generates a large amount of monitoring metrics, log information, and tracing data during operation, accurately characterizing the system's operational state from high-dimensional, continuous, and interconnected observational information has become a fundamental issue in backend microservice intelligent operation and maintenance research.

Anomaly detection is a core component of backend microservice operational assurance. Its goal is to promptly identify state changes deviating from normal behavior patterns from complex operational data, providing a basis for subsequent fault localization, performance optimization, and risk warning. Compared to traditional centralized systems, anomalies in microservice environments exhibit stronger contextual dependencies and cross-component coupling. The operational characteristics of different service instances also vary significantly across different time periods. Some anomalies manifest as short-term, sudden fluctuations, while others show performance degradation after long-term accumulation [7]. Still others are accompanied by coordinated changes across multiple services, leading to more blurred anomaly boundaries and more complex patterns. In this context, anomaly detection no longer relies solely on judging single metric thresholds but requires a comprehensive consideration of temporal evolution patterns, service interaction structures, and historical state correlations to establish a modeling approach that better reflects the actual operational characteristics of backend microservices.

## 3. Methodology

In order to characterize the nonlinear evolution of backend microservice states under complex service dependencies, the method first organizes multivariate observations collected from monitoring metrics, trace statistics, and log-derived quantitative signals into a unified temporal representation, so that heterogeneous runtime evidence can be projected into a common latent space before anomaly discrimination is performed.

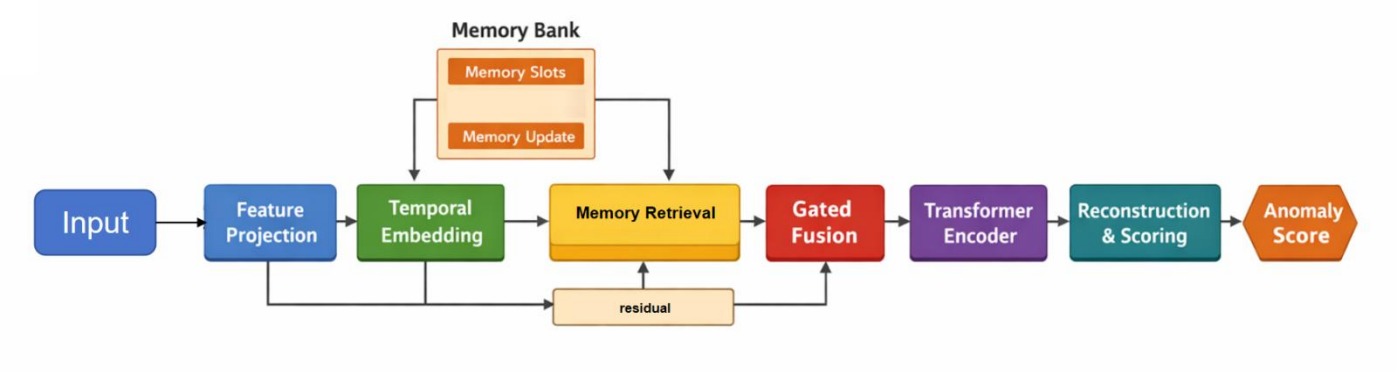


Figure 1. Overall model architecture

Let the observation sequence within a time window be denoted by:

$$\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T], \quad \boldsymbol{x}_t \in \mathbb{R}^d \tag{1}$$

where $T$ denotes the window length and $d$ represents the feature dimension of each timestamp. Rather than directly feeding raw observations into the sequence encoder, a trainable projection is introduced to enhance semantic consistency across heterogeneous signals, which is beneficial for reducing scale discrepancy and improving the separability between normal fluctuations and true abnormal variations. The latent embedding at time step $t$ is formulated as:

$$\boldsymbol{h}_t^0 = \boldsymbol{W}_e \boldsymbol{x}_t + \boldsymbol{p}_t \tag{2}$$

in which $\boldsymbol{W}_e$ is the feature projection matrix and $\boldsymbol{p}_t$ is the positional encoding that preserves temporal order information. This design makes the model not only aware of instantaneous service conditions but also sensitive to the sequential arrangement of state transitions, which is important because anomalies in microservice systems are often reflected through progressive context drift rather than isolated pointwise deviation.

Beyond conventional sequence modeling, the architecture introduces an external memory mechanism to retain historically salient operational patterns that may reappear across distant timestamps, thereby alleviating the tendency of standard attention models to weaken early abnormal cues when the observation horizon becomes long. A memory bank with $K$ slots is defined as:

$$\boldsymbol{M} = [\boldsymbol{m}_1, \boldsymbol{m}_2, ..., \boldsymbol{m}_K], \quad \boldsymbol{m}_k \in \mathbb{R}^{d_h} \tag{3}$$

where $d_h$ is the hidden dimension of the latent state. For each encoded token, the relevance between the current hidden representation and each memory slot is computed to retrieve historically compatible context, and the corresponding addressing weights are written as:

$$\alpha_{t,k} = \frac{exp(\boldsymbol{h}_t^\top \boldsymbol{W}_m \boldsymbol{m}_k)}{\sum_{j=1}^{K} exp\,(\boldsymbol{h}_t^\top \boldsymbol{W}_m \boldsymbol{m}_j)} \tag{4}$$

with $\boldsymbol{W}_m$ serving as the learnable similarity transformation. The retrieved memory-enhanced context is then aggregated by:

$$\tilde{\boldsymbol{m}}_t = \sum_{k=1}^{K} \alpha_{t,k}\, \boldsymbol{m}_k \tag{5}$$

so that long-range historical prototypes can participate in the current decision process. Such a mechanism is meaningful because transient noise and stable abnormal precursors may exhibit similar local amplitudes, whereas the reference to persistent historical structures provides an additional criterion for distinguishing short-term perturbation from genuine anomaly accumulation.

After memory retrieval, the model couples the current token state with the recalled historical context and injects the fused representation into a Transformer encoder to model cross-time dependencies at different semantic levels. Instead of treating memory as an auxiliary branch detached from self-attention, a gated fusion strategy is adopted to regulate how much historical information should enter the current temporal interaction, which prevents irrelevant memory content from overwhelming local evidence. The fusion process is expressed as:

$$\boldsymbol{z}_t = \sigma\big(\boldsymbol{W}_g[\boldsymbol{h}_t; \tilde{\boldsymbol{m}}_t] + \boldsymbol{b}_g\big) \odot \boldsymbol{h}_t + \Big(1 - \sigma\big(\boldsymbol{W}_g[\boldsymbol{h}_t; \tilde{\boldsymbol{m}}_t] + \boldsymbol{b}_g\big)\Big) \odot \tilde{\boldsymbol{m}}_t \tag{6}$$

where $\sigma(\cdot)$ denotes the sigmoid activation, $[\cdot\,;\cdot]$ is the concatenation operator, and $\odot$ represents element-wise multiplication. The resulting state $\mathbf{z}_t$ is subsequently processed through self-attention to capture global temporal interactions among all timestamps in the current window, allowing abnormal propagation paths across service states to be represented more completely than with local-context modeling alone. This stage is essential because microservice anomalies frequently emerge through delayed response chains, resource contention diffusion, and cascading performance degradation, all of which require the encoder to jointly perceive current evidence, historical traces, and cross-step structural relevance.

Finally, anomaly scoring is performed by reconstructing the normal behavior manifold in latent space and measuring the discrepancy between encoded states and their predicted normal counterparts, since abnormal

samples usually deviate from regular temporal organization more significantly than normal observations. Let $\hat{\boldsymbol{z}}_t$ denote the reconstructed representation generated by the decoding layer, then the anomaly score at time step $t$ is defined as:

$$s_t = \|\boldsymbol{z}_t - \hat{\boldsymbol{z}}_t\|_2^2 \tag{7}$$

so that larger reconstruction deviations indicate stronger abnormality intensity. To stabilize representation learning and preserve meaningful memory evolution, the training objective jointly minimizes reconstruction error and memory regularization, leading to:

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T} \|\boldsymbol{z}_t - \hat{\boldsymbol{z}}_t\|_2^2 + \lambda\|\boldsymbol{M}\|_F^2 \tag{8}$$

where $\lambda$ controls the contribution of memory regularization and $\|\cdot\|_F$ denotes the Frobenius norm. By combining heterogeneous feature projection, historical memory retrieval, gated temporal fusion, and deviation-based anomaly scoring within a unified framework, the method strengthens the capability of capturing latent abnormal semantics in backend microservice systems and provides a more robust modeling basis for identifying complex anomalies hidden in long-range operational dynamics.

## 4. Experimental Results and Analysis

### 4.1 Dataset

This paper selects LO2: Microservice API Anomaly Dataset of Logs and Metrics as the dataset foundation. This dataset is designed for anomaly detection and architectural degradation analysis in microservice systems. The data originates from the production-grade open-source microservice system Light-OAuth2, and effectively addresses the complex service calls, dynamic changes in runtime states, and cross-component propagation of anomaly signals in backend microservice scenarios. Compared to data resources containing only a single metric or log modality, LO2 provides logs, runtime metrics, and link-related information simultaneously, thus offering more comprehensive data support for time-series correlation modeling, historical context preservation, and multi-source anomaly representation based on memory-enhanced Transformers. The dataset paper and accompanying open resources are publicly available; the Zenodo page provides sample data and reproduction scripts, meeting the open-source access requirements.

From a data organization perspective, LO2 covers service log files and large-scale monitoring metric files collected during multiple system runs, and includes publicly available scripts for reproducing the data collection process. Therefore, it is suitable for characterizing microservice state evolution over time and for analyzing the formation process of anomaly patterns from a multimodal perspective. Since backend microservice anomalies are often not isolated offsets at a single moment, but complex behaviors formed by the combined effects of request chains, service interactions, and historical states, the multi-source runtime data contained in LO2 can better support the method design of this paper, which focuses on long-term dependencies, memory retrieval, and context fusion, and ensure that the selection of datasets is highly consistent with the theme of the paper.

### 4.2 Data Preprocessing

In the data preprocessing stage, the log information, monitoring metrics, and timestamp records in the original microservice runtime data are first uniformly organized, aligned, and segmented according to service instances and time order to ensure consistent semantic correspondence between observations of different modalities within the same time window. Subsequently, missing values, abnormal noise points, and duplicate records are cleaned, and standardization methods are used to eliminate dimensional differences between different metrics to enhance the model's sensitivity to multidimensional feature changes. For log data, key event patterns or statistical features are further extracted through structured parsing and fused with numerical monitoring metrics to form a unified time-series input sequence. Finally, sample fragments are constructed based on a fixed sliding window strategy, and normal and abnormal samples are distinguished using data labels,

providing a standardized, continuous, and computable input foundation for subsequent time-series modeling and anomaly detection of the memory-enhanced Transformer.

### 4.3 Experimental Setup

To ensure that the backend microservice anomaly detection method based on memory-enhanced Transformer could be trained and validated under unified and reproducible conditions, the experimental environment was uniformly configured in terms of hardware platform, software framework, number of training epochs, batch size, optimizer, and learning rate. Consistency was also maintained in data partitioning, random seed, and input window settings to reduce interference from non-model factors. Considering the significant temporal correlation and multi-source heterogeneity of microservice runtime data, a fixed-length time window was used to construct input samples during the experiment, and log, metric, and link observation data were uniformly aligned before being fed into the model. Meanwhile, to enhance the stability of the training process, an adaptive optimization strategy was adopted for model parameter updates, and the same initialization method and random control settings were used during training. Specific experimental configurations are shown in Table 1.

Table 1. Hyperparameter Settings

| Parameter Category | Parameter Name | Value |
|---|---|---|
| **Data Settings** | Time Window Length | 50 |
| **Data Settings** | Sliding Step Size | 10 |
| **Training Settings** | Batch Size | 32 |
| **Training Settings** | Epochs | 100 |
| **Training Settings** | Optimizer | AdamW |
| **Training Settings** | Initial Learning Rate | 1e-4 |
| **Training Settings** | Weight Decay | 1e-5 |
| **Training Settings** | Dropout | 0.1 |
| **Model Settings** | Hidden Dimension | 128 |
| **Model Settings** | Number of Attention Heads | 8 |
| **Model Settings** | Number of Transformer Layers | 4 |
| **Model Settings** | Number of Memory Slots | 32 |

### 4.4 Experimental Results and Analysis

To illustrate the technological context of our proposed method from a related research perspective, we selected representative works closely related to backend microservice anomaly detection, log and link joint modeling, indicator fusion analysis, and graph representation learning for comparison. The listed methods cover different observation sources such as logs, indicators, and distributed links, and can comprehensively reflect the typical research ideas in the current field of microservice anomaly detection in terms of multi-source information fusion, structural modeling, and temporal representation. The relevant comparison results are shown in Table 2.

Table 2. Experimental Results Compared with Other Models

| Method | ACC | Precision | Recall | F1 |
|---|---|---|---|---|
| **Zhang et al. [8]** | 95.68 | 94.71 | 93.84 | 94.27 |
| **Pawar et al. [9]** | 94.92 | 93.88 | 92.76 | 93.31 |
| **Xie et al. [10]** | 95.14 | 94.02 | 93.67 | 93.81 |
| **Xie et al. [11]** | 94.57 | 93.21 | 92.94 | 93.07 |
| **Ren et al. [12]** | 95.83 | 94.95 | 94.28 | 94.61 |
| **Panahandeh et al. [13]** | 96.21 | 95.34 | 94.87 | 95.10 |
| **Yang et al. [14]** | 96.08 | 95.12 | 94.96 | 95.03 |
| **Ours** | 97.36 | 96.58 | 96.11 | 96.34 |

Overall, the proposed algorithm demonstrates strong discriminative ability and good stability in anomaly detection tasks, indicating that the constructed memory-enhanced modeling mechanism can adequately characterize key anomaly features in the operational state of backend microservices. This method not only effectively extracts latent semantic information from multi-source observation data but also maintains continuous awareness of the anomaly context in complex temporal dependencies, thus exhibiting higher accuracy and better overall recognition quality in the classification process. This demonstrates that the designed model structure has strong applicability in microservice anomaly detection scenarios and can provide a reliable basis for the analysis of complex system operational states.

Furthermore, the proposed method maintains a relatively coordinated performance in precision, recall, and overall evaluation results, reflecting that the model not only has good sensitivity in identifying anomaly samples but also effectively suppresses false positives. This result is attributed to the memory-enhanced mechanism strengthening the model's ability to retain historical key states, the Transformer structure improving the modeling ability for long-range correlation information, and the multi-source data fusion process enhancing the completeness of anomaly pattern representation. The combined effect of the above factors enables the model to achieve good detection results in the dynamic and complex environment of backend microservices, which further demonstrates that the method presented in this paper has high practical application value.

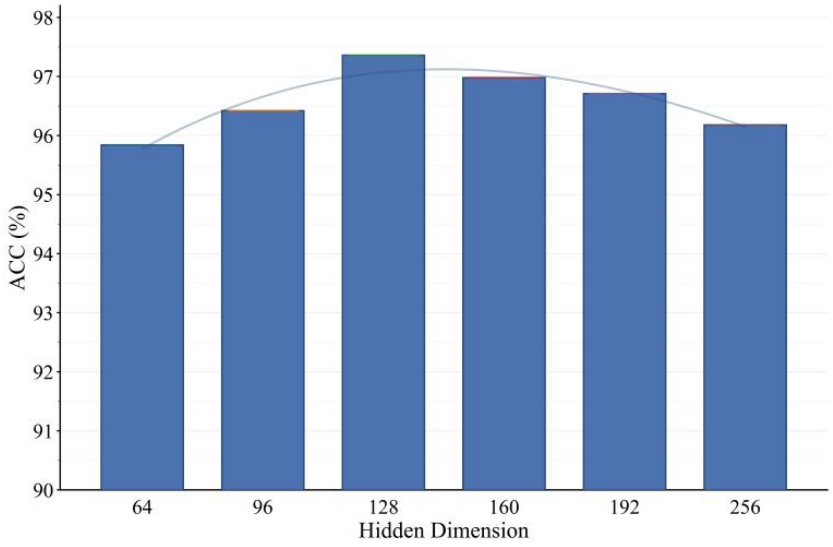


Figure 2. The impact of the hidden layer dimension on ACC

As shown in Figure 2, the proposed algorithm maintains strong classification and discrimination capabilities under the specified hidden layer dimension, indicating good stability and effectiveness of the model's internal feature representation. The memory enhancement mechanism and the Transformer structure work closely together during feature mapping, enabling the model to fully absorb key contextual information from the microservice execution sequence and encode anomaly-related patterns relatively completely into the latent space representation. Therefore, the proposed method still exhibits good recognition performance under these parameter conditions, demonstrating the adaptability and reliability of the model design in backend microservice anomaly detection tasks.

This result also shows that the proposed method has good information-carrying capacity in hidden representation construction, retaining key anomaly clues from the original observations while strengthening the expression of important historical states through memory retrieval and gating fusion. This modeling approach helps improve the model's ability to characterize complex temporal dependencies and potential anomaly semantics, thus ensuring good accuracy and consistency in the anomaly recognition process. Overall, the proposed algorithm achieved good results under this experimental setting, further validating its application value in microservice anomaly detection scenarios.

The visualization experiment of feature distribution before and after memory enhancement aims to characterize the internal changes of the model during feature reconstruction and semantic aggregation from the perspective of representation space, so as to more intuitively demonstrate the impact of the memory mechanism on the organization of the latent space. This experiment focuses on the latent representation formed by the proposed method in the feature encoding stage and observes the distribution and structural morphology of anomaly-related information in the mapping space through visualization. Combined with this process, the role of the memory enhancement module in context preservation, historical pattern injection, and anomaly semantic aggregation can be further presented, and the experimental results are shown in Figure 3.

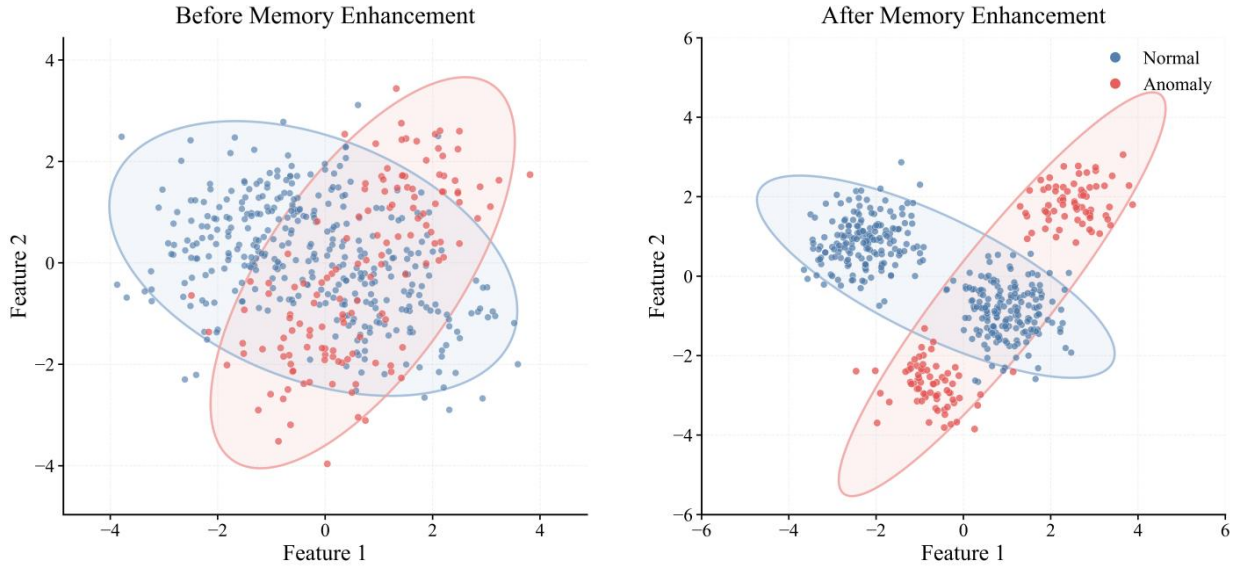


Figure 3. Visualization experiment of feature distribution before and after memory enhancement

The visualization results show that the feature distribution before memory augmentation still exhibits some overlap and mixing, and the boundary between normal and abnormal samples in the latent space is not clear enough. This indicates that while the original encoding can reflect some abnormal semantics, it is still insufficient in aggregating complex contextual information. After memory augmentation, the spatial organization of the feature representation becomes more regular, the clustering of samples from different categories in the latent space is significantly enhanced, the intra-class distribution is more compact, and the inter-class boundaries are clearer. This demonstrates that the memory mechanism can effectively supplement the historical key information that is easily weakened during temporal modeling, enabling the model to better preserve the abnormal-related context during the feature expression stage, thereby improving the discriminative ability of the latent space representation.

## 5. Conclusion

This paper addresses the challenges of hidden anomaly patterns, complex temporal dependencies, and tight coupling of multi-source observation information in backend microservice systems. It proposes an anomaly detection method based on memory-enhanced Transformers. To overcome the limitations of traditional methods, such as insufficient long-range context preservation, inadequate utilization of historical key states, and limited ability to model complex anomaly semantics, this study models the anomaly from two levels: temporal representation learning and historical information preservation. It organically integrates memory enhancement mechanisms with the Transformer structure, thus constructing a unified anomaly detection framework for backend microservice scenarios. This method can more meticulously characterize the system state evolution process based on multi-source runtime data and enhances the model's ability to perceive potential anomaly signals through memory retrieval, context fusion, and anomaly representation learning, providing a new research approach for intelligent anomaly detection in complex dynamic environments.

From a research perspective, the value of this work lies not only in the structural design at the model level but also in its expansion of the modeling methods for the operational mechanisms of backend microservices. Anomalies in microservice systems are often not caused by a single service state mutation but are closely related to service dependency propagation, runtime context accumulation, and the continuation of historical disturbances. Therefore, anomaly detection tasks inherently possess significant long-term dependency characteristics and complex semantic features. Based on this understanding, this paper introduces a memory enhancement mechanism to enable the model to more stably retain key historical clues during global temporal modeling and continue to play a role in subsequent state discrimination, thereby enhancing the completeness and discriminativeness of anomaly representation. This design approach, oriented towards historical context modeling, provides a more solid methodological foundation for microservice anomaly detection to move from local fluctuation identification to global behavior understanding, and also provides a technical reference with promotional value for intelligent operation and maintenance research of complex software systems.

From an application perspective, the method proposed in this paper has strong practical significance for cloud-native platforms, distributed service architectures, enterprise-level backend systems, and high-concurrency Internet service scenarios. As system scale continues to expand and service coupling deepens, traditional detection methods relying on human experience, static rules, or shallow statistical thresholds are no longer sufficient to meet the real-time monitoring needs of highly complex operating environments. Anomaly detection methods with the ability to retain historical states and model global dependencies are more likely to achieve stable, fine-grained, and forward-looking state recognition in complex scenarios. This research explores anomaly detection in backend microservices, contributing to improved system fault warning efficiency, reduced anomaly propagation risks, and shorter operation and maintenance response times. It also further promotes the development of anomaly detection technology towards automation, intelligence, and context awareness. Therefore, this research not only serves the specific anomaly detection task itself but also has a positive impact on the construction of intelligent operation and maintenance systems, service reliability assurance, and the secure operation of software systems.

Future research can be further deepened in several directions. First, as the sources of observational data for microservice systems become increasingly abundant, subsequent work can further enhance the joint modeling capabilities of the deep coupling relationships between log semantics, indicator time series, and call chains, thereby improving the model's adaptability to more complex anomaly patterns. Second, addressing the issues of continuous evolution of anomaly types, dynamic drift in distribution, and high label acquisition costs in real-world production environments, we can explore more suitable continuous learning, adaptive updating, and weakly supervised anomaly recognition mechanisms for open scenarios, thereby enhancing the model's generalization ability and stability during long-term deployment. Furthermore, by combining interpretable representation learning and anomaly root cause analysis, the model's capabilities in interpreting detection results, characterizing anomaly propagation paths, and locating service-level faults can be further enhanced, enabling anomaly detection to move beyond simple identification and become a diagnostic aid and decision support tool. Overall, research on backend microservice anomaly detection based on memory-reinforced Transformers still has broad development prospects, with the potential for continued expansion and in-depth application in areas such as intelligent operation and maintenance, complex system monitoring, and cloud service reliability assurance.

## References

[1] S. Zhang, Z. Pan, H. Liu, et al., "Efficient and Robust Trace Anomaly Detection for Large-Scale Microservice Systems", Proceedings of the 2023 IEEE 34th International Symposium on Software Reliability Engineering, pp. 69-79, 2023.

[2] S. Ding, Y. E, J. Zhang, et al., "Trace Anomaly Detection for Microservice Systems via Graph-Based Semi-Supervised Learning", Proceedings of the 2024 27th International Conference on Computer Supported Cooperative Work in Design, pp. 2375-2380, 2024.

[3] P. Wang, X. Zhang, Z. Cao, et al., "MADMM: Microservice System Anomaly Detection via Multi-Modal Data and Multi-Feature Extraction", Neural Computing and Applications, vol. 36, no. 25, pp. 15739-15757, 2024.

[4] Y. Xue, "State-Space Temporal Modeling and Feature Representation Learning for Anomaly Detection in Backend Microservice Systems", 2024.

[5] K. Wu, "Mamba-Based Temporal Feature Learning for Anomaly Detection in Distributed Microservice Environments", 2024.

[6] J. Chen, R. Zhang, P. Chen, et al., "MTG_CD: Multi-Scale Learnable Transformation Graph for Fault Classification and Diagnosis in Microservices", Journal of Cloud Computing, vol. 13, no. 1, Art. no. 103, 2024.

[7] J. Nobre, E. J. S. Pires and A. Reis, "Anomaly Detection in Microservice-Based Systems", Applied Sciences, vol. 13, no. 13, Art. no. 7891, 2023.

[8] C. Zhang, X. Peng, C. Sha, et al., "DeepTraLog: Trace-Log Combined Microservice Anomaly Detection Through Graph-Based Deep Learning", Proceedings of the 44th International Conference on Software Engineering, pp. 623-634, 2022.

[9] P. P. Pawar and C. Hartung, "TraceGra: A Trace-Based Anomaly Detection for Microservices Using Graph Neural Networks", Computer Communications, vol. 204, pp. 109-117, 2023.

[10] X. Xie, S. Jian and C. Huang, "LogTraceAD: Anomaly Detection from Both Logs and Traces with Graph Representation Learning", Proceedings of the 2023 2nd International Conference on Networks, Communications and Information Technology, pp. 116-121, 2023.

[11] Z. Xie, H. Xu, W. Chen, et al., "Unsupervised Anomaly Detection on Microservice Traces Through Graph VAE", Proceedings of the ACM Web Conference 2023, pp. 2874-2884, 2023.

[12] R. Ren, Y. Wang, F. Liu, et al., "TRIPLE: The Interpretable Deep Learning Anomaly Detection Framework Based on Trace-Metric-Log of Microservice", Proceedings of the 2023 IEEE/ACM 31st International Symposium on Quality of Service, pp. 1-10, 2023.

[13] M. Panahandeh, A. Hamou-Lhadj, M. Hamdaqa, et al., "ServiceAnomaly: An Anomaly Detection Approach in Microservices Using Distributed Traces and Profiling Metrics", Journal of Systems and Software, vol. 209, Art. no. 111917, 2024.

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