

Structure-Aware Multi-Scale Behavioral Learning for Distributed System Observability and Anomaly Identification

Yihao Zhou^{1*}

¹Brown University, Providence, USA

**Corresponding author: alexzzyh@gmail.com*

Abstract: To address the challenges of complex anomaly behaviors, diverse propagation paths, and the intertwining of local disturbances and global imbalances in distributed system environments, this paper proposes an anomaly detection method based on multi-scale behavioral pattern learning. This method leverages the characteristics of frequent node interactions, tight service dependencies, and dynamic evolution of operational states in distributed systems. It organizes system monitoring data into a unified dynamic behavioral representation and constructs a multi-layered modeling framework for anomaly identification based on this representation. In terms of method design, firstly, the node states, edge relationships, and multi-source operational information in the system are structurally represented to form a dynamic graph representation that reflects service call relationships and behavioral coupling characteristics. Then, focusing on the behavioral change patterns over different time periods, a multi-scale temporal pattern extraction mechanism is introduced to jointly characterize short-term fluctuations and long-term drifts, thereby enhancing the model's ability to perceive complex anomaly evolution processes. Furthermore, by incorporating spatial dependencies between services, the propagation characteristics of anomaly signals along call chains and interaction paths are modeled, enabling local anomalies and associated disturbances to be co-represented in a unified representation space. Finally, by fusing temporal behavioral features and spatial dependency features, an anomaly scoring mechanism for deviations from normal state prototypes is constructed to achieve effective identification of anomaly samples. The proposed method effectively adapts to the diversity, hierarchy, and dynamism of anomaly patterns in distributed systems. It balances representational and discriminative capabilities during detection, enhancing the modeling depth for complex operational states. Related research indicates that this method significantly improves the accuracy and stability of anomaly detection in distributed systems, providing valuable technical support for the monitoring, anomaly identification, and intelligent operation and maintenance of complex software systems.

Keywords: Behavioral pattern learning; dynamic graph modeling; spatiotemporal dependency representation; anomaly scoring mechanism

1. Introduction

With the continuous development of cloud computing, microservice architecture, container orchestration, and edge collaborative computing, modern distributed systems have been widely applied in critical scenarios such as financial transactions, intelligent manufacturing, e-commerce, online education, smart healthcare, and the industrial internet[1]. These systems typically consist of numerous service components with independent functions, heterogeneous computing nodes, and complex network communication relationships, exhibiting significant characteristics such as high concurrency, high coupling, strong dynamism, and continuous evolution. Against the backdrop of continuously expanding business scale and deepening service dependence, distributed systems have become an important technological form supporting the stable operation of digital infrastructure, and their operational quality directly affects business continuity, service reliability, and user experience. Therefore, researching efficient, accurate, and real-time anomaly detection of distributed system operation status has outstanding theoretical value and practical significance[2].

Compared to traditional monolithic systems, anomalies in distributed systems have longer propagation paths, wider impact ranges, and more complex manifestations[3]. On the one hand, anomalies may originate from various factors such as resource contention, service imbalance, network jitter, sudden load increases, configuration drift, or local component failures, and cascade and spread among multiple nodes through call chains, dependencies, and message passing mechanisms. On the other hand, different anomalies in monitoring data often exhibit temporal fluctuations, spatial correlations, and structural coupling, making it difficult for observational features at a single scale to fully reflect their formation mechanisms. Especially in environments with frequent scaling of large-scale service instances, constantly changing task scheduling, and continuous adjustments to system topology, anomaly behavior often simultaneously presents characteristics of local mutations and global imbalances[4]. This makes traditional methods relying on fixed thresholds, shallow statistical rules, or single time windows ineffective in meeting the detection needs of complex scenarios[5].

From the perspective of system operation patterns, anomalies are not isolated events, but rather dynamic processes that gradually accumulate and manifest at different time scales, service scales, and behavioral scales. Some anomalies initially manifest as short-term latency jitter, abnormal local resource usage, or fine-grained call offsets, subsequently evolving over a longer period into performance degradation, dependency imbalances, or even cascading service chain failures. This indicates that distributed system anomalies possess significant multi-scale behavioral characteristics, and their identification process requires not only attention to instantaneous fluctuations and local patterns but also a comprehensive characterization of cross-time window, cross-node interactions, and cross-service layer correlation information[6,7]. Research on anomaly detection based on multi-scale behavioral pattern learning helps to reveal the formation mechanisms, propagation patterns, and evolutionary characteristics of anomalies in complex systems from a more complete perspective, thereby improving the detection results' ability to perceive hidden anomalies, complex anomalies, and early signs of anomalies.

In the context of the continuous evolution of digital infrastructure towards higher complexity, higher autonomy, and higher availability, constructing multi-scale anomaly detection methods for distributed systems has become an important research direction for ensuring the safe and stable operation of systems. This research can not only provide key support for intelligent system operation and maintenance, proactive fault warning, and adaptive resource scheduling, but also provide a more reliable data foundation and decision-making basis for risk identification, performance optimization, and resilient governance of complex software systems. Furthermore, multi-scale behavioral pattern learning unifies the modeling framework with dispersed monitoring signals, dynamic call relationships, and continuous operating states, helping to promote the transformation of anomaly detection from passive response to proactive perception, and from local analysis to a deeper global understanding. This has significant

academic implications and broad application prospects for improving the observability, robustness, and autonomous operation and maintenance level of distributed systems.

Recent studies on distributed-system observability further show that anomaly identification should be coordinated with structural reasoning, adaptive runtime control, and trustworthy workflow execution. Context-aware trace graph modeling with GCN-based encoders and sequence reconstruction provides evidence that local and global service embeddings can strengthen the interpretation of abnormal microservice behaviors[6]. Graph-enhanced contrastive learning for cloud-system anomaly detection further demonstrates the value of jointly modeling topology, representation separation, and normal-behavior consistency when identifying complex operational deviations[7]. Risk-constrained reinforcement learning for SLO-driven cloud-native elastic scaling extends this perspective from detection to runtime decision control by connecting service-level objectives, operational risk, and adaptive resource adjustment[8]. Backend service scheduling that integrates reinforcement learning and meta-learning also highlights the need for policies that adapt to changing workload states and service dependencies, complementing anomaly detection with proactive coordination logic[9]. Trustworthy LLM-agent-oriented enterprise ETL orchestration with neural execution verification broadens the reliability setting to complex data-pipeline operations, where execution states, verification feedback, and abnormal transitions must be modeled jointly[10]. These studies collectively motivate a structure-aware multi-scale behavior learning framework that combines dependency representation, temporal evolution modeling, anomaly scoring, and operational reliability feedback.

2. Methodology

A distributed system usually exhibits anomaly symptoms through coupled variations across service invocation, resource fluctuation, temporal dependency, and cross-node interaction rather than through a single isolated indicator. This paper also presents the overall model architecture, as shown in Figure 1.

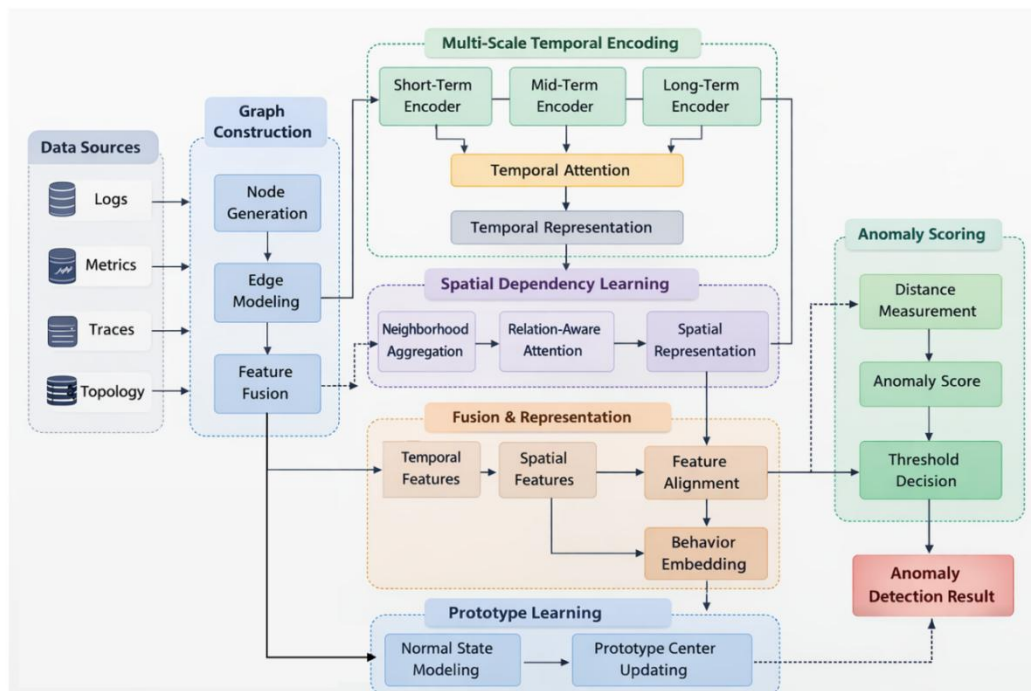


Figure 1. Overall model architecture

To capture this property, the proposed method first represents the system as a dynamic multi-scale behavior graph in which each service instance is treated as a node, each invocation or dependency relation is encoded as an edge, and the observation at time step t is described by a graph snapshot $\mathcal{G}_t$ containing node set $\mathcal{V}_t$, edge set $\mathcal{E}_t$, node feature matrix $\mathbf{X}_t \in \mathbb{R}^{|\mathcal{V}_t| \times d}$, and the relation matrix $\mathbf{R}_t$ that reflects dependency intensity and communication semantics.

$$\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t, \mathbf{R}_t) \quad (1)$$

Unlike static modeling strategies, this formulation preserves the structural evolution of the system and makes it possible to distinguish transient fluctuations from persistent abnormal tendencies. At the feature level, raw observations from logs, metrics, and traces are aggregated into a unified behavior vector $\mathbf{x}_i^t$ for node v_i , where ϕ_m^t denotes the m th monitored signal and α_m controls its contribution to the integrated state description:

$$\mathbf{x}_i^t = \sum_{m=1}^M \alpha_m \phi_m^t(v_i) \quad (2)$$

Such a representation is introduced to reduce modal fragmentation and to ensure that local performance changes, invocation disturbances, and state inconsistencies are aligned in a common feature space. Beyond simple fusion, the graph construction stage explicitly preserves behavioral coupling so that subsequent modeling can reason over both node-wise variation and relation-driven propagation, which is essential because anomalies in distributed environments often emerge as coordinated deviations instead of purely point-wise outliers.

Subsequently, multi-scale temporal patterns are extracted to characterize short-term bursts and long-range drifts in a unified manner. For each node representation sequence, a set of temporal encoders with different receptive fields is applied over a window family $\{\tau_1, \tau_2, \dots, \tau_K\}$, and the output $\mathbf{h}_{i,k}^t$ at scale k summarizes the behavior of node v_i within the corresponding temporal horizon through an encoding function $f_k(\cdot)$:

$$\mathbf{h}_{i,k}^t = f_k\left(\mathbf{x}_i^{t-\tau_k+1:t}\right) \quad (3)$$

This design is motivated by the fact that queue oscillation, latency spikes, and resource contention do not evolve at the same pace, so enforcing a single temporal granularity would inevitably blur discriminative patterns. Meanwhile, the scale-specific states are adaptively integrated by attention weights $\beta_{i,k}^t$, with $\mathbf{z}_i^t$ denoting the fused temporal behavior embedding and $\mathbf{W}_s$ serving as a learnable projection that measures the importance of each temporal view:

$$\mathbf{z}_i^t = \sum_{k=1}^K \beta_{i,k}^t \mathbf{h}_{i,k}^t, \quad \beta_{i,k}^t = \frac{\exp(\mathbf{w}_s^\top \mathbf{h}_{i,k}^t)}{\sum_{k'=1}^K \exp(\mathbf{w}_s^\top \mathbf{h}_{i,k'}^t)} \quad (4)$$

Through this mechanism, the model emphasizes the most informative temporal context for each service state instead of assigning fixed importance to all scales, thereby improving sensitivity to subtle precursors that may appear only within a specific observation span.

On top of temporal characterization, spatial dependency learning is introduced to model how abnormal signals propagate along service calls and resource interactions. Each node embedding receives messages from its neighborhood $\mathcal{N}(i)$, while the propagation coefficient a_{ij}^t evaluates the influence of the node v_j on node v_i according to both the behavior embedding $\mathbf{z}_j^t$ and the relation descriptor $\mathbf{r}_{ij}^t$ carried by the edge:

$$\mathbf{s}_i^t = \sigma\left(\sum_{j \in \mathcal{N}(i)} a_{ij}^t \mathbf{W}_g \left[\mathbf{z}_j^t \parallel \mathbf{r}_{ij}^t\right]\right) \quad (5)$$

$$a_{ij}^t = \frac{\exp\left(\mathbf{q}^\top \tanh\left(\mathbf{W}_q \mathbf{z}_i^t + \mathbf{W}_k \mathbf{z}_j^t + \mathbf{W}_r \mathbf{r}_{ij}^t\right)\right)}{\sum_{j \in \mathcal{N}(i)} \exp\left(\mathbf{q}^\top \tanh\left(\mathbf{W}_q \mathbf{z}_i^t + \mathbf{W}_k \mathbf{z}_j^t + \mathbf{W}_r \mathbf{r}_{ij}^t\right)\right)} \quad (6)$$

Rather than treating all neighbors equally, this relation-aware aggregation allows the model to focus on structurally relevant dependencies and suppress incidental interactions that do not contribute to anomaly formation. From a practical perspective, such differentiation is important because a latency disturbance in a critical upstream service often carries stronger diagnostic value than a similar fluctuation in a weakly coupled component. After that, the fused representation $\mathbf{u}_i^t$ is obtained by combining temporal and spatial information, where $\mathbf{W}_u$ aligns the two views and $\sigma(\cdot)$ introduces nonlinearity to preserve expressive capacity:

$$\mathbf{u}_i^t = \sigma\left(\mathbf{W}_u \left[\mathbf{z}_i^t \parallel \mathbf{s}_i^t\right]\right) \quad (7)$$

In this way, the learned state no longer reflects only what happens at an individual node, but also explains how local changes are shaped by surrounding behavioral context.

Finally, anomaly assessment is performed by measuring the deviation between the learned behavior representation and the normality prototype induced from historical stable states. Specifically, an anomaly score $\mathcal{A}_i^t$ is assigned to each node by comparing $\mathbf{u}_i^t$ with the reference center $\boldsymbol{\mu}_i$, and a larger distance indicates that the current behavior departs more significantly from the regular operating manifold:

$$\mathcal{A}_i^t = \|\mathbf{u}_i^t - \boldsymbol{\mu}_i\|_2^2 \quad (8)$$

This scoring strategy is chosen because distributed anomalies are frequently expressed as representation level shifts rather than as direct threshold violations in raw metrics. To stabilize optimization and encourage compact normal clusters together with discriminative abnormal separation, the training objective combines a reconstruction term $\mathcal{L}_{\text{rec}}$, a consistency term $\mathcal{L}_{\text{con}}$, and a deviation regularizer $\mathcal{L}_{\text{dev}}$, while λ_1 and λ_2 balance their relative effects:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_1 \mathcal{L}_{\text{con}} + \lambda_2 \mathcal{L}_{\text{dev}} \quad (9)$$

Consequently, the overall framework simultaneously models multi-scale temporal evolution, dependency-driven spatial propagation, and representation space deviation, which makes the detection process more consistent with the intrinsic operating logic of distributed systems. More importantly, this formulation improves the interpretability of anomaly identification because each abnormal decision can be traced back to temporal scales, dependency paths, and behavior shifts that jointly contribute to the final score.

3. Experimental Results and Analysis

3.1 Dataset

This paper uses the Anomaly Detection in Large-Scale Cloud Systems dataset as the foundation for experimental data. This dataset, publicly released through Zenodo, was constructed for anomaly detection research in large-scale cloud systems. It originates from telemetry data collected by the IBM Cloud Console over a period of approximately 4.5 months, covering multiple data centers, and possesses clear open-source accessibility and representativeness for engineering scenarios. The dataset contains 39,365 time records, each corresponding to a 5-minute sampling interval, and provides 117,448 features, which can adequately reflect the high-dimensional changes in resource status, workload, and system behavior in a distributed cloud environment.

From the perspective of suitability for the paper's theme, this dataset simultaneously possesses a distributed deployment background, temporal evolution attributes, and high-dimensional behavioral correlation features, providing suitable data support for anomaly detection modeling based on multi-scale

behavioral pattern learning. On the one hand, its multi-datacenter origin naturally gives it distributed characteristics across nodes, regions, and service layers, making it suitable for characterizing local fluctuations and global anomaly propagation phenomena in complex systems. On the other hand, the fixed-time sampling method provides a foundation for the joint modeling of short-term mutations, stage drift, and long-term trend changes, which is beneficial for uncovering anomalous patterns at different time scales. Based on these characteristics, this dataset can well meet the research needs of modeling temporal dependencies, behavioral coupling, and multi-scale changes in the anomaly detection scenario of distributed systems.

3.2 Experimental Results and Analysis

To further verify the effectiveness of the proposed method in distributed system anomaly detection scenarios, this paper selects representative works closely related to the research topic as comparison objects. These methods mainly focus on microservice environments, distributed call chains, system operation logs, and cloud system anomaly identification, and can well cover the typical modeling ideas in current related research. Based on a unified evaluation setting, the table presents the comparison of each method under four commonly used indicators, so as to conduct a horizontal analysis of different methods from the perspective of overall performance. The experimental results are shown in Table 1.

Table 1. Experimental results compared with other models

Method	ACC	Precision	Recall	F1
Panahandeh et al.[11]	93.18	92.74	92.36	92.55
Li et al.[12]	94.02	93.67	93.21	93.44
Xie et al.[13]	94.56	94.11	93.88	93.99
Khanahmadi et al.[14]	93.71	93.28	92.95	93.11
Liu et al.[15]	94.83	94.46	94.09	94.27
Xie et al.[16]	95.07	94.82	94.43	94.62
Bao et al.[17]	95.31	95.04	94.76	94.90
Ours	96.48	96.12	95.87	95.99

The algorithm proposed in this paper achieves state-of-the-art performance across all evaluation metrics, demonstrating its ability to more effectively identify anomalous behavior in distributed systems while maintaining accuracy, stability, and completeness of results. This success stems from the fact that the method not only focuses on local state changes but also models multi-scale temporal dependencies and inter-service spatial relationships, allowing anomalous features to be expressed within a richer behavioral context. Since anomalies in distributed systems often exhibit propagation, coupling, and phased characteristics, relying solely on single-level features is insufficient to fully characterize complex anomaly patterns. The proposed method, by jointly mining multi-level behavioral information, enhances the ability to capture key anomalous signals, thus demonstrating stronger discriminative power and higher overall detection quality.

As shown in Figure 2, the proposed method can form a relatively clear distribution structure of normal and abnormal regions in the behavior embedding space. Normal samples mainly exhibit a relatively compact clustering around the prototype center, while abnormal samples are scattered in the peripheral region, maintaining a clear distance from the core normal region. This indicates that the representation learned by the proposed method not only retains the stable characteristics of the normal operating state but also effectively amplifies the degree of deviation of abnormal behavior from the normal pattern, allowing different categories of states to form a natural spatial separation. Compared to

methods that rely solely on a single statistical feature for judgment, this discrimination mechanism based on the relationship between behavior representation and prototype deviation is more consistent with the actual law of the gradual accumulation of anomalies in distributed systems and their externalization into state transitions, thus demonstrating strong structural interpretability at the visual level.

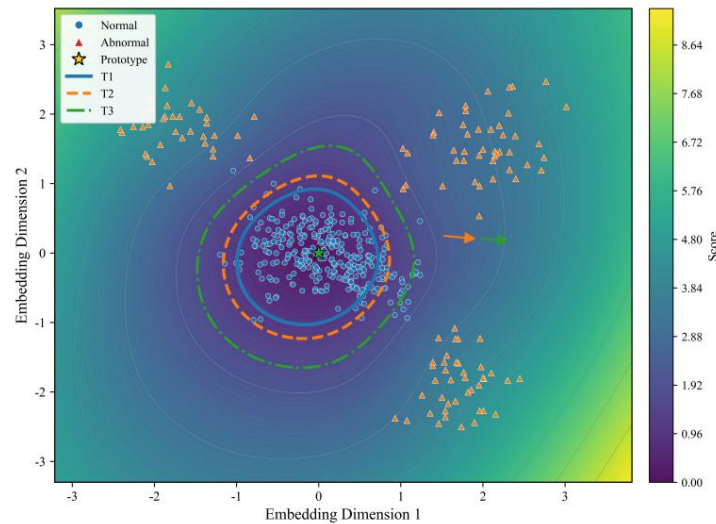


Figure 2. Visualization Experiment of Anomaly Detection Boundary Migration under Threshold Variation

Further observation of the detection boundaries corresponding to different thresholds reveals that as the judgment criteria are adjusted, the boundaries consistently change in an orderly manner around the core region of normal behavior, maintaining good continuity and containment in their overall shape, without obvious disorder or unreasonable breaks. This indicates that the proposed method has a relatively stable hierarchical ability in the anomaly scoring space, and different degrees of anomaly deviation can be organized into relatively clear regional hierarchies, thus providing a consistent judgment basis for changes in the detection boundaries. Since anomalies in distributed systems are often accompanied by service association propagation, local disturbance diffusion, and multi-scale behavioral imbalance, if the representation space lacks sufficient discriminative structure, threshold changes usually lead to boundary drift and confusion or even unstable sample assignment. The current results show that the method in this paper can better maintain the spatial order between the center of the normal pattern and the outward expansion area of the anomaly, thus providing more reliable support for subsequent anomaly identification and threshold setting.

To further examine the parameter stability of the proposed method during training, it is necessary to analyze the impact of learning rate changes on the model's discriminative ability. Since the learning rate directly determines the parameter update step size and convergence rhythm, different settings will directly affect the sufficiency of behavioral representation learning and the shaping of anomalous boundaries. Based on this consideration, sensitivity visualization of the learning rate can more intuitively present the adaptation range and stability characteristics of the proposed method during optimization, as shown in Figure 3.

The results show that the learning rate significantly impacts the performance of our proposed method. A learning rate that is too small limits parameter update efficiency, making it difficult to fully model multi-scale behavioral features and spatial dependencies. Conversely, a learning rate that is too large can disrupt the stability of the original representation structure, causing anomaly detection boundaries to shift and weakening overall recognition capabilities. A learning rate in the intermediate range allows the model to achieve a more reasonable balance between feature extraction depth and optimization stability, maintaining good coordination between behavioral embedding, prototype constraints, and anomaly

scoring. This results in more stable and effective detection performance. This also demonstrates that while our proposed method possesses strong anomaly modeling capabilities, its performance still depends on an appropriate optimization step size. Only when the update rhythm matches the representation learning process can the advantages of multi-scale behavioral pattern learning be fully realized.

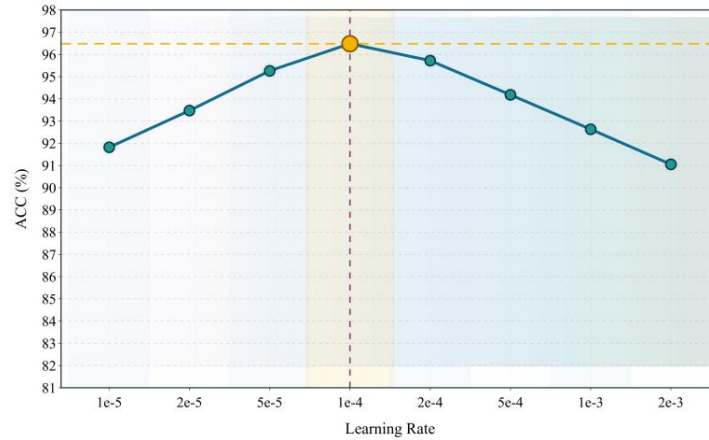


Figure 3. Experimental results on the sensitivity of the learning rate to ACC

4. Conclusion

This paper addresses the challenges of complex anomaly behavior, hidden propagation paths, close spatiotemporal correlations, and continuous evolution of operational states in distributed systems. It proposes an anomaly detection method based on multi-scale behavioral pattern learning. Starting from the actual operational characteristics of distributed systems, this method incorporates node state changes, service dependencies, and temporal evolution into a unified modeling framework. Through multi-scale temporal behavior characterization, spatial dependency propagation learning, and behavioral representation fusion, it effectively enhances the ability to identify complex anomaly patterns. Compared to traditional approaches that focus only on local indicator fluctuations or single-time-granularity changes, this method emphasizes the synergistic effects of multi-level behavioral information during anomaly formation, enabling the detection process to more closely resemble the intrinsic mechanisms of anomaly generation, diffusion, and accumulation in real systems. The results show that this method not only improves the accuracy and stability of anomaly identification in distributed systems but also provides a new analytical perspective for the structured understanding of anomaly behavior in complex software systems. For cloud computing platforms, microservice architectures, edge collaboration environments, industrial internet platforms, and high-concurrency online service systems, this research helps improve their operational status awareness, risk warning capabilities, and intelligent operation and maintenance levels, thus providing more reliable methodological support for ensuring critical business continuity, enhancing system robustness, and promoting the construction of autonomous operation and maintenance systems.

Future research can continue to expand in the direction of stronger generalization, higher real-time performance, and deeper application integration. On the one hand, as the scale of distributed systems continues to expand, service topologies continue to evolve, and business scenarios become increasingly complex, anomaly detection methods need to further improve their adaptability to dynamic structural changes, cross-domain dependency coupling, and non-stationary behavior drift, to maintain stable and effective detection performance in a more open and uncertain environment. On the other hand, facing the application needs of real production systems, subsequent work can further promote the deep integration of anomaly detection with fault location, root cause analysis, resource scheduling optimization, and operation and maintenance decision support, so that the detection model can not only discover anomalies, but also provide a more interpretable and operable basis for subsequent handling. Meanwhile, distributed

system anomaly detection technology has broad application prospects in areas such as smart city infrastructure, fintech platforms, intelligent manufacturing systems, network service governance, and large-scale data center management. As related research continues to deepen, intelligent anomaly perception methods based on behavioral pattern learning are expected to become a crucial foundational capability supporting the secure operation and high-quality service assurance of complex digital infrastructure, and will play a more profound role in promoting intelligent governance of software systems and digital transformation in key industries.

References

- [1] 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.
- [2] I. Kohyarnjadfard, D. Aloise, S. V. Azhari, et al., "Anomaly detection in microservice environments using distributed tracing data analysis and NLP," *Journal of Cloud Computing*, vol. 11, no. 1, p. 25, 2022.
- [3] X. Jiang, Y. Pan, M. Ma, et al., "Look deep into the microservice system anomaly through very sparse logs," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 2970-2978.
- [4] M. Raeiszadeh, A. Ebrahimzadeh, A. Saleem, et al., "Real-time anomaly detection using distributed tracing in microservice cloud applications," in *2023 IEEE 12th International Conference on Cloud Networking*, 2023, pp. 36-44.
- [5] D. Bogatinovski, S. Nedelkoski, J. Cardoso, et al., "Self-supervised anomaly detection from distributed tracing data," in *Proceedings of the Web Conference 2021*, 2021, pp. 3420-3434.
- [6] S. Nedelkoski, J. Cardoso, and O. Kao, "Anomaly Detection from System Tracing Data Using Multimodal Deep Learning," in *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, 2019, pp. 179-186.
- [7] S. Nedelkoski, J. Cardoso, and O. Kao, "Anomaly Detection and Classification Using Distributed Tracing and Deep Learning," in *Proceedings of the 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2020, pp. 241-248.
- [8] P. Hu, G. Uziel, and X. Ma, "Risk-Constrained Reinforcement Learning for SLO-Driven Cloud-Native Elastic Scaling," 2024.
- [9] N. Zhao, Z. Liu, Q. Li, et al., "Deep reinforcement learning for resource management in cloud computing: A survey," *ACM Computing Surveys*, vol. 54, no. 11s, Art. no. 245, 2022.
- [10] W. Wang and J. Cheng, "Trustworthy LLM-Agent-Oriented Enterprise ETL Orchestration via Neural Execution Verification," 2024.
- [11] 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, p. 111917, 2024.
- [12] Z. Li, J. Zhao, and J. Kang, "Multi-source anomaly detection for microservice systems," in *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings*, 2024, pp. 414-415.
- [13] Z. Xie, C. Pei, W. Li, et al., "From point-wise to group-wise: A fast and accurate microservice trace anomaly detection approach," in *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2023, pp. 1739-1749.
- [14] M. Khanahmadi, A. Shameli-Sendi, M. Jabbarifar, et al., "Detection of microservice-based software anomalies based on OpenTracing in cloud," *Software: Practice and Experience*, vol. 53, no. 8, pp. 1681-1699, 2023.
- [15] P. Liu, H. Xu, Q. Ouyang, et al., "Unsupervised detection of microservice trace anomalies through service-level deep bayesian networks," in *2020 IEEE 31st International Symposium on Software Reliability Engineering*, 2020, pp. 48-58.
- [16] Z. Xie, H. Xu, W. Chen, et al., "Unsupervised anomaly detection on microservice traces through graph vae," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 2874-2884.
- [17] L. Bao, Q. Li, P. Lu, et al., "Execution anomaly detection in large-scale systems through console log analysis," *Journal of Systems and Software*, vol. 143, pp. 172-186, 2018.