Transactions on Computational and Scientific Methods | Vol. 4, No. 7, 2024
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Heterogeneous Dynamic Dependency Graph Learning for Anomaly Detection in Large-Scale Backend Systems

Fangyi Chen
Northeastern University, Seattle, USA
fangyichen0219@gmail.com

Abstract: This paper addresses the anomaly detection needs of microservice systems characterized by continuously evolving dependencies, diverse call semantics, and heterogeneous, multi-source observation data. It constructs a dynamic dependency anomaly detection framework based on heterogeneous graph neural networks. The method represents the system runtime as a time-organized heterogeneous dynamic graph, explicitly characterizing service entities, interaction relationships, and their type semantics within a unified structure. Observational features such as logs, metrics, and links are mapped to a shared representation space to achieve cross-source alignment. To enhance the characterization of dependency reorganization and propagation effects, the framework employs a message passing and attention aggregation mechanism under relation type constraints, selectively converging information along semantically consistent dependency channels, thereby strengthening the contribution of key interactions to anomaly detection. For time-series non-stationarity and short-term disturbances in dynamic environments, the method introduces learnable temporal representations and adjusts the intensity of temporal information injection through gating fusion, reducing noise amplification and enhancing the stability of state representation. Furthermore, the framework constructs edge representations at the relation layer and outputs anomaly scores to consider the joint discrimination of node states and dependency semantics. Comparative experimental results show that the proposed method outperforms other methods on multiple evaluation metrics, verifying the effectiveness and robustness of heterogeneous dependency modeling and temporal information fusion in dynamic dependency anomaly detection tasks.

Keywords: Dynamic dependency modeling; heterogeneous graph representation; relation semantic aggregation; anomaly scoring mechanism

1. Introduction

Cloud-native and microservice architectures are widely adopted in enterprise systems. Service decomposition brings greater agility and scalability, but also introduces more complex operational patterns and dependencies. Microservices work together through various mechanisms such as synchronous and asynchronous calls, message queues, shared storage, and configuration centers. Frequent interactions between services, spanning multiple component boundaries, lead to a strongly coupled propagation effect and non-linear behavior in the overall system[1]. With the normalization of fluctuating business traffic, version iterations, and elastic scaling, dependency topologies and call paths continuously change. Traditional monitoring and alerting based on static structures struggle to capture this continuously evolving interaction pattern, thus reducing the timeliness and reliability of anomaly detection[2].

Changes in dynamic dependencies are not equivalent to anomalies. Real-world systems exhibit both normal and risky change mechanisms. Normal changes stem from scaling up/down, canary releases, load balancing strategy adjustments, and cache hit changes, causing short-term dependency migrations and metric disturbances. Risky changes, on the other hand, may be triggered by cascading failures, configuration drift, resource contention, network jitter, and service degradation, manifesting as abnormal dependency strength, shifted interaction semantics, and reorganized propagation paths[3]. The two types of changes highly overlap in terms of time scale and observation noise, making it difficult for detection methods relying solely on threshold rules or single-indicator statistics to distinguish between acceptable fluctuations and potential fault signals, leading to frequent false alarms or severe missed alarms. Anomaly detection for dynamic dependencies requires understanding both the evolution of dependency structures and the collaborative changes in cross-service observations to form a structured characterization of the system's operational state[4].

Observational data from microservice systems naturally exhibits multi-source heterogeneity, including log events, indicator time series, and link tracing, with significant differences in the functional roles, resource types, and interaction protocols of different services. While unifying this information into a graph structure can preserve inter-service relationships and propagation paths, the isomorphic graph assumption often struggles to express the differences between various node types and semantic edges, and also fails to simultaneously model the complementary relationships between different observation sources[5]. Heterogeneous graph modeling can use various entities, such as services, interfaces, call chains, log templates, and indicator channels as nodes, and various relationships, such as calls, dependencies, co-occurrences, associations, and temporal adjacencies as edges, thereby characterizing the system structure and semantics within a unified framework. Representation learning based on heterogeneous graph neural networks can transmit and aggregate information under the constraints of relation type and node type, improving the ability to express complex interactive semantics and cross-domain associations, and providing stronger structural priors and interpretable association paths for dynamic dependency anomaly detection.

To meet the needs of reliability governance and intelligent operation and maintenance, it is of great significance to construct anomaly detection methods that can adapt to the evolution of dynamic dependencies. On the one hand, accurately identifying abnormal changes in dependencies helps to capture weak signals in the early stages of fault propagation, reducing the risk of cascading failures and shortening recovery time. On the other hand, detection results based on structured heterogeneous representations can present potential risk sources in the form of associated links and key relationships, enhancing the operability of localization and decision-making. Furthermore, research on dependency anomaly detection helps to shift operation and maintenance from passive alarms to proactive risk perception, providing a data-driven basis for capacity planning, release control, and elasticity strategies, and laying the foundation for building an interpretable intelligent monitoring system for cloud-native systems.

2. BackGround

Microservice systems break down business capabilities into multiple independently deployable and evolving service units, achieving elastic operation through infrastructure such as service discovery, gateway routing, configuration management, and container orchestration[6]. The lifecycle of service instances is driven by scheduling and scaling strategies; the number of instances and their deployment locations change over time, causing the same business request to potentially traverse different instance sets and call paths at different times. Simultaneously, runtime strategies such as rate limiting, circuit breaking, retries, degradation, and caching alter request distribution and call frequency, dynamically adjusting dependencies in strength and direction. This dynamic dependency nature means the system state is no longer determined solely by single-point metrics but is shaped by multi-service collaboration patterns and propagation chains. Runtime understanding requires expression and reasoning based on the relational structure.

Anomaly detection in microservice scenarios typically relies on three types of observation signals: logs, metrics, and traces. However, each signal differs in granularity, noise characteristics, and semantic expression.

Log events emphasize discrete semantics and context; metrics emphasize continuous trends and periodic disturbances; and traces emphasize cross-service causal links and latency decomposition. Processing them in isolation will result in the loss of crucial relational information[7]. Graph representation learning provides a unified structured representation approach for multi-source observations. Heterogeneous graphs, by distinguishing entity types and relational semantics, can express multiple relationships between services and interfaces, events and indicator channels, and call segments and resource dimensions. Heterogeneous graph neural networks perform message passing and representation fusion under type constraints, mapping local observations to consistent representations in a global dependency structure. This provides a methodological foundation for characterizing anomaly propagation, identifying anomalous relationship changes, and supporting subsequent root cause analysis.

3. Methods

Facing the characteristics of continuously evolving dependencies and multi-source heterogeneous observation signals in microservice systems, the method design takes the dependency structure as the main thread and unifies temporal perturbations and semantic differences into the same representation space, thereby avoiding simplifying anomalies as single-metric shifts while ignoring critical clues of cross-service propagation and relationship reconfiguration. This paper presents the overall model architecture, as shown in Figure 1.

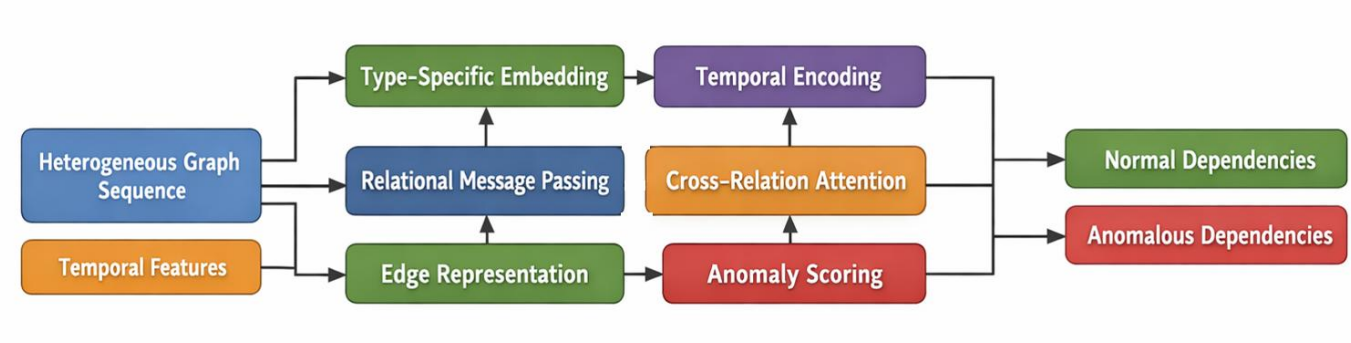


Figure 1. Overall model architecture

The microservice runtime state is first abstracted as a heterogeneous dynamic graph with types and relational semantics, and the significance of this step lies in mapping service instance changes and call-chain rerouting into learnable structural variations while providing an explicit inductive bias for subsequent relation-level modeling. The graph representation at time t is defined as:

$$\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \boldsymbol{X}_t, \mathcal{T}, \mathcal{R})$$

and the type and direction of edges are used to express invocation and dependency semantics, so that observations from different sources can be aligned consistently within node features. The input of a node $v \in \mathcal{V}_t$ is composed of structural features and runtime features, letting $\boldsymbol{x}_{v,t} \in \mathbb{R}^d$ denote the concatenated vector, and a type-specific mapping matrix is used to project heterogeneous inputs into a shared space to ensure cross-type comparability:

$$\boldsymbol{h}_{v,t}^{(0)} = \boldsymbol{W}_{\tau(v)} \boldsymbol{x}_{v,t}$$

where $\tau(v) \in \mathcal{T}$ identifies the node type, and the introduction of $\boldsymbol{W}_{\tau(v)}$ strengthens the separability of type differences and suppresses dimensional interference caused by features at different scales.

Considering that dependency anomalies often appear as a mixture of temporal drift and short-term abrupt changes, time information is injected into node states via learnable continuous embeddings, thereby separating periodicity and non-stationary perturbations into different dimensions to improve detection

sensitivity. The time encoding is constructed using a set of trainable-frequency basis functions, and given a timestamp t it yields a vector $\phi(t) \in \mathbb{R}^{2m+1}$:

$$\phi(t) = \left[t, \sin(\omega_1 t), \cos(\omega_1 t), \ldots, \sin(\omega_m t), \cos(\omega_m t)\right]$$

Then, the time embedding is fused with the initial node representation, and a gate is used to control the influence strength of temporal perturbations on different service states, avoiding excessive fluctuations during stable stages:

$$\widetilde{\boldsymbol{h}}_{v,t}^{(0)} = \boldsymbol{h}_{v,t}^{(0)} + \sigma \quad \left(\boldsymbol{U}\phi(t)\right) \odot \boldsymbol{V}\phi(t)$$

where $\sigma(\cdot)$ acts as a smooth gating function and $\odot$ denotes element-wise multiplication, and such a fusion treats time as a conditional signal rather than a noise source, mechanistically enhancing adaptability to concept drift.

Message passing on the heterogeneous graph is used to characterize dependency propagation and cross-domain coupling, and its core significance is to let anomalous signals aggregate along semantically consistent relational channels instead of being indiscriminately mixed in homogeneous adjacency. For a relation type $r \in \mathcal{R}$, relation-specific information is aggregated from the neighbor set $\mathcal{N}_r(v,t)$, and key dependencies are selectively emphasized through attention weights:

$$\boldsymbol{m}_{v,t,r}^{(l)} = \sum_{u \in \mathcal{N}_r(v,t)} \alpha_{u \to v,r,t}^{(l)} \boldsymbol{W}_r^{(l)} \widetilde{\boldsymbol{h}}_{u,t}^{(l)}$$

The attention coefficients are jointly determined by content similarity and relation constraints, enabling the same service to obtain differentiated aggregation results under different dependency semantics:

$$\alpha_{u \to v,r,t}^{(l)} = \frac{exp\left(LeakyReLU\left(\boldsymbol{a}_r^{(l)\top}\left[\boldsymbol{W}_r^{(l)}\widetilde{\boldsymbol{h}}_{u,t}^{(l)} \,\|\, \boldsymbol{W}_r^{(l)}\widetilde{\boldsymbol{h}}_{v,t}^{(l)}\right]\right)\right)}{\sum_{u' \in \mathcal{N}_r(v,t)} exp\left(LeakyReLU\left(\boldsymbol{a}_r^{(l)\top}\left[\boldsymbol{W}_r^{(l)}\widetilde{\boldsymbol{h}}_{u',t}^{(l)} \,\|\, \boldsymbol{W}_r^{(l)}\widetilde{\boldsymbol{h}}_{v,t}^{(l)}\right]\right)\right)}$$

where $\|$ denotes vector concatenation, and $\boldsymbol{W}_r^{(l)}$ and $\boldsymbol{a}_r^{(l)}$ equip each relation type with its own transformation and discrimination criterion, thereby distinguishing normal invocation migration from abnormal dependency surges at the structural level.

Furthermore, anomaly detection needs to lift node-level information to relation-level discrimination to capture abnormal changes in dependency strength and direction, thus a relation representation is constructed for each edge $e = (u,v,r,t) \in \mathcal{E}_t$ and an anomaly score is produced, using a differentiable scoring function that fuses endpoint representations and relation embeddings:

$$s(u,v,r,t) = \|\boldsymbol{W}_s\left[\widetilde{\boldsymbol{h}}_{u,t}^{(L)} \,\|\, \widetilde{\boldsymbol{h}}_{v,t}^{(L)} \,\|\, \boldsymbol{e}_r\right]\|_2$$

The training objective is constrained by structural consistency and emphasizes sensitivity to anomalous relations, learning a robust decision boundary by maximizing the compatibility of normal relations while enlarging the margin from perturbed relations:

$$\mathcal{L} = \sum_{(u,v,r,t)\in\mathcal{E}_t} \log\sigma\left(-s\left(u,v,r,t\right)\right) + \sum_{\left(u,\bar{v},r,t\right)\in\tilde{\mathcal{E}}_t} \log\sigma\left(s\left(u,\bar{v},r,t\right)\right)$$

where $\tilde{\mathcal{E}}_t$ denotes the set of perturbed edges generated according to relation semantics, and the contrastive structure of the loss drives representation learning to focus on relation-level discriminative differences, thereby maintaining stable identification capability for anomalous reconfigurations even in scenarios with frequently changing dependency topology.

4. Experimental Results and Analysis

4.1 Dataset

This paper uses the open-source AIOps2022 microservice root cause analysis dataset as the experimental data source. This dataset is designed for anomaly detection and dependency diagnosis scenarios in cloud-native microservice systems under multiple fault injection conditions, providing multi-source observable data and annotation information organized by fault samples. The data content covers three types of observations: metrics, logs, and traces. All observations are aligned to the same fault window through a unified timestamp, thus supporting the construction of a time-varying heterogeneous dependency graph in the context of dynamic service calls and cross-service propagation, and simultaneously characterizing service state evolution and dependency semantic changes at both the node and relationship levels. The dataset also provides fault type and root cause service-level labels, which can be used for anomaly scoring and localization evaluation in the candidate service space. This paper divides the samples according to training, validation, and test sets, and conducts comparative experiments on the detection performance and localization metrics of the model under the same partition to verify the effectiveness of the proposed microservice dynamic dependency anomaly detection method based on heterogeneous graph neural networks.

4.2 Experimental Results and Analysis

To systematically characterize the methodological differences and capability boundaries of heterogeneous graph neural networks in the scenario of dynamic dependency anomaly detection in microservices, representative studies highly relevant to the task set in this paper were selected as comparison objects, and a unified comparative framework was formed from dimensions such as dependency modeling, cross-modal fusion, and temporal characterization. The comparison table summarizes the performance records of each method on the same task using consistent evaluation metrics, which facilitates subsequent replication, verification, and mechanism analysis under the same data and evaluation process. The experimental results are shown in Table 1.

Table 1: Comparative experimental results

Method	AUC	F1	Precision	Recall
Zhang et al.[8]	0.88	0.79	0.81	0.77
Chen et al.[9]	0.86	0.77	0.80	0.74
Wang et al.[10]	0.89	0.81	0.83	0.79
Zhao et al.[11]	0.90	0.82	0.85	0.80
Roy et al.[12]	0.84	0.75	0.78	0.73
Tang et al.[13]	0.87	0.78	0.82	0.75
Akmeemana et al.[14]	0.91	0.83	0.86	0.81
Ours	0.94	0.86	0.89	0.84

The methods listed in Table 1 exhibit clear differentiation across the four evaluation metrics, indicating that different modeling assumptions substantially impact the effectiveness of anomaly detection for dynamic dependencies in microservices. Overall, the baseline methods can capture some anomaly patterns, but in scenarios involving rapid dependency reorganization and multi-source observation coupling, they remain susceptible to noise disturbances and semantic drift, resulting in insufficient identification stability and unclear discrimination boundaries. This phenomenon suggests that representations relying solely on a single signal or weak structural constraints struggle to simultaneously consider relational semantics and temporal non-stationarity, thus limiting the ability to characterize anomaly propagation and relation-level changes.

Ours performs better across all four metrics, reflecting that heterogeneous dependency modeling and temporal information injection more fully preserve cross-service interaction semantics and highlight key dependency channels in relation-level aggregation, making anomaly signals less prone to dilution during propagation. Thanks to message passing under type and relational constraints and selective aggregation of key dependencies, the model can more accurately distinguish between normal and risky changes, reducing misjudgments of short-term jitter as anomalies, while improving sensitivity to weak and early anomalies. The overall results show that fusing dynamic dependency structures with multi-source observations in a unified representation space helps to form more reliable anomaly discrimination criteria and improve the overall detection quality.

The temporal encoding dimension determines the model's capacity to represent periodic and non-stationary perturbations, thus affecting the separability of dependent states in the shared representation space. This ablation is used to examine the effectiveness boundary of temporal information injection under different capacity configurations and to assess whether excessively small or large representation dimensions weaken the stability of relation-level discrimination. Plotting the temporal encoding dimension versus AUC around this setting can be used to determine whether a more suitable capacity range exists to support the subsequent methodological description; the experimental results are shown in Figure 2.

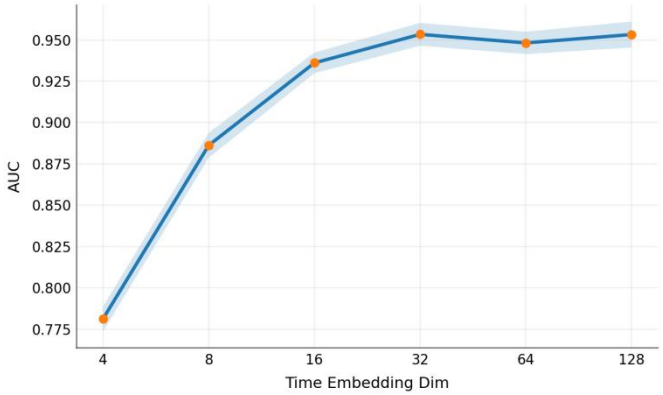


Figure 2. The impact of the time encoding dimension on AUC

The temporal encoding dimension directly affects the carrying capacity of temporal information in the representation space, thus impacting the separability of dependency states and the clarity of relation-level discrimination. The figure shows that in low-dimensional configurations, the compressed temporal pattern struggles to simultaneously express periodicity and non-stationary perturbations, limiting discriminative ability. Entering the medium-dimensional range, the temporal context can participate more fully in heterogeneous relation aggregation, and the proposed algorithm achieves a more ideal AUC, indicating that the temporal injection mechanism and relational attention can form a more stable collaborative representation.

In higher-dimensional configurations, the AUC remains high with minimal variation, demonstrating the algorithm's strong robustness to temporal encoding capacity. This phenomenon suggests that the model does not rely on excessively large temporal embedding dimensions for effectiveness, but rather injects effective temporal signals into key dependency channels through gated fusion and relational semantic constraints,

thereby reducing noise amplification and overfitting risks caused by redundant dimensions. Overall, these sensitivity results provide a clear basis for the selection of temporal encoding dimensions and enhance the reliability and feasibility of the proposed algorithm in dynamic dependency anomaly detection scenarios.

5. Conclusion

It proposes a dynamic dependency anomaly detection framework centered on heterogeneous graph neural networks, integrating service interactions, relational semantics, and temporal perturbations into a structured representation learning process. The method uses a heterogeneous dynamic graph to characterize the runtime state, performing message passing and selective aggregation under type and relational constraints. This ensures that anomalous signals are amplified along semantically consistent dependency channels rather than being diluted by homogenization. This modeling approach transforms dependency topology changes from unusable runtime noise into learnable structural evidence, providing a more realistic representational foundation for characterizing dependency reorganization, propagation link reshaping, and cross-service coupling.

At the application level, dynamic dependency anomaly detection has direct value for reliability governance in cloud-native systems. Dependency anomalies often precede large-scale failures and can spread rapidly through call chains. Structured detection can capture weak cross-service signals earlier and reduce over-reliance on single-point thresholds. For production-oriented incident response processes, dependency-level anomaly profiling helps quickly focus on key interaction channels in complex topologies, improving the certainty of risk identification and handling decisions, and reducing the costs of repeated trial and error in fault localization. Furthermore, this framework provides a dependency-side risk measurement perspective for release control, capacity planning, and resilience strategies, enabling system management to gradually shift from passive response based primarily on indicator monitoring to proactive protection centered on structural risks.

From a broader industry perspective, microservices have become the mainstream architectural form in fields such as finance, e-commerce, online education, content distribution, and the industrial internet. Abnormal changes in dependencies are often highly correlated with user experience degradation, transaction failures, and service-level protocol breaches. The structured anomaly detection paradigm proposed in this paper can serve as a fundamental capability module in intelligent operations and maintenance systems, linking with alarm aggregation, root cause analysis, and automated repair to shorten the failure window and improve service continuity. For multi-tenant cloud platforms, dependency-level risk identification helps to more accurately distinguish between tenant load fluctuations and platform-side anomalies, improving the effectiveness of resource scheduling and isolation strategies. For edge cloud and hybrid cloud scenarios, heterogeneous relationship modeling can more naturally absorb the characteristics of cross-domain links and heterogeneous infrastructure, providing a scalable representation framework for stable operation under complex deployment patterns.

Future work can be further advanced in three directions. First, for the access and continuous evolution of new services in open environments, we can explore representation learning mechanisms with adaptive expansion capabilities in type and relationship spaces to reduce reliance on prior type definitions and relationship enumeration, and improve the generalization ability to novel dependency semantics. Second, for the availability requirements of real-world operation and maintenance processes, we can further combine causal constraints and structural consistency priors to more tightly couple the anomaly attribution process with the detection process, enabling structural evidence to naturally support the generation of subsequent localization and handling strategies. Third, for the engineering implementation of large-scale systems, we can study more efficient dynamic graph update and incremental inference schemes to reduce computational and storage overhead while ensuring detection quality, and achieve low-latency integration with online monitoring links, thereby promoting the continuous deployment and long-term operation of dynamic dependency anomaly detection in cloud-native production systems.

References

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

[2] Li Y, Liu C, Zhao X, et al. LogFlash: Real-time anomaly detection and diagnosis from system logs for large-scale software systems[C]//Proceedings of the 33rd IEEE/ACM International Conference on Automated Software Engineering. 2018: 80-90.

[3] Zhu Y, Wang J, Li B, et al. Microirc: Instance-level root cause localization for microservice systems[J]. Journal of Systems and Software, 2024, 216: 112145.

[4] Chen Y, Xu D, Chen N, et al. FRL-MFPG: Propagation-aware fault root cause location for microservice intelligent operation and maintenance[J]. Information and Software Technology, 2023, 153: 107083.

[5] Chen R, Ren J, Wang L, et al. Microegrcl: An edge-attention-based graph neural network approach for root cause localization in microservice systems[C]//International Conference on Service-Oriented Computing. Cham: Springer Nature Switzerland, 2022: 264-272.

[6] Zhou J, Li Z, Wu J, et al. Graph neural networks: A review of methods and applications[J]. AI Open, 2020, 1: 57-81.

[7] Zheng L, Chen Z, Chen H, et al. OCEAN: Online Multi-modal Root Cause Analysis for Microservice Systems[J], 2024.

[8] Zhang C, Peng X, Sha C, et al. Deeptralog: Trace-log combined microservice anomaly detection through graph-based deep learning[C]//Proceedings of the 44th international conference on software engineering. 2022: 623-634.

[9] Chen J, Liu F, Jiang J, et al. TraceGra: A trace-based anomaly detection for microservice using graph deep learning[J]. Computer Communications, 2023, 204: 109-117.

[10] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[C]//International Conference on Learning Representations. 2017.

[11] Zhao Z, Wang Z, Zhang T, et al. CHASE: A Causal Hypergraph based Framework for Root Cause Analysis in Multimodal Microservice Systems[J]. arXiv preprint arXiv:2406.19711, 2024.

[12] Roy D, Zhang X, Bhave R, et al. Exploring llm-based agents for root cause analysis[C]//Companion proceedings of the 32nd ACM international conference on the foundations of software engineering. 2024: 208-219.

[13] Tang H, Guo Y, Yang J, et al. Microservice anomaly diagnosis with graph convolution network based on implicit microservice dependency[C]//Proceedings of the 2023 9th International Conference on Computing and Artificial Intelligence. 2023: 437-441.

[14] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[C]//International Conference on Learning Representations. 2018.