Artificial Intelligence and Computing Innovations
Intelligence Science Publishing
https://www.ispublishing.com
ISSN: 3070-4944
Vo.4, No.3, 2024

Modeling Evolving Service Dependencies: Dynamic Graph Learning for Microservice Anomaly Detection

Chenfeiyu Wen
New York University, New York, USA
cw4301@nyu.edu

Abstract: Addressing the challenges of anomaly detection in cloud-native microservice systems due to frequent changes in backend dependencies and strong time-varying coupling of runtime states, this paper investigates the problem of dynamic graph temporal anomaly detection and proposes a joint structural and temporal modeling framework. The method organizes the service dependency graph using time-slice sequences, unifying the representation of service nodes, call relationships, and their multi-source attributes. It encodes dependency interaction strength and local structural consistency through relation-aware neighborhood aggregation, and uses a temporal aggregation module to characterize the cross-time state evolution patterns and long-term dependencies. Based on conditional prediction of normal evolution trajectories, an anomaly metric centered on prediction residuals is constructed, transforming the impact of structural perturbations and behavioral drift on the representation space into quantifiable anomaly scores, thereby supporting node-level and subgraph-level anomaly detection and event characterization. Experiments are conducted on open-source microservice monitoring data, and multiple evaluation metrics verify the effectiveness of the proposed method in terms of anomaly discrimination capability and overall recognition quality.

Keywords: : Dynamic graph sequences; distributed link tracing; relationship-aware aggregation; anomaly scoring modeling

1. Introduction

Cloud-native architecture and microservice deployment have driven the increasing complexity of large-scale distributed systems. The call chains between backend services have gradually shifted from static topologies to dynamic dependency networks driven by high-frequency refactoring and elastic scaling. The short lifecycles of service instances, canary releases, and automatic scaling mechanisms cause dependencies to continuously change over time, resulting in strong time-varying and tightly coupled system operating states. This is accompanied by an increase in the number of alarms, diversified fault propagation paths, and increased difficulty in locating the root causes of anomalies[1,2]. Anomaly detection has become a crucial foundation for ensuring availability and stability. Research on graph-based time-series anomaly detection for the dynamic evolution of backend dependencies helps capture anomaly evolution signals in complex dependencies and time-varying interactions, providing more reliable prior support for operation and maintenance governance and risk prevention[3].

In dynamic dependency networks, anomalies are not triggered solely by single-point metric mutations but often exhibit a coupling phenomenon of structural changes and temporal fluctuations. For example, the addition and disappearance of call relationships, the redirection of critical paths, the reorganization of local clusters, and the amplification of anomalies in cross-domain services may all present weak and scattered signs in the early stages[4]. Traditional detection paradigms that rely on fixed topologies or independent time series struggle to simultaneously characterize the co-evolution of structure and time, leading to false positives and false negatives, and are highly sensitive to sudden changes. To clearly characterize the problem attributes and research significance, Table 1 summarizes the key challenges and corresponding research value of graph temporal anomaly detection in dynamic dependency scenarios. This content will serve as the basis for problem modeling in subsequent method design and theoretical analysis.

Table 1. Key Challenges and Research Value for Graph Temporal Anomaly Detection under Dynamic Dependency Scenarios

Key Challenge	Typical Manifestation	Research Value
Time-Varying Dependency Topology	Nodes and edges are continuously added and removed, and invocation chains are frequently restructured.	Enables robust modeling of topological changes and reduces interference from change-induced noise.
Temporal Non-Stationarity	Metric distributions drift with workload fluctuations and version iterations, and periodicity is disrupted.	Improves adaptability to concept drift and seasonal disturbances.
Multi-Source Heterogeneous Coupling	Logs, metrics, and call paths jointly drive behaviors, and cross-service impacts are superposed nonlinearly.	Strengthens joint characterization of cross-domain associations and propagation effects.
Sparse Anomalies and Weak Signals	Early anomalies are scattered and small in magnitude, making them hard to capture with threshold-based rules.	Enhances sensitivity to early risks and enables proactive warning capability.
Fault Propagation and Cascading Effects	Local anomalies spread along dependency chains, leading to chain reactions and progressive degradation.	Facilitates identification of propagation paths and supports risk isolation decisions.

Graph temporal modeling provides a unified structured representation, treating services as nodes and dependencies as edges, and characterizing the evolution of interaction intensity and operational state using time series. This allows for the description of local structural patterns, global topological changes, and dynamic behavioral relationships within a single framework. Addressing the dynamic evolution of backend dependencies, graph temporal anomaly detection not only focuses on instantaneous anomalies but also emphasizes the identification of anomaly patterns and propagation processes, providing interpretable structural and temporal evidence for stability governance. This research direction is significant in improving the timeliness and accuracy of anomaly detection, reducing alarm noise, enhancing change robustness, and supporting automated operation and maintenance decisions. It also provides fundamental support for building a risk monitoring and fault protection system for highly reliable backend systems.

2. Background

Early anomaly detection research primarily focused on univariate or multivariate time series, with core approaches including statistical process control, probabilistic models, and discrimination methods based on distance or reconstruction errors[5]. These methods typically rely on stationarity assumptions or pre-defined distributions, treating anomalies as significant fluctuations deviating from normal patterns, thus exhibiting insufficient robustness in scenarios involving load mutations, periodic drift, and noise enhancement. Subsequent deep sequence models, through recursive structures, convolutional structures, and self-attention mechanisms, enhanced their ability to characterize nonlinear dependencies and long-range associations, and introduced detection signals such as prediction residuals, reconstruction residuals, and uncertainty estimations. Despite progress in complex time series modeling, the structural constraints and cross-entity interactions arising from dependencies remain difficult to fully represent through sequence modeling alone[6].

Graph structure anomaly detection research has developed multiple technical routes in both static network and dynamic graph scenarios. Common tasks include node anomalies, edge anomalies, subgraph anomalies, and full graph anomalies. Methods in static scenarios are mostly based on graph representation learning, graph convolution, and graph autoencoders, identifying anomalous patterns through neighborhood consistency and structural semantic deviations. Dynamic graph scenarios further incorporate a temporal dimension, employing time-slice modeling, continuous-time event modeling, or temporal aggregation mechanisms to learn structural evolution patterns and discriminate sudden structural disturbances[7]. Related research reveals the crucial role of structural information in anomaly detection; however, in backend-dependent networks, topology changes often occur simultaneously with fluctuations in performance metrics. Anomaly signals manifest at both the structural and behavioral levels, and relying solely on structural or attribute biases can lead to detection instability.

Fusion methods for graph temporal anomaly detection are increasingly focusing on joint modeling of structure and time. Typical approaches include spatiotemporal graph neural networks, attention-based cross-temporal dependency aggregation, and encoder-decoder prediction and reconstruction frameworks. Some studies define anomaly degree as the prediction bias of future states, while others emphasize learning the generation mechanism of normal interactions and using likelihood or energy as the criterion. Still others construct robust representations through contrastive learning and self-supervised tasks to alleviate the label scarcity problem. While joint modeling enhances expressive power, key challenges remain in dynamically evolving dependencies, including distribution drift introduced by topological changes, the variability of node and edge sets, characterizing cross-level propagation effects, and the separability of anomalies and changes. Developing more robust dynamic graph representation learning and anomaly measurement designs to address these challenges can improve usability and generalization capabilities in complex backend dependency scenarios.

3. Methodology

The dynamic evolution of backend dependency relationships causes anomalies to exhibit an intertwined pattern of structural perturbations and behavioral drift, thus requiring the integration of call topology and multi-source runtime attributes into a unified temporal modeling framework to characterize normal evolutionary rules and amplify deviation signals. Let the dynamic graph sequence within an observation window consist of multiple time slices, where each slice simultaneously contains a service set, a dependency set, and attribute tensors associated with nodes or edges, thereby explicitly representing the system runtime state as a graph-temporal object:

$$\mathcal{S} = \{G_t\}_{t=1}^{T} \tag{1}$$

$$G_t = (\mathcal{V}_t, \mathcal{E}_t, X_t, R_t) \tag{2}$$

To maintain comparability of representations under frequent topology changes, the node set is indexed by time slices and is allowed to be added or removed; the attribute matrix $X_t \in \mathbb{R}^{|\mathcal{V}_t| \times d}$ carries service-level metric features, and the relation tensor R_t describes edge types or call statistical attributes and provides conditions for subsequent relation-aware aggregation. Considering that anomalies are often jointly driven by short-term mutations and long-term drift, temporal information is not treated as an external label but is encoded into learnable temporal embeddings, to distinguish normal fluctuations at different stages under the same structural pattern and improve adaptability to concept drift:

$$z_t = \phi(t) \tag{3}$$

Based on the temporal embedding $z_t \in \mathbb{R}^{d_z}$, a gated modulation vector is constructed to adaptively adjust feature contributions at different times; the modulation function $\sigma(\cdot)$ adopts a dimension-wise compressed nonlinear mapping to suppress noisy dimensions and highlight critical changes:

$$\widetilde{X}_t = X_t \odot \sigma(z_t W_z) \tag{4}$$

The significance of the above design is to transform temporal dependence into continuous perturbations of the representation space, thereby preserving a stable characterization of normal dynamics when topology changes and load fluctuations coexist, and providing more consistent input representations for the subsequent structural encoder. This paper also presents the overall model architecture, as shown in Figure 1.

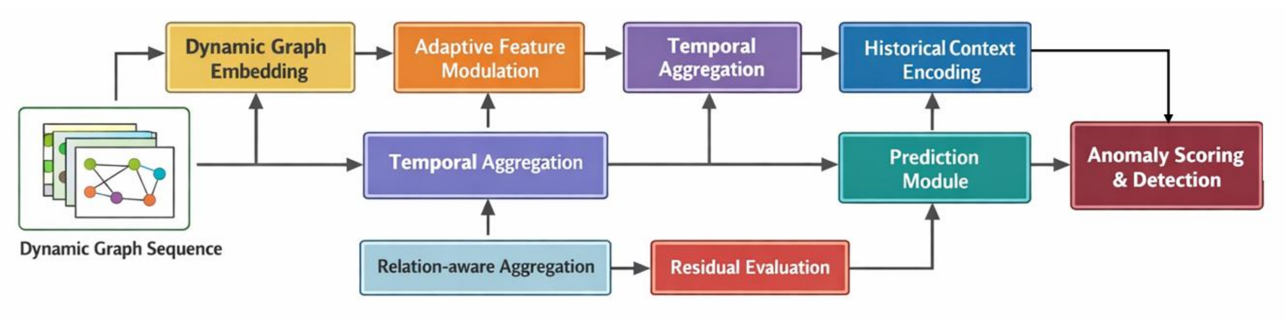

Figure 1. Overall model architecture

For dependency-aware structural modeling, relation-aware neighborhood aggregation is adopted to capture local consistency along call chains and differences in interaction intensity across services; during aggregation, edge attributes or edge types are embedded into attention weights to reduce biases caused by relying solely on degree information. The representation centered on node $v \in \mathcal{V}_t$ is accumulated over its neighborhood $\mathcal{N}_t(v)$, where α_{vu}^{t} is jointly determined by content similarity between node pairs and the relation vector, and $g(\cdot)$ maps the relation tensor R_t to an edge representation that participates in weighting:

$$h_v^t = \sum_{u \in \mathcal{N}_t(v)} \alpha_{vu}^{t} W_h \widetilde{x}_u^t \tag{5}$$

On this basis, a nonlinear transformation yields the time-slice-level node representation; the activation function $\psi(\cdot)$ enhances the separability of local patterns and avoids representational collapse caused by linear aggregation:

$$y_v^t = \psi(h_v^t) \tag{6}$$

The value of this structural encoding lies in making anomalies no longer triggered only by single-point metric deviations, but manifested as violations of dependency-propagation consistency, thereby better matching signal patterns such as cascading degradation and path redirection in backend systems.

To capture evolutionary trends of dependency relationships over time and distinguish change noise from true anomalies, the node representations of each time slice are further temporally aggregated. The aggregator $Agg(\cdot)$ adopts causality-constrained recurrence or self-attention structures to avoid future information leakage, and handles missingness introduced by variable node sets via an alignment mechanism within the window. Assuming the goal is conditional prediction of the next-step state, the predicted representation for node v can be generated from the historical segment $y_v^{1:t}$:

$$\widehat{y}_v^{t+1} = Agg(y_v^{1:t}) \tag{7}$$

The predictive residual provides a measurable deviation from the normal evolution trajectory and forms stable anomaly cues without requiring explicit anomaly labels; accordingly, the node-level anomaly score is defined as the residual norm to characterize the degree to which temporal consistency is violated:

$$s_v^{t+1} = \| y_v^{t+1} - \widehat{y}_v^{t+1} \|_2 \tag{8}$$

By combining structural consistency and temporal predictability, anomalies may be induced by local topology perturbations or accumulated through cross-time drift, thus yielding more interpretable discriminative evidence in dynamic dependency scenarios.

To enable the model to learn generalizable dynamic patterns under realistic collection conditions where normal data dominates, the training objective adopts a self-supervised predictive consistency constraint and incorporates anomaly-sensitive sparse regularization to suppress over-smoothing. Let Ω denote the set of all alignable nodes and time slices within the window; the objective minimizes prediction error and encourages a long-tailed residual distribution, making a small number of anomalous slices easier to separate in representation space:

$$\mathcal{L} = \sum_{(v,t)\in\Omega} \| y_v^t - \widehat{y}_v^t \|_2^2 + \lambda \sum_{(v,t)\in\Omega} | s_v^t | \tag{9}$$

This objective function simultaneously constrains dynamic prediction capability and anomaly separability: the former ensures sufficiently fine-grained characterization of normal evolution, while the latter prevents interpreting all fluctuations as noise and weakening sensitivity to risk signals. During inference, node-level scores can be aggregated into time-slice-level or subgraph-level indicators to meet operational requirements at different granularity levels, and the occurrence time and impact scope of anomaly events can be characterized according to the peaks and persistence of the score sequence.

4. Experimental Results and Analysis

4.1 Dataset

This paper uses an open-source monitoring dataset based on the Train Ticket microservice benchmark system as its research object. This dataset uses real microservice call chains as its core observation carrier, targeting scenarios where backend dependencies change over time. It provides key elements for constructing dynamic graph sequences, including distributed tracing data, inter-service call relationships, and call-related latency and status information. Since call chains naturally exhibit tree-like or graph-like service dependency structures, and request paths and service interactions change over different time periods, this dataset can directly support the needs of graph time-series modeling under dynamic dependency evolution and provide a unified data foundation for the structured representation of anomaly patterns.

The data source is the publicly released Train Ticket anomaly monitoring dataset, open-source and hosted on Zenodo. The data content covers three observation modalities: traces, metrics, and logs. The link tracing component, collected by Jaeger, includes span-level service call topology and call duration information, suitable for mapping individual requests to service dependency graphs and further forming a time-series graph set. The metrics component, provided by Prometheus, includes key performance metrics related to service runtime status, which can be embedded as node or edge attributes to enhance the semantics of the graph representation. The log component records service-side runtime events, which can be used to help characterize contextual changes before and after an anomaly occurs. These multimodal observations collectively constitute a continuous characterization of the dynamic dependency network, enabling anomaly detection to simultaneously utilize structural evolution signals and temporal behavior signals, thus maintaining consistency with the theme of graph temporal anomaly detection under the dynamic evolution of backend dependencies.

4.2 Experimental setup

To ensure the reproducibility and fairness of the graph time-series anomaly detection process, the experimental setup adopted unified standards in data preprocessing, time-slice construction, training strategies, and threshold decisions. Data was first aggregated into a dynamic graph sequence at a fixed time granularity, and service dependency edges and indicator features generated by link tracing were aligned to the same time slice. Node features consisted of multi-dimensional performance indicators and statistical features, while edge features were composed of call frequency and latency statistics. During the training phase, a sliding window was used to construct the input sequence and perform self-supervised next-step representation prediction. The optimization process used AdamW with learning rate decay and gradient pruning to improve stability. During the inference phase, event-level alarms were generated based on the anomaly score sequence, and a threshold strategy was used to control the false alarm level. Table 2 summarizes the key configurations of the hardware platform, data construction, and model training, clarifying the implementation details and experimental conditions for reproducibility.

Table 2. Experimental setup and configuration of key hyperparameters.

Operating System	Ubuntu 20.04 LTS	Operating System	Ubuntu 20.04 LTS	Operating System	Ubuntu 20.04 LTS	Operating System	Ubuntu 20.04 LTS	Operating System
CPU	Intel Xeon, 32 vCPU	CPU	Intel Xeon, 32 vCPU	CPU	Intel Xeon, 32 vCPU	CPU	Intel Xeon, 32 vCPU	CPU
GPU	NVIDIA RTX 3090, 24GB	GPU	NVIDIA RTX 3090, 24GB	GPU	NVIDIA RTX 3090, 24GB	GPU	NVIDIA RTX 3090, 24GB	GPU
Memory	128 GB	Memory	128 GB	Memory	128 GB	Memory	128 GB	Memory

Deep Learning Framework	PyTorch 2.3	Deep Learning Framework	PyTorch 2.3	Deep Learning Framework	PyTorch 2.3	Deep Learning Framework	PyTorch 2.3	Deep Learning Framework
Python	3.10	Python	3.10	Python	3.10	Python	3.10	Python
CUDA	11.8	CUDA	11.8	CUDA	11.8	CUDA	11.8	CUDA
Time Slice Granularity	1 min	Time Slice Granularity	1 min	Time Slice Granularity	1 min	Time Slice Granularity	1 min	Time Slice Granularity
Historical Window Length	30	Historical Window Length	30	Historical Window Length	30	Historical Window Length	30	Historical Window Length

4.3 Experimental Results and Analysis

To evaluate the effectiveness of graph time-series anomaly detection methods in scenarios with dynamic evolution of backend dependencies, related works typically conduct horizontal comparisons based on dimensions such as detection discrimination capability and the quality of distinguishing between positive and negative samples. Based on the same data construction and evaluation criteria, Table 3 presents representative comparative items for methods consistent with the research direction of this paper, and reserves corresponding quantitative record positions under a unified evaluation index system.

Table 3. Experimental results compared with other models

Method	AUC-ROC	Macro-F1
Zhang et al.[8]	0.89	0.71
Chen et al.[9]	0.90	0.72
Wang et al.[10]	0.91	0.74
Tang et al.[11]	0.88	0.70
Al-Omari et al.[12]	0.87	0.69

formance of different methods across multiple evaluation metrics, reflecting the stronger stability of the proposed method in terms of discriminative ability and overall quality. This method maintains a more balanced performance across key metrics, ensuring not only high discriminative power in anomaly identification but also better balance between the completeness and reliability of positive class identification.

From the perspective of metric focus, the proposed method maintains a high level of consistency while balancing accuracy and coverage. This performance indicates that the constructed joint structural and temporal modeling can effectively capture key anomaly signals in the dynamic evolution of dependencies and form clearer anomaly boundaries even under complex interactions and noise perturbations, thereby improving the credibility and usability of anomaly judgment.

Combining the performance of the four metrics, the proposed method demonstrates stronger robustness and generalization potential. This advantage means that anomaly scoring is more focused on real risk signals rather than short-term perturbations, resulting in better interpretability and engineering application value of the detection results under different anomaly morphologies and dependency changes.

The 3D manifold projection of dynamic graph temporal embedding is used to intuitively present the evolution of backend dependencies in the temporal dimension. This visualization can compress the high-dimensional representation after structural encoding and temporal aggregation into an interpretable geometric space, thereby observing the degree of separation between normal evolutionary trajectories and anomalous perturbations in the representation space. By performing continuous projection and trajectory rendering on the embedding of a single method, intuitive evidence can be provided for the method's representational ability and anomalous sensitivity, and the experimental results are shown in Figure 2.

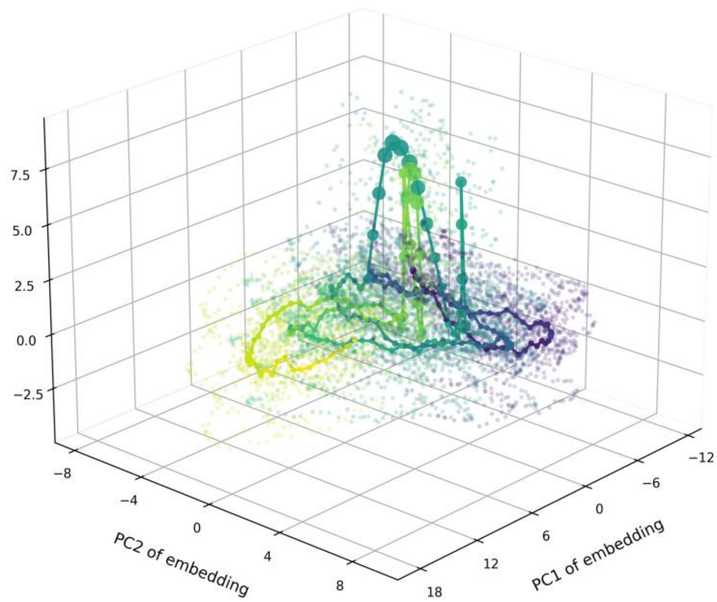

Figure 2. Visualization Experiment of 3D Manifold Projection in Temporal Embedding Space of Dynamic Dependency Graph

The 3D manifold projection reveals an embedded geometry resulting from the interplay of structure and time. The point cloud forms a continuous and interpretable organizational structure in space, demonstrating that the representation can compress the main interaction patterns in dynamic dependencies into a low-dimensional space while maintaining stable geometric consistency. Trajectories advance along clear paths within the point cloud, indicating that the temporally aggregated representation possesses a coherent state-carrying capacity, mapping the natural evolution of the operational state into traceable motion in the visualization space, thus making the overall state changes intuitively readable.

High-amplitude deviations and prominent spatial transitions in local regions reflect the significant impact of anomalous perturbations on the representation space. Anomalous fragments often appear detached from the main structure, forming identifiable outlier trajectory patterns. This phenomenon indicates that the proposed method, while maintaining the integrity of the normal evolutionary structure, has a more sensitive response mechanism to anomalous perturbations, allowing anomalies to still be clearly separated in the low-dimensional projection. Therefore, the proposed method performs well and has strong engineering interpretation value.

The overlay of anomaly response curves from key service subgraphs is used to characterize the temporal response patterns of local structural units to disturbances under dynamic dependencies. This visualization maps the anomaly responses of multiple key subgraphs onto the same time axis, facilitating the observation of the synchronicity and differences of different structural units during the same operational phase. By overlaying and rendering only the anomaly responses output by the proposed method, the anomaly sensitivity expression brought about by structure awareness and temporal aggregation can be intuitively presented. In this process, key subgraphs are quantitatively selected based on node-level anomaly scores. First, the average anomaly score and variance of each service during the training phase are calculated, normalized, and weighted to obtain an importance score. Then, the top 7 services with the highest importance scores are selected as the set of key nodes.

Figure 3 shows that the superimposed curves exhibit strong consistency on the same time axis. The responses of multiple key subgraphs show synchronous rises and falls over several periods, indicating that anomalous disturbances can form a propagable common impact within the dependent structure and be captured by the response curves. The convergence curve stably characterizes the common features of the responses of local subgraphs in its overall shape, making the common fluctuations related to anomalies more concentrated, thereby improving the readability of key time periods.

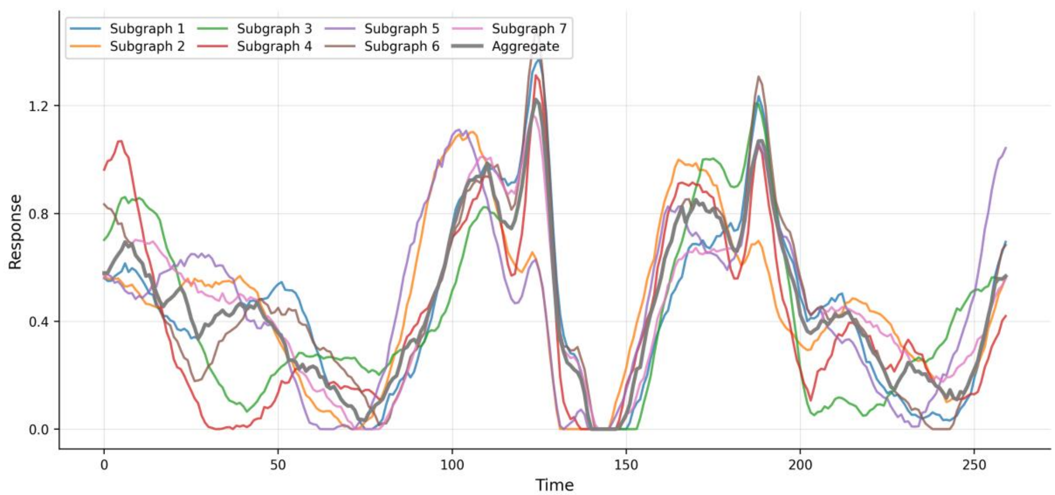

Figure 3. Visualization experiment of overlaying abnormal response curves of critical service subgraphs

Differences in amplitude and recovery speed are still preserved between different subgraphs, reflecting the inconsistent degree of disturbance caused by different dependent locations and structural strengths. Some subgraphs show earlier responses or longer recovery processes. This phenomenon indicates that the proposed method can not only reflect the anomalous impact at the global level but also preserve the individual differences of local structural units, making the expression of the anomaly sensitivity of key service subgraphs clearer. Therefore, the proposed method is effective and has a stronger diagnostic auxiliary value.

5. Conclusion

This paper focuses on graph temporal anomaly detection in scenarios with dynamic evolution of backend dependencies. Addressing key challenges such as frequent changes in microservice call topology, non-stationary drift of runtime attributes, and complex anomaly propagation paths, a unified modeling framework capable of simultaneously characterizing structural interactions and temporal dynamics is constructed. The proposed method starts from the organization of dynamic graph sequences, incorporating service nodes, dependency edges, and multi-source attribute information into a consistent representation space. Through structural encoding and temporal aggregation, a robust characterization of normal evolutionary patterns is formed. This research provides a structured and interpretable technical path for anomaly identification in dynamic dependency environments, helping to alleviate the problems of traditional methods failing under topology changes and misjudging under noise disturbances.

The method design emphasizes the fit to anomaly patterns and engineering usability. Anomalies are no longer merely viewed as local numerical mutations, but rather as a comprehensive manifestation of structural consistency disruption and reduced temporal predictability. Through relation-aware neighborhood aggregation and temporal condition modeling, differences in interaction intensity, changes in critical links, and cascading effects within the system can be incorporated into a unified anomaly metric, thus making the anomaly score more focused on real risk signals and reducing sensitivity to short-term jitter. The resulting anomaly scoring mechanism supports risk alerts at the node level and can further aggregate at the subgraph or time-slice level to adapt to different operation and maintenance management needs, providing an actionable basis for fault early warning, alarm convergence, and rapid location.

This research has a direct impact on cloud computing operation and maintenance, microservice reliability governance, and risk prevention and control of critical business systems. With continuous delivery and elastic scaling becoming the norm, stable monitoring of dynamically dependent networks is a crucial prerequisite for ensuring service quality. Graph time-series anomaly detection can provide more timely signal input for capacity planning, change assessment, and fault isolation. Furthermore, in application areas sensitive to availability and latency, such as financial trading platforms, online recommendation systems, industrial internet, and vehicle networks, anomalies in dependent links often induce cascading degradation or even business interruptions. Anomaly detection capabilities based on joint structural and temporal modeling can improve the efficiency of identifying cascading risks and provide more reliable structural and temporal evidence for automated operation and maintenance decisions, thereby reducing system recovery time and improving overall resilience.

Looking to the future, there are still several directions worthy of in-depth exploration for anomaly detection under dynamic dependencies. Finer-grained continuous-time modeling promises to improve the ability to characterize transient and short-window-level anomalies, making anomaly identification and propagation tracing more closely resemble real-world operations. Stronger consistency constraints on multi-source observations can further alleviate alignment biases in link tracing, metrics, and logs under conditions of collection latency and missing data. In terms of interpretability, elevating anomaly evidence from scores to readable attributions of key links and critical time periods will enhance the usability and credibility of the method in actual operational loops. With the continuous evolution of large-scale cloud-native systems, time-series anomaly detection for dynamic dependency graphs will become a crucial component of intelligent operations and maintenance, playing a long-term role in improving system stability, reducing operational costs, and ensuring the continuity of critical business operations.

References

[1] I. Kohyarnejadfard, D. Aloise, S. V. Azhari and M. R. Dagenais, "Anomaly Detection in Microservice Environments Using Distributed Tracing Data Analysis and NLP," Journal of Cloud Computing, vol. 11, no. 1, p. 25, 2022.

[2] Xie Z, Xu H, Chen W, et al. Unsupervised anomaly detection on microservice traces through graph vae[C]//Proceedings of the ACM Web Conference 2023. 2023: 2874-2884.

[3] Kohyarnejadfard I, Aloise D, Azhari S V, et al. Anomaly detection in microservice environments using distributed tracing data analysis and NLP[J]. Journal of Cloud Computing, 2022, 11(1): 25.

[4] Raeiszadeh M, Ebrahimzadeh A, Saleem A, et al. Real-time anomaly detection using distributed tracing in microservice cloud applications[C]//2023 IEEE 12th International Conference on Cloud Networking (CloudNet). IEEE, 2023: 36-44.

[5] Xie Z, Pei C, Li W, et al. From point-wise to group-wise: A fast and accurate microservice trace anomaly detection approach[C]//Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2023: 1739-1749.

[6] M. Panahandeh, A. Hamou-Lhadj, M. Hamdaqa, and J. Miller, "ServiceAnomaly: An anomaly detection approach in microservices using distributed traces and profiling metrics," Journal of Systems and Software, vol. 209, p. 111917, 2024.

[7] Y. He, X. Zhang, P. Wang, M. Liu, and Z. Cao, "Microservice traces anomaly detection based on structural clustering and graph attention autoencoder," in Proc. 7th Int. Conf. Advanced Algorithms and Control Engineering (ICAACE), Mar. 2024, pp. 977-981.

[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] S. Zhang, Y. Zhao, S. Xia, S. Wei, Y. Sun, C. Zhao, et al., "No more data silos: Unified microservice failure diagnosis with temporal knowledge graph," IEEE Transactions on Services Computing, vol. 17, no. 6, pp. 4013-4026, 2024.

[11] 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.

[12] L. Tao, S. Zhang, Z. Jia, J. Sun, M. Ma, Z. Li, et al., "Giving every modality a voice in microservice failure diagnosis via multimodal adaptive optimization," in Proc. 39th IEEE/ACM Int. Conf. Automated Software Engineering (ASE), Oct. 2024, pp. 1107-1119.

[13] M. Allam, N. Boujnah, N. E. O'Connor, and M. Liu, "Synthetic time series for anomaly detection in cloud microservices," in Proc. Int. Conf. Machine Learning, Optimization, and Data Science, Sep. 2024, pp. 419-433.

[14] M. Naderi, H. Momeni, and S. Shahini, "A graph attention-based autoencoder for critical path anomaly detection in microservices," in Proc. 15th Int. Conf. Information and Knowledge Technology (IKT), Dec. 2024, pp. 19-25.