

AI-driven fault diagnosis from textual system logs

Áron Kiss, Károly Nehéz, Olivér Hornyák

University of Miskolc
aron.kiss@uni-miskolc.hu
karoly.nehez@uni-miskolc.hu
oliver.hornyak@uni-miskolc.hu

Abstract. The increasing complexity and scale of microservice-based systems pose major challenges for ensuring reliability and operational continuity. Multimodal fault diagnosis integrating logs, metrics, and traces has emerged as a key approach for improving anomaly detection, failure type identification, and root cause localization. Graph Neural Networks (GNNs) show strong potential for modeling intricate service dependencies and fault propagation patterns in such systems. This study presents a systematic review of state-of-the-art graph-based multimodal diagnostic frameworks. We compare existing methods in terms of diagnostic accuracy, scalability, computational cost, and implementation complexity, and analyze representative public datasets and benchmark systems. We highlight key challenges, including generalization, explainability, online applicability, and outline promising directions for future research. In addition, we report preliminary findings from our own experiments, which suggest that Transformer-based models provide a promising foundation for multimodal fault diagnosis in enterprise microservice systems. These early results motivate our ongoing work toward hybrid architectures that combine the strengths of Transformers and GNNs.

Keywords: microservice system, fault diagnosis, anomaly detection, root-cause localization, Graph Neural Network, Transformer

AMS Subject Classification: Primary: 68T05, Secondary: 68T30

1. Introduction

The microservice architecture is increasingly prevalent in modern information systems. However, ensuring reliability in such environments – composed of numerous

interdependent services – remains challenging, as faults can propagate in complex ways [13]. To monitor system health, logs, performance metrics, and traces are typically collected. Different anomalies may appear in different modalities: a failure may cause only a metric spike or manifest as log entries indicating a crash [29].

Multimodal fault diagnosis is thus essential and has gained growing attention in recent years [7, 30]. Recent studies focus on combining heterogeneous telemetry data to improve fault detection and root cause analysis.

Graph-based methods are particularly promising [7, 11, 30], since microservices form natural graphs, with nodes as service instances and edges denoting dependencies. Graph Neural Networks (GNNs) leverage this structure to model system-wide interactions [12, 23, 26, 29, 31, 32].

This paper reviews multimodal fault diagnosis methods using GNNs. We compare approaches that integrate log, metric, and trace data across multiple dimensions such as accuracy, scalability, and complexity. We also examine common datasets and benchmarks, and identify research gaps.

In addition, we present preliminary findings from our own work. We first analyze log anomaly detection, showing the benefit of Transformer architectures, then introduce a Graph Transformer-based model for enhanced root cause localization and failure type identification.

2. Methodology

The studies reviewed in this paper were identified using keyword-based searches on platforms of Google Scholar and Semantic Scholar. To ensure comprehensive coverage, we used a diverse set of keywords, including but not limited to: “multimodal fault diagnosis,” “microservice anomaly detection,” “graph neural network,” “data fusion,” “root cause analysis,” and “AIOps benchmark.” The keywords were combined with Boolean operators and adapted iteratively during the search process. Initial selections were filtered based on relevance, citation impact, and publication recency, followed by citation chaining. Although the process was exploratory, we aimed to ensure broad and representative coverage of methodologies and key contributions across both survey and empirical research papers.

3. Multimodal fault diagnosis in microservice systems

Artificial Intelligence for IT Operations (AIOps) aims to enhance IT systems using AI techniques. Despite advancements, challenges in adaptivity, efficiency, and scalability remain, partly due to the lack of standardized taxonomies for data handling, target tasks, and system requirements [13]. This complicates the comparison and integration of diagnostic approaches.

Early fault diagnosis methods in microservice systems typically relied on a single telemetry source, which proved insufficient. Multimodal fault diagnosis addresses

this by jointly analyzing logs, metrics, and traces, capturing complementary system perspectives [12, 29].

For example, a complex outage may manifest through spikes in CPU usage, increased log severity, and timeout traces. Analyzing only one modality could lead to missed or misattributed faults.

The most commonly used data types are:

- **Logs:** Semi-structured textual entries containing system and application events [26]. Log-based methods rely on parsing and pattern mining [20].
- **Metrics:** Time-series data capturing quantitative performance indicators (e.g., CPU usage, request counts) [32]. Diagnostics often involve anomaly detection on time series [20].
- **Traces:** Sequences of spans representing end-to-end request paths. Useful for identifying bottlenecks and fault propagation [30].

Key diagnostic tasks include:

- **Anomaly Detection (AD):** Binary classification to detect abnormal system behavior.
- **Failure Type Identification (FTI):** Multi-class classification to determine the nature of the fault.
- **Root Cause Localization (RCL):** Identifying the component responsible for the fault, often ranked by likelihood; advanced methods also reconstruct propagation paths.

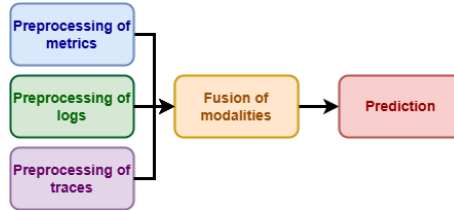


Figure 1. Multimodal fault diagnosis pipeline.

As shown in Figure 1, the diagnosis process typically involves: (a) preprocessing and unifying telemetry data; (b) fusing multimodal features into a model input; and (c) applying predictive models to perform AD, FTI, and RCL.

Modern approaches utilize deep learning to model complex system dynamics. Early works like DeepLog [5] applied sequential modeling to log data. Similarly, LSTM-based models and statistical methods have been used for metric anomaly detection. However, unimodal analysis often lacks sufficient insight, prompting recent methods to integrate all three modalities [26, 29], yielding improved FTI and RCL performance.

4. Graph Neural Networks in fault diagnosis of microservice systems

Graph Neural Networks (GNNs) have become a key trend in fault analysis of microservice systems. These deep learning models are designed to process graph-structured data, making them ideal for modeling microservices, where nodes represent instances and edges denote service dependencies. GNNs can capture both topological and contextual information using techniques like message passing and attention [8, 9, 27].

In multimodal diagnosis, GNNs are used to fuse heterogeneous telemetry data or to process node-level features within a system graph.

Several recent models adopt similar GNN-based architectures [7], differing mainly in the graph encoders and supported diagnostic tasks. Table 1 summarizes their key characteristics.

Table 1. Overview of GNN-based multimodal fault diagnosis models. M, L, T are metrics, logs, and traces, respectively.

Model	Year	Modalities	Graph Encoder	AD	FTI	RCL
Eadro [12]	2023	M,L,T	GAT [2]	✓	✗	✓
DiagFusion [29]	2023	M,L,T	TAGConv [4]	✓	✓	✓
UniDiag [31]	2024	M,L,T	R-GCN [22]	✓	✓	✗
DeepHunt [23]	2024	M,L,T	GraphSAGE [8]	✓	✗	✓
TVDiag [26]	2025	M,L,T	GraphSAGE [8]	✓	✓	✓
CHASE [32]	2025	M,L,T	HGT [9]	✓	✗	✓

Eadro [12] uses GAT over fused multimodal inputs, but its complexity may cause overfitting on small systems, where simpler models sometimes perform better [7].

DiagFusion [29] embeds alerts using fastText and fuses features via a multi-part neural architecture. However, ignoring modality-specific traits may harm performance when data is incomplete [24].

UniDiag [31] builds temporal knowledge graphs and uses R-GCN for efficient stream-based diagnosis, introducing a microservice-specific embedding strategy.

TVDiag [26] avoids uniform fusion by using task-specific GNN encoders and contrastive learning, improving robustness via graph augmentation.

DeepHunt [23] introduces Root Cause Score (RCS), combining reconstruction error and propagation patterns. It also supports adaptive refinement via feedback.

CHASE [32] employs a causal hypergraph framework with modality weighting and inference to identify root causes. While accurate, its resource demands may limit scalability.

In summary, GNN-based models effectively integrate multimodal data and capture complex patterns. Still, simpler models with well-designed features can match

or exceed GNN performance in constrained settings [7].

5. Public datasets and benchmark environments

A major challenge in fault diagnosis research is the scarcity of large-scale, publicly available datasets from real-world microservice systems [13]. Consequently, many studies rely on proprietary data or synthetic environments. In recent years, several datasets have been released to support model development and evaluation.

Table 2 summarizes widely used datasets for FTI and RCL tasks, including both single and multimodal sources.

Table 2. Overview of widely used, openly available information system anomaly datasets. M, L, and T are metrics, logs, and traces, respectively.

Name	Year	Modalities	Description
AIOps Challenge 2020 [16]	2020	M, T	Data about a real-world production microservice system, collected by China Mobile Zhejiang.
AIOps Challenge 2021 [1]	2021	M, L, T	Data about two large commercial banking systems, collected by Tsinghua University.
Generic AIOps Atlas (GAIA) [3]	2022	M, L, T	Detailed dataset with simulated anomalies and injection records for root cause analysis evaluation, collected by CloudWise.
Loghub [33]	2023	L	Large volume of logs from various systems.
RCAEval-RE1 [17, 18]	2024	M	Metric-only failure cases from three microservice systems, supporting metric-based root cause analysis.
RCAEval-RE2, RCAEval-RE3 [18]	2025	M, L, T	Multi-source telemetry data for root cause analysis, with RE2 covering general failure cases and RE3 focusing on code-level fault cases across three microservice systems.

These resources facilitate benchmarking and comparison of diagnostic models. However, most datasets involve synthetic fault injection, which may fail to capture the complexity and spontaneity of real production failures. As a result, models trained exclusively on these may face generalization issues in large-scale industrial settings.

In addition to datasets, open-source benchmark systems have been developed to emulate microservice environments with integrated telemetry tools (e.g., OpenTelemetry, Prometheus, Jaeger, Loki). Table 3 lists commonly used examples.

Several studies inject faults into these systems and monitor the telemetry to assess model performance [12, 26]. While effective for controlled experimentation, their scale – typically 8–12 services – is far from enterprise-grade systems. For context, Netflix operated over 200 microservices by 2016 [6], while Uber reported 4,500 by 2023 [25].

Table 3. Overview of widely used microservice benchmark systems.

Name	Release	Microservices	URL
SockShop	2016	8	https://github.com/microservices-demo/microservices-demo
Train Ticket	2018	41	https://github.com/FudanSELab/train-ticket
Hotel Reservation	2019	10	https://github.com/delimitrou/DeathStarBench/tree/master/hotelReservation
Social Network	2019	12	https://github.com/delimitrou/DeathStarBench/tree/master/socialNetwork
Online Boutique	2020	12	https://github.com/GoogleCloudPlatform/microservices-demo

6. Comparison of existing methods

Evaluating multimodal fault diagnosis methods requires careful consideration of several criteria. In this section, we summarize the most important dimensions for comparison, with a particular focus on detection accuracy, computational cost and scalability, real-time applicability, and implementation complexity. We also discuss how recent models perform with respect to each of these aspects.

6.1. Detection and localization accuracy

In general, multimodal GNN-based models tend to achieve the improved reported accuracy, compared to single modal approaches in published studies. A recent evaluation [7] showed that state-of-the-art models such as Eadro, DiagFusion, DeepHunt, TVDdiag, and CHASE achieved top-1 hit rates between 30% and 60% on the GAIA dataset, while their top-3 hit rates ranged from approximately 59% to 88%. Precision is another critical metric, particularly when assessing false positives. Multimodal models tend to outperform single-modality detectors in this regard, as they require consistent signals across multiple modalities to trigger an alert – resulting in fewer spurious detections.

However, it is important to note that the advantage of multimodal GNNs in accuracy may be highly dependent on the dataset and fault types. Some studies suggest that in certain scenarios, this advantage may not be statistically significant [7].

6.2. Computational cost

Graph-based deep learning models are generally more computationally expensive than simpler architectures such as MLPs or LSTMs. This is due to the complexity of GNN layers (e.g., graph convolution or attention operations that involve neighborhood aggregation with time complexity $O(V + E)$ depending on the graph structure), and the additional overhead introduced by multimodal data preprocessing (e.g., log parsing, metric filtering, dimensionality reduction). A key scalability concern is that most GNN-based diagnostic models have not been tested in online setups, and on systems with hundreds of service instances.

Although current datasets are manageable with GNNs, it remains unclear whether these methods can scale effectively to larger environments. Future progress may require more scalable architectures, as the computational cost of running complex multimodal models may be prohibitive in smaller organizational settings lacking adequate infrastructure.

6.3. Real-time applicability

A practical requirement for fault diagnosis methods is their ability to operate in real time or near real time. Most published multimodal models are trained offline and applied in a near-real-time setting, where the model processes telemetry aggregated over a fixed interval (e.g., every second or minute) and raises alerts as needed. Offline training is not inherently problematic if model updates are infrequent. However, microservice environments are highly dynamic, raising the need for continual learning or online inference capabilities. Real-time applicability is further constrained by the latency of preprocessing steps such as log parsing, metric filtering, and trace clustering. Stream-based tools like *Drain3* for logs and sliding-window aggregation for metrics are essential in online systems.

There is broad consensus in the literature that current deep learning models require further optimization, such as model pruning or knowledge distillation to be truly viable in real-time AIOps settings. For now, multimodal GNN-based models are best suited for near real time detection.

6.4. Engineering complexity

Implementing multimodal GNN-based diagnostic systems demands significant engineering effort. Key challenges include handling and synchronizing three different telemetry streams (logs, metrics, and traces), cleaning and storing diverse data sources, and integrating multiple specialized algorithms (e.g. NLP for log analysis, time-series modeling for metrics, and graph processing for traces). In many implementations, the diagnostic pipeline comprises multiple interdependent components that must be individually tuned [26, 29]. For instance, DiagFusion and TVDiag include a full preprocessing framework before the neural network model begins learning. This involves filtering logs, normalizing metrics, and deriving service dependency graphs from trace data.

GNN models typically require dedicated libraries such as *PyTorch Geometric* or *Deep Graph Library*, which introduce a steeper learning curve for operations teams. Moreover, debugging end-to-end pipelines can be difficult: when the diagnosis fails, it is not always clear whether the error occurred during log parsing, feature fusion, or within the GNN itself. This may prevent the utilization of such methods for small-medium sized or non-technology-focused organizations with less AI expertise.

In conclusion, there is no single best method that dominates across all dimensions. GNN-based deep learning models offer high accuracy and effectively leverage multimodal data. However, they come with considerable resource demands, scalability limitations, and operation complexity.

7. Limitations and open challenges

Despite recent progress in multimodal fault diagnosis for microservice systems, several important challenges and limitations remain. These arise both from the limitations of current methods and the shortcomings of existing datasets and benchmarks. Below, we highlight the most critical issues identified.

7.1. Generalizability

Most published models have been evaluated on relatively small and simple fault scenarios. It is common for studies to assess their methods on 1-2 microservice benchmarks with just 2-5 injected fault types [12, 29, 31]. As a result, the generalizability of these models to real-world systems remains unclear. In production environments, dozens of distinct root causes may exist, including hardware failures, code-level bugs, and network anomalies, many of which are not represented in the training data. Moreover, operational environments are constantly evolving

due to factors of sudden peaks in user demand, ongoing software development, and the introduction of new hardware components, further complicating fault diagnosis and model robustness.

For example, a GNN model tuned primarily for detecting CPU spikes may completely miss a database deadlock scenario if neither the metrics nor the log anomaly detection modules provide a clear signal. Overfitting to the limited fault cases present in the training set is a common risk – as demonstrated by Eadro, which performed well on trained scenarios but underperformed in unseen environments [7]. Current deep learning models lack proven generalization capabilities, and there is a strong need for larger, more diverse benchmarks that reflect real production complexity.

7.2. Lack of real-world data

Although public datasets exist, many studies still rely on synthetic fault injection. While such datasets can approximate real-world conditions, they are inherently artificial. In actual operations, failures often arise from overlapping and interdependent causes, such as a minor memory leak combined with a configuration change, which are difficult to reproduce under controlled lab conditions. This limits the external validity of many evaluation results.

A major reason for the lack of real-world data is the sensitivity of production environments. Sharing logs and failure cases may pose risks related to business confidentiality, intellectual property, or security. However, it is also in the interest of organizations that researchers develop methods targeting real operational challenges. Without access to realistic data, proposed models may remain overfitted to synthetic scenarios and fail to generalize to production-scale systems.

Furthermore, synthetic datasets typically include ground truth labels for the faulty component, allowing models to be trained to recognize known faults rather than to discover novel root causes in unfamiliar environments. This introduces potential bias. To truly evaluate and improve fault diagnosis systems, publicly available datasets based on real-world incidents and complex causal chains are needed.

7.3. Interpretability and explainability

Deep learning models, particularly GNNs, often lack transparent decision-making mechanisms. In operational contexts, it is critical for human operators to understand why a service was flagged as faulty. Although recent studies have begun to address this issue, most state-of-the-art models still fall short in terms of explainability.

For instance, a GNN might output *Node X is anomalous (confidence = 0.9)*, but this information is of limited practical use, unless the system can also indicate which features contributed to this decision. Addressing this challenge requires either post hoc interpretation techniques (e.g., feature attribution or counterfactual analysis), or the integration of inherently interpretable components into the

model architecture. While attention mechanisms have been used in some models to provide a degree of transparency, their alignment with human-understandable explanations remains limited and context-dependent. Methods such as SHAP [15], LIME [21], and GNNExplainer [28] have been proposed to improve model interpretability by identifying the most influential features or substructures contributing to a prediction.

7.4. Scalability

Most current methods have not been tested on industrial-scale systems with at least 50–100 microservices. GNN complexity typically grows linearly with the number of nodes and edges, but can become quadratic with operations such as self-attention over large node sets. In large-scale environments, the volume of telemetry data and the complexity of service interactions may significantly impact inference time and memory usage, raising concerns about real-time applicability and deployment feasibility.

Furthermore, large microservice systems are likely to experience multiple concurrent faults, which may be independent or interacting. Supporting multi-fault scenarios increases the diagnostic complexity of models, as they must disambiguate overlapping symptoms and attribute them to distinct root causes. This additional burden further challenges the scalability of current approaches and highlights the need for more robust, efficient, and fault-tolerant diagnostic architectures.

7.5. Preprocessing and pipeline fragility

Multimodal diagnostic pipelines depend on complex, often computationally intensive preprocessing steps, such as log parsing, metric normalization, trace correlation, and graph construction. These can strongly influence downstream model performance. The diversity of current approaches has led to heterogeneous workflows, complicating reproducibility and fair comparison.

Moreover, errors or inconsistencies during preprocessing can severely degrade diagnostic accuracy. For instance, log parsing failures may cause misclassification, while incomplete traces from sampling or missing instrumentation lead to fragmented service graphs. Noisy or outlier-heavy metric series can distort model behavior and degrade learned representations.

Current models often lack robustness to such imperfections. Beyond improving resilience via adaptive preprocessing or uncertainty-aware learning, there is a clear need for standardization. Common data formats and pipelines would support more meaningful comparisons and enable sharing of preprocessed datasets alongside raw telemetry. This would lower the entry barrier for new researchers and promote transparency and reproducibility.

8. Preliminary explorations to address challenges

To assess how different AI models perform in detecting faults based on event sequences, we conducted an initial comparative evaluation of machine learning approaches for log anomaly detection. Our focus was on event sequence-based methods that transform logs into sequences of event template identifiers and classify them as either normal or anomalous. This process is depicted in Figure 2.

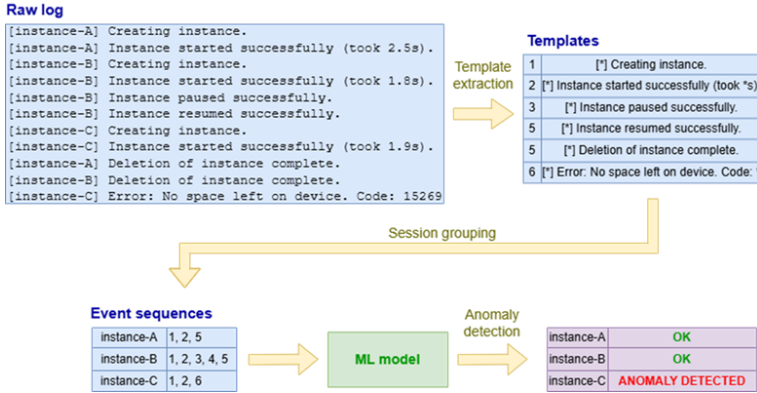


Figure 2. Process of event sequence based log anomaly detection.

Using the HDFS dataset [33], we benchmarked four models: the original LSTM-based DeepLog implementation [5], as well as our own variations based on a bi-directional LSTM, a Random Forest classifier, and a Transformer encoder.

In our experiments, the Transformer model outperformed the LSTM baseline, Bi-LSTM, and Random Forest in terms of overall accuracy, precision, and F1-score. Its ability to capture long-range dependencies resulted in $\sim 6\%$ gain in F1-score over the LSTM and a $\sim 25\%$ gain over the Random Forest. Notably, it also achieved about 11% relative improvement over the baseline LSTM model in precision. This indicates the model’s superior ability to reduce false positives while maintaining high recall. These findings suggest that Transformer-based architectures are a promising choice for anomaly detection, as they effectively capture dependencies in event sequences. Motivated by these results, we began extending this approach to a multimodal diagnostic setting by integrating logs, metrics, and traces into a unified model.

As a first step toward this goal, we developed a prototype system that leverages Transformer-based graph encoders to integrate logs, metrics, and traces for fault diagnosis in microservice systems. Our design choices were informed by key limitations identified in the literature, including the lack of robust multimodal fusion strategies, the limited scalability of conventional GNNs in large microservice architectures, and the absence of interpretability mechanisms suitable for operational use.

Our method builds on the TVDiag framework [26], replacing its GraphSAGE-based encoder with the Graphormer architecture [27], a Transformer model adapted for graphs. As illustrated in Figure 3, telemetry anomalies are detected using tailored strategies: the 3σ method [19] for metrics, Isolation Forest [14] for traces, and pattern-based filtering for logs. Alerts are embedded using fastText [10] and mapped to nodes in a graph representing service instances. Edges are formed based on runtime communication patterns between the instances.

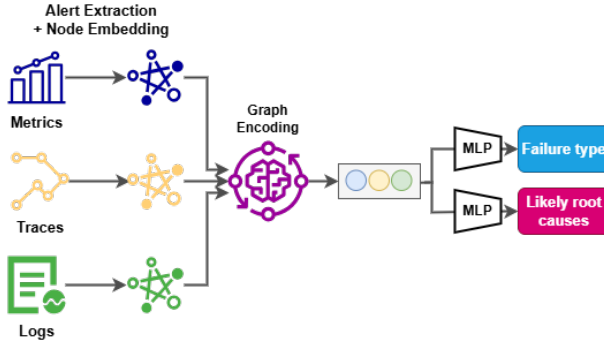


Figure 3. Overview of our prototype framework for multimodal fault diagnosis.

To overcome the lack of inherent node ordering in graphs, we adopted encoding strategies introduced by [27]. Node centrality – computed from node degrees – is used to highlight structurally important service instances, which are more likely to play critical role in failures. Additionally, spatial encoding captures shortest-path distances between nodes, allowing the model to account for fault propagation patterns within the service graph. Together, these encodings offer a richer structural context that can support more informed fault reasoning.

The encoded graph is passed through Graphormer layers, followed by two MLP heads performing FTI and RCL. We performed preliminary evaluation using the Generic AIOps Atlas dataset, which features fault scenarios in a QR-based access control system. Compared to the baseline TVDiag model, our method yielded 3–6% higher RCL top-1 to top-3 hit rate and approximately 5% improvement in F1-score for FTI, while reducing inference time by up to 80%, thanks to the model’s parallelizable architecture.

Although further validation is needed, results show that combining Transformer-based graph representation learning with multimodal data can address certain key challenges in fault diagnosis. In particular, it improves both diagnostic accuracy and runtime efficiency, while providing interpretable features like centrality and spatial locality that can assist system administrators. Our work is still in an early stage and should not be interpreted as a complete solution. Instead, it offers encouraging evidence to justify deeper investigation. Future work will focus on deploying the model in dynamic, real-time settings and addressing issues such as data imbal-

ance and preprocessing overhead. Understanding how the model performs under online conditions and dynamically evolving service topologies will be essential for evaluating its practical value in production systems.

9. Conclusion

In recent years, multimodal fault diagnosis in microservice systems has made significant progress, especially in combining logs, metrics, and traces to improve fault detection and root cause localization. GNNs have shown strong potential for modeling fault propagation across complex service topologies, offering notable gains in detection accuracy and root cause localization.

Despite these benefits, GNNs demand significant computational resources and are highly sensitive to data quality. Interestingly, simpler models, such as multilayer perceptrons can also be competitive when paired with well-designed preprocessing and feature engineering.

Current research indicates that no single model consistently outperforms others across all scenarios. Choosing the right approach involves trade-offs between accuracy, scalability, automation, and interpretability. In industrial settings, GNN-based models are still rarely deployed for real-time fault detection. A promising strategy is to use hybrid architectures: lightweight models for routine monitoring, complemented by GNN-based diagnostics for critical or ambiguous cases.

Achieving practical breakthroughs will require greater focus on addressing real-world constraints, such as data availability, benchmark realism, and model transparency, rather than solely increasing model complexity. Our ongoing research, starting from single-modal Transformer-based log anomaly detection and advancing toward a graph-based multimodal framework, provides early evidence that appropriately adapted Transformer models may offer substantial benefits. However, thorough experimentation and validation remain necessary to evaluate the robustness and generalizability of this approach.

References

- [1] AIOps-NANKAI: *AIOps2021*, 2021, URL: <https://www.aiops.cn/gitlab/aiops-nankai/data/trace/aiops2021/-/tree/main> (visited on 06/15/2025).
- [2] S. BRODY, U. ALON, E. YAHAV: *How Attentive are Graph Attention Networks?*, 2022, arXiv: [2105.14491](https://arxiv.org/abs/2105.14491) [cs.LG], URL: <https://arxiv.org/abs/2105.14491>.
- [3] CLOUDWISE: *Generic AIOps Atlas*, 2022, URL: <https://github.com/CloudWise-OpenSource/GAIA-DataSet> (visited on 06/15/2025).
- [4] J. DU, S. ZHANG, G. WU, J. M. F. MOURA, S. KAR: *Topology Adaptive Graph Convolutional Networks*, 2018, arXiv: [1710.10370](https://arxiv.org/abs/1710.10370) [cs.LG], URL: <https://arxiv.org/abs/1710.10370>.
- [5] M. DU, F. LI, G. ZHENG, V. SRIKUMAR: *DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning*, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17, ACM, Oct. 2017, pp. 1285–1298, DOI: [10.1145/3133956.3134015](https://doi.org/10.1145/3133956.3134015), URL: <http://dx.doi.org/10.1145/3133956.3134015>.

- [6] Y. GAN, Y. ZHANG, D. CHENG, A. SHETTY, P. RATHI, N. KATARKI, A. BRUNO, J. HU, B. RITCHKEN, B. JACKSON, K. HU, M. PANCHOLI, Y. HE, B. CLANCY, C. COLEN, F. WEN, C. LEUNG, S. WANG, L. ZARUVINSKY, M. ESPINOSA, R. LIN, Z. LIU, J. PADILLA, C. DELIMITROU: *An Open-Source Benchmark Suite for Microservices and Their Hardware-Software Implications for Cloud - Edge Systems*, in: Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '19, ACM, Apr. 2019, pp. 3–18, doi: [10.1145/3297858.3304013](https://doi.org/10.1145/3297858.3304013), URL: <http://dx.doi.org/10.1145/3297858.3304013>.
- [7] F. GAO, R. XIN, X. LI, Y. ZHANG: *Are GNNs Actually Effective for Multimodal Fault Diagnosis in Microservice Systems?*, 2025, doi: [10.48550/ARXIV.2501.02766](https://doi.org/10.48550/ARXIV.2501.02766), URL: <https://arxiv.org/abs/2501.02766>.
- [8] W. L. HAMILTON, R. YING, J. LESKOVEC: *Inductive Representation Learning on Large Graphs*, 2018, arXiv: [1706.02216](https://arxiv.org/abs/1706.02216) [cs.SI], URL: <https://arxiv.org/abs/1706.02216>.
- [9] Z. HU, Y. DONG, K. WANG, Y. SUN: *Heterogeneous Graph Transformer*, in: Proceedings of The Web Conference 2020, WWW '20, ACM, Apr. 2020, pp. 2704–2710, doi: [10.1145/3366423.3380027](https://doi.org/10.1145/3366423.3380027), URL: <http://dx.doi.org/10.1145/3366423.3380027>.
- [10] A. JOULIN, E. GRAVE, P. BOJANOWSKI, T. MIKOLOV: *Bag of Tricks for Efficient Text Classification*, 2016, doi: [10.48550/ARXIV.1607.01759](https://doi.org/10.48550/ARXIV.1607.01759), URL: <https://arxiv.org/abs/1607.01759>.
- [11] L. KOVÁCS, E. BAKSÁNE VARGA, P. MILEFF: *Prediction of Complex Event Graphs with Neural Networks*, Computing and Informatics 43.1 (2024), pp. 181–212, ISSN: 2585-8807, doi: [10.31577/cai_2024_1_181](https://doi.org/10.31577/cai_2024_1_181), URL: http://dx.doi.org/10.31577/cai_2024_1_181.
- [12] C. LEE, T. YANG, Z. CHEN, Y. SU, M. R. LYU: *Eadro: An End-to-End Troubleshooting Framework for Microservices on Multi-source Data*, 2023, doi: [10.48550/ARXIV.2302.05092](https://doi.org/10.48550/ARXIV.2302.05092), URL: <https://arxiv.org/abs/2302.05092>.
- [13] Z. LI, N. ZHAO, S. ZHANG, Y. SUN, P. CHEN, X. WEN, M. MA, D. PEI: *Constructing Large-Scale Real-World Benchmark Datasets for AIOps*, 2022, doi: [10.48550/ARXIV.2208.03938](https://doi.org/10.48550/ARXIV.2208.03938), URL: <https://arxiv.org/abs/2208.03938>.
- [14] F. T. LIU, K. M. TING, Z.-H. ZHOU: *Isolation Forest*, in: 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 413–422, doi: [10.1109/ICDM.2008.17](https://doi.org/10.1109/ICDM.2008.17).
- [15] S. M. LUNDBERG, S.-I. LEE: *A Unified Approach to Interpreting Model Predictions*, in: Advances in Neural Information Processing Systems 30, ed. by I. GUYON, U. V. LUXBURG, S. BENGIO, H. WALLACH, R. FERGUS, S. VISHWANATHAN, R. GARNETT, Curran Associates, Inc., 2017, pp. 4765–4774, URL: <http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf>.
- [16] NETMANAIOps: *AIOps-Challenge-2020-Data*, 2020, URL: <https://github.com/NetManAIOps/AIOps-Challenge-2020-Data> (visited on 06/15/2025).
- [17] L. PHAM, H. HA, H. ZHANG: *Root Cause Analysis for Microservice System based on Causal Inference: How Far Are We?*, in: Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering, ASE '24, ACM, Oct. 2024, pp. 706–715, doi: [10.1145/3691620.3695065](https://doi.org/10.1145/3691620.3695065), URL: <http://dx.doi.org/10.1145/3691620.3695065>.
- [18] L. PHAM, H. ZHANG, H. HA, F. SALIM, X. ZHANG: *RCAEval: A Benchmark for Root Cause Analysis of Microservice Systems with Telemetry Data*, in: Companion Proceedings of the ACM on Web Conference 2025, WWW '25, ACM, May 2025, pp. 777–780, doi: [10.1145/3701716.3715290](https://doi.org/10.1145/3701716.3715290), URL: <http://dx.doi.org/10.1145/3701716.3715290>.
- [19] F. PUKELSHEIM: *The three sigma rule*, The American Statistician 48.2 (1994), pp. 88–91, URL: <https://d-nb.info/1186632208/34>.
- [20] Y. REMIL, A. BENDIMERAD, R. MATHONAT, M. KAYTOUE: *AIOps Solutions for Incident Management: Technical Guidelines and A Comprehensive Literature Review*, 2024, arXiv: [2404.01363](https://arxiv.org/abs/2404.01363) [cs.OS], URL: <https://arxiv.org/abs/2404.01363>.

- [21] M. T. RIBEIRO, S. SINGH, C. GUESTRIN: “Why Should I Trust You?”: Explaining the Predictions of Any Classifier, 2016, arXiv: [1602.04938 \[cs.LG\]](#), URL: <https://arxiv.org/abs/1602.04938>.
- [22] M. SCHLICHTKRULL, T. N. KIPF, P. BLOEM, R. VAN DEN BERG, I. TITOV, M. WELLING: *Modeling Relational Data with Graph Convolutional Networks*, in: The Semantic Web, ed. by A. GANGEMI, R. NAVIGLI, M.-E. VIDAL, P. HITZLER, R. TRONCY, L. HOLLINK, A. TORDAI, M. ALAM, Cham: Springer International Publishing, 2018, pp. 593–607, ISBN: 978-3-319-93417-4.
- [23] Y. SUN, Z. LIN, B. SHI, S. ZHANG, S. MA, P. JIN, Z. ZHONG, L. PAN, Y. GUO, D. PEI: *Interpretable Failure Localization for Microservice Systems Based on Graph Autoencoder*, ACM Transactions on Software Engineering and Methodology 34.2 (Jan. 2025), pp. 1–28, ISSN: 1557-7392, DOI: [10.1145/3695999](#), URL: <http://dx.doi.org/10.1145/3695999>.
- [24] L. TAO, S. ZHANG, Z. JIA, J. SUN, M. MA, Z. LI, Y. SUN, C. YANG, Y. ZHANG, D. PEI: *Giving Every Modality a Voice in Microservice Failure Diagnosis via Multimodal Adaptive Optimization*, in: Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering, ASE ’24, ACM, Oct. 2024, pp. 1107–1119, DOI: [10.1145/3691620.3695489](#), URL: <http://dx.doi.org/10.1145/3691620.3695489>.
- [25] UBER ENGINEERING: *Up: Portable Microservices Ready for the Cloud*, Uber blog post, Sept. 2023, URL: <https://www.uber.com/en-HU/blog/up-portable-microservices-ready-for-the-cloud/> (visited on 06/16/2025).
- [26] S. XIE, J. WANG, H. HE, Z. WANG, Y. ZHAO, N. ZHANG, B. LI: *TVDiag: A Task-oriented and View-invariant Failure Diagnosis Framework with Multimodal Data*, 2024, arXiv: [2407.19711v1 \[cs.SE\]](#), URL: <https://arxiv.org/abs/2407.19711v1>.
- [27] C. YING, T. CAI, S. LUO, S. ZHENG, G. KE, D. HE, Y. SHEN, T.-Y. LIU: *Do Transformers Really Perform Bad for Graph Representation?*, 2021, arXiv: [2106.05234 \[cs.LG\]](#), URL: <https://arxiv.org/abs/2106.05234>.
- [28] R. YING, D. BOURGEOIS, J. YOU, M. ZITNIK, J. LESKOVEC: *GNNEExplainer: Generating Explanations for Graph Neural Networks*, 2019, arXiv: [1903.03894 \[cs.LG\]](#), URL: <https://arxiv.org/abs/1903.03894>.
- [29] S. ZHANG, P. JIN, Z. LIN, Y. SUN, B. ZHANG, S. XIA, Z. LI, Z. ZHONG, M. MA, W. JIN, D. ZHANG, Z. ZHU, D. PEI: *Robust Failure Diagnosis of Microservice System through Multimodal Data*, 2023, DOI: [10.48550/ARXIV.2302.10512](#), URL: <https://arxiv.org/abs/2302.10512>.
- [30] S. ZHANG, S. XIA, W. FAN, B. SHI, X. XIONG, Z. ZHONG, M. MA, Y. SUN, D. PEI: *Failure Diagnosis in Microservice Systems: A Comprehensive Survey and Analysis*, 2024, DOI: [10.48550/ARXIV.2407.01710](#), URL: <https://arxiv.org/abs/2407.01710>.
- [31] S. ZHANG, Y. ZHAO, S. XIA, S. WEI, Y. SUN, C. ZHAO, S. MA, J. KUANG, B. ZHU, L. PAN, Y. GUO, D. PEI: *No More Data Silos: Unified Microservice Failure Diagnosis With Temporal Knowledge Graph*, IEEE Transactions on Services Computing 17.6 (Nov. 2024), pp. 4013–4026, ISSN: 2372-0204, DOI: [10.1109/tsc.2024.3489444](#), URL: <http://dx.doi.org/10.1109/TSC.2024.3489444>.
- [32] Z. ZHAO, Z. WANG, T. ZHANG, Z. SHEN, H. DONG, Z. LEI, X. MA, G. XU, Z. DING, Y. YANG: *CHASE: A Causal Hypergraph based Framework for Root Cause Analysis in Multimodal Microservice Systems*, 2025, arXiv: [2406.19711 \[cs.LG\]](#), URL: <https://arxiv.org/abs/2406.19711>.
- [33] J. ZHU, S. HE, P. HE, J. LIU, M. R. LYU: *Loghub: A Large Collection of System Log Datasets for AI-driven Log Analytics*, in: 2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE), IEEE, Oct. 2023, pp. 355–366, DOI: [10.1109/issre59848.2023.00071](#), URL: <http://dx.doi.org/10.1109/ISSRE59848.2023.00071>.