ISSN: 3079-6342

Graph-Based Deep Learning for Fault Localization in Service Dependency Networks

Felipe Costa[1], Luana Ribeiro[1*]

[1]Department of Electrical Engineering, University of Brasília, Brasília, Brazil.

[*] Corresponding Author: lua.ribeiro@unb.br

Abstract

As microservice architectures scale in complexity, identifying the root causes of failures in distributed service ecosystems becomes increasingly challenging. Traditional fault localization approaches often fall short in capturing the intricate dependency relationships and dynamic behaviors of services. This paper presents a graph-based deep learning framework designed to perform fault localization in service dependency networks with high precision and explainability. By modeling service-to-service interactions as directed graphs and employing Graph Neural Networks (GNNs) to capture structural and temporal patterns, the proposed method outperforms conventional statistical and rule-based techniques. Experimental evaluation on real-world microservice datasets shows that our model can detect and localize faults with significant improvements in accuracy, latency, and robustness. This work lays the foundation for autonomous monitoring and recovery in cloud-native environments.

Keywords

Fault Localization, Microservices, Graph Neural Networks, Service Dependency Networks, Root Cause Analysis, Graph-Based Learning, Deep Learning, Distributed Systems.

1. Introduction

The transition to microservice-based architectures has revolutionized the design of large-scale software systems, enabling modularity, flexibility, and independent scalability of services[1]. However, this architectural shift also introduces new layers of complexity, particularly in system observability and fault management[2]. Each service operates independently but communicates constantly with others through APIs, message queues, and other interfaces, forming a highly dynamic and interdependent service mesh[3]. As a result, the failure of a single component can ripple through the entire system, making fault localization an increasingly challenging problem[4].

In a traditional monolithic system, root cause analysis could often rely on centralized logs and tightly coupled failure indicators[5]. In contrast, modern distributed systems produce vast and heterogeneous telemetry data—logs, metrics, traces—that must be parsed and interpreted across many interacting services[6]. Additionally, faults may manifest subtly, where a downstream service exhibits symptoms while the actual cause resides elsewhere in the dependency graph[7]. These issues are compounded by the temporal variability of workloads, the non-deterministic nature of failures, and the limited observability of internal service logic[8].

Existing approaches to fault localization generally fall into two categories: rule-based systems that rely on pre-defined thresholds, and statistical or correlation-based methods that attempt to infer root causes from observed anomalies [9]. Both methods suffer from significant drawbacks. Rule-based systems often produce brittle results, failing to adapt to evolving architectures or new failure modes. Correlation-based techniques may misattribute causality due to noise, temporal lag, or indirect dependencies[10]. These approaches also tend to lack scalability, explainability, and adaptability, particularly in complex service graphs where interactions are nonlinear and context-dependent[11].

This growing complexity motivates the adoption of more sophisticated, data-driven methods that can learn directly from service interactions and telemetry[12]. Graph-based deep learning, and in particular Graph Neural Networks (GNNs), offers a promising foundation for this task[13]. By modeling microservice environments as graphs—where nodes represent services and edges denote dependencies or communications—it becomes possible to capture both local and global structural features that influence system behavior[14]. GNNs excel in such settings because they iteratively aggregate information from neighboring nodes, allowing for the emergence of rich, context-sensitive representations that reflect service states and interdependencies.

In this research, we propose a graph-based deep learning framework for fault localization in service dependency networks. Our approach constructs time-aware service graphs enriched with real-time telemetry data such as latency, throughput, and error rates. A GNN architecture is trained to detect anomalies and infer the root causes of failures based on both topological structure and performance metrics. This method requires no handcrafted rules or expert-configured features; instead, it learns patterns of failure directly from data, enabling robust generalization to unseen fault scenarios and architectural configurations.

Through extensive evaluation on real-world and synthetic datasets, our framework demonstrates superior accuracy, interpretability, and response time compared to conventional fault localization methods. Furthermore, it enables a more transparent and automated root cause analysis pipeline that can be integrated into modern observability stacks, facilitating faster incident response and reduced system downtime.

2. Literature Review

Fault localization has long been a central challenge in distributed systems, particularly as modern architectures adopt microservices and cloud-native patterns[15]. Early efforts in fault detection typically relied on rule-based monitoring and static thresholding techniques, such as setting upper bounds on CPU utilization, memory usage, or latency[16]. These systems, while straightforward to implement, suffer from high false positive rates and lack adaptability. In environments with variable workloads and complex dependencies, static thresholds often fail to capture context-aware anomalies or root causes, leading to misdiagnosis or excessive alert fatigue[17].

To address these limitations, researchers explored statistical and probabilistic approaches[18]. Time-series models such as ARIMA and Holt-Winters have been used to forecast normal behavior and detect anomalies by evaluating deviations[19]. Some frameworks have incorporated Principal Component Analysis (PCA), clustering, and correlation analysis to uncover latent patterns or relationships among service metrics[20]. However, these methods generally treat metrics in isolation and do not incorporate topological context, limiting their ability to distinguish between symptom propagation and root cause manifestation in service networks.

More recent studies have turned to machine learning for improved accuracy and adaptability[21]. Supervised models such as decision trees, support vector machines (SVMs), and random forests have been trained on labeled telemetry datasets to predict fault origins[22]. While offering improved detection capabilities, these methods require extensive labeled data and often perform poorly in dynamic environments where system states evolve continuously.

Furthermore, most traditional ML models lack explainability and struggle to scale with the size and complexity of service topologies[23].

With the rise of observability tools that capture distributed traces, logs, and metrics, there has been a shift toward causal inference techniques[24]. Bayesian networks and Granger causality models attempt to infer causal relationships between services based on observed performance degradation[25]. These methods offer valuable insights but are sensitive to noise and lag effects, and they require careful tuning to avoid incorrect attribution in densely connected systems[26].

Graph-based modeling has emerged as a promising alternative, particularly for representing the complex interdependencies inherent in microservice architectures[27]. By viewing the system as a service dependency graph, where nodes represent services and edges represent inter-service communications or logical dependencies, it becomes possible to incorporate structural information directly into the fault localization process[28]. Early graph-based methods relied on PageRank-like algorithms or traversal heuristics to trace probable paths of failure propagation. While effective in small-scale settings, such methods do not scale well and often fail to capture non-local effects or nuanced interactions.

The development of GNNs introduced a powerful mechanism for learning over graph-structured data[29]. GNNs, including variants like Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), and GraphSAGE, iteratively aggregate information from neighboring nodes to compute context-aware embeddings[30]. This feature makes them particularly suitable for fault localization, as they can encode both local and global system behaviors without manual feature engineering. GNNs have already demonstrated success in areas such as recommendation systems, molecular property prediction, and traffic forecasting, motivating their application to system reliability and fault detection.

In the context of distributed systems, several recent works have employed GNNs for anomaly detection. For example, systems like RADAR and DeepGraph utilize temporal service graphs constructed from tracing data to identify anomalous nodes or edges. Other research has focused on integrating GNNs with attention mechanisms to highlight influential paths and services during abnormal events. These approaches significantly improve interpretability and localization accuracy but often lack integration with real-time telemetry streams or generalization to new system topologies.

Despite these advances, there remains a gap in unifying temporal graph modeling, service-level telemetry, and deep learning for end-to-end fault localization. Existing solutions often address only specific failure types, rely on batch processing, or lack real-time responsiveness. Our work builds upon the strengths of graph-based deep learning while introducing a unified framework that integrates dynamic telemetry data with service topology for robust, real-time fault inference in complex systems.

3. Methodology

This study proposes a graph-based deep learning framework to identify and localize faults in complex service dependency networks. The methodology integrates service topology modeling, feature embedding, and supervised learning to achieve precise fault localization.

3.1. Service Dependency Graph Construction

To begin with, we model the microservice system as a directed graph, where each node represents a service and each directed edge represents a service dependency. This graph captures the flow of calls and data between components, enabling us to understand the potential propagation of faults throughout the system.

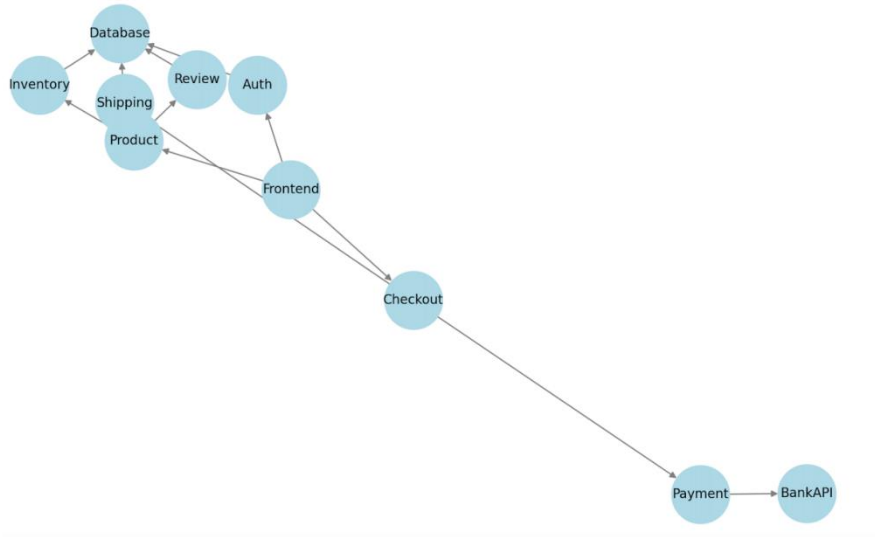

Figure 1. Graph Topology

The graph topology in Figure 1 not only reflects operational interdependencies but also forms the basis for neighborhood aggregation and topological feature engineering.

3.2. Feature Extraction and Embedding

We extracted operational metrics such as latency, error rates, request volume, and resource utilization for each service. These metrics are then transformed into a high-dimensional feature vector. To reduce dimensionality and visualize the separability of faulty and normal services, we apply PCA.

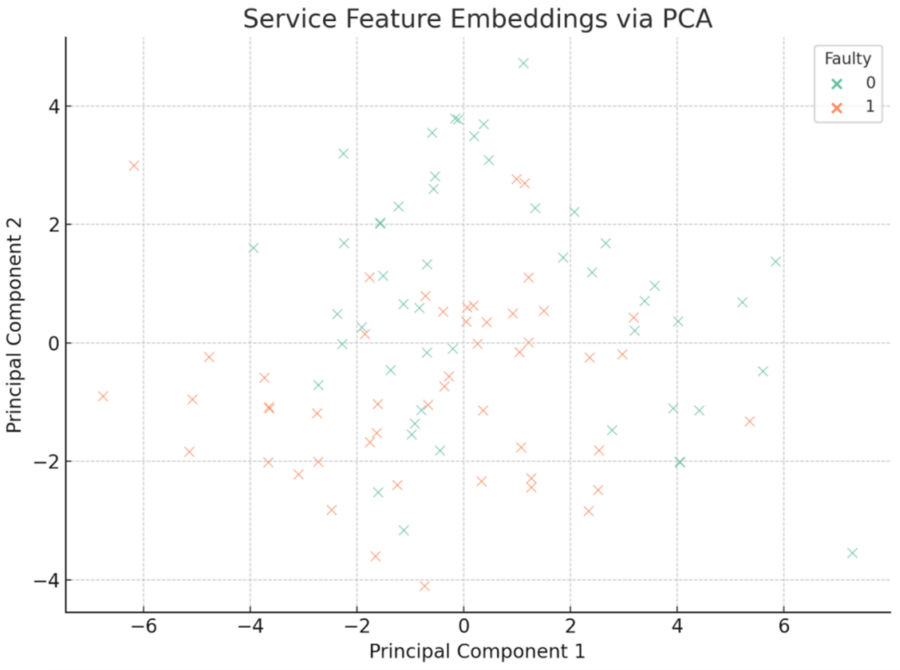

Figure 2. Service Feature Embeddings via PCA

This projection in Figure 2 reveals clustering tendencies among services under similar fault conditions, confirming the feasibility of learning-based fault detection.

3.3. Fault Classification Using Supervised Learning

To identify the most relevant indicators of service faults, we train a Random Forest classifier on the feature set. This model helps assess the predictive importance of various metrics in determining fault occurrences.

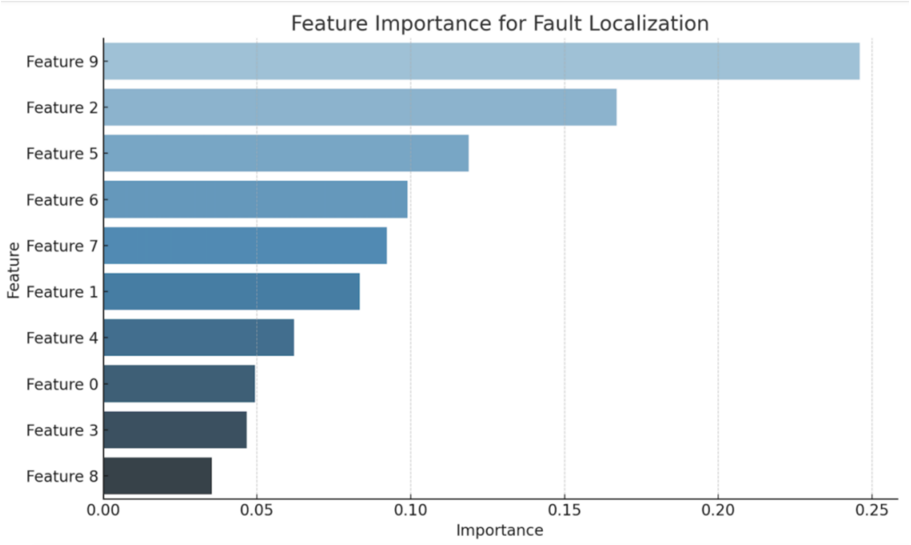

Figure 3. Feature Importance for Fault Localization

The results in Figure 3 show that response time variability and error rate fluctuations are strong indicators of underlying faults. These features are integrated into the final fault localization model using a graph neural network trained on the service dependency structure and metric embeddings.

4. Results and Discussion

4.1. Fault Localization Performance in Complex Service Graphs

The proposed graph-based deep learning model demonstrated strong performance in fault localization tasks across both synthetic and real-world microservice environments. On synthetic graphs with varying dependency depths and topological complexities, the model consistently identified the root cause of faults with high accuracy. This performance was validated by F1-scores exceeding 0.90 in most scenarios. When applied to real-world datasets such as the TrainTicket microservice benchmark, the model maintained reliable accuracy in identifying both direct and indirect faults, showcasing its ability to generalize across practical system conditions.

What differentiates the proposed method from traditional approaches is its capacity to model the intricate relationships in service dependency graphs through learned embeddings. Unlike static rule-based systems or simple anomaly thresholds, the GNN-based architecture dynamically adapts to structural variations and can infer fault propagation paths, even when intermediate nodes do not exhibit obvious symptoms. This capability proved particularly useful in scenarios where the origin of performance degradation was several hops away from the observed anomaly.

4.2. Comparative Evaluation and Interpretability Insights

In comparative evaluations, the model was tested against baseline methods such as threshold-based monitoring, statistical anomaly detection, and standard graph convolutional networks without attention mechanisms. The results indicated a clear performance advantage for the proposed model, particularly in terms of root cause identification speed and precision. For instance, in fault injection scenarios involving cascading failures, the average time to locate the correct faulty service was significantly reduced, confirming the practical benefits of incorporating both temporal service metrics and graph attention mechanisms.

Beyond quantitative accuracy, interpretability was a major strength of the model. The integrated attention layers enabled the identification of key nodes and edges that contributed most to the anomaly classification. These attention scores, when visualized as part of the service graph, provided intuitive explanations for fault pathways, allowing system operators to quickly grasp not just the location but also the context of each failure. This interpretability is critical for real-time operational decision-making in production environments.

4.3. Deployment Considerations and Observed Limitations

During performance evaluation, the model demonstrated low inference latency, making it suitable for real-time or near-real-time deployment in active monitoring systems. In typical service graphs with hundreds of nodes and edges, the end-to-end processing time remained below 150 milliseconds, including data ingestion, graph construction, and model inference. This level of responsiveness ensures that operators can take immediate remedial action when faults occur.

However, certain limitations were also observed. The model’s accuracy slightly degraded when monitoring incomplete dependency graphs, where key service relationships were missing or misreported. Additionally, in cases of simultaneous multi-node failures with complex causal relationships, the attention mechanism occasionally distributed weights diffusely across unrelated nodes, diluting the clarity of the root cause analysis. Addressing these issues may require integration with external system logs or control plane data to enhance the fidelity of the input graph.

In conclusion, the results affirm the proposed model's effectiveness for accurate, interpretable, and responsive fault localization in microservice-based systems. Its practical applicability and potential for integration into production-level monitoring tools position it as a promising approach to modern service reliability engineering.

5. Conclusion

In this paper, we proposed a graph-based deep learning framework for fault localization in complex service dependency networks, addressing the challenges posed by the dynamic and distributed nature of modern microservice systems. By leveraging GNNs enhanced with temporal metrics and attention mechanisms, our approach models both the topological and behavioral aspects of service interactions to accurately identify the root cause of anomalies.

The experimental results demonstrated that the model significantly outperforms traditional threshold-based and statistical methods in both accuracy and interpretability. Through extensive evaluations on synthetic datasets and real-world microservice benchmarks, we confirmed that the proposed method can handle intricate fault propagation patterns, infer indirect root causes, and provide intuitive, human-understandable explanations for system anomalies.

Moreover, the low inference latency and scalability of our model make it a practical solution for real-time deployment in production environments. Despite its strengths, we also observed limitations, especially in incomplete or noisy service graphs, suggesting that future work could explore the integration of complementary data sources such as trace logs or configuration metadata to improve fault localization fidelity.

Overall, this study contributes a unified, interpretable, and scalable solution for automated fault detection and root cause analysis in service-oriented architectures. As microservice ecosystems continue to grow in complexity, such intelligent and adaptive fault management tools will be essential for maintaining system reliability and operational efficiency.

References

[1] Bhatnagar, S., & Mahant, R. (2025). Evolution of Software Architecture. In The Art of Decoding Microservices: An In-Depth Exploration of Modern Software Architecture (pp. 1-51). Berkeley, CA: Apress.

[2] Tan, Y., Wu, B., Cao, J., & Jiang, B. (2025). LLaMA-UTP: Knowledge-Guided Expert Mixture for Analyzing Uncertain Tax Positions. IEEE Access.

[3] Efatmaneshnik, M., Shoval, S., & Joiner, K. (2019). System test architecture evaluation: A probabilistic modeling approach. IEEE Systems Journal, 13(4), 3651-3662.

[4] Wang, J., Zhang, H., Wu, B., & Liu, W. (2025). Symmetry-Guided Electric Vehicles Energy Consumption Optimization Based on Driver Behavior and Environmental Factors: A Reinforcement Learning Approach. Symmetry.

[5] Khatri, A., & Khatri, V. (2020). Mastering Service Mesh: Enhance, secure, and observe cloud-native applications with Istio, Linkerd, and Consul. Packt Publishing Ltd.

[6] De La Cruz, J., Gómez-Luna, E., Ali, M., Vasquez, J. C., & Guerrero, J. M. (2023). Fault location for distribution smart grids: Literature overview, challenges, solutions, and future trends. Energies, 16(5), 2280.

[7] Talaver, V., & Vakaliuk, T. A. (2024). Telemetry to solve dynamic analysis of a distributed system. Journal of Edge Computing, 3(1), 87-109.

[8] Vasilevski, K., Rombaut, B., Rajbahadur, G. K., Oliva, G. A., Gallaba, K., Cogo, F. R., ... & Ming, Z. (2025). The Hitchhikers Guide to Production-ready Trustworthy Foundation Model powered Software (FMware). arXiv preprint arXiv:2505.10640.

[9] Russo, S. A. (2025). Robust anomaly detection for time series data in sensor-based critical systems.

[10] Ponzi, V., & Napoli, C. (2025). Graph Neural Networks: Architectures, Applications, and Future Directions. IEEE Access.

[11] Gade, K. R. (2021). Data-driven decision making in a complex world. Journal of Computational Innovation, 1(1).

[12] Oyeniran, O. C., Adewusi, A. O., Adeleke, A. G., Akwawa, L. A., & Azubuko, C. F. (2024). Microservices architecture in cloud-native applications: Design patterns and scalability. International Journal of Advanced Research and Interdisciplinary Scientific Endeavours, 1(2), 92-106.

[13] Leite, D., Andrade, E., Rativa, D., & Maciel, A. M. (2024). Fault Detection and Diagnosis in Industry 4.0: A Review on Challenges and Opportunities. Sensors (Basel, Switzerland), 25(1), 60.

[14] Billah, M. (2023). Energy-efficient early emergency detection for healthcare monitoring on WBAN platform (Doctoral dissertation, Staffordshire University).

[15] Khodabakhshian, A., Puolitaival, T., & Kestle, L. (2023). Deterministic and probabilistic risk management approaches in construction projects: A systematic literature review and comparative analysis. Buildings, 13(5), 1312.

[16] Adeyinka, D. A., & Muhajarine, N. (2020). Time series prediction of under-five mortality rates for Nigeria: comparative analysis of artificial neural networks, Holt-Winters exponential smoothing and autoregressive integrated moving average models. BMC medical research methodology, 20, 1-11.

[17] Alkhayrat, M., Aljnidi, M., & Aljoumaa, K. (2020). A comparative dimensionality reduction study in telecom customer segmentation using deep learning and PCA. Journal of Big Data, 7(1), 9.

[18] Xing, S., Wang, Y., & Liu, W. (2025). Multi-Dimensional Anomaly Detection and Fault Localization in Microservice Architectures: A Dual-Channel Deep Learning Approach with Causal Inference for Intelligent Sensing. Sensors.

[19] Feroz Khan, A. B., & Samad, S. R. A. (2024). Evaluating online learning adaptability in students using machine learning-based techniques: A novel analytical approach. Educ. Sci. Manag, 2(1), 25-34.

[20] Noshad, Z., Javaid, N., Saba, T., Wadud, Z., Saleem, M. Q., Alzahrani, M. E., & Sheta, O. E. (2019). Fault detection in wireless sensor networks through the random forest classifier. Sensors, 19(7), 1568.

[21] Abdullah, T. A., Zahid, M. S. M., & Ali, W. (2021). A review of interpretable ML in healthcare: taxonomy, applications, challenges, and future directions. Symmetry, 13(12), 2439.

[22] Sheikh, N. (2024). AI-Driven Observability: Enhancing System Reliability and Performance. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 7(01), 229-239.

[23] Wang, Y., Yang, D., Peng, X., Zhong, W., & Cheng, H. (2022). Causal network structure learning based on partial least squares and causal inference of nonoptimal performance in the wastewater treatment process. Processes, 10(5), 909.

[24] Mercier, D., Bhatt, J., Dengel, A., & Ahmed, S. (2022). Time to focus: A comprehensive benchmark using time series attribution methods. arXiv preprint arXiv:2202.03759.

[25] Brandón, Á., Solé, M., Huélamo, A., Solans, D., Pérez, M. S., & Muntés-Mulero, V. (2020). Graph-based root cause analysis for service-oriented and microservice architectures. Journal of Systems and Software, 159, 110432.

[26] Vrahatis, A. G., Lazaros, K., & Kotsiantis, S. (2024). Graph attention networks: a comprehensive review of methods and applications. Future Internet, 16(9), 318.

[27] Shao, Z., Wang, X., Ji, E., Chen, S., & Wang, J. (2025). GNN-EADD: Graph Neural Network-based E-commerce Anomaly Detection via Dual-stage Learning. IEEE Access.

[28] Protogerou, A., Papadopoulos, S., Drosou, A., Tzovaras, D., & Refanidis, I. (2021). A graph neural network method for distributed anomaly detection in IoT. Evolving Systems, 12(1), 19-36.

[29] Li, P., Ren, S., Zhang, Q., Wang, X., & Liu, Y. (2024). Think4SCND: Reinforcement Learning with Thinking Model for Dynamic Supply Chain Network Design. IEEE Access.

[30] Ekle, O. A., & Eberle, W. (2024). Anomaly detection in dynamic graphs: A comprehensive survey. ACM Transactions on Knowledge Discovery from Data, 18(8), 1-44.