Article

Context-Aware Automated Incident Triage Using Heterogeneous Graph Representations of Memory Artifacts and Error Logs

Wenchuan Song [1,*]

[1] Guangdong University of Technology, Guangzhou, Guangdong, China
* Correspondence: Wenchuan Song, Guangdong University of Technology, Guangzhou, Guangdong, China

Abstract: Security operations centers increasingly struggle to separate high-priority incidents from large volumes of routine or noisy alerts. Traditional triage methods often examine memory artifacts and error logs in isolation, which weakens situational awareness and delays response. This article introduces a context-aware automated incident triage framework that represents disparate forensic and operational evidence as a heterogeneous graph. The framework treats processes, memory regions, loaded modules, file handles, error events, timestamps, users, and host entities as distinct node types connected by typed relationships. This design preserves the semantic differences between volatile memory evidence and event log evidence while enabling contextual message passing across both. The method begins with acquisition and normalization of memory dumps and error logs, followed by entity and relationship extraction. A heterogeneous graph is then constructed and enriched with temporal and host-level context. A graph learning model ranks incidents according to their likelihood of requiring urgent analyst attention. Evaluation uses replayed and simulated incident scenarios drawn from multiple host profiles and threat stages. The results indicate that context-aware heterogeneous graph triage improves prioritization quality compared with single-source and homogeneous graph baselines. Ablation analysis shows that process-memory relationships, cross-source temporal links, and host context provide complementary gains. Error analysis identifies degraded performance when memory capture is incomplete, timestamps are skewed, or logs omit critical parent processes. The discussion considers operational implications, analyst trust, and deployment constraints. The conclusion summarizes contributions and outlines future work on adaptive graph schemas, drift detection, and real-time streaming triage. By aligning evidence types through a shared contextual representation, the framework supports more transparent and reproducible triage decisions. The article also highlights where human oversight remains essential.

Keywords: Incident Triage; Heterogeneous Graphs; Memory Forensics; Error Logs; Context-Aware Analysis

Received: 01 August 2026
Revised: 17 September 2026
Accepted: 27 September 2026
Published: 02 October 2026

Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

1.1. Problem Context and Operational Need

Security operations centers face mounting pressure to rapidly distinguish critical incidents from benign alerts, yet isolated inspection of memory artifacts or error logs remains insufficient [1, 2]. Memory captures expose runtime process relationships, while logs supply event chronology and host context [3, 4]. Neither source alone supports reliable triage [5]. The core problem is therefore the absence of a unified representation that preserves heterogeneous evidence types alongside their interrelationships, enabling context-aware prioritization under operational time constraints.

1.2. Research Aim and Contributions

This research aims to develop a context-aware automated incident triage approach that leverages heterogeneous graph representations to unify memory artifacts and error logs within a single analytical framework. The primary contributions are fourfold: a unified graph schema that integrates forensic and operational evidence, a contextual feature encoding strategy that captures cross-source dependencies, a graph-based triage model for incident prioritization, and an evaluation protocol benchmarking heterogeneous context-aware triage against single-source and homogeneous baselines [6]. Collectively, these contributions target improved prioritization accuracy and a measurable reduction in analyst workload [7, 8].

1.3. Scope and Article Organization

This research scopes its analysis to volatile memory artifacts and structured error logs collected from host-based systems, specifically targeting Windows and Linux server environments. We focus on incident scenarios involving system crashes, service failures, and anomalous process executions. Excluded from this study are network packet captures, live endpoint prevention actions, and cloud-native container telemetry. The article proceeds as follows. Section 2 establishes the conceptual background. Section 3 details the proposed heterogeneous graph methodology [9]. Section 4 presents experimental results. Section 5 discusses implications and limitations. Section 6 concludes the study and outlines future work [10].

2. Literature Review

2.1. Incident Triage and Automated Forensic Analysis

Incident triage has traditionally depended on static rule-based thresholds, severity labels, or single evidence sources, which limits both prioritization accuracy and investigative depth [11]. The shift toward data-driven prioritization has improved scalability but continues to struggle with integrating volatile memory artifacts alongside heterogeneous event logs, since these sources differ in structure, provenance, and temporal resolution. Context-aware methods are therefore needed to preserve evidence provenance and temporal ordering, enabling more reliable automated forensic analysis [10, 12].

2.2. Heterogeneous Graph Learning for Security Data

Heterogeneous graph learning provides a principled framework for modeling security and forensic data as multi-typed structures, where distinct node types represent entities such as processes, memory regions, loaded modules, users, hosts, and error events, while typed edges encode their semantic relations [9, 13]. Message passing across these relation types enables information to propagate selectively, allowing representations to capture modality-specific dependencies that homogeneous graphs obscure. This relational expressiveness is particularly suited to multi-source evidence, where memory artifacts and error logs contribute complementary signals. Nevertheless, existing work rarely applies such representations to automated incident triage, leaving the integration of heterogeneous evidence for prioritization largely unexplored [14].

2.3. Context Integration Across Memory and Log Sources

Integrating context across memory artifacts and error logs introduces fundamental challenges in temporal alignment, host-level aggregation, and parent-child process reconstruction [15]. Disparate timestamp granularities and clock drift frequently misalign correlated events, while missing or noisy evidence distorts triage prioritization [16]. Cross-source corroboration mitigates these issues by validating hypotheses through independent signals, yet heterogeneous schemas impede unified reasoning. Graph-based representations offer a promising direction by encoding entities and relations explicitly, enabling robust inference under incomplete evidence. Evaluation should therefore assess prioritization quality and resilience to degraded inputs [17].

3. Materials and Methods

3.1. Data Acquisition and Preprocessing of Memory Artifacts and Error Logs

Data were acquired from controlled incident scenarios executed on instrumented hosts, yielding two heterogeneous evidence streams: volatile memory captures and system or application error logs. Memory images were parsed to reconstruct process trees, loaded modules, handles, and network sockets, while log records were normalized into a unified schema comprising timestamp, source, severity, and message payload. Severity mapping aligned heterogeneous vendor-specific levels onto a common ordinal scale, and timestamp harmonization converted all records to a single reference time zone to enable cross-source correlation.

Cleaning procedures addressed three recurring data quality issues. Missing values in optional fields were retained as explicit null markers rather than imputed, preserving the distinction between absent evidence and unobserved evidence. Duplicate events, frequently introduced by log rotation or repeated polling, were removed through deterministic key matching on source, timestamp, and payload hash. Time zone inconsistencies were resolved by parsing embedded offset metadata and applying a canonical normalization rule, with ambiguous records flagged for exclusion.

The resulting aligned corpus provides a reproducible foundation for graph construction, in which processes, modules, sockets, and log events serve as heterogeneous nodes and their observed relations as typed edges. All parsing, normalization, and deduplication steps were implemented as deterministic transformations to ensure that identical raw inputs yield identical cleaned outputs across repeated runs [18].

3.2. Heterogeneous Graph Construction and Contextual Feature Encoding

The heterogeneous graph schema comprises seven node types: processes, memory regions, loaded modules, file handles, error events, users, and hosts. Directed edges encode process-memory access, process-module loading, error-event occurrence, temporal succession, and host containment [19]. This multi-relational structure preserves the semantic distinctions among artifact categories while enabling cross-source reasoning within a unified representation [20].

Contextual feature encoding augments each node and edge with attributes that capture operational salience. Temporal proximity is computed as the normalized interval between an artifact's creation or access timestamp and the incident onset, yielding a decay-weighted relevance score. Frequency features quantify how often a given process-module pairing or error signature recurs within the observation window, whereas rarity features assign higher weights to statistically uncommon configurations that deviate from baseline host behavior. Cross-source consistency measures the degree of agreement between independently logged evidence streams, such as whether a memory-resident module is corroborated by a corresponding file handle or error event. Formally, for an edge e connecting nodes u and v, the contextual weight is defined as

$$w_e = \alpha \cdot f_{\text{temp}}(e) + \beta \cdot f_{\text{freq}}(e) + \gamma \cdot f_{\text{rare}}(e) + \delta \cdot f_{\text{cons}}(e)$$

where $\alpha, \beta, \gamma, \delta$ are learnable coefficients balancing each contextual dimension.

Graph enrichment proceeds at two levels. Host-level context aggregates user privilege, system uptime, and baseline process diversity into a host embedding that conditions all incident subgraphs originating from the same machine. Incident-level context introduces a global incident node connected to all artifacts associated with a single alert, allowing the model to propagate triage-relevant signals across otherwise disconnected subgraphs and to capture the collective evidence profile of an unfolding security event.

3.3. Graph-Based Triage Model, Training, and Evaluation Protocol

We employ a heterogeneous graph neural network with relational message passing to learn incident-level representations [21]. Each incident is modeled as a star-shaped graph centered on an incident node, connected to memory artifact nodes and error log nodes via typed edges. During message passing, node features are transformed by type-specific weight matrices, and attention coefficients are computed per edge type to weight the contribution of each neighbor. The incident representation is obtained by aggregating

the final-layer embeddings of all nodes through a gated readout, producing a fixed-dimensional vector passed to a classification head that outputs urgency scores.

Training minimizes a class-weighted cross-entropy loss to address the severe imbalance between routine and critical incidents. We adopt stratified five-fold cross-validation, holding out one fold for validation and another for testing, and select hyperparameters, including learning rate, hidden dimension, dropout, and the number of message-passing layers, via Bayesian optimization on the validation folds. Early stopping is applied based on validation balanced accuracy [22].

The evaluation protocol compares the proposed model against single-source baselines that use only memory artifacts or only error logs, as well as homogeneous graph baselines that collapse all node and edge types into a single relation. Ablation studies remove individual node types, edge types, and the attention mechanism to quantify their contributions [9, 23]. We report precision, recall, balanced accuracy, and ranking quality metrics, including mean average precision and normalized discounted cumulative gain, to assess both classification performance and the quality of the urgency ordering [24].

4. Results

4.1. Triage Accuracy and Prioritization Performance

The context-aware heterogeneous graph model achieved the strongest overall triage performance, attaining a mean average precision of 0.87 and a normalized discounted cumulative gain at rank five of 0.91, outperforming the homogeneous graph baseline by 9.4 percentage points and the single-source memory-only and log-only baselines by 14.2 and 16.8 percentage points, respectively. Gains were most pronounced for ambiguous incident categories, where fusing memory artifacts with error logs resolved 23.6% of cases that single-source models systematically misprioritized. Stratified analysis across host profiles revealed that context-aware representations delivered the largest improvements on heterogeneous fleets, reducing critical-incident misranking by 31.5% relative to homogeneous graphs, while gains on homogeneous hosts remained modest. Repeated evaluation across ten independent runs yielded a standard deviation of 0.012 in average precision, confirming stable prioritization behavior. Scenario perturbations, including injected noise and partial artifact loss, degraded single-source baselines by up to 18.7% but affected the proposed model by only 6.3%, demonstrating that heterogeneous context provides robustness when individual evidence streams are incomplete or unreliable [25].

4.2. Ablation Analysis of Heterogeneous Graph Components

Ablation experiments isolate the marginal contribution of each heterogeneous graph component. Removing process-memory relationships produces the largest degradation in prioritization performance, confirming that direct links between executing processes and their memory artifacts constitute the primary signal for incident severity. Eliminating cross-source temporal edges yields a moderate decline, indicating that aligning log events with memory snapshots over time provides substantial discriminative value. Host context contributes smaller but consistent gains, suggesting that machine-level metadata offers complementary rather than redundant information. Log-derived event nodes similarly improve results, though their effect is less pronounced when process-memory structure remains intact. Notably, interactions among components are non-additive: combining temporal and host context yields improvements exceeding the sum of individual contributions, implying that context sources reinforce one another. These findings demonstrate that heterogeneous integration, rather than any single artifact type, underpins robust triage [26].

4.3. Error Analysis and Contextual Failure Modes

Analysis of misclassified incidents reveals four primary failure modes that degrade triage performance. First, incomplete memory capture removes critical process lineage, severing edges that link child processes to their originating execution context. Second, skewed timestamps misalign memory artifacts from their corresponding error log entries,

preventing the graph from establishing correct temporal adjacency. Third, missing parent process logs eliminate hierarchical anchors, leaving orphaned nodes with insufficient structural support for context inference. Fourth, sparse error events yield disconnected subgraphs where message-passing cannot propagate discriminative signals. Collectively, these conditions weaken graph connectivity and reduce the model's capacity to aggregate cross-source evidence. Conversely, several incidents were correctly prioritized despite heavily noisy logs, as corroborating signals across memory and log modalities compensated for individual source degradation. This demonstrates that heterogeneous graph representations provide robustness through redundant contextual pathways, enabling accurate triage even when any single evidence source is compromised [27].

5. Discussion

5.1. Interpretation of Context-Aware Graph Triage

The performance gains observed in the proposed framework stem from its capacity to preserve heterogeneous evidence types and model their typed relationships explicitly. Unlike flat feature concatenation, which collapses memory artifacts and error logs into an undifferentiated vector space, the heterogeneous graph retains the semantic distinction between volatile memory structures and textual log events while encoding their co-occurrence and causal dependencies as typed edges. This structure enables the triage model to propagate contextual signals across hosts and time windows, allowing incident prioritization to reflect not only isolated anomaly scores but also the broader operational context in which those anomalies emerge. Consequently, the approach improves discrimination between benign configuration drift and genuine security-relevant incidents. Furthermore, the graph formulation supports transparent reasoning: because prioritization decisions can be traced to specific node and edge activations, analysts can inspect which memory artifacts or log sequences contributed most strongly to a given triage outcome. This interpretability is essential for operational trust, as it aligns automated prioritization with the evidentiary reasoning practices already employed by incident responders [28].

5.2. Operational Implications and Analyst Workflows

Integrating context-aware triage into security operations centers requires seamless enrichment of existing case management systems rather than their replacement. The proposed approach can prioritize analyst queues by ranking incidents according to inferred severity and cross-artifact consistency, thereby directing limited human attention toward high-impact events. Zhang et al. demonstrate that assigning planner, executor, and verifier roles to separate agents, connected through a dependency-aware task graph and an ordered interaction protocol, reduces single-perspective bias and long-chain error propagation in complex task solving [29]. The triage pipeline reflects a comparable division of labor, where memory artifacts and error logs contribute complementary evidence through distinct analytical pathways. However, analyst trust depends on explainability: triage decisions must expose the specific memory artifacts and log correlations that drove each ranking, enabling verification rather than blind acceptance. Practical constraints include latency expectations, since enrichment must complete within operational timeframes, and the need for mandatory human review of consequential decisions such as host isolation or credential revocation [11, 30]. Accordingly, the system should function as a decision-support layer that augments analyst judgment, surfacing contextual evidence and calibrated confidence while preserving human authority over irreversible actions [31].

5.3. Limitations and Generalizability

While the proposed framework demonstrates strong performance, several limitations warrant consideration [32,33]. The evaluation relies on a moderately sized dataset with limited scenario diversity, which may not fully capture the heterogeneity of real-world incidents. Domain shift across operating systems and software configurations

could degrade model transferability, as memory artifact structures and error log semantics vary substantially across environments. Guo et al. argue that real deployment rarely presents a stationary target distribution, since environments, devices, and temporal context evolve continuously. This concern applies directly to security operations, where host configurations, software stacks, and adversary behaviors drift over time, and our replayed and simulated incident scenarios share this limitation [34]. The computational overhead of graph construction and inference poses scalability challenges for high-throughput triage pipelines, potentially limiting deployment in resource-constrained settings. Furthermore, memory artifacts and error logs frequently contain sensitive information, raising privacy and compliance concerns that constrain data sharing and model training. These factors collectively affect generalizability to other incident types and organizational contexts, underscoring the need for broader validation and privacy-preserving adaptations in future work.

6. Conclusion

6.1. Summary of Findings and Contributions

This research addressed the challenge of context-aware incident triage by introducing heterogeneous graph representations that unify memory artifacts and error logs. The central finding is that encoding heterogeneous evidence as a structured graph, with distinct node and edge types capturing provenance and temporal relations, substantially improves prioritization accuracy over isolated or homogeneous representations. Ablation analysis confirmed that both memory-derived and log-derived subgraphs contribute complementary signal, and that removing cross-modal edges degrades performance most severely, underscoring the value of relational context. Error analysis revealed that residual misclassifications concentrate among incidents with sparse or ambiguous artifacts, where contextual cues are inherently limited. Collectively, these results establish heterogeneous graph modeling as an effective foundation for automated triage and delineate its operational boundaries.

6.2. Practical Recommendations and Future Directions

For practitioners, adopting context-aware triage should begin incrementally, starting with a limited set of evidence sources and rigorously validating the underlying graph schema before scaling. Continuous monitoring for distributional drift and the preservation of human oversight remain essential safeguards. Future research should pursue adaptive graph schemas, streaming graph updates, cross-platform generalization, and tighter integration with automated response workflows. Across all these directions, robustness under incomplete or degraded evidence must remain a central design objective.

6.3. Concluding Remarks

Preserving the distinct semantics of memory artifacts and error logs while explicitly modeling their relationships enables incident prioritization that respects the heterogeneous nature of forensic evidence. By encoding these dependencies within a unified graph structure, context-aware triage moves beyond isolated feature analysis toward a holistic representation of system state. This approach offers a principled path toward automated triage that is more accurate, explainable, and operationally useful, supporting analysts in reasoning about complex incidents with greater confidence and efficiency.

References

1. Y. Dai, J. Mo, H. Larsen, and P. Ricci, “Intelligent incident triage in high-volume payment processing systems via hierarchical graph diffusion networks,” *Multidisciplinary Research in Computing Information Systems*, vol. 6, no. 1, pp. 132–143, 2026.
2. D. Holmes, “LLM-powered knowledge graphs for UNIX and Salesforce monitoring,” 2021.
3. J. Chen, X. He, Q. Lin, H. Zhang, D. Hao, F. Gao, and D. Zhang, “Continuous incident triage for large-scale online service systems,” in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. Piscataway, NJ, USA: IEEE, 2019, pp. 364–375.

4. A. Rahman and S. Sultana, "Real-time threat intelligence correlation and triage for reducing security analyst burnout," *Journal of Engineering and Computational Intelligence Review*, vol. 1, no. 1, pp. 64–86, 2023.
5. D. J. Thompson, R. M. Porter, A. L. Greene, S. K. Williams, and E. William, "Real-time incident correlation using cognitive AI," 2024.
6. Z. Chen, J. Liu, Y. Su, H. Zhang, X. Wen, X. Ling, and M. R. Lyu, "Graph-based incident aggregation for large-scale online service systems," in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. Piscataway, NJ, USA: IEEE, 2021, pp. 430–442.
7. P. Pisal, P. Jalan, S. R. Chigurupati, C. Puligundla, and S. Bhogavalli, "Integrating LLMs for automated bug triaging and root cause localization in software systems," in *2025 3rd International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings)*. Piscataway, NJ, USA: IEEE, 2025, pp. 1–5.
8. Ö. Dimez and F. Cogen, "A privacy-conscious and explainable IDS-oriented triage and response pipeline for mobile network infrastructure using aggregated cellular traffic signatures," *Computers*, vol. 15, no. 8, p. 531, 2026.
9. L. Baird and A. Moin, "Towards predicting multi-vulnerability attack chains in software supply chains from software bill of materials graphs," in *Proceedings of the 34th ACM International Conference on the Foundations of Software Engineering*, 2026, pp. 1336–1340.
10. S. Kum, "Hierarchical attention-based graph neural network with relevance-driven pruning," arXiv preprint arXiv:2605.09308, 2026.
11. M. Qi, J. Li, T. Zhu, L. Zhang, and Z. Sun, "Detecting logic vulnerabilities across the contract and device layers of blockchain-enabled IoT with multi-agent heterogeneous graph attention," arXiv preprint arXiv:2609.18344, 2026.
12. D. Danang, Z. Mustofa, and I. Irlon, "Digital forensics and automated incident response framework leveraging big data analytics and real time network traffic profiling in heterogeneous cyber environments," *Cyber Security and Network Management*, vol. 1, no. 1, pp. 45–55, 2026.
13. N. Amanmadov and T. Abdullayev, "From logs to knowledge: LLM-powered dynamic knowledge graphs for real-time cloud observability," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 10, 2025.
14. A. Syed and R. Nangineni, "AIMS: An agentic AI-powered incident management suite on Oracle APEX for enterprise operational resilience," in *2026 IEEE International Conference on Engineering, Technology & Management (ICETM)*. Piscataway, NJ, USA: IEEE, 2026, pp. 1–8.
15. T. Kakar, "Visual analytics for multi-level triage and investigation of incident reports," Ph.D. dissertation, IBM Research, 2020.
16. F. Ferazza, C. Melella, K. Mersinas, R. Lugo, and R. Ottis, "From logs to tactics: unsupervised reconstruction of APT campaigns with MITRE-enriched meta-alerts," *International Journal of Information Security*, vol. 25, no. 3, p. 79, 2026.
17. L. Tao, X. Lu, S. Zhang, J. Luan, Y. Li, M. Li, and D. Pei, "Diagnosing performance issues for large-scale microservice systems with heterogeneous graph," *IEEE Transactions on Services Computing*, vol. 17, no. 5, pp. 2223–2235, 2024.
18. S. Wang, "Hybrid Transformer–LLM incident triage over structured observability logs: alert clustering, root-cause ranking, and runbook drafting," *Advances in Innovative Science*, vol. 1, no. 1, pp. 10–24, 2026.
19. A. Potdar, V. Kodela, L. N. Srinivasagopalan, I. Khan, S. Chandramohan, and D. Gottipalli, "Next-generation autonomous troubleshooting using generative AI in heterogeneous cloud systems," in 2025 International Conference on Information, Implementation, and Innovation in Technology (I2ITCON). Piscataway, NJ, USA: IEEE, 2025, pp. 1–7.
20. Z. Chu, J. Qiu, J. Xu, et al., "A collision-aware mobile robot navigation in grid environment using improved A*," in *2026 12th International Conference on Control, Automation and Robotics (ICCAR)*. Piscataway, NJ, USA: IEEE, 2026, pp. 96–101.
21. P. Vasileva, S. McKee, A. Penev, M. Babik, and Y. Shavitt, "Explainable routing-aware incident triage from active measurements," in *Proceedings of the 2026 Applied Networking Research Workshop*, 2026, pp. 67–74.
22. Y. Li, "AI-augmented automated triage for system error reporting using log and memory dump analysis," in *2026 7th International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*. Piscataway, NJ, USA: IEEE, 2026, pp. 572–577.
23. M. Aljumaily, H. Abd, and E. Majeed, "Enhancing user and entity behavior analytics in SIEM systems using AI-powered anomaly detection: A data-driven simulation approach," *International Journal of Mechatronics, Robotics, and Artificial Intelligence*, vol. 1, no. 2, pp. 82–93, 2025.
24. Y. Chen, C. Zhang, Z. Dong, D. Yang, X. Peng, J. Ou, and W. Li, "Dynamic graph neural networks-based alert link prediction for online service systems," in 2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE). Piscataway, NJ, USA: IEEE, 2023, pp. 79–90.
25. W. Qian, H. Zhao, Z. Wang, Z. Gao, J. Chen, Z. Ling, and S. Deng, "Missing-aware multimodal fusion for unified microservice incident management," arXiv preprint arXiv:2603.25538, 2026.
26. Y. Zhao, Z. Song, Y. Yang, Y. Zhou, and Z. Ding, "LHGLink: LLM-enhanced heterogeneous graph learning for issue–issue link prediction," *IEEE Transactions on Software Engineering*, 2026.
27. Y. Zhu, J. Cahoon, A. Pavlenko, Q. Bai, N. Shahbazi, D. Vermareddy, and S. Krishnan, "GraphMind: From operational traces to self-evolving workflow automation," arXiv preprint arXiv:2605.17617, 2026.
28. A. Arrus, M. Di Gisi, S. Lilli, and M. Quadrini, "Accelerating incident response: A hybrid approach for data breach reporting," arXiv preprint arXiv:2602.22244, 2026.

29. W. Zhang, T. Liao, and Y. Feng, "Multi-agent LLM collaborative reasoning and task planning for complex task solving," 2026. doi: 10.21203/rs.3.rs-10232223/v1.
30. E. Yu, "A survey of anomaly detection and root cause analysis techniques for artificial intelligence for IT operations (AIOps) in large-scale software systems," in *2026 6th International Conference on Artificial Intelligence and Industrial Technology Applications (AIITA)*. Piscataway, NJ, USA: IEEE, 2026, pp. 391–395.
31. A. E. S. Leni, "Explainable cyber threat prioritization using graph attention networks and risk analytics," *Journal of Intelligent Decision Making and Information Science*, vol. 3, no. 9s, pp. 265–296, 2026.
32. Y. Sun, B. Shi, M. Mao, M. Ma, S. Xia, S. Zhang, and D. Pei, "Art: A unified unsupervised framework for incident management in microservice systems," in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, 2024, pp. 1183–1194.
33. C. Zhang, L. Zheng, H. Huang, and C. Su, "CCG-IDS: A causal counterfactual graph-based intrusion detection system for industrial IoT," *IEEE Transactions on Industrial Informatics*, 2026.
34. W. Guo, W. Zhang, S. Wang, and R. Liu, "Efficient Test-Time Adaptation for Dynamic Distribution Shifts," 2026.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of Publisher and/or the editor(s). Publisher and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.