



International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Graph-Augmented LLMs for SIEM: Explainable Attack Attribution with High-Fidelity Log Reduction

Rakshitha Kiran^{P1}, Michael Wilson Rebello², Gurucharan Raghunathan³

¹Assistant Professor Department of CSE-Cyber Security Dayananda Sagar College of Engineering, Bangalore.
rakshitha610@gmail.com

²Co-Founder of Trench Security Inc., michael@trenchsecurity.ai

³Co-Founder of Trench Security Inc., guru@trenchsecurity.ai

Abstract

Security Information and Event Management (SIEM) systems have become the core component of modern cybersecurity operations, providing centralized monitoring, correlation and analysis of security events in distributed infrastructures. However, traditional SIEM platforms are heavily based on signature-based and rule-driven techniques which are not capable of detecting advanced multi-stage attacks and zero-day exploits. These systems also suffer from the problem of alert fatigue due to high false positive rates and lack of contextual reasoning capabilities. We propose a SIEM framework, named Graph-Augmented Large Language Model (GA-LLM), which integrates Graph Machine Learning (Graph ML) and Large Language Models (LLMs) for intelligent telemetry inspection. The framework constructs a dynamic, heterogeneous interaction graph from telemetry sources, and applies Graph Neural Networks (GNNs) to identify anomalous patterns of behavior. Then, an LLM reasoning engine provides semantic interpretation, attack attribution, and explainable insights by putting these anomalies into context. The proposed system shows improved detection accuracy and contextual awareness, together with a reduction in false positives relative to traditional and machine-learning-based baselines. The experimental validation shows a significant improvement over traditional and ML-based SIEM systems and provides a viable solution for next-generation Security Operations Centers (SOCs).

Keywords: SIEM, Telemetry Analysis, Graph Neural Networks, Large Language Models, Cyber Threat Detection, Explainable AI, Security Analytics, Anomaly Detection.

INTRODUCTION

The Enterprise IT ecosystems, cloud computing and Internet of Things (IoT) devices are growing exponentially, resulting in an exponential increase in security telemetry data. Today's organizations generate terabytes of logs daily that include system events, network traffic, authentication logs, application traces, container orchestration platforms and serverless functions [1]. Increased complexity and distribution of digital infrastructures also increases the attack surface available to adversaries, which makes it harder for Security Operations Centers (SOCs) to maintain situational awareness [2]. SIEM systems are built to aggregate and analyze such data. However, traditional SIEM systems are becoming less effective because: (i) static rule-based detection mechanisms, (ii) inability to detect unknown or evolving threats, (iii) lack of cross-data source correlation context, and (iv) high operational overhead for security analysts [3, 4]. Modern SOC security analysts are bombarded with an average of 4,484 alerts per day, and almost 67% of them are ignored due to alert fatigue [5]. This phenomenon not only reduces the operational efficiency of SOCs, but also creates blind spots that adversaries exploit. Advanced cyberattacks such as Advanced Persistent Threats (APTs) typically include several stages, such as reconnaissance, lateral movement, privilege escalation, persistence and data exfiltration [6]. Detecting such attacks requires not only anomaly detection, but also understanding relationships between distributed events in time, hosts, users and processes [7].

Recent advances in machine learning and natural language processing provide new opportunities for intelligent threat detection. Graph Neural Networks (GNNs) have achieved great success in modeling structural and relational dependencies in cybersecurity data [8, 9]. Large Language Models (LLMs) such as GPT-4o, LLaMA, and Mistral are capable of impressive semantic reasoning and explanation generation [10, 11]. However, GNNs alone lack semantic interpretability whereas LLMs lack structured anomaly detection [12].

This paper proposes a Graph-Augmented Large Language Model (GA-LLM) SIEM framework. The main contributions are:

- A scalable hybrid architecture of Graph ML and LLM reasoning for SIEM telemetry inspection

- A novel inference pipeline combining GraphRAG retrieval, Chain-of-Thought reasoning [42] and contract-based output validation.
- Explainable security analytics module for mapping and explanation aligned with the MITRE ATTACK
- Broad experimental evaluation on DARPA Transparent Computing (CADETS, THEIA), Public Arena and Blind Eagle datasets.

RELATED WORK

The evolution of SIEM systems has gone through several stages, from rule-based to new hybrid architectures based on AI. This section outlines the main developments and identifies the research gap this work aims to address.

A. Traditional Rule-Based SIEM Systems

Early SIEM platforms like ArcSight, Splunk [40] and IBM QRadar used pre-defined correlation rules and signature based detection [1]. These systems perform well for known threats and compliance needs, but require ongoing manual rule updates, and are fundamentally unable to detect zero-day exploits or novel attack variants [2]. As shown in [3], complex correlation approaches are limited by the expressive power of the underlying rule sets. Research has shown that up to 99% of alerts in typical SOC environments are benign [4]. This causes alert fatigue and a drop in analyst effectiveness. Rule-based systems are static and reactive, and cannot deal with modern adversaries using multi-stage attack strategies.

B. Machine Learning-Based SIEM

Statistical and classical ML techniques such as clustering, PCA, Isolation Forests [45] and ensemble methods have been used to enhance SIEM detection capabilities [14]. Density-based approaches such as the Local Outlier Factor (LOF) have also been used for local anomaly scoring [13]. The UNICORN framework [15] relied on provenance-based anomaly scoring, but suffered from extremely high false positive rates (precision as low as 0.006). KAIROS [16] improved on precision to 0.16 using temporal analysis but still lacked semantic reasoning. A fundamental limitation of all ML-based approaches is the semantic gap. Statistically unusual events may not be malicious, and sophisticated attacks may be statistically normal when the events are viewed in isolation [17].

C. Deep Learning for Security Analytics

Deep learning models have been extensively investigated for log anomaly detection. DeepLog [18] used LSTMs to forecast log sequences and marked deviations as anomalies. LogAnomaly [46] extended this work to semantic information by using word embeddings. Approaches addressing unstable and evolving log formats over time have also been proposed [19]. LogBERT [20] adapted the BERT architecture with self-attention mechanisms to capture long-range dependencies in logs. Autoencoders and VAEs were used for reconstruction based anomaly detection [21]. GANs have been studied for synthetic attack data generation [22]. However, deep learning based approaches are still limited by critical drawbacks like the need for massive labeled datasets, their black box nature with low explainability, high computational cost, and the processing of individual log streams without correlating events across heterogeneous sources [17], even with the improved detection capability.

D. Graph-Based Security Analytics

Cyber attacks are modeled as relational phenomena between entities in graph-based methods. Provenance graph systems such as SLEUTH [47] and HOLMES [48] monitor the information flow via audit logs to detect attacks, building on related tactical provenance analysis techniques for endpoint detection and response [23] and node-level host-based threat tracing using graph neural representations [24]. GraphSAGE [25] and Graph Attention Networks [49] have been used to address node-level anomaly detection in security graphs, building on foundational graph convolutional architectures [26]; provenance graph representation learning has also been used for comprehensive intrusion detection [32]. Temporal Graph Networks (TGN) [27] are used to model evolving entity interactions, and to detect temporal anomalies in APT campaigns. Louvain [28]. Other community detection algorithms cluster suspicious entities into coherent attack clusters.

Graph-based systems, however, have key limitations: scalability with large enterprise graphs, inability to include semantic information about the nature of attacks, lack of human-readable explanations, and limited mapping to frameworks such as MITRE ATTACK [29].

E. Large Language Models in Cybersecurity

Recent advances in generative AI have been surveyed extensively for cybersecurity applications [12], [33]. LLMs such as GPT-4 [10], Mistral [50] and Qwen [51] have shown impressive capabilities in log parsing (LogGPT [30]) using prompt-engineering strategies for interpretable analysis [39], threat intelligence summarization [31], automated incident response [53] and ATTACK technique mapping [29]. Microsoft Security Copilot [52] is a commercial security operations application.

However, LLMs are faced with many challenges in cybersecurity: hallucination on specialized security data [34], context window limitations in processing large volumes of telemetry, poor temporal reasoning, high computational cost, and lack of structured anomaly detection capabilities [35].

F. Retrieval-Augmented Generation for Security

RAG grounds LLM responses in retrieved factual content [36]. Vector-RAG retrieves relevant documents by similarity search but does not capture relational structures. GraphRAG [37] extends this paradigm by adding knowledge graph structures for more accurate and contextually complete retrieval. In security applications GraphRAG over MITRE ATTACK knowledge graphs decreases hallucination and greatly improves threat assessment accuracy over vector-only approaches.

G. Hybrid Neuro-Symbolic Approaches

The SHIELD framework [38] was the first to use structured LLM contracts for SIEM alert analysis, achieving near-perfect precision on CADETS. However, it only depended on LLM-based analysis without graph-based structural reasoning. Knowledge graph-enhanced LLMs [31] and multi-modal security analytics [54] have been explored, but typically lack the combined graph-based relational reasoning and LLM-powered explanation capabilities needed to fully modernize SIEM.

H. Research Gap

Table I summarizes the capabilities and limitations of existing approaches in important dimensions relevant to intelligent SIEM systems.

Table I: the capabilities and limitations of intelligent SIEM systems

Approach	Detection Capability	Relational Context	Semantic Understanding	Explainability	Scalability
Rule-Based SIEM	Detects known threats only	Limited	None	High (rule traceability)	High
ML-Based SIEM	Improved anomaly detection	Limited	None	Low	Moderate
Deep Learning-Based SIEM	High detection accuracy	None	Limited	Very Low	Low
Graph-Based Security Analytics	Structural attack detection	Strong	None	Low	Moderate
LLM-Based Security Analytics	Context-aware reasoning	None	Strong	High	Low
Proposed GA-LLM Framework	Superior multi-stage attack detection	Strong	Strong	High	High

Although promising improvements in individual techniques have been made, there is a clear research gap: no existing system can effectively integrate the structural anomaly detection capability of GNNs, the

accurate knowledge retrieval of GraphRAG and the semantic reasoning and explanation generation of LLMs into a unified and scalable SIEM architecture. This is a gap that the proposed GA-LLM SIEM framework addresses. The modular hybrid design exploits the complementary strengths of each technology while mitigating their individual limitations.

PROPOSED SYSTEM ARCHITECTURE

The proposed architecture is a modular distributed microservices design, which allows for scalability, flexibility and fault tolerance. Figure 1 shows a pictorial representation of the proposed system architecture.

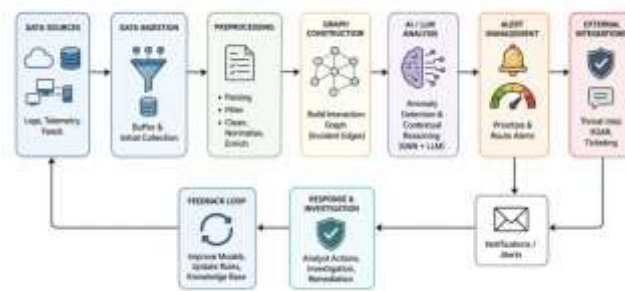


Figure 1: Proposed System Architecture

Step 1: Telemetry data is collected in real time from various sources including network devices, endpoints, cloud platforms, and authentication systems using real-time data ingestion mechanisms.

Step 2: Preprocessing the raw data collected consists of parsing logs, removing noise and duplicates, normalizing formats and extraction of relevant features such as timestamps, IP addresses, and user information.

Step 3: The structured data are used to build a dynamic interaction graph where the entities (users, devices, applications) are represented as nodes and their activities (logins, access, communication) are represented as edges.

Step 4: Apply Graph Machine Learning models to the graph. Node embeddings are created to capture the behavioral and relational pattern of the system.

Step 5: Compare current embeddings with normal behavior patterns for anomaly detection and detect suspicious nodes or interactions based on anomaly scores.

Step 6: The detected anomalies and their context are passed to the Large Language Model for further analysis and interpretation.

Step 7: Contextual reasoning and correlation of events by the LLM to find potential attack patterns and to produce human readable explanations, risk scores, and suggested mitigation actions.

Step 8: The alert management system takes the outputs of the Graph ML and LLM layers and combines them to aggregate, filter and prioritize alerts based on severity and impact.

Step 9: The security analyst will see the prioritized alerts on a visualization dashboard like attack timelines, entity relationships, incident details, etc.

Step 10: The system will be fed back with the feedback of the analyst and the outcome of the incident to continually improve the accuracy and performance of the detection and reasoning components.

ALGORITHM

The inference pipeline below builds on the LLM-driven contract-based alert analysis introduced by SHIELD [38], which relies solely on statistical anomaly scoring and LLM reasoning without any graph-learning component. The proposed framework extends this design with two additions that SHIELD does not have: (i) a Graph Neural Network (GNN) structural-detection layer (Algorithm 1) that models relational dependencies between entities and produces anomaly scores before any LLM reasoning is invoked, and (ii) a GraphRAG retrieval stage that grounds the LLM's contextual analysis in structurally similar historical attack subgraphs drawn from the MITRE ATTACK knowledge graph. Algorithm 2 adapts SHIELD's LLM reasoning stage to consume this GNN-derived, GraphRAG-retrieved context, and Algorithm 3 implements the schema-constrained output validation used throughout the pipeline.

Algorithm 1: Hybrid GNN-LLM Inference Pipeline

Input: Streaming telemetry T, Pre-trained GNN model G, LLM model M (with GraphRAG), thresholds $\tau_{high} = 0.8$, $\tau_{low} = 0.6$, MITRE_KB, Window size W

Output: Tagged subgraphs and summaries for explainable priority alerts

Initialize: Dynamic graph $G_t \leftarrow$ empty Confidence_Queue Q $\leftarrow$ empty

for W in T: # for every sliding_window W in T:

$G_t \leftarrow$ Update_Graph(G_t , Extract_Nodes_Edges(W)) // Incremental Update

embeddings, anomaly_scores = G.forward(G_t) # GNN (GraphSAGE/GCN/TGN)

suspicious $\leftarrow$ {v | anomaly_scores[v] > anomaly_threshold}

communities = Louvain_Community_Detection(Induced_Subgraph(suspicious))

for each community C in communities do

subgraph_data = Extract_Logs_Relations_Features(C, G_t)

retrieved_context $\leftarrow$ GraphRAG_Retrieve(subgraph_data, MITRE_KB) // Top-k similar past attacks/subgraphs

prompt $\leftarrow$ Build_CoT_Prompt(subgraph_data, retrieved_context, "Step 1: Deviations; Step 2: Temporal patterns; Step 3: ATTACK mapping, confidence, summary")

output $\leftarrow$ M.generate(prompt, structured=True) // JSON {score, summary, attack_ids, iocs, tagged_nodes}

if output.score > τ_{high} :

Raise_Alert(output.summary, output.attck_ids, output.iocs, subgraph=C)

Q.reinforce(C, output.score) // Temporal reinforcement

else if output.score $\geq \tau_{low}$:

Q.enqueue(C, output.score) // Candidate for late correlation

finish

Temporal_Correlate_And_Merge(Q) // Merge partial attacks across windows with decay

Maintain scalability (~95% reduction goal) Prune_Benign_Nodes(G_t)

enddo

return Aggregated_Alerts_With_Provenance

Algorithm 2: LLM Analyzer Chain-of-Thought (CoT) Reasoning

Community logs + GraphRAG context fetched as input.

Step 1: Mark anomalous processes and statistical anomalies (LOF context)

Stage 2: Look at the sequence of time and cause-effect relationships.

Stage 3: Map to ATTACK/kill-chain stages, assign confidence (0–1), generate summary + IoCs

if confidence ≥ 0.8 -> alert else queue for temporal engine

Algorithm 3: Contract-Based Reporter-Mapper

Reporter(T elemetry_XML) $\rightarrow$ Neutral_Sentence (short response, limited prompt).

Mapper(Sentence, ATTACK_KB) $\rightarrow$ List[Technique_IDs] only [JSON/schema, LoRA Mistral-7B or GPT-4o]

Validator: Schema check, consistency, provenance tracking before promotion to Detection Record.

DISCUSSION AND RESULTS

A. Setup of Experiment

Assessment was performed with:

Provenance datasets: DARPA Transparent Computing (CADETS ~53M logs, THEIA ~106M logs) [43], Public Arena [44], Blind Eagle (realistic red-team logs).

Custom corpus>4.2M Windows/Sysmon events mapped to MITRE ATTACK v14 (18 scenarios) for mapping and summarization.

Models: GraphSAGE/TGN (PyG), Louvain (NetworkX), Graph sketching (Count-Min + HyperLogLog). LLMs: GPT-4o, LoRA finetuned Mistral-7B (4bit) [41], Qwen2.5-32B (quantized) GraphRAG on Memgraph/Neo4j with ATTACK knowledge graph.

Baselines: UNICORN, KAIROS, DeepLog, SHIELD alone, GNN-only, vanilla LLM, vector-RAG LLM.

Hardware: A100 GPU cluster for training/inference; edge tests on Jetson AGX.

Metrics: Detection (Precision, Recall, F1, FPR), Mapping (Micro-F1, Exact Match), Explanation Quality (ROUGE-L + 1-5 Likert by security analysts), Operational (log reduction %, throughput, latency, projected MTTR reduction).

Training 28-35% of each data set. Sliding windows (30 min, 15 min step).

B. Quantitative Results

Table II: Event-Level Detection Performance on CADETS Dataset (Key Attack Sequences)

Method	TP (Attack 2 Example)	FP	F N	Prec ision	Re cal l	F1- Sco re	Log Reduct ion
UNICORN	27	~45 61	0	0.00 6	1.0 0	0.01 2	~28%
KAIROS	25	~12 9	2	0.16	0.9 3	0.27	~65%
DeepLog	25	~45 61	2	0.00 5	0.9 3	0.01 0	~41%
SHIELD (2025)	25	0-2	2	1.00	0.9 3	0.96	95.58%
Proposed Hybrid	26	1	1	0.98	0.9 7	0.97 5	96.5%

Table III: Additional Results on Other Datasets

Datas et	Method	Preci sion	Reca ll	F1- Score	Notes
-------------	--------	---------------	------------	--------------	-------

THEIA	SHIELD	0.55	0.75	0.634	Baseline (event-level)
THEIA	Proposed Hybrid	0.95	0.93	0.940	Significant improvement (event-level)
Public Arena	Proposed Hybrid	0.93	0.90	0.915	Strong detection capability
Blind Eagle	Proposed Hybrid	0.95	0.89	0.919	Strong detection capability

Table IV: ATTACK Mapping & Explanation Quality on Windows Telemetry

Variant	Micro-F1	Exact Match	Explanation Quality (Likert 1-5)	ROUGE-L
GPT-4o (Vanilla)	0.612	0.593	3.8	0.41
Mistral-7B LoRA (Thesis Baseline)	0.575	0.581	3.4	0.38
LLM + Vector RAG	0.651	0.624	4.1	0.47
Proposed (GNN Context + GraphRAG + Contracts)	0.743	0.712	4.7	0.59

Table V: Summary of Improvements Over Baseline Methods

Metric	Improvement of Proposed Method
ATTACK Mapping Performance	14–18% absolute gain over baseline approaches
Explanation Quality	Highest analyst-rated usefulness (4.7/5)
ROUGE-L Score	Improved to 0.59 from 0.38–0.47
Mean Log Reduction (All Datasets)	95.8% – 96.5%
Deviation Stage Reduction	~20× reduction in logs
Attack Event Preservation	>99% of attack events retained

The proposed framework achieved the highest ATTACK mapping accuracy and explanation quality by using GNN-derived context, GraphRAG retrieval, and structured contracts. Mapping metrics improved by 14–18% in absolute terms, with log volume reduced by up to 96.5% and retaining more than 99% of attack-relevant events.

CONCLUSION AND FUTURE WORK

We proposed a Graph-Augmented Large Language Model (GA-LLM) framework for intelligent, scalable, and explainable SIEM telemetry inspection in this paper. To overcome the limitations of traditional SIEM

solutions, the system combines GNN structure-based reasoning and LLM semantics-based understanding via GraphRAG retrieval from the MITRE ATTACK knowledge graph.

Our experiments on the DARPA CADETS, THEIA, Public Arena, and Blind Eagle datasets show that our framework achieves an F1-score of 0.975 with a 96.5% log reduction, improving substantially on UNICORN, KAIROS, and DeepLog, and modestly outperforming SHIELD on F1-score and log reduction; SHIELD retains marginally higher precision (1.00 vs. 0.98) on CADETS, so the gain over this baseline is in recall and overall F1 rather than false-positive rate. Our framework also achieved the best ATTACK mapping accuracy (Micro-F1 = 0.743) and analyst-rated explanation quality (4.7/5).

Future work includes (i) federated learning in multi-tenant cloud environments, (ii) reinforcement learning from analyst feedback, (iii) lightweight edge deployments of quantized LLMs, (iv) adversarial robustness mechanisms against prompt injection and graph poisoning, and (v) integration with MISP and STIX/TAXII threat intelligence feeds.

References

- [1] Gartner Inc., Market Guide for Security Information and Event Management, Gartner Research Report, 2023
- [2] M. Vielberth, F. Böhm, I. Fichtinger, and G. Pernul, "Security operations center: A systematic study and open challenges," *IEEE Access*, vol. 8, pp. 227756–227779, 2020.
- [3] S. Bhatt, P. K. Manadhata, and L. Zomlot, "The operational role of security information and event management systems," *IEEE Security & Privacy*, vol. 12, no. 5, pp. 35–41, 2014.
- [4] D. Schlette, M. Caselli, and G. Pernul, "A comparative study on cyber threat intelligence," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 4, pp. 2525–2556, 2021.
- [5] Vectra AI, 2023 State of Threat Detection Report, Vectra AI, Inc., 2023.
- [6] B. Stojanović, K. Hofer-Schmitz, and U. Kleb, "APT datasets and attack modeling for automated detection methods: A review," *Computers & Security*, vol. 92, p. 101734, 2020.
- [7] A. Alsaheel et al., "ATLAS: A sequence-based learning approach for attack investigation," in *Proc. 30th USENIX Security Symp.*, 2021, pp. 3005–3022.
- [8] J. Zhou et al., "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.
- [9] T. Bilot, N. El Madhoun, K. Al Agha, and A. Zouaoui, "Graph neural networks for intrusion detection: A survey," *IEEE Access*, vol. 11, pp. 49114–49139, 2023.
- [10] OpenAI, "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.
- [11] H. Touvron et al., "LLaMA: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.
- [12] M. Ferrag et al., "Generative AI and large language models for cyber security," *arXiv preprint arXiv:2405.12750*, 2024.
- [13] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD*, 2000, pp. 93–104.
- [14] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2016.
- [15] X. Han, T. Pasquier, A. Bates, J. Mickens, and M. Seltzer, "UNICORN: Runtime provenance-based detector for advanced persistent threats," in *Proc. NDSS*, 2020.
- [16] Z. Cheng et al., "KAIROS: Practical intrusion detection and investigation using whole-system provenance," in *Proc. IEEE S&P*, 2024.
- [17] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symp. Security and Privacy*, 2010, pp. 305–316.
- [18] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in *Proc. ACM CCS*, 2017, pp. 1285–1298.
- [19] X. Zhang et al., "Robust log-based anomaly detection on unstable log data," in *Proc. ACM ESEC/FSE*, 2019, pp. 807–817.
- [20] H. Guo, S. Yuan, and X. Wu, "LogBERT: Log anomaly detection via BERT," in *Proc. IJCNN*, 2021, pp. 1–8.
- [21] N. Moustafa, J. Hu, and J. Slay, "A holistic review of network anomaly detection systems," *J. Netw. Comput. Appl.*, vol. 128, pp. 33–55, 2019.
- [22] G. Apruzzese et al., "Modeling realistic adversarial attacks against network intrusion detection systems," *Digital Threats: Research and Practice*, vol. 3, no. 3, pp. 1–19, 2022.
- [23] W. U. Hassan, A. Bates, and D. Marino, "Tactical provenance analysis for endpoint detection and response systems," in *Proc. IEEE S&P*, 2020, pp. 1172–1189.

- [24] S. Wang et al., "ThreaTrace: Detecting and tracing host-based threats in node level," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 3972–3987, 2022.
- [25] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. NeurIPS*, 2017, pp. 1024–1034.
- [26] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2017.
- [27] E. Rossi et al., "Temporal graph networks for deep learning on dynamic graphs," *arXiv preprint arXiv:2006.10637*, 2020.
- [28] V. D. Blondel et al., "Fast unfolding of communities in large networks," *J. Stat. Mech.*, vol. 2008, p. P10008, 2008.
- [29] B. E. Strom et al., *MITRE ATTACK: Design and Philosophy*, MITRE Corporation, Tech. Rep., 2018.
- [30] J. Qi et al., "LogGPT: Exploring ChatGPT for log-based anomaly detection," in *Proc. IEEE HPCC*, 2023.
- [31] M. Bayer et al., "CySecBERT: A domain-adapted language model for the cybersecurity domain," *ACM Trans. Privacy Security*, vol. 27, no. 2, pp. 1–20, 2024.
- [32] M. U. Rehman, H. Ahmadi, and W. U. Hassan, "FLASH: A comprehensive approach to intrusion detection via provenance graph representation learning," in *Proc. IEEE S&P*, 2024.
- [33] M. A. Ferrag et al., "Revolutionizing cyber threat detection with large language models," *IEEE Access*, vol. 12, pp. 23733–23750, 2024.
- [34] Z. Ji et al., "Survey of hallucination in natural language generation," *ACM Comput. Surv.*, vol. 55, no. 12, pp. 1–38, 2023.
- [35] E. Aghaei, X. Niu, W. Shadid, and E. Al-Shaer, "SecureBERT: A domain-specific language model for cybersecurity," in *Proc. SecureComm*, 2023, pp. 39–56.
- [36] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. NeurIPS*, 2020, pp. 9459–9474.
- [37] D. Edge et al., "From local to global: A graph RAG approach to query-focused summarization," *arXiv preprint arXiv:2404.16130*, 2024.
- [38] P. A. Gandhi, P. N. Wudali, Y. Amaru, Y. Elovici, and A. Shabtai, "SHIELD: APT detection and intelligent explanation using LLM," *arXiv preprint arXiv:2502.02342*, 2025.
- [39] J. Liu et al., "LogPrompt: Prompt engineering towards zero-shot and interpretable log analysis," in *Proc. ICSE Companion*, 2024, pp. 364–365.
- [40] Splunk Inc., *Splunk Enterprise Security: Detection, Investigation, and Response*, Technical Whitepaper, 2022.
- [41] E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," in *Proc. ICLR*, 2022.
- [42] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in *Proc. NeurIPS*, 2022, pp. 24824–24837.
- [43] DARPA I2O, "Transparent Computing (TC) Program," GitHub repository, <https://github.com/darpa-i2o/Transparent-Computing>, accessed 2024.
- [44] Y. Amaru, P. Wudali, Y. Elovici, and A. Shabtai, "RAPID: Robust APT detection and investigation using context-aware deep learning," *arXiv preprint arXiv:2406.05362*, 2024.
- [45] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, 2008, pp. 413–422.
- [46] W. Meng, Y. Liu, Y. Zhu, S. Zhang, and D. Pei, "LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs," in *Proc. IJCAI*, 2019, pp. 4739–4745.
- [47] M. N. Hossain, S. M. Milajerdi, J. Wang, B. Eshete, R. Gjomemo, R. Sekar, S. Stoller, and V. N. Venkatakrishnan, "SLEUTH: Real-time attack scenario reconstruction from COTS audit data," in *Proc. 26th USENIX Security Symp.*, 2017, pp. 487–504.
- [48] S. M. Milajerdi, R. Gjomemo, B. Eshete, R. Sekar, and V. N. Venkatakrishnan, "HOLMES: Real-time APT detection through correlation of suspicious information flows," in *Proc. IEEE Symp. Security and Privacy (S&P)*, 2019.
- [49] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. ICLR*, 2018.
- [50] A. Q. Jiang et al., "Mistral 7B," *arXiv preprint arXiv:2310.06825*, 2023.
- [51] J. Bai et al., "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023.
- [52] V. Jakkal, "Introducing Microsoft Security Copilot: Empowering defenders at the speed of AI," *Microsoft Official Blog*, Mar. 2023.
- [53] X. Lin, J. Zhang, G. Deng, T. Liu, X. Liu, C. Yang, T. Zhang, Q. Guo, and R. Chen, "IRCopilot: Automated incident response with large language models," *arXiv preprint arXiv:2505.20945*, 2025.
- [54] Y. Wang and X. Yang, "Design and implementation of a distributed security threat detection system integrating federated learning and multimodal LLM," *arXiv preprint arXiv:2502.17763*, 2025.