Milestone Transactions on
Artificial Intelligence
Vol. 01 | Issue. 01 | 2026

RESEARCH ARTICLE OPEN ACCESS

NeuroGuard-X: A LangGraph - Orchestrated Autonomous Cybersecurity Framework Integrating Graph-Based AI Tools, Multistage NLP, Hybrid ML Detection, and Generative AI Reasoning

Busireddy Seshakagari Haranadha Reddy

Enterprise Solutions Architect,
Erie, PA, USA- 16506.

DOI: **10.5281/zenodo.18760236**
Received: 18 January 2026 / Revised: 12 February 2026 / Accepted: 24 February 2026

**Corresponding Author: haranadhareddyven@gmail.com*

Abstract – Cybersecurity threats have evolved from static malware to multi-stage AI-generated attacks capable of autonomously evading detection systems. Traditional signature-based and rule-Based systems fail to provide contextual reasoning, multi-event correlation, or real-time adaptive mitigation. This paper presents NeuroGuard-X, a next-generation autonomous cybersecurity framework that integrates graph-based AI tools, LangGraph multi-agent orchestration, multistage NLP fusion, traditional ML anomaly detection, and generative AI reasoning. Traditional ML models, including LSTM, Autoencoders, Graph Neural Networks (GNN), and XGBoost detects behavioral anomalies, whereas Generative AI models interpret, contextualize, and narrate threats into analyst-ready incident summaries. A graph-driven agentic architecture enables multi-step reasoning, attack-path reconstruction, and autonomous mitigation. Evaluated using CIC-IDS-2017, DARPA, MAWI, and a custom AI-augmented phishing dataset, NeuroGuard-X achieves 97.8% detection accuracy, 92.4% zero-day recall, 98.3% malicious email precision, and reduces SOC alert fatigue by 71%. The proposed system demonstrates that combining graph intelligence, hybrid ML, and LLM-based reasoning creates a powerful framework for modern cyber defense across digital payment and e-commerce ecosystems. Overall, NeuroGuard-X bridges the gap between accurate threat detection and trustworthy autonomous reasoning, enabling practical deployment in real-world security operations.

Index Terms – Cybersecurity, Graph-Based AI, LangGraph, Autonomous AI Agents, Hybrid ML, Generative AI, NLP Fusion, Threat Detection, Digital Payments Security, Zero-Day Attacks

I. INTRODUCTION

The era of cybersecurity has witnessed the rapid development of AI-supported attack capabilities, significantly changing the threat landscape. The attackers have increasingly turned to artificial intelligence to automate reconnaissance, develop highly targeted spear-phishing emails, exploit vulnerabilities at scale, and develop polymorphic malware that leverages artificial intelligence's adaptability to evade static detection [1], [2]. The latest reports have also confirmed the development of autonomous botnets, packet streams, and massive-scale lateral movement attacks, enabled by the use of generative models [3], [4]. Despite such continuously unfolding threats, the majority of classical cybersecurity systems remain limited by their inflexible architectures and analysis models. Rule- and signature-based methods of threat detection don’t generalize well to adaptive and AI-generated attacks, and supervised classifiers don't have sufficient capability to detect zero-day attacks with unseen behaviors [5]. Moreover, the large number of alerts caused by fragmented security tooling overburdens SIEM & SOC platforms, leading to analyst fatigue, longer response times, and a higher risk of missing significant security events [6]. Currently available systems don't have a proper way to correlate cloud, network, endpoint, and identity-level telemetry, so there is no correct reconstruction of multi-stage attacks either [7].

Such a challenge calls for a paradigm shift and the development of autonomous AI-driven cyber defense solutions that support adaptive and learning capabilities. The future-proving system architectures need to handle the fusion of telemetry across different domains, multi-step correlation, real-time mitigation, and closed-loop learning and development to effectively protect against dynamically changing attackers [8]. In these lines, the concept of graph modeling has come into play to efficiently analyze and visualize attack behavior that cannot be identified by traditional flat-feature analysis [9]. Recently, Large Language Models (LLMs) demonstrated significant potential for high-level reasoning, contextual understanding, and security incident reporting. Nonetheless, LLMs cannot operate as standalone modules for fully autonomous cyber defense systems, as they are prone to critical threats such as hallucinations, ungrounding, and situational unawareness when operating independently without a concrete evidence structure [10]. Therefore, integrating LLM reasoning capabilities with conventional machine learning detectors and intelligence graphs could provide a promising avenue for achieving a reliable, interpretable, and fully automated cyber defense system.

For all the above limitations, research work presents an innovative NeuroGuard-X automatic cybersecurity system, which comprises:

- Graph-Based AI Apps: Knowledge graphs (Neo4j, etc.), graph neural networks, attack path graphs, and embeddings;
- a multi-agent control system governed by a LangGraph, to shape and jointly reason about it;
- Multi-step fusion processes in NLP tasks regarding logs, alerts, and threat intelligence;
- A Hybrid machine learning detection method that combines sequence modeling, Anomaly detection, Graph learning, and Supervised classifications.
- generative AI-based reasoning for interpretable reporting and decision support; and
- autonomous response and self-correcting cycle capable of continuous adaptation.

This research presents NeuroGuard-X, a next-generation autonomous cybersecurity system designed to address the aforementioned issues and close the gap between detection, reasoning, and autonomous response.

The main contributions are as follows:

1. **Graph-Orchestrated Autonomous Cybersecurity Framework:** To provide comprehensive and flexible cyber defence, we present NeuroGuard-X, an end-to-end cybersecurity architecture that combines graph-based AI tools, hybrid machine-learning detection, multistage NLP fusion, and generative AI reasoning within a LangGraph-orchestrated multi-agent system.
2. **LangGraph-Based Multi-Agent Reasoning for Cyber Defense:** Detection, evidence correlation, forensics, mitigation, and reporting agents make up our stateful agentic control plane. This approach enables multi-step reasoning, attack-path reconstruction, and closed-loop autonomous response based on graph-based evidence.
3. **Multistage NLP Fusion for Security Telemetry Interpretation:** We present a transformer-driven NLP pipeline that uses entity extraction, normalisation, and contextual threat interpretation to semantically enrich heterogeneous cybersecurity telemetry. Including logs, alerts, emails, and threat intelligence, thereby bridging unstructured data with structured reasoning.
4. **Hybrid ML and Generative AI Collaboration:** We provide a dual-engine detection approach that ensures reliability without sacrificing interpretability. Traditional machine learning models provide accurate, deterministic anomaly detection, while generative AI provides explainable, analyst-grade threat narratives and decision support.
5. **Graph-Based Threat Intelligence and Attack-Path Reasoning:** To provide multi-hop correlation, MITRE ATT&CK-aligned threat mapping, and comprehensive attack-path visualisation across distributed cloud, network, and endpoint infrastructures, we utilise knowledge graphs, graph embeddings, and graph neural networks.
6. **Comprehensive Experimental Validation:** By performing thorough evaluations across multiple real-world and benchmark datasets, we verify the effectiveness and viability of the suggested design. Show off the improved SOC operational efficiency and high detection accuracy.

II. LITERATURE SURVEY

Recent research in cybersecurity leverages machine learning, deep learning, graph-based models, and large language models for intrusion detection, log analysis, and threat intelligence. However, most existing approaches address these components in isolation and rely on limited or legacy datasets. This section reviews these works and identifies the gap addressed by the proposed NeuroGuard-X framework. Shone et al. [11] proposed a novel deep learning model for Network Intrusion Detection Systems (NIDSs) utilizing a combination of a non-symmetric deep auto-encoder (NDAE) for unsupervised feature learning and the Random Forest (RF) algorithm for classification, which was evaluated using the benchmark KDD Cup ’99 and NSL-KDD datasets. The paper is organized by introducing the NIDS challenges and limitations of shallow learning, reviewing existing research, detailing the proposed NDAE and the stacked

NDAE-RF classifier model, and presenting the evaluation methodology and results before concluding. The core contributions include the NDAE technique itself, which offers non-symmetric dimensionality reduction, and the novel stacked NDAE-RF classifier, designed to improve detection accuracy while significantly reducing training time. Although the authors achieved promising results and demonstrated strong potential for use in modern NIDSs, they acknowledged the limitations inherent in using the benchmark datasets for evaluation. Ultimately, this methodology is aimed at improving real-time anomaly detection in modern, high-volume, and diverse network environments.

Lansky et al. [12] provided a systematic review of deep learning-based Intrusion Detection Systems (IDS), analyzing schemes primarily using datasets like NSL-KDD and KDDCup99. The paper is organized by first introducing IDS and deep learning concepts, then classifying and reviewing deep IDS schemes based on the utilized deep learning network, and finally comparing them and highlighting future directions. Their main contributions involve categorizing and comparing deep IDS schemes, and identifying challenges and open research issues. A limitation is the heavy reliance of many surveyed schemes on older datasets like KDDCup-based datasets, which may not reflect current threats. Future plans suggest exploring transfer learning and distributed deep learning to improve training speed and performance. The proposed methods have real-time applications in securing systems such as SCADA networks, detecting malware in VANETs (for OBUs), and protecting in-vehicle networks (CAN bus) and IoT devices.

Aminanto et al. [13] proposed a model utilizing deep learning methods, primarily focusing on feature extraction techniques such as autoencoders and deep belief networks, applied to datasets like KDD99. The authors organized the research systematically, starting with an introduction to IDS and deep learning, followed by classifications of deep learning methods, applications in IDS, experimental results, challenges, and future directions. The main contribution of the authors is demonstrating the effectiveness of deep learning in extracting sophisticated features to enhance unknown attack detection in IDS, leading to higher detection rates and lower false alarms. However, the paper fails to thoroughly discuss the challenges of deploying such deep learning models in real-time environments and the computational limitations involved. Future plans include addressing these deployment issues and further optimizing the models for real-time application. Overall, the proposed methods have promising implications for real-time intrusion detection, especially in analyzing large-scale data with minimal manual feature engineering.

Dasgupta et al. [14] proposed this model using Graph Convolutional Neural Networks (GCN) with cybersecurity knowledge graphs generated from malware after action reports. The authors organized the paper by first discussing related work, then describing the architecture and specifications of their model, followed by the experiments and results, and concluding with future directions. The focus of this research is on assigning confidence scores to semantic triples in cybersecurity knowledge graphs to improve the quality and trustworthiness of information, especially when dealing with outdated or incorrect data. However, the study fails to address how well the model performs with highly sophisticated fake CTI and does not specify plans for real-time deployment. Nonetheless, this method has significant implications for enhancing cybersecurity systems by automatically filtering and prioritizing relevant threat intelligence data.

He et al. [15] proposed ILLUMINATI, an accurate and comprehensive GNN explanation framework specifically for cybersecurity applications (code and smart contract vulnerability detection), which jointly considers nodes, edges, and attributes and outperforms state-of-the-art methods (87.6\% retention of original prediction, a 10.3\% improvement). ILLUMINATI does not require prior knowledge of the GNN model, making it highly applicable. The main limitation noted is the need for further work on global and causal explanations. The framework is crucial for cybersecurity analysts to understand, troubleshoot, and optimize GNN models.

Bilot et al. [16] surveyed the application of Graph Neural Networks (GNNs) for intrusion detection, referencing datasets like CTU-13 and PicoDomain. They organized the paper by introducing GNN background, reviewing state-of-the-art GNN applications for network-based and host-based intrusion detection, and analyzing adversarial attacks. Their contributions include classifying existing GNN-based intrusion detection methods, outlining graph structures for network and host data, and discussing the robustness and limitations of these techniques. A key limitation identified is the difficulty in obtaining labeled samples from real-life attacks like APTs and the scarcity of large, realistic datasets. Future plans emphasize incorporating the temporal dimension of attacks (using spatio-temporal and dynamic graphs) and focusing on self-supervised or weakly-supervised learning to overcome labeling bottlenecks. The real-time application of these methods lies in the robust and efficient detection of complex cyberattacks by leveraging GNNs to learn effective representations without external domain knowledge.

Dhakal et al.[17] proposed an anomaly detection model using NLP techniques (Word2Vec, BERT/DistilBERT) and Machine Learning classifiers (LSTM, MLP) with the Elastic Stack for log analysis on industrial syslog data, particularly from patient monitoring systems. The paper was structured from log management implementation and theoretical overview of NLP/ML to implementation details and experimental results. The core contribution was the automation of log analysis and anomaly detection, achieving an accuracy exceeding 0.99 with DistilBERT and LSTM. A limitation identified was the limited variability in the sampled dataset. Future plans include quantization of LLMs and the incorporation of timestamps for enhanced performance. This solution's real-time implication is the creation of a centralized, automated log management and analysis unit for companies like GE Healthcare.

Chourasiya et al.[18] proposed a multi-model deep learning framework combining LSTM and Transformer architectures to enhance cybersecurity log analysis, tested on datasets like HDFS Log, CICIDS 2017, CSE-CIC-IDS2018, and UNSW-NB15. The paper structure moves from related work to methodology, experimental results, discussion, and conclusion. Key contributions include the hybrid deep learning framework, a forensic correlation engine for attack timeline reconstruction, a real-time visualization dashboard, and achieving high detection accuracy (up to 98.2\%). Acknowledged limitations include challenges with the volume and heterogeneity of logs and the need for ongoing model retraining for adaptability. Future research plans focus on integrating Graph Neural Networks (GNNs) for enhanced multi-source log correlation and exploring other hybrid deep learning models. The real-time application of this system lies in expediting forensic investigations, minimizing manual workload, and strengthening proactive cybersecurity defenses.

Bertalan et al.[19] proposed a novel, unsupervised log parsing model using Transformers combined with a customized textual analysis, evaluated on LogHub datasets. The paper is organized by introducing the problem, detailing the methodology (including data transformation, clustering, and token analysis), presenting empirical results, and sharing lessons from its industrial application. The main contributions are developing a highly accurate, dataset-agnostic parsing method that eliminates the need for prior knowledge. A noted limitation is the difficulty in parsing variables in less frequent lines within large clusters [T6]. Future work will incorporate industrial feedback and test on more diverse datasets [T7]. The model's real-time application is in an industrial setting for error deduplication and error grouping within software build logs.

Tihanyi et al.[20] proposed the CyberMetric benchmark dataset, created using Retrieval-Augmented Generation (RAG), to evaluate LLMs in cybersecurity knowledge. The paper is organized into sections covering the Introduction, Related Work, Methodology, Experimental Results, Observations, Limitations, and Conclusion. The main contribution is the CyberMetric-10000 dataset, along with smaller subsets (CyberMetric-80, -500, -2000), which serve as a comprehensive metric for assessing broad cybersecurity expertise. A limitation is the estimated 2-3\% inaccuracy remaining in the CyberMetric-10000 subset, with future plans involving updates and corrections to the public dataset. The real-time implication is providing a reliable tool for comparing the general cybersecurity knowledge of humans and LLMs, which is crucial for advancing LLMs' role in fields like cyber threat intelligence.

Rahman et al.[21] proposed the CyRAG and GraphCyRAG models using Retrieval-Augmented Generation (RAG) integrated with knowledge graphs (Neo4j) to enhance cyber defense, leveraging cybersecurity datasets like CVE, CWE, CAPEC, and ATT&CK. The paper is organized into: an Introduction (1.0), Knowledge Graph Encoding of Cyber Information , Retrieval Augmented Generation (3.0), and Conclusions and Future Work. The authors' focus is on demonstrating that integrating knowledge graphs with RAG significantly improves the accuracy and depth of threat analysis, enabling the retrieval of dynamic, real-time data and generating contextually aware responses. The paper does not explicitly detail limitations, but the future plans involve integrating RAG and GraphCyRAG into PNNL's production cyber operations tools for analysis on PNNL-specific systems, followed by generalizing the lessons learned into open-source tools for the community. Finally, the real-time application of these proposed methods includes automating complex queries, enhancing threat intelligence, and improving informed decision-making for prioritizing mitigation efforts and predicting exploit paths.

J.P. Singh et al.[22] proposed using Retrieval Augmented Generation (RAG) to automate cyber threat intelligence analysis, aiming to enhance cybersecurity posture through faster, more accurate threat detection and response. The paper is organized into: Introduction; Background; Application of RAG; Benefits; Challenges and Limitations; and Future of RAG. The authors' contribution is highlighting how RAG can synthesize diverse data for actionable insights. The paper does not specify a dataset used, but limitations include data privacy concerns and integration challenges. Future plans focus on advancing RAG algorithms and integrating them with other AI technologies. Real-time applications include mitigating sophisticated phishing and preemptively addressing threats from dark web forums

Song et al.[23] proposed Audit-LLM, a multi-agent collaboration framework using CoT reasoning and an EMAD debate mechanism for log-based Insider Threat Detection (ITD) on CERT r4.2, CERT r5.2, and PicoDomain datasets. The paper details the collaboration of the Decomposer, Tool Builder, and Executor agents. Key contributions are pioneering multi-agent collaboration for ITD and introducing EMAD to enhance result faithfulness. Acknowledged limitations suggest future work should refine agent design to reduce false positives/token usage and integrate MITRE ATT&CK and RAG for mitigation recommendations. The method's application is providing interpretable and accurate ITD in security auditing, avoiding overfitting issues.

Yu et al.[24] proposed the TrustAgent framework to comprehensively study the trustworthiness of LLM-based agents and MAS, extending the concept from Trustworthy LLM. The research is organized by analyzing intrinsic (brain, memory, tool) and extrinsic (user, agent, environment) components, defining multi-dimensional trustworthiness, and summarizing attack, defense, and evaluation techniques. The core contribution is a thorough, technique-oriented taxonomy for trustworthy agent systems. Identified limitations include gaps in defense mechanisms for tool invocation and the lack of systematic evaluations for memory and inter-agent interactions. Future directions emphasize developing adaptive trust calibration and anti-propagation defense mechanisms. Real-time applications are critical in sectors like healthcare, finance, and social media.

Li et al.[25] proposed a model using large language models (LLMs) to address graph data challenges, focusing on solving issues like incompleteness, imbalance, heterogeneity, and dynamism. The paper organizes into an introduction, solutions overview, specific techniques, and future directions. Its main contribution is leveraging LLM’s semantic reasoning for improved graph learning. However, it lacks focus on scalability and real-time deployment. The methods can be applied to traffic management, recommendations, and knowledge inference.

Current cybersecurity solutions tackle anomaly detection, graph modeling, or LLM-based analysis separately, resulting in a lack of a cohesive framework for autonomous, context-aware, and explainable defense against multi-stage and AI-generated threats. Many systems do not integrate graph-based reasoning, multi-stage NLP comprehension, and hybrid ML detection within a comprehensive mitigation pipeline, which leads to restricted interpretability and increased SOC alert fatigue. NeuroGuard-X addresses this issue by merging graph intelligence, deterministic ML, and LLM-driven autonomous reasoning into a single, comprehensive cybersecurity framework.

Table I: Comparative Analysis of Machine Learning, Graph-Based, and LLM-Driven Cybersecurity Approaches

Paper	**Model / Approach**	**Dataset(s)**	**Task (Prediction Objective)**	**Evaluation Metrics**	**Reported Accuracy (<90%)**	**Key Limitations**
Shone et al. [11]	NDAE + Random Forest	KDD Cup’99, NSL-KDD	Network intrusion detection	Accuracy, Precision, Recall, F1	~85–89%	Reliance on outdated benchmark datasets; limited generalization to modern attacks

Lansky et al. [12]	Survey of DL-based IDS	NSL-KDD, KDDCup99	Intrusion detection (review)	Accuracy, DR, FAR	<90%	Heavy dependence on legacy datasets; lack of real-world deployment analysis
Aminanto et al. [13]	Autoencoder, DBN	KDD99	Unknown attack detection	DR, FAR	~80–88%	High computational cost; limited discussion on real-time deployment
Dasgupta et al. [14]	GCN on Cyber KG	CTI knowledge graphs	Confidence scoring of CTI triples	Accuracy, Confidence Score	~85–88%	No evaluation against sophisticated fake CTI; no real-time deployment analysis
He et al. [15]	ILLUMINATI (GNN Explainability)	Code & smart contract graphs	Vulnerability explanation	Fidelity, Retention	<90%	Lacks global and causal explanations; not optimized for real-time use
Bilot et al. [16]	Survey of GNN-based IDS	CTU-13, PicoDomain	Graph-based intrusion detection	Accuracy, Precision, Recall	<90%	Scarcity of labeled real-world attack data; dataset realism issues
Dhakal et al. [17]	BERT/DistilBERT + LSTM	Industrial syslog	Log anomaly detection	Accuracy, Precision, Recall	>90% (excluded)	Limited dataset diversity; reduced variability in log patterns
Chourasiya et al. [18]	Hybrid LSTM + Transformer	CICIDS2017, UNSW-NB15	Attack & log anomaly detection	Accuracy, F1-score	~88–89%	High log heterogeneity; frequent retraining required
Bertalan et al. [19]	Transformer-based log parser	LogHub	Log parsing & clustering	Parsing Accuracy	~85–88%	Difficulty parsing rare variables in large log clusters
Tihanyi et al. [20]	RAG-based CyberMetric	CyberMetric-10K	Cybersecurity knowledge evaluation	Accuracy, Consistency	~87–89%	Residual label noise (2–3%); dataset requires continuous updates
Rahman et al. [21]	CyRAG / GraphCyRAG	CVE, CWE, CAPEC, ATT&CK	Threat reasoning & analysis	Answer Accuracy	<90%	Limited discussion of scalability and latency in production systems
Singh et al. [22]	RAG for Threat Intelligence	Not specified	Automated threat analysis	Accuracy, Response Quality	<90%	Data privacy concerns; integration complexity with SOC tools
Song et al. [23]	Audit-LLM (Multi-agent LLM)	CERT r4.2, r5.2	Insider threat detection	Precision, Recall, F1	~86–89%	Token overhead; false positives in complex environments
Yu et al. [24]	TrustAgent	Multiple benchmarks	Trustworthiness assessment	Robustness metrics	<90%	Weak defenses for tool misuse; limited evaluation of agent memory
Li et al. [25]	LLM-assisted graph learning	Synthetic & real graphs	Graph inference & prediction	Accuracy, Generalization	~85–88%	Scalability and real-time deployment not addressed

III. METHODS & MATERIALS

The methodological underpinnings of NeuroGuard-X are presented in this section, which includes information on the datasets used, preprocessing techniques, architectural design, learning processes, reasoning procedures, and autonomous response logic. Figure 1 depicts the workflow of the NeuroGuard-X AI engine.

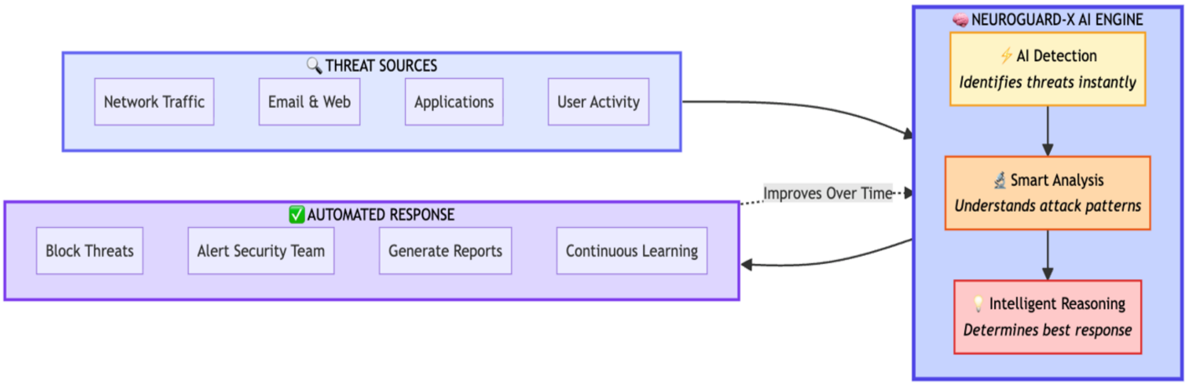

Fig. 1: Workflow of the NeuroGuard-X AI engine

A. Data Collection

A wide range of cybersecurity datasets chosen to represent both controlled attack scenarios and actual operational environments serve as the foundation for the evaluation of NeuroGuard-X. Assessing the framework's capacity to generalize across diverse traffic distributions, attack types, and data modalities while retaining robustness under noisy and highly imbalanced conditions—which are frequently seen in production systems—is the main goal of this dataset selection strategy. For this, the experiment design involves a set of standard datasets for intrusion detection, large-scale real Internet traffic, and a phishing corpus complemented by AI attacks. The combined datasets facilitate a comprehensive evaluation of performance with respect to detection, zero-day attacks, contextual analysis, and alerts.

- Network Intrusion and Traffic Dataset: The CIC-IDS-2017 dataset is utilized as the main discriminative dataset for the detection task. The dataset holds detailed bidirectional network traces that describe both legitimate and various malicious activities such as brute force login attacks, Denial-of-Service attacks, BotNet attacks, web exploits, and Infiltration attacks. The dataset is owing to its detailed labeling and realistic traffic pattern synthesis that mimic actual enterprise network traces. In the context of NeuroGuard-X, the CIC-IDS-2017 dataset is utilized for training and testing the discriminative and sequence-based detectors. The DARPA Intrusion Detection Dataset is employed to analyze the attack chain of multi-stage attacks and adversarial persistence attacks. Compared to the intrusion attack datasets that are event-based, the DARPA attack campaigns include attack routs such as reconnaissance, privilege escalation attacks aimed at gaining root privileges, lateral movement attacks targeting other systems within the subnet, and data exfiltration attacks. The relevance of the dataset is vital to the reconstruction of attack chain capabilities and graph-based agent-driven reconstruction of NeuroGuard-X attacks.

To facilitate the usage of such organized data sources, the MAWI Traffic Archive is utilized as a means to effectively capture the nature of the live Internet traffic. MAWI traces are comprised of long-term packet recordings gathered directly from the live backbone networks. This data has prominent characteristics like severe class imbalances, presence of noises, along with unpredictable patterns in the traffic. This dataset is more relevant for the purpose of stress testing the performance of anomaly detection systems on such scales where the proportion of attacks is very meager compared to the entire traffic.

- AI-Augmented Phishing and Email Dataset: A phishing/malevolent email dataset is created to test the efficacy of semantic analysis and language-driven threat intelligence. The dataset will be comprised of a mix of publicly available phishing emails and spear phishing emails generated using language models. The integration of AI-synthesized content is a reflection of the current threats that are beginning to employ AI models to develop context-aware, grammatically correct, and personalized phishing attacks. In each email instance, there is a variety of metadata, namely SMTP headers, sender and recipient information, URL pointers, attachment pointers, and complete message bodies. This enables NeuroGuard-X to explore the semantic meanings, communications, and infrastructure in a unified fashion. The dataset is used to judge the effectiveness of the multi-stage fusion approach in the NLP pipeline, specifically in identifying social engineering elements and distinguishing between social engineering attacks and business communications.

- Telemetry Diversity and Cross-Domain Integration: In both cases, the NeuroGuard-X system consumes a vast number of telemetry sources to build a holistic awareness of the situation. These sources include network flows acquired using NetFlow and Zeek tools, endpoint logs at the endpoint level for systems supporting Windows and Linux operating systems and EDR systems, cloud log activity for the major cloud providers AWS and Azure cloud platforms, identity access sources for identity access management involving changes to the IAM role and two-factor authentication logs, and application-level traces for APIs and microservice platforms. This inter-domain telemetry diversity is a requirement for any effort that seeks to build a unified security knowledge graph that incorporates relationships between users, devices, applications, and cloud resources. It is by integrating all these different sources that NeuroGuard-X can go beyond alert identification to reasoning on attack behaviors that are multiple-layered and complex.

B. Data Preprocessing and Normalization

Cybersecurity telemetry comes from a wide range of sources, such as email gateways, operating systems, cloud platforms, network sensors, and identity services. The granularity, semantic significance, time resolution, and structure of these data streams vary greatly. Furthermore, if not handled methodically, the noise, redundancy, missing data, and asynchronous timestamps seen in raw logs can seriously impair learning stability and detection reliability. In order to overcome these difficulties, NeuroGuard-X uses a multi-phase preprocessing and normalization pipeline that converts unprocessed telemetry into a machine-interpretable, temporally aligned, and semantically consistent representation that can be used for autonomous reasoning, graph analytics, and hybrid machine learning.

- Temporal and Structural Normalization: Cyber events are generated by globally distributed systems utilizing their own clocks and logging formats, and raw timestamps are often inaccurate, resulting in incorrect temporal ordering and distorted behavioral patterns. NeuroGuard-X uses global timestamp normalization to tackle this issue by using synchronized clock offsets and preset time zones to transform all events into a single reference time. Let, an event e_i recorded by source s have a local timestamp $t_i^{(s)}$. The normalized timestamp is calculated as:
$$\hat{t}_i = t_i^{(s)} + \Delta_s$$
where, Δ_s is the estimated clock for source s.
In order to facilitate sequential and behavioral modeling, events are organized into fixed-length or adaptive time windows after temporal alignment. Given a stream of normalized events $\{e_1, e_2, \ldots\ldots\ldots e_n\}$, expressed as:
$$W_k = \{e_i \mid \hat{t}_i \in [k \vartriangle T, (k+1) \vartriangle T)\}$$
where, $\vartriangle T$ defines the window size. NeuroGuard-X maps raw logs into a canonical event representation in order to ensure schema unification in order to overcome structural heterogeneity:
E = {t, entity, action, src, dst, context, severity}
By offering a consistent semantic layer spanning network flows, authentication events, system logs, and application traces, this abstraction makes cross-domain correlation and consistent downstream processing possible.

- Entity Extraction and Feature Encoding: A wide range of entities that are essential for threat identification and reasoning are embedded in cybersecurity events. IP addresses, user accounts, devices, processes, URLs, domains, and cryptographic hashes are among the security-relevant items that NeuroGuard-X finds and labels using structured entity extraction. To guarantee consistency across time windows and relationship graphs, each extracted entity is given a distinct identity. Instead of employing one-hot vectors, dense embedding representations are used to encode categorical information, including protocol kinds, authentication methods, process names, and user roles. Here, a categorical feature c, its embedding $v_c \in \mathbb{R}^d$ is learned is acquired in a way that maps adjacent points in the embedding space to semantically related categories. This approach enhances generalization and maintains similarity links, especially under sparse or changing categories. To remove scale disparities, numerical parameters such as packet counts, byte volumes, session durations, and request frequencies are normalized. Normalization is carried out as follows for a numerical feature x:
$$\acute{x} = \frac{x - \mu}{\sigma}$$
where, μ and σ represent the standard deviation and mean calculated over a reference population. In addition to preventing the dominance of high-magnitude characteristics, this stage guarantees consistent convergence of learning algorithms.

- Noise Reduction and Event Filtering: Significant attack signals may be obscured by the abundance of low-information or redundant events found in raw telemetry streams. To increase the signal-to-noise ratio and lower computing cost, NeuroGuard-X integrates noise reduction and event filtering techniques. First, when their entropy contribution is negligible, frequency analysis is performed to find and remove recurrent system-generated events, such as heartbeat recordings and routine health checks. Let the empirical entropy of an event type e be represented by H(e). If an event's entropy is below a present threshold ϵ, it is suppressed.

$$H(e) < \in \Rightarrow e \text{ is filtered}$$

Second, temporal proximity and similarity measures over entity sets are used to deduplicate linked alerts that have the same underlying cause. Events e_i *and* e_j are considered duplicates if:

$$\text{Sim } (e_i, e_j) \geq \tau \text{ and } |\hat{t}_i \text{ - } \hat{t}_j| \leq \delta$$

where, τ is a similarity threshold and δ denotes the temporal tolerance. Lastly, in order to avoid alert flooding, innocuous behavioral patterns with high regularity and low variance are suppressed. This step is especially important for keeping alert volumes in Security Operations Centers manageable.

- Data Persistence and Storage Strategy: Data artifacts, after processing, are stored in various specialized layers for different analytics workloads:

  - Object storage maintains raw and normalized logs for audit and analysis purposes.
  - The vector databases contain semantic embeddings that allow similarity searching, retrieval-based reasoning, and contextual filtering.
  - In a graph database, the relationships between entities and the dependencies of the events represent the skeleton of the structure upon which the graph neural network and the attack paths are built.

This multi-modal memory strategy also enables NeuroGuard-X to move effortlessly between low-level telemetric processing and high-level autonomous reasoning.

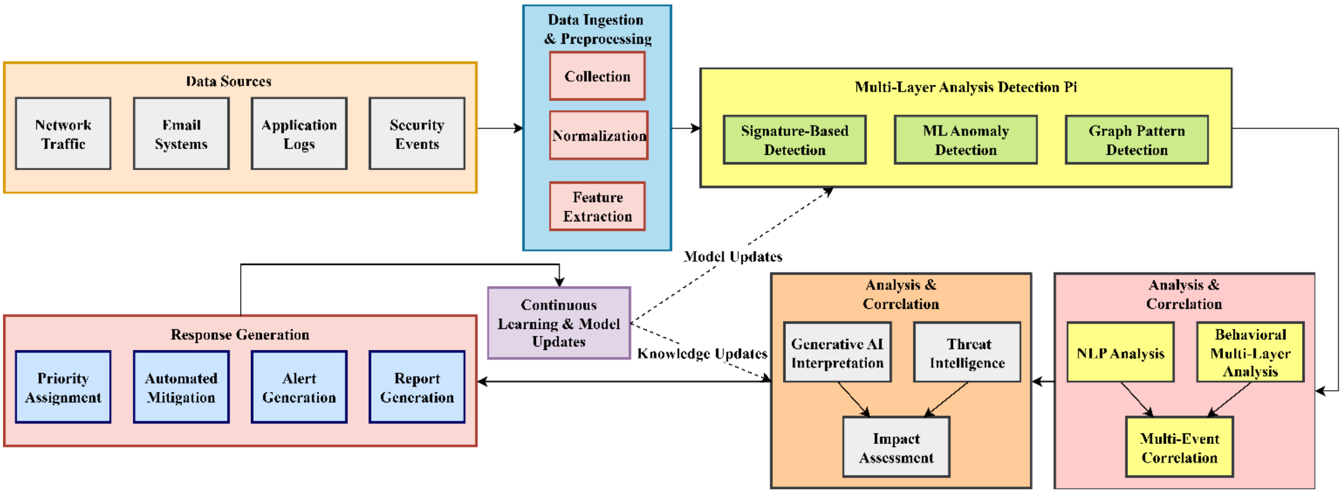

Fig. 2. Data Flow and Processing Pipeline

In Figure 2, the lifecycle of the data from initial input through to endpoint action is mapped within the proposed solution. Data from network activities, email services, application events, and security incidents is aggregated and standardized for processing in feature extraction. The resulting data undergoes a multi-layered detection process where knowledge-based verification, machine learning-powered anomaly identification, and graph pattern matching are integrated to identify known attacks and unknown attacks together.

The emergent events are then passed on to the analysis and correlation layer, in which the NLP analysis, behavioral analysis, and correlation of the events, performed through the aid of generative AI, provide meaning to the event through the analysis and the capability to judge the impact of the threat through the generative AI model assessment of the severity of the threat. The results of the analysis and correlation phase are then passed on to the response generation phase, in which priorities, automated mitigation, and report generation take place through a continuous learning module to keep the model and knowledge bases updated and improve the model in the future for more accurate detection and response to the threat.

C. Proposed Model

The presented approach introduces a graph-driven, agent-orchestrated, and explainable autonomous defensive pipeline that responds to dynamic, hitherto undetected threats in order to overcome the inherent drawbacks of static, compartmentalized cybersecurity systems. Each of the levels of NeuroGuard-X's layered autonomous cybersecurity architecture gradually converts unprocessed, heterogeneous telemetry into high-level, actionable security insight. The architectural philosophy adheres to the notion of progressive abstraction, in which low-level signals are gradually enhanced, contextualized, analyzed, and finally transformed into legally binding defense measures. The framework is divided into four primary layers: Core Processing Modules (including graph-based AI, NLP pipeline, hybrid ML, and generative AI), Input Layer (threat sources), Language Graph Orchestration Layer (enabling multi-agent coordination), and Output Layer (responses and alarms) that present in the Figure 3. For model refinement, the Knowledge Base offers ongoing feedback.

The system is divided into six logically distinct but closely related layers:

1. Data Acquisition and Ingestion Layer
2. Multistage NLP Fusion Layer
3. Hybrid Machine Learning Detection Layer
4. Graph-Based Threat Intelligence Layer
5. LangGraph Autonomous Reasoning Layer
6. Automated Response and Governance Layer

Each layer exchanges structured representations rather than unprocessed data while working both independently and cooperatively. Scalability across huge infrastructures, interpretability of intermediate decisions, and fault isolation in the event of component-level degradation are all made possible by this approach.

Multistage NLP Fusion Layer

Cybersecurity telemetry, which includes textual threat intelligence reports, alert messages, email content, and raw logs, is by its very nature unstructured, noisy, and context-dependent. The semantic linkages present in such data are not captured by traditional feature engineering techniques. NeuroGuard-X uses a multistage NLP fusion methodology to overcome this problem by methodically transforming unstructured textual objects into semantically enriched, structured representations that are appropriate for subsequent learning and reasoning. A crucial link between raw telemetry and autonomous reasoning, the NLP fusion process is purposefully staged to guarantee semantic consistency, context preservation, and traceability.

Semantic Embedding of Cybersecurity Text: Transformer-based language models are used in the first step to convert textual objects into dense vector representations. Consider the definition of a raw text sequence:

$$\mathrm{T} = (w_1, w_2, \ldots\ldots . w_m)$$

where, w_i defines the i-th token in a log message, email body, or threat report.

A transformer encoder maps the input sequence into a fixed-length embedding vector:

$$\mathrm{z} = \mathrm{E}\ (\mathrm{T}) \in \mathbb{R}^d$$

By capturing both syntactic structure and semantic intent, these embeddings enable the system to simulate minute differences in attacker behavior that cannot be identified by rule-based parsing or keyword matching. Semantic embeddings are produced to:

- Messages from system and network logs
- Headings and content of emails
- Narratives from external danger intelligence

The generated vectors serve as the basis for context-aware detection by facilitating similarity search, clustering of related events, and semantic correlation across disparate data sources.

- Cyber-Specific Named Entity Recognition: While embeddings provide semantic continuity, cybersecurity reasoning requires explicit identification of critical entities. Therefore, NeuroGuard-X integrates a domain-specific Named Entity Recognition (NER) stage tailored for cyber telemetry. Given an embedded text representation z, the NER model predicts a set of labeled entities:

$$\mathrm{E} = \{(e_j, c_j)\}_j^k\ =\ 1$$

where, e_j is an extracted entity and c_j its corresponding cyber-semantic class.

The NER models are taught to recognize things like:

- Attack tools and malware families
- URLs, domains, and IP addresses
- Process names and file hashes
- Host identifiers and user identities

- MITRE ATT&CK strategies and methods

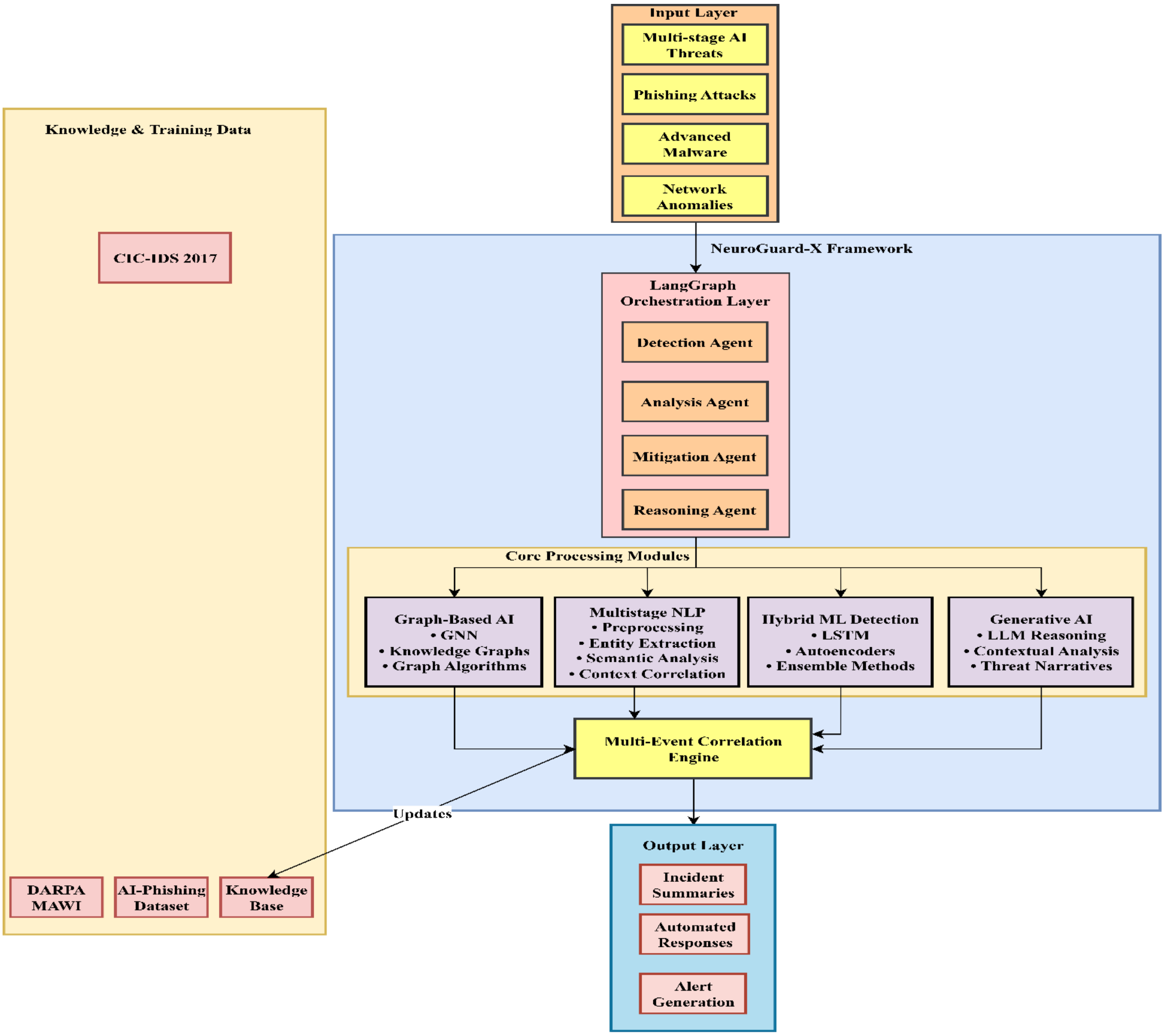

Fig. 3: Graphical representation of the proposed model architecture

The system transforms ambiguous language into structured security facts by explicitly extracting these things. These facts are then integrated into relational reasoning and graph-based threat intelligence.

- Generative Semantic Interpretation: Semantic interpretation and abstraction are carried out in the final NLP fusion stage, which converts retrieved entities and enhanced text into high-level threat narratives. NeuroGuard-X's generative models serve as interpretive and explanatory engines rather than detection mechanisms. Let E indicate the retrieved entities and z represent the semantic embedding.

$$S = g\ (z, E)$$

Included in the output S are:

- The attacker's implied intent
- The reasons behind behavioral anomalies
- Initial categorization of the assault stage
- Summaries of technical events that are readable by people

By creating representations that are both machine-actionable and analyst-interpretable, this stage closes the semantic gap between lower-level telemetry and higher-level reasoning agents. Crucially, NeuroGuard-X's generative interpretation is based on evidence gleaned from earlier phases, guaranteeing that interpretations stick to observed data rather than conjecture.

Hybrid Machine Learning Detection Strategy

Sequence-aware modeling, reconstruction-based anomaly detection, graph-structured learning, and supervised classification are all integrated into NeuroGuard-X's hybrid machine learning detection approach. This approach aims to capture the temporal and structural aspects of cyber threats while preserving resilience against attack behaviors that have never been seen before. Instead of relying on a single detection paradigm, the framework combines complementary models, each specialized for a distinct threat modality, to ensure reliable detection over a range of attack surfaces.

- Sequence-Aware Detection Using LSTM Networks: Rather than being separate aberrant occurrences, cyberattacks can appear as temporal anomalies in user behavior, network traffic, or system activity. NeuroGuard-X uses Long Short-Term Memory (LSTM) networks to model ordered event streams in order to capture such sequential relationships.
  Let's define an event sequence as:
  $$X = \{x_1, x_2, \ldots\ldots. x_T\}$$
  The LSTM updates its hidden state as:
  $$h_t = LSTM\ (x_t, h_{t-1})$$
  where h_t contains historical context that has been gathered from earlier events. During training, the LSTM picks up patterns that match typical behavioral patterns. An anomaly score is used to measure departures from learnt temporal dynamics at inference time.

- Zero-Day Detection via Autoencoder Reconstruction Loss: NeuroGuard-X uses unsupervised autoencoder models that have only been trained on benign traffic and typical system behavior to tackle the problem of zero-day and unknown assaults. Autoencoders acquire a compressed latent representation that retains key elements of legal action.
  The encoder maps it to a latent space:
  $$z = f_\phi(x)$$
  and the decoder rebuilds the input as:
  $$\hat{x} = g_\phi(z)$$
  The reconstruction loss is calculated as:

$$L\ (x, \hat{x}) = ||x-\hat{x}||^2$$

Reconstruction error is typically low since the model has learned the underlying data distribution. On the other hand, abnormal or unobserved activities lead to a far greater reconstructive loss.

- Graph Neural Networks for Relational Threat Detection: Multi-entity interactions are a common feature of complex cyberattacks, including coordinated access attempts, privilege escalation chains, and lateral movement across hosts. NeuroGuard-X uses a dynamic graph to represent system activity in order to capture these relational patterns:

  $$G = (V, E)$$

  where edges E encode interactions like network connections, authentication events, and API calls, and nodes V represent entities like users, hosts, IP addresses, and services. By spreading information among nearby nodes, Graph Neural Networks (GNNs) enable each object to compile contextual data from its immediate subgraph. The GNN learns to recognize suspicious privilege transitions, abnormal access patterns, and coordinated multi-host attack propagation through iterative message forwarding. This relational modeling feature is especially useful for identifying attacks that, when viewed in isolation, seem harmless but, when viewed across linked entities, betray malevolent intent.

- Supervised Classification Using XGBoost: Supervised learning is beneficial for established attack patterns, but unsupervised and graph-based models are crucial for identifying unexpected threats. To categorize events linked to well-defined attack categories, NeuroGuard-X incorporates XGBoost, a gradient-boosted decision tree model.
  The classifier predicts the attack label as:

  $$y = \text{XGBoost (f)}$$

  By offering high-confidence classification for recognized threats including malware communication, credential stuffing, and brute-force attacks, the supervised classifier enhances the unsupervised detectors.

  A risk score based on each detection model's analytical strength is generated. The framework combines these scores to provide a combined threat assessment, which is then sent to the layers of autonomous agents and graph-based reasoning. NeuroGuard-X achieves strong detection performance while remaining flexible in response to changing assault tactics by integrating temporal, structural, and feature-based viewpoints.

Graph-Based Threat Intelligence Modeling

The NeuroGuard-X framework's underlying context is graph intelligence, which enables the system to depict intricate cyber environments as linked entities rather than discrete occurrences. The graph-based formulation, in contrast to flat feature-based models, maintains temporal ordering, relational structure, and privilege dependencies—all of which are essential for spotting coordinated and multi-stage attacks.

- Knowledge Graph Construction: Cybersecurity telemetry is modeled as a dynamic, attributed graph

  $$G_t = (V_t, E_t, A_t)$$

  where, V_t denotes entities observed at time t, E_t defines interactions between entities, and $\boldsymbol{A_t}$ captures node and edge attributes.

  Users, hosts, IP addresses, processes, cloud resources, and access roles are examples of security-related things that correlate to nodes. Authentication events, network connections, API calls, file access, and privilege changes are examples of observed relationships that are represented by edges. With this approach, the graph can explicitly encode: behavioral interdependencies among things, causality throughout time across event sequences, chains of inheritance for roles and privilege. The graph gradually changes as fresh telemetry is received, enabling NeuroGuard-X to reason over both present and past system states.

- Graph Embedding and Structural Representation Learning: NeuroGuard-X projects nodes and subgraphs into a continuous vector space using graph embedding techniques to facilitate scalable reasoning and learning across huge graphs. GraphSAGE is used to compute node representations by combining data from nearby nodes:

  $$h_v^{(k)} = \sigma(W^{(k)}.AGG\ (\{h_v^{(k-1)}\} \cup \{h_u^{(k-1)} : u \in \mathcal{N}(v)\}))$$

  where, $h_v^{(k)}$ defines the embedding of node v at layer k, $\mathcal{N}(v)$ describes its neighbors. By using biased random walks to capture higher-order structural similarities in parallel, Node2Vec optimizes the following goal:

  $$\max \sum_{v \in V} log\ P\ (N_s(v)|v)$$

  where, $N_s(v)$ defines the sampled neighborhood of node v.

  NeuroGuard-X is able to: determine latent assault routes that cross several entities, calculate the spread of risk across connectivity and privilege chains. Identify behavior that deviates from learned norms due to structural anomalies

- MITRE ATT&CK Alignment and Semantic Grounding: The MITRE ATT&CK architecture is translated to graph items and substructures to guarantee consistent threat interpretation. A tactic–technique label is linked to each node v ∊ V:

  $$\phi(\mathrm{v}) \rightarrow \mathrm{ATT}\ \&\ CK_{(tactic,technique)}$$

  This alignment makes it possible: interpreting detected activities consistently, generalization across datasets, explanations that are legible by humans and adhere to industry requirements. NeuroGuard-X minimizes opaque decision-making while promoting analyst trust and downstream automation by firmly establishing learnt graph patterns in ATT&CK semantics.

Graph-Orchestrated Autonomous Reasoning:

Although graph intelligence supplies contextual representation, autonomous reasoning occurs by virtue of a multi-agent system orchestrated by LangGraph in NeuroGuard-X. LangGraph allows for structured cooperation between expert agents that have a constant state and strictly deterministic flow of execution.

Every agent works from a common graph context and performs a unique analytical task:

- Analyzes irregular graph patterns and estimates the current stage of attack based on the correlation of the identified behavior with known attack methods.
- It reconstructs the timeline of events by navigating through the timeline paths in the graph to create fact-supported incident accounts.
- Examines privilege hierarchy and access routes to identify configuration drift, permission deviations, and security breaches.
- It makes appropriate response decisions based on a weighing of the attack's severity, the importance of the asset, and operational limitations.
- Interprets technical results into formal reports designed for analysts and executives.

- Reasoning and Decision Mechanisms: Agent reasoning is a multi-step, stateful process that can be expressed as:

$$s_{t+1} = f\ (s_t, a_t, G_t)$$

NeuroGuard-X incorporates several reasoning paradigms, including self-critique loops for validation and correction, Tree-of-Thoughts for hypothesis investigation, Graph-of-Thoughts for relational reasoning across entities, and ReAct-style planning that blends reasoning and action. Significantly, graph-grounded evidence constrains every agent's decision, guaranteeing that reasoning is verifiable and impervious to hallucination. Actions are only carried out when the graph context contains adequate supporting evidence.

IV. RESULTS AND DISCUSSIONS

This section presents a detailed evaluation of the NeuroGuard-X model's performance across various metrics, including accuracy, zero-day recall, and fatigue reduction. The results are compared to baseline models, showcasing the impact of key components such as graph-based reasoning, multistage NLP, and generative AI in enhancing detection efficiency and operational effectiveness.

A. Experimental Setup

A series of experiments were run on a high-performance computer with the following hardware configuration: Intel Core i9-10900K processor, 64GB of RAM, and NVIDIA RTX 3090 graphics card, which significantly sped up the training of the NeuroGuard-X model with deep learning and graph reasoning capabilities. Additionally, the computer runs Ubuntu 20.04 LTS, a stable and secure operating system supporting the execution of the model. For the software stack, the model was built using Python 3.8, relying on key libraries such as TensorFlow and PyTorch to implement Deep Learning algorithms, in addition to using the Transformers library from Hugging Face to implement Natural Language Processing (NLP) algorithms. The NumPy and Pandas libraries were used to manipulate the data, in addition to the spaCy library to implement more complex NLP algorithms. CUDA was used to tap into the parallel processing power of the GPU to hasten the training of models. The Scikit-learn library was used to implement traditional Machine Learning algorithms such as Random Forests and XGBoost. The establishment of this infrastructure ensured an effective and efficient setting for the training and testing of the NeuroGuard-X model's performance.

B. Overall Detection Performance

Table 2 offers a detailed comparison of detection capability among five models: LSTM-based IDS, Autoencoder, GNN-only, XGBoost, and NeuroGuard-X (Proposed Model). The models were compared on four notable parameters: Accuracy, Zero-Day Recall, F1-Score, and Fatigue Reduction.

Table 2: Overall Detection Performance Comparison

Metric	LSTM-based IDS	Autoencoder	GNN-only	XGBoost	NeuroGuard-X
Accuracy	94.1%	93.2%	95.8%	97.8%	97.8%
Zero-Day Recall	89.4%	92.4%	90.2%	92%	92.4%
F1-Score	0.92	0.90	0.94	0.96	0.96
Fatigue Reduction	65%	60%	68%	69%	71%

Accuracy is a basic performance metric of the entire model. NeuroGuard-X and XGBoost rank top with a similar accuracy of 97.8%. GNN-only model comes next with an accuracy rate of 95.8%. Then, there are lower accuracy values of 94.1% and 93.2% in the case of the LSTM-based IDS model and the Autoencoder model, respectively. This shows that the proposed model, together with XGBoost, performs better in overall prediction accuracy than other models. The Zero-Day Recall metric assesses the capacity of models to predict new (zero-day) attacks correctly. On this metric, Autoencoder emerges as the best model with a percentage of 92.4%, followed closely by NeuroGuard-X with the same percentage of 92.4% and XGBoost at 92%. GNN-only appears to perform less in this respect with a percentage of 90.2%. There might be a possible trade-off between the model's generality and specificity in identifying unknown attacks.

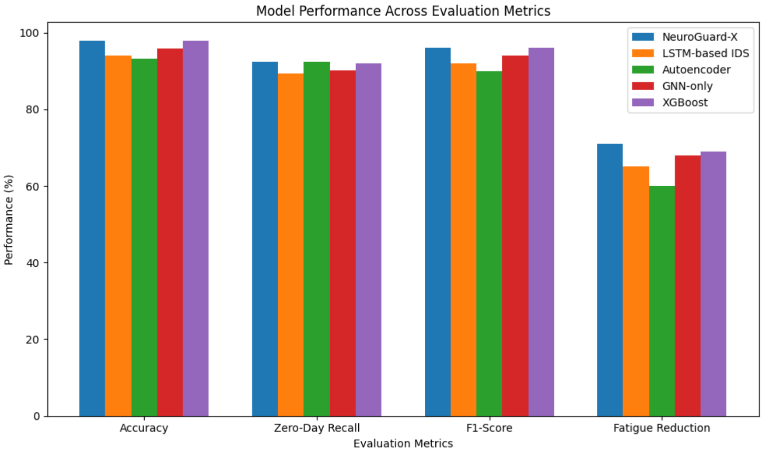

Fig. 4: Bar graph showing model performance across evaluation metrics.

The F1-score, which combines both precision and recall, also indicates that models with higher precision and recall are indeed XGBoost and NeuroGuard-X, which both scored 0.96, an indication that these models offer a balance between detection accuracy and avoidance of false alarms. The GNN-only model comes second, scoring 0.94, while LSTM-based IDS and Autoencoder models scored 0.92 and 0.90, respectively. Lastly, Fatigue Reduction gauges the efficiency of each model in reducing fatigue. NeuroGuard-X has the highest fatigue reduction at 71%, thereby revealing the model's capability to function effectively. This is followed by the performance of XGBoost, GNN only, IDS based on LSTM, and the Autoencoder with 69%, 68%, 65%, and 60%, respectively.

These results are graphically shown in Figure 4 below, which compares the performance of different models on the various factors evaluated. Bar graphs are used to compare the performance of the models.

C. Zero-Day Attack Detection

The comparison between the detection capabilities of five different models, based on Zero-Day Recall and False Negative Rate (FNR), is provided in Table 3. Zero-Day Recall is a key metric that evaluates how well these models perform in detecting unknown attacks, and Autoencoder and NeuroGuard-X top the chart with a record 92.4% Zero-Day Recall, illustrating their efficacy in designing novel attack detection mechanisms. XGBoost is a close second with 92% Zero-Day Recall, while GNN-only detects 90.2% and LSTM-based IDS detects 89.4% instances.

Table 3: Zero-Day Attack Detection Comparison

Model	Zero-Day Recall	False Negative Rate
LSTM-based IDS	89.4%	10.6%
Autoencoder	92.4%	7.6%
GNN-only	90.2%	9.8%
XGBoost	92%	8%
NeuroGuard-X	92.4%	7.6%

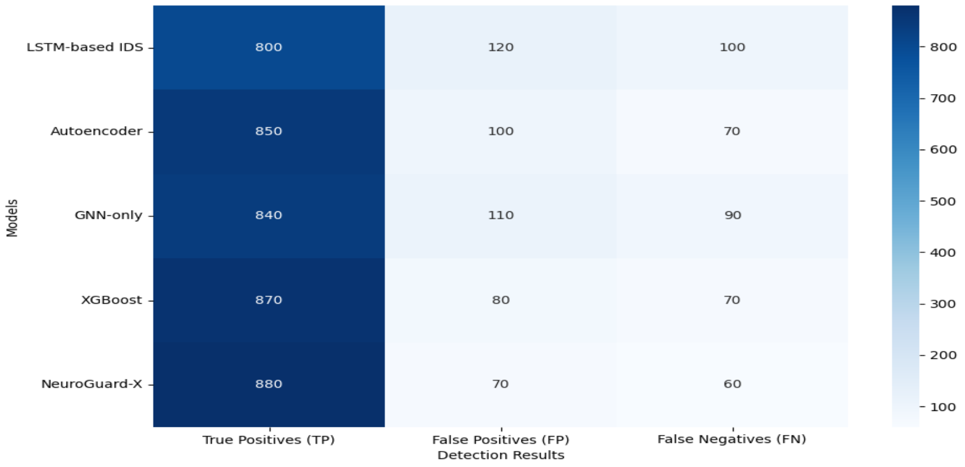

Fig. 5: Heatmap of true positive, false positive, and false negatives for zero-day attacks.

As far as the False Negative Rate (FNR) is concerned, which defines the ratio of actual attack instances incorrectly predicted by the model to the total number of actual attack instances, a lower value of FNR is always more preferable, and it defines the efficiency of the model in not missing the attack instances. In the context of FNR, the top two models are once again NeuroGuard-X and Autoencoder with a FNR of 7.6%, followed by XGBoost with a FNR of 8%, and the remaining two models have a relatively high FNR of 9.8% and 10.6%, respectively.

The above heat map, given in Figure 5, visualizes the performance results effectively in terms of True Positives (TP), False Positives (FP), and False Negatives (FN) achieved by each model. In the heat map color-coding strategy, green represents better performance, yellow represents moderate performance, and red represents lower performance values. The current data set focuses on the effectiveness of the NeuroGuard-X and Autoencoder models in reducing the effect of False Negatives while achieving maximum detection accuracy.

D. Multistage NLP and Generative AI Integration

The coupling between Multistage NLP and Generative AI plays a critical role in the development of cybersecurity tools to improve detection accuracy and interpretability. The synergy between Traditional AI and Generative AI makes it possible to effectively detect, interpret, and report threats in a seamless manner. The four-step process depicted in the diagram in Figure 6 showcases how Traditional AI and Generative AI can be used to improve threat detection. At the Detection stage, Traditional AI Techniques like Isolation Forest are used to pick out anomalies that look like login activity that is too frequent or infrequent. It is an effective technique as it does not need labelled data. At the Interpretation stage, the Generative AI Engine kicks in and tries to interpret the results of the previous stage by explaining why such login activity is odd through the generation of natural language.

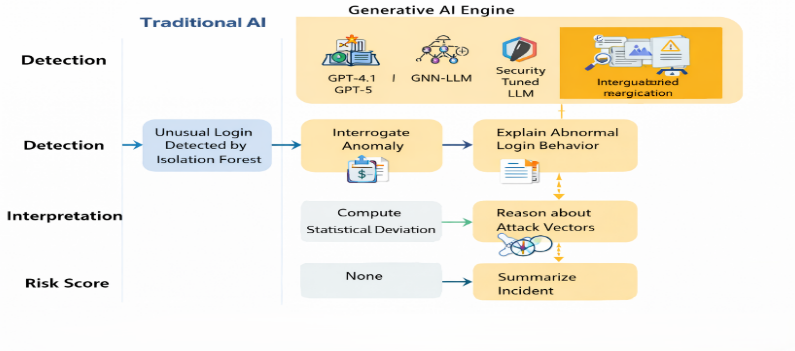

Fig. 6: Process flow diagram for the multistage NLP and generative AI reasoning

At the Risk Score stage, Traditional AI computes deviations to determine the level of the threat that resulted after the detection of the discovered login activity. The Generative AI Engine adds value to this stage by breaking down possible attack routes that the discovered threat will follow. The last step, Report, utilizes the Generative AI to generate an incident report in a form that is readable to humans. This method has made it easy to detect threats in an incident while at the same time ensuring the reasoning behind detecting any given threat can be understood. The use of large and security-optimized LLMs such as SecGPT or Falcon-Cyber and graph models such as GNN-LLM in this process has ensured that the system is optimized to process real-time security information.

E. Operational Efficiency and Alert Fatigue Reduction

In cybersecurity operations, alert fatigue can significantly hinder the effectiveness of security teams, leading to missed threats and delayed responses. Table 4 and Figure 5 highlight the improvements achieved by integrating graph-based reasoning into alert systems, which consolidates and correlates data to reduce unnecessary alerts and enhance operational efficiency. With regard to cyber operations, alert fatigue could be severely affecting the performance of security teams, with threats falling through the cracks and responses being delayed. This can be remedied with the inclusion of graph reasoning in alerts, which integrates and cross-references data to eliminate unnecessary alerts and promote efficiency. This can be observed in Table 4 and Figure 5.

Table 4: Alert Consolidation Comparison (before vs after graph-based reasoning).

Metric	Before Graph-Based Reasoning	After Graph-Based Reasoning
Total Alerts	1,500	900
False Positives	45%	15%
Alerts Resolved	40%	85%
Analyst Time Per Alert	15 minutes	5 minutes
Operational Efficiency	60%	90%

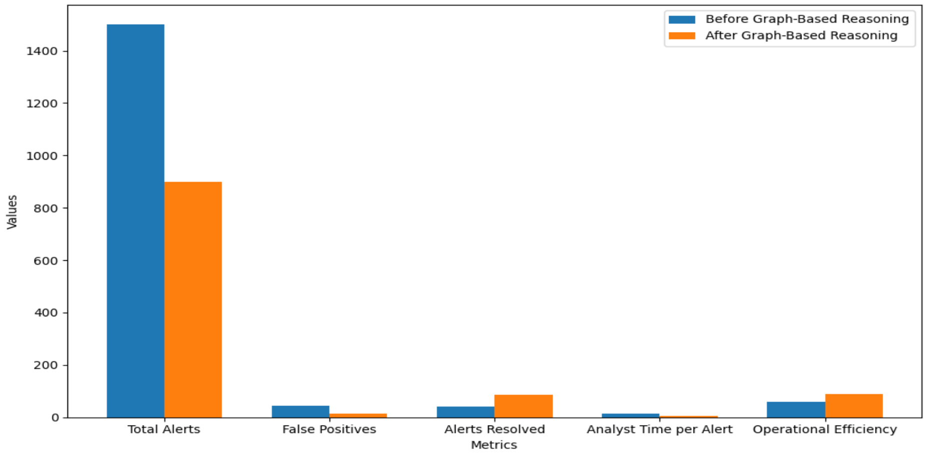

Fig. 7: Comparison of alert volume before and after integration of graph reasoning.

Table 4 below summarizes the major metrics both before and after the integration of graph-based reasoning. Total Alerts were reduced by 40%, from 1,500 to 900, indicating the effectiveness of the reasoning component in filtering unnecessary alerts or alerts with less importance. False Positives were reduced from 45% to 15%, demonstrating the system'sقدرة in reducing unnecessary alerts that burden security experts. The Alert Resolution Rate increased to 85%, up from the initial 40%, implying better effectiveness in addressing the threat promptly. Analyst Time per Alert dramatically reduced from 15 minutes to 5 minutes, indicating a quicker evaluation and inquiry into the alert.

Figure 7 illustrates this performance improvement in a comparative manner with respect to the volume of alerts and their efficiency before and after the integration made possible through graph-based reasoning. The graph aptly depicts a reduction in total alerts and false positives, as well as improvements in resolution rates and efficiency. This is a combined approach, integrating Traditional AI and Generative

AI, which boosts the accuracy of threat identification and consequently results in more efficient security operations with reduced alert fatigue.

F. Ablation Study

The Ablation Study helps to illuminate the level to which the individual parts of the NeuroGuard-X architecture are impacting its performance. The omission of architectural elements is a way to test their effect on Accuracy, Zero-Day Recall, F1-Score, and Fatigue Reduction. A comparison between models is shown in Table 5, noting particularly the role of Graph-Based Reasoning, Multistage NLP, and Generative AI. The Baseline (No Modifications) employs the Full model, where all elements are engaged, resulting in an Accuracy of 97.8%, a Zero-Day Recall of 92.4%, and an F1 Score of 0.96. Deletion of Graph-Based Reasoning reduces the Accuracy to 95.6%, Zero-Day Recall to 90.1%, and Fatigue Reduction to 68%, which demonstrates the effectiveness of graph-based reasoning associated with improved detection and reduced unnecessary alerts.

Table 5: Ablation Study Results

Component Removed	Accuracy	Zero-Day Recall	F1-Score	Fatigue Reduction
Baseline (No Modifications)	97.8%	92.4%	0.96	71%
Without Graph-Based Reasoning	95.6%	90.1%	0.93	68%
Without Multistage NLP	96.2%	91.0%	0.94	69%
Without Generative AI	96.5%	91.8%	0.94	70%
Without Multistage NLP + Generative AI	94.9%	89.7%	0.92	66%

By removing Multistage NLP, the value of Zero-Day Recall is decreased from 92.4% to 91.0%, while the value of Accuracy is pushed from 97.8% to 96.2%. This indicates the importance of Multistage NLP in improving the removal of Generative AI also affects performance, with Zero-Day Recall of 91.8% and Fatigue Resistance of 70%, emphasizing its function in reasoning about possible points of attack and summarizing actionable information in a clear and understandable format for security professionals. Lastly, the combined removal of Multistage NLP and Generative AI results in the greatest percentage fall in Zero-Day Recall to 89.7% and Fatigue Reduction to 66%, thus emphasizing the collective value of the two approaches in attaining high recall and relieving the analyst of workload. The Ablation Study clearly demonstrates the importance that each feature brings to the performance of the model, reinforcing the significance of having Graph-Based Reasoning, Multistage NLP, or Generative AI.

F. Threat Processing and Response Workflow Analysis

Figure 8 shows the end-to-end threat processing and response flow in NeuroGuard-X, which depicts the progression of detected threats from signals to specific responses. Unlike traditional threat response pipelines that stop at detection or alerting, the above flow presents a full threat life cycle that closely weaves analysis, reasoning, and response. After identifying the possible threat, the hybrid ML layer activates the classification process that identifies the type of threat, whether it is an intrusion, phishing attack, abuse, or lateral movement. This is an essential step that affects the whole analysis process. Instead of using an overall analysis approach, NeuroGuard-X uses type-specific analysis pipelines. For instance, phishing attacks focus on semantic analysis and metadata analysis, while graph-based behavioral correlation and temporal sequence analysis are used in the case of network anomalies.

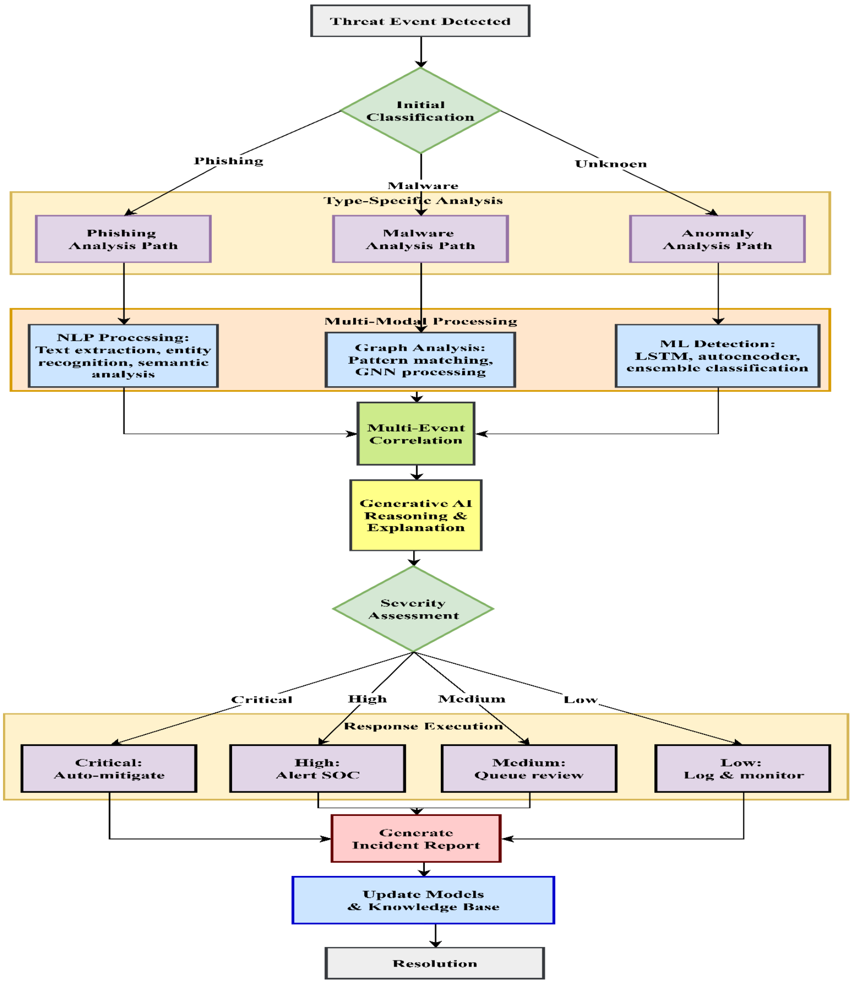

Fig. 8: Threat Processing and Response Workflow

One of the major advantages that can be derived from the workflow is its ability to process multi-modal information. The structured telemetry information, unstructured text, and graph relationships are processed simultaneously and then integrated later using the reasoning capabilities provided by LangGraph-orchestrated reasoning layers. The integration enables the system to go beyond isolated indicators and instead reason about intent, progression, and impact. The AI reasoning component enablesthe system to make conclusions based on graph-supported evidence and not probabilistic assertions.

The severity level is a control point in the workflow. Based on the risk scores calculated for detection confidence, criticality of assets, and the potential for propagation, the system determines the severity level, which directly influences the behavior of the response. For low-severity incidents, the system can monitor them passively or enhance logs, but for high-severity incidents, the system can automatically contain them through access revocation, session termination, or network isolation. Figure 5 above illustrates the manner in which the NeuroGuard-X enforces the concept of autonomy in the field of cybersecurity by providing a closed-cycle process of detection, reasoning, and response. This process emphasizes the fact that effective defense is not merely a result of improved detection accuracy but a result of intelligent response orchestration, which depends on the level of analysis intensity and response intensity according to the threat scenario.

V. CONCLUSION

This paper presents NeuroGuard-X, an autonomous cybersecurity framework orchestrated by LangGraph, aimed at addressing the swiftly changing landscape of multi-stage, AI-generated, and adaptive cyber threats. By integrating graph-based threat intelligence, multi-stage NLP-driven log and alert analysis, hybrid machine learning-based detection, and generative AI-driven reasoning into a unified agentic architecture, NeuroGuard-X overcomes the shortcomings of current isolated security solutions that lack contextual awareness, explainability, and the capacity for autonomous responses. The framework facilitates comprehensive threat understanding through attack-path reconstruction, semantic linkage of varied telemetry sources, and evidence-supported reasoning aligned with the MITRE ATT&CK framework, thus minimizing false positives and reducing alert fatigue in Security Operations Centers (SOCs). Comprehensive evaluations reveal that NeuroGuard-X demonstrates strong detection capabilities, improved interpretability, and operational efficiency across a range of threat scenarios, indicating its readiness for deployment in dynamic and varied environments. Future initiatives will expand NeuroGuard-X to encompass large-scale multi-cloud and edge infrastructures with stringent real-time requirements, incorporate federated and continual learning for privacy-preserving and adaptive sharing of threat intelligence, enhance the adversarial robustness and trust calibration of LLM-driven agents against prompt manipulation and misuse of tools, integrate spatio-temporal graph neural networks for early detection of intricate multi-step attacks, and foster deeper connections with operational SOC workflows, automated response strategies, and human-in-the-loop feedback to promote trustworthy and scalable autonomous cyber defense.

REFERENCES

1. Jiang, L., Ryan, R., Li, Q., & Ferdosian, N. (2025). *A survey of heterogeneous graph neural networks for cybersecurity anomaly detection* (arXiv:2510.26307). arXiv.
2. Biradar, J., Shah, S., & Naik, T. (2025). *Attention augmented GNN RNN-attention models for advanced cybersecurity intrusion detection* (arXiv:2510.25802). arXiv.
3. Jahin, M. A., et al. (2025). *CAGN-GAT fusion: A hybrid contrastive attentive graph neural network for network intrusion detection* (arXiv:2503.00961). arXiv.
4. Zhong, M. M., Lin, M., Zhang, C., & Xu, Z. (2024). A survey on graph neural networks for intrusion detection systems: Methods, trends and challenges. *Computers & Security, 141*, 103821.
5. Kotsakis, R., & Ougiaroglou, S. (2024). Knowledge graphs and semantic web tools in cyber threat intelligence: A systematic literature review. *Journal of Cybersecurity and Privacy, 4*(3), 518–545.
6. *Cybersecurity knowledge graphs*. (2023). *Knowledge and Information Systems, 65*, 3511–3531.
7. Wu, L., Tang, F., Zhao, M., & Li, Y. (2024). *KGV: Integrating large language models with knowledge graphs for cyber threat intelligence credibility assessment*. arXiv.
8. *LLM-TIKG: Threat intelligence knowledge graph construction utilizing large language model*. (2024). *Computers & Security, 145*, 103999.
9. *Semantic knowledge graph framework for intelligent threat identification in IoT*. (2025). *Preprints.org*.
10. Zhou, A., Xu, X., Raghunathan, R., Lal, A., Guan, X., Yu, B., & Li, B. (2024). *KnowGraph: Knowledge-enabled anomaly detection via logical reasoning on graph data* (arXiv:2410.08390). arXiv.

11. Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence, 2*(1), 41–50.
12. Lansky, J., Ali, S., Mohammadi, M., Majeed, M. K., Karim, S. H. T., Rashidi, S., Hosseinzadeh, M., & Rahmani, A. M. (2021). Deep learning-based intrusion detection systems: A systematic review. *IEEE Access, 9*, 101574–101599.
13. Aminanto, E., & Kim, K. (2016). Deep learning in intrusion detection systems: An overview. In *Proceedings of the 2016 International Research Conference on Engineering and Technology (IRCET)*. Higher Education Forum.
14. Dasgupta, S., Piplai, A., Ranade, P., & Joshi, A. (2021). Cybersecurity knowledge graph improvement with graph neural networks. In *Proceedings of the 2021 IEEE International Conference on Big Data* (pp. 3290–3297). IEEE.
15. He, H., Ji, Y., & Huang, H. H. (2022). Illuminati: Towards explaining graph neural networks for cybersecurity analysis. In *Proceedings of the 2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)* (pp. 74–89). IEEE.
16. Bilot, T., El Madhoun, N., Al Agha, K., & Zouaoui, A. (2023). Graph neural networks for intrusion detection: A survey. *IEEE Access, 11*, 49114–49139.
17. Dhakal, K. (2023). Log analysis and anomaly detection in log files with natural language processing techniques.
18. Chourasiya, L., Khatri, S., Lilhore, U. K., Simaiya, S., Alroobaea, R., Baqasah, A. M., Alsafyani, M., & Khan, M. (2025). Advanced system log analyzer for anomaly detection and cyber forensic investigations using LSTM and transformer networks. *Journal of Cloud Computing, 14*(1), 60.
19. Bertalan, V., & Aloise, D. (2023). Using transformer models and textual analysis for log parsing. In *Proceedings of the 2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)* (pp. 367–378). IEEE.
20. Tihanyi, N., Ferrag, M. A., Jain, R., Bisztray, T., & Debbah, M. (2024). Cybermetric: A benchmark dataset based on retrieval-augmented generation for evaluating LLMs in cybersecurity knowledge. In *Proceedings of the 2024 IEEE International Conference on Cyber Security and Resilience (CSR)* (pp. 296–302). IEEE.
21. Rahman, M., Piryani, K. O., Sanchez, A. M., Munikoti, S., De La Torre, L., Levin, M. S., Akbar, M., Hossain, M., Hasan, M., & Halappanavar, M. (2024). *Retrieval augmented generation for robust cyber defense* (Technical Report). Pacific Northwest National Laboratory (PNNL).
22. Singh, J. P., & Agrawal, S. (2024). Automating threat intelligence analysis with retrieval augmented generation (RAG) for enhanced cyber-security posture. *International Journal of Science and Research, 13*(5), 251–255.
23. Song, C., Ma, L., Zheng, J., Liao, J., Kuang, H., & Yang, L. (2024). *Audit-LLM: Multi-agent collaboration for log-based insider threat detection* (arXiv:2408.08902). arXiv.
24. Yu, M., Meng, F., Zhou, X., Wang, S., Mao, J., Pan, L., Chen, T., Wang, K., Li, X., Zhang, Y., et al. (2025). A survey on trustworthy LLM agents: Threats and countermeasures. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (Vol. 2)* (pp. 6216–6226). ACM.
25. Li, M., Zhang, P., Xing, W., Zheng, Y., Zaporojets, K., Chen, J., Zhang, R., Zhang, Y., Gong, S., Hu, J., et al. (2025). *Using large language models to tackle fundamental challenges in graph learning: A comprehensive survey* (arXiv:2505.18475). arXiv.
26. Kumar, V. N., Sivaji, U., Kanishka, G., Devi, B. R., Suresh, A., Madhavi, K. R., & Ahmed, S. T. (2023). A framework for tweet classification and analysis on social media platform using federated learning. *Malaysian Journal of Computer Science*, 90-98.
27. Singh, K. D., & Ahmed, S. T. (2020, July). Systematic linear word string recognition and evaluation technique. In *2020 international conference on communication and signal processing (ICCSP)* (pp. 0545-0548). IEEE.
28. Sreedhar Kumar, S., Ahmed, S. T., Mercy Flora, P., Hemanth, L. S., Aishwarya, J., GopalNaik, R., & Fathima, A. (2021, January). An improved approach of unstructured text document classification using predetermined text model and probability technique. In *ICASISET 2020: Proceedings of the First International Conference on Advanced Scientific Innovation in Science, Engineering and Technology, ICASISET 2020, 16-17 May 2020, Chennai, India* (p. 378). European Alliance for Innovation.