



EXPLAINABLE AI FOR AUTONOMOUS CYBER THREAT DETECTION USING GRAPH NEURAL NETWORKS AND LARGE LANGUAGE MODELS

THANGARAJU S B, Dr.S.Bharathiraja

Research Supervisor, Assistant Professor, PG & Research Department of Electrical Engineering, The Standard Fireworks Rajaratnam college for women (Autonomous), Sivakasi

ABSTRACT

The rapid expansion of interconnected digital infrastructures has significantly increased the complexity, volume, and sophistication of cyber threats. Conventional intrusion detection systems frequently depend on predefined signatures, manually engineered features, or isolated machine-learning models, limiting their ability to recognize evolving and previously unseen attack behaviors. At the same time, deep-learning-based cybersecurity systems can provide strong detection capabilities while offering limited explanations regarding why a particular event is considered malicious. This paper proposes an Explainable Artificial Intelligence (XAI) framework for autonomous cyber threat detection that integrates Graph Neural Networks (GNNs) with Large Language Models (LLMs). The proposed architecture represents network entities and their interactions as dynamic graphs, enabling GNNs to identify structural, behavioral, and temporal attack patterns. The resulting graph embeddings, threat scores, and security events are subsequently supplied to an LLM-based reasoning layer, which correlates alerts with contextual information and generates human-readable explanations. An XAI layer further identifies influential nodes, edges, features, and behavioral patterns responsible for the detection decision. The framework incorporates a threat-intelligence knowledge base based on structured cybersecurity information such as MITRE ATT&CK techniques and supports autonomous alert prioritization, incident summarization, and response recommendation. A security guardrail layer is proposed to reduce hallucination, prompt injection, unsafe automated actions, and unauthorized response behavior. The research establishes an evaluation methodology covering detection accuracy, precision, recall, F1-score, false-positive rate, detection latency, explanation fidelity, explanation consistency, LLM hallucination rate, computational cost, and autonomous-response reliability. The proposed framework aims to bridge the gap between high-performance graph-based threat detection and human-understandable cybersecurity reasoning, providing a foundation for trustworthy, explainable, and semi-autonomous cyber defense.

Keywords: Explainable Artificial Intelligence, Graph Neural Networks, Large Language Models, Cybersecurity, Autonomous Threat Detection, Intrusion Detection, Cyber Threat Intelligence, Security Operations Center, Graph Learning, Generative AI

INTRODUCTION

Digital transformation has resulted in highly interconnected computing environments containing servers, endpoints, cloud services, Internet of Things (IoT) devices, industrial systems, applications, users, and network infrastructure. Although interconnected environments improve efficiency and availability, they also create complex attack surfaces. Modern cyberattacks may involve multiple stages, devices, credentials, communication channels, and attack techniques rather than a single observable malicious event.

Traditional cybersecurity systems generally rely on rule-based detection, signatures, statistical thresholds, or conventional machine-learning algorithms. These approaches remain important, but they can have difficulty identifying sophisticated attacks that do not exactly match previously known patterns.

Artificial intelligence has therefore become increasingly important in cybersecurity research. Deep-learning methods can automatically learn representations from network traffic, system logs, endpoint telemetry, and other security data. However, many deep-learning models operate as black boxes. A security analyst may



receive an alert indicating that an event is malicious without understanding which network relationships or behavioral characteristics caused the prediction.

This problem is particularly significant in security operations. An analyst needs to determine not only:

“Is this event malicious?”

but also:

“Why was it detected, which entities are involved, what attack behavior does it represent, and what should be investigated next?”

Explainable Artificial Intelligence addresses this problem by providing information about the reasoning behind an AI prediction. NIST's AI Risk Management Framework identifies explainability, interpretability, transparency, security, privacy, and reliability as important characteristics of trustworthy AI.

At the same time, Graph Neural Networks provide an effective mechanism for representing relationships among cybersecurity entities. A network can naturally be modeled as a graph in which computers, users, applications, IP addresses, and services are nodes while communications and interactions are edges.

Large Language Models introduce another capability: contextual reasoning over heterogeneous textual and structured information. Recent cybersecurity research has investigated LLMs for threat detection, threat intelligence, security operations, vulnerability analysis, incident response, and security automation.

This research proposes combining GNNs and LLMs rather than treating them as competing approaches.

The proposed architecture assigns different responsibilities to each component:

- **GNN:** detects structural and behavioral anomalies.
- **XAI:** identifies influential features, nodes, and relationships.
- **LLM:** interprets detections and correlates them with cybersecurity knowledge.
- **Threat-intelligence layer:** provides structured contextual information.
- **Autonomous orchestration layer:** prioritizes alerts and proposes or executes controlled responses.

RESEARCH PROBLEM

The research addresses five major limitations in existing cybersecurity systems.

2.1 Limited Relational Understanding

Many traditional intrusion-detection models process network events independently. However, sophisticated attacks frequently involve relationships among multiple entities.

2.2 Limited Explainability

Deep-learning models can detect suspicious behavior without providing sufficient evidence for analysts.

2.3 Alert Overload

Security operations centers may receive large numbers of alerts, making manual investigation difficult.

2.4 Fragmented Security Information

Security decisions may require simultaneous analysis of:

- network traffic;
- authentication logs;
- endpoint events;
- DNS activity;
- threat intelligence;
- vulnerability information;
- historical incidents.



2.5 Unsafe Automation

Autonomous AI systems can introduce additional risks if an LLM hallucinates information, follows malicious instructions embedded in data, or performs an inappropriate security action.

Therefore, the research problem can be expressed as:

How can GNN-based relational threat detection and LLM-based contextual reasoning be integrated into an explainable and controlled autonomous cybersecurity architecture?

RESEARCH OBJECTIVES

The objectives of this study are:

1. To develop a graph-based representation of cyber environments.
2. To design a GNN-based cyber threat detection mechanism.
3. To integrate GNN-generated security representations with an LLM.
4. To develop an XAI mechanism for explaining GNN predictions.
5. To use cybersecurity knowledge bases to improve LLM reasoning.
6. To develop autonomous alert correlation and prioritization.
7. To introduce guardrails for safe AI-assisted security actions.
8. To establish an experimental framework for evaluating detection and explanation performance.

RESEARCH CONTRIBUTIONS

The proposed research makes the following conceptual contributions:

1. **GNN-based relational threat detection:** cyber entities are modeled as dynamic graphs.
2. **LLM-based security reasoning:** graph-based detections are translated into contextual threat narratives.
3. **Integrated XAI:** explanations identify important features, nodes, and edges.
4. **Threat-intelligence augmentation:** external cybersecurity knowledge is used to contextualize detections.
5. **Controlled autonomy:** autonomous recommendations are separated from high-impact actions.
6. **Multi-dimensional evaluation:** detection performance and explainability are evaluated together.

RELATED WORK

5.1 AI-Based Cybersecurity

Machine learning has been applied to:

- intrusion detection;
- malware detection;
- phishing detection;
- anomaly detection;
- user behavior analytics;
- vulnerability assessment;
- threat intelligence.

A recent systematic survey of LLM applications in cybersecurity identifies applications across attack analysis, cyber threat intelligence, and security operations while also highlighting risks associated with LLM deployment.

However, conventional machine-learning systems often focus on classification performance rather than contextual reasoning.



GRAPH NEURAL NETWORKS FOR CYBER THREAT DETECTION

A cybersecurity environment can be modeled as a graph:

$$G=(V,E,X) \quad G=(V,E,X)$$

where:

- V represents entities;
- E represents relationships;
- X represents entity and relationship features.

Possible nodes include:

- IP addresses;
- hosts;
- users;
- applications;
- processes;
- domains;
- cloud resources;
- IoT devices.

Edges may represent:

- network connections;
- authentication;
- file access;
- process execution;
- DNS resolution;
- API calls;
- remote sessions.

A GNN can aggregate information from neighboring entities.

For a graph convolutional layer:

$$H^{(l+1)} = \sigma \left(\frac{1}{2} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right) \quad H^{(l+1)} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right)$$

where:

- $\tilde{A}$ is the adjacency matrix with self-connections;
- $\tilde{D}$ is the degree matrix;
- $H^{(l)}$ is the node representation;
- $W^{(l)}$ is a trainable parameter matrix;
- σ is an activation function.

The resulting representation captures information from both the entity and its neighborhood.

DYNAMIC CYBERSECURITY GRAPH

Real networks change continuously. Therefore, a static graph is insufficient.

The proposed system constructs:

$$G_t=(V_t,E_t,X_t) \quad G_t=(V_t,E_t,X_t)$$

for time interval t .

A temporal graph allows the system to identify:

- new communication relationships;
- unusual connection sequences;
- sudden increases in traffic;
- abnormal authentication chains;
- lateral movement;
- coordinated attacks.



For example, a workstation communicating with a database server may be normal. However, if the workstation suddenly communicates with several previously unrelated servers after a suspicious authentication event, the **relationship pattern** itself can become an important security signal.

PROPOSED GNN THREAT DETECTION MODEL

The GNN produces an embedding for each entity:

$$z_i = \text{GNN}(x_i, N_i) \quad z_i = \text{GNN}(x_i, N_i)$$

where:

- x_i represents the features of entity i ;
- N_i represents its neighborhood;
- z_i is the learned representation.

The classification layer generates:

$$P(y_i|G) = \text{Softmax}(Wz_i + b) \quad P(y_i|G) = \text{Softmax}(Wz_i + b)$$

The output may contain classes such as:

- benign;
- reconnaissance;
- brute-force attack;
- denial-of-service;
- credential abuse;
- malware activity;
- lateral movement;
- command-and-control;
- data exfiltration.

The exact classes depend on the dataset used for evaluation.

EXPLAINABLE AI LAYER

9.1 Need for XAI

A cybersecurity model should not simply output:

Threat probability = 0.97

It should also provide evidence.

For example:

The alert was generated because the host exhibited an unusual authentication sequence, established communication with previously unseen endpoints, and formed a communication pattern associated with lateral movement.

Such explanations can improve analyst understanding and facilitate incident investigation.

NIST notes that explainability can support debugging, monitoring, documentation, auditing, and understanding of AI behavior, while also warning that explanation methods can themselves have limitations such as insufficient fidelity or consistency.

FEATURE-LEVEL EXPLANATION

For a prediction $f(x)$, an attribution method can estimate:

$$f(x) \approx \phi_0 + \sum_{j=1}^m \phi_j f(x) \approx \phi_0 + \sum_{j=1}^m \phi_j$$

where ϕ_j represents the contribution of feature j .

Potential influential features include:

- connection frequency;
- packet rate;



- destination-port distribution;
- authentication failures;
- process behavior;
- DNS frequency;
- session duration;
- unusual protocol usage.

SHAP- or LIME-style approaches can be used for feature-level interpretation.

GRAPH-LEVEL EXPLANATION

Feature importance alone is insufficient for a graph model.

The proposed system also identifies important graph components:

$$V^* \subseteq V \wedge V^* \subseteq V$$

and

$$E^* \subseteq E \wedge E^* \subseteq E$$

where:

- V^* represents influential nodes;
- E^* represents influential relationships.

For example:

Primary suspicious node: Host A

Associated node: Host B

Important relationship: $A \rightarrow B$

Reason: unexpected communication following authentication anomaly.

This provides an analyst with a structural explanation of the alert.

LARGE LANGUAGE MODEL INTEGRATION

The LLM does not directly replace the GNN.

Instead, it acts as a **contextual reasoning and communication layer**.

The GNN produces structured evidence:

$$Z = \{z_i, E^*, V^*, P(y|G), F^*\} \quad Z = \{z_i, E^*, V^*, P(y|G), F^*\}$$

This information is combined with:

- security logs;
- asset information;
- vulnerability information;
- threat intelligence;
- historical incidents;
- MITRE ATT&CK knowledge.

The LLM receives a structured security context rather than unrestricted raw network traffic.

LLM-BASED THREAT REASONING

The LLM performs several functions.

13.1 Alert Summarization

It converts technical events into concise analyst-oriented summaries.

13.2 Alert Correlation

Multiple alerts can be grouped into a potential attack campaign.



13.3 Attack-Stage Identification

The system can map evidence to possible attack stages.

MITRE ATT&CK provides a knowledge base of adversary tactics and techniques and is designed to support threat-informed cyber defense.

13.4 Threat Explanation

The LLM transforms GNN explanations into natural language.

13.5 Investigation Assistance

The LLM can generate investigation questions such as:

- Which account initiated the suspicious session?
- Did the host communicate with additional systems?
- Was there an unusual process execution?
- Are vulnerabilities present on the affected host?

RETRIEVAL-AUGMENTED THREAT INTELLIGENCE

A major weakness of standalone LLMs is that their responses may lack current or organization-specific security context.

The proposed system therefore uses Retrieval-Augmented Generation (RAG).

The retrieval layer can access:

- MITRE ATT&CK;
- internal security policies;
- incident reports;
- vulnerability databases;
- asset inventories;
- threat-intelligence feeds;
- security playbooks.

The process is:

Query→Retriever→Relevant Knowledge→LLM→Security Explanation
Query \rightarrow Retriever \rightarrow Relevant Knowledge \rightarrow LLM \rightarrow Security Explanation

This reduces dependence on the LLM's internal knowledge.

PROPOSED GNN-LLM-XAI ARCHITECTURE

The proposed architecture contains seven layers.

Layer 1 — Data Collection

Sources include:

- network flows;
- firewall logs;
- authentication logs;
- endpoint telemetry;
- DNS logs;
- cloud events;
- application logs.

Layer 2 — Graph Construction

Events are transformed into:

$G_t = (V_t, E_t, X_t)$ $G_{-t} = (V_{-t}, E_{-t}, X_{-t})$



Layer 3 — GNN Detection

The GNN identifies:

- anomalous nodes;
- anomalous relationships;
- attack patterns;
- threat probabilities.

Layer 4 — XAI

The explanation engine identifies:

- influential features;
- influential nodes;
- influential edges;
- temporal evidence.

Layer 5 — Threat Intelligence

The system retrieves relevant cybersecurity knowledge.

Layer 6 — LLM Reasoning

The LLM:

- correlates alerts;
- explains findings;
- identifies possible attack techniques;
- summarizes incidents;
- recommends investigation steps.

Layer 7 — Security Orchestration

The system assigns:

- severity;
- priority;
- recommended response;
- analyst escalation status.

AUTONOMOUS THREAT DETECTION WORKFLOW

The proposed workflow is:

Data→Graph→GNN→XAI→Threat Intelligence→LLM→Risk Assessment→ResponseData \rightarrow Graph \rightarrow GNN \rightarrow XAI \rightarrow Threat\ Intelligence \rightarrow LLM \rightarrow Risk\ Assessment \rightarrow Response

Step 1

Collect security telemetry.

Step 2

Construct a temporal cybersecurity graph.

Step 3

Run GNN inference.

Step 4

Identify suspicious graph structures.

Step 5

Generate an explanation.

Step 6

Retrieve relevant threat intelligence.



Step 7

Send structured evidence to the LLM.

Step 8

Generate a threat narrative.

Step 9

Assign a risk score.

Step 10

Prioritize the incident.

Step 11

Recommend a response.

Step 12

Require human approval for high-impact actions unless a predefined automated policy explicitly permits the action.

RISK SCORING

A combined risk score can be defined as:

$$R = w_1 P_g + w_2 S_e + w_3 C_t + w_4 V_a + w_5 I_m$$

where:

- P_g = GNN threat probability;
- S_e = explanation-supported evidence score;
- C_t = threat-intelligence confidence;
- V_a = asset vulnerability;
- I_m = estimated impact;
- $w_1, \dots, w_5$ are configurable weights.

The resulting score can be mapped to operational categories such as:

- informational;
- low;
- medium;
- high;
- critical.

These categories should be configured according to organizational risk policies rather than treated as universally valid thresholds.

AUTONOMOUS RESPONSE ARCHITECTURE

The proposed system distinguishes between **recommendation** and **execution**.

Low-Risk Events

The system may automatically:

- enrich the alert;
- group duplicate alerts;
- update an investigation record;
- request additional telemetry.

Medium-Risk Events

The system may:

- recommend blocking;
- recommend credential reset;
- recommend endpoint isolation;
- increase monitoring.



High-Risk Events

The system should generally require an explicit authorization mechanism before destructive or disruptive actions.

This design reduces the risk associated with autonomous AI decisions.

LLM SECURITY GUARDRAILS

The introduction of an LLM creates new attack surfaces.

19.1 Prompt Injection

Attackers may insert malicious instructions into:

- log files;
- emails;
- web pages;
- threat-intelligence documents;
- endpoint metadata.

Therefore, retrieved content should be treated as **untrusted data**, not instructions.

19.2 Hallucination

An LLM may generate an apparently plausible but unsupported explanation.

The system should therefore require claims to be grounded in available evidence.

19.3 Tool Misuse

If the LLM can interact with security tools, its permissions should be restricted.

A principle of least privilege should be applied.

19.4 Response Validation

Before an automated response is executed:

Action→Policy Check→Risk Check→Authorization→ExecutionAction \rightarrow Policy\ Check \rightarrow Risk\ Check \rightarrow Authorization \rightarrow Execution

The LLM should not have unrestricted control over security infrastructure.

NIST's Generative AI Profile specifically addresses risks associated with generative AI and provides a framework for incorporating trustworthiness considerations into the design, development, deployment, and evaluation of such systems.

PROPOSED ALGORITHM

Algorithm 1: Explainable Autonomous Cyber Threat Detection

Input: Network telemetry DD

Output: Threat report TT

1. Collect security events from multiple data sources.
2. Normalize and preprocess the events.
3. Construct temporal graph GtG_t.
4. Generate node and edge features.
5. Apply the trained GNN.
6. Calculate threat probabilities.
7. Identify suspicious nodes and edges.
8. Generate graph-level and feature-level explanations.
9. Retrieve relevant cybersecurity knowledge.
10. Construct a structured evidence package.



11. Submit the evidence package to the LLM.
12. Generate threat interpretation.
13. Map the event to possible attack techniques.
14. Calculate risk score.
15. Validate the LLM output against available evidence.
16. Assign alert priority.
17. Generate response recommendation.
18. Apply authorization and security-policy checks.
19. Execute only permitted automated actions.
20. Record the decision and explanation for auditing.

THREAT MODEL

The proposed framework considers four categories of threats.

21.1 Network Attacks

Examples include:

- scanning;
- denial-of-service;
- credential attacks;
- lateral movement;
- command-and-control;
- data exfiltration.

21.2 Adversarial ML Attacks

Examples include:

- data poisoning;
- adversarial examples;
- graph manipulation;
- evasion attacks;
- model poisoning.

21.3 LLM-Specific Attacks

Examples include:

- prompt injection;
- indirect prompt injection;
- malicious retrieval content;
- hallucination exploitation;
- tool-use manipulation.

21.4 Insider and Supply-Chain Risks

The framework should also consider compromised:

- endpoints;
- credentials;
- software dependencies;
- security tools;
- data sources.



EXPERIMENTAL METHODOLOGY

Because this paper proposes a framework rather than reporting a completed experiment, numerical performance results should be obtained through implementation rather than invented.

22.1 Datasets

Potential datasets include:

- CICIDS2017;
- CSE-CIC-IDS2018;
- UNSW-NB15;
- Bot-IoT;
- TON_IoT;
- suitable enterprise or industrial cybersecurity datasets.

A dataset should be selected according to the intended deployment environment.

DATA PREPROCESSING

The preprocessing pipeline consists of:

1. missing-value handling;
2. duplicate removal;
3. categorical encoding;
4. numerical normalization;
5. feature selection;
6. temporal ordering;
7. graph construction;
8. train/validation/test separation.

Care must be taken to prevent temporal or entity leakage.

BASELINE MODELS

The proposed system should be compared with:

1. Random Forest
2. Support Vector Machine
3. XGBoost
4. MLP
5. CNN
6. LSTM
7. GNN
8. GNN + XAI
9. LLM-based cybersecurity analysis
10. GNN + LLM
11. **Proposed GNN + XAI + LLM framework**

The comparison should use the same test partition and preprocessing protocol wherever possible.

EVALUATION METRICS

25.1 Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$



25.2 Precision

$$\text{Precision} = \frac{TP}{TP + FP}$$

25.3 Recall

$$\text{Recall} = \frac{TP}{TP + FN}$$

25.4 F1-Score

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

25.5 False Positive Rate

$$\text{FPR} = \frac{FP}{FP + TN}$$

25.6 Detection Latency

$$DL = T_{\text{alert}} - T_{\text{attack}}$$

25.7 Explanation Fidelity

Explanation fidelity measures how accurately the explanation reflects the model's actual decision process.

25.8 Explanation Stability

The explanation should remain reasonably consistent when small irrelevant changes are introduced into the input.

25.9 LLM Hallucination Rate

A useful metric is:

$$HR = \frac{\text{Unsupported claims}}{\text{Total factual claims}}$$

A lower value indicates better grounding.

25.10 Autonomous Action Reliability

The system should measure:

- correct action rate;
- unsafe action rate;
- unauthorized action rate;
- analyst override rate.

ABLATION STUDY

The contribution of each component can be investigated using the following configurations:

Model GNN XAI LLM Threat Intelligence

M1	No	No	No	No
M2	Yes	No	No	No
M3	Yes	Yes	No	No
M4	No	No	Yes	Yes
M5	Yes	No	Yes	No
M6	Yes	Yes	Yes	No
M7	Yes	Yes	Yes	Yes

M7 represents the complete proposed architecture.



The objective is to determine the individual and combined contribution of graph learning, explainability, LLM reasoning, and threat-intelligence retrieval.

EXPECTED RESEARCH OUTCOMES

The proposed framework is expected to provide improvements in several dimensions, subject to empirical validation.

Detection

GNNs can exploit relationships that are difficult to capture with independent feature-vector models.

Contextual Understanding

LLMs can convert heterogeneous security evidence into contextual incident narratives.

Explainability

The XAI layer can identify the evidence supporting a GNN prediction.

Analyst Productivity

Automated correlation and summarization may reduce repetitive manual investigation.

Threat Intelligence Integration

RAG can connect detected behavior with structured cybersecurity knowledge.

Controlled Autonomy

Policy-based orchestration can enable automation while restricting high-impact actions.

These are research hypotheses rather than measured results and should be validated experimentally.

DISCUSSION

The proposed GNN-LLM-XAI architecture addresses a fundamental limitation of many AI cybersecurity systems: the separation between **detection** and **interpretation**.

A GNN is well suited to the structural characteristics of cybersecurity data. Network environments naturally contain relationships among machines, users, applications, domains, and services.

An LLM, in contrast, is particularly useful for contextualizing heterogeneous information and communicating conclusions in natural language.

The combination therefore creates a division of responsibilities:

GNN→Detection GNN \rightarrow Detection XAI→Evidence XAI \rightarrow Evidence
RAG→Knowledge RAG \rightarrow Knowledge LLM→Reasoning LLM \rightarrow Reasoning
Policy Engine→Control Policy\ Engine \rightarrow Control

This separation is important because the LLM should not be treated as the sole authority for determining whether an event is malicious.

The AI Risk Management Framework emphasizes that trustworthy AI involves multiple characteristics, including validity and reliability, safety, security and resilience, accountability and transparency, explainability and interpretability, privacy, and fairness.



Consequently, a cybersecurity architecture based on LLMs should evaluate not only detection capability but also reliability, security, explainability, and operational safety.

CHALLENGES

29.1 Computational Complexity

Large graph models and LLMs can require substantial computational resources.

29.2 Real-Time Processing

Security operations require rapid decisions, whereas complex reasoning pipelines may introduce latency.

29.3 Data Quality

Incomplete or noisy logs can lead to incorrect graph representations.

29.4 Concept Drift

Network behavior changes over time, potentially reducing model performance.

29.5 Explanation Reliability

An explanation generated by an LLM may sound convincing even when the underlying evidence is weak.

29.6 LLM Security

The reasoning layer introduces prompt injection, hallucination, data leakage, and tool-use risks.

29.7 Lack of Standardized Evaluation

There is currently no single universally accepted metric covering detection accuracy, explainability, LLM reasoning quality, and autonomous response safety simultaneously.

FUTURE RESEARCH DIRECTIONS

30.1 Temporal GNNs

Temporal graph architectures could improve detection of multi-stage attacks.

30.2 Multimodal Security Models

Future systems could jointly process:

- network graphs;
- text logs;
- endpoint telemetry;
- malware behavior;
- threat-intelligence documents.

30.3 Smaller Security-Specific LLMs

Compact domain-specific models could reduce latency and deployment costs.

30.4 Federated GNN-LLM Systems

Organizations could collaborate without sharing raw security telemetry.

30.5 Continual Learning

Continuous adaptation could improve performance against emerging threats.



30.6 Human-AI Collaboration

Future research should examine how analysts interact with AI explanations and how analyst feedback can improve model reliability.

30.7 Autonomous Cyber Defense

Future systems may progress from:

Detection→Explanation→Recommendation→Controlled Automation
Detection \rightarrow Explanation
\rightarrow Recommendation \rightarrow Controlled Automation
while maintaining human oversight for high-impact decisions.

CONCLUSION

This paper proposed an Explainable AI framework for autonomous cyber threat detection based on the integration of Graph Neural Networks and Large Language Models. The architecture combines the structural learning capability of GNNs with the contextual reasoning and communication capabilities of LLMs.

The proposed framework represents cybersecurity environments as dynamic graphs, enabling detection of suspicious relationships and behavioral patterns. An XAI layer identifies the features, nodes, and edges contributing to security predictions. Threat-intelligence retrieval then provides contextual information, while an LLM converts structured evidence into human-readable explanations, incident summaries, and investigation recommendations.

The framework also recognizes that autonomous cybersecurity introduces new risks. Prompt injection, hallucination, model manipulation, unauthorized tool use, and incorrect automated responses must therefore be addressed through evidence grounding, retrieval controls, policy enforcement, authorization mechanisms, and auditing.

Rather than treating the LLM as an unrestricted autonomous security authority, the proposed architecture assigns complementary roles to the GNN, XAI, retrieval system, LLM, and security-policy engine. This separation provides a foundation for developing cybersecurity systems that are not only capable of identifying complex threats but can also explain their decisions and support security analysts in understanding and responding to incidents.

Future empirical research should implement the proposed architecture on representative cybersecurity datasets and evaluate detection performance, explanation fidelity, LLM reliability, computational overhead, robustness, and autonomous-response safety. Such evaluation will determine the practical effectiveness of GNN-LLM-XAI systems for real-world cyber defense.

REFERENCES

1. Tabassi, E. (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1. National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.AI.100-1>.
2. Autio, C., Schwartz, R., Dunietz, J., Jain, S., Stanley, M., Tabassi, E., Hall, P., & Roberts, K. (2024). *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile*. NIST AI 600-1. <https://doi.org/10.6028/NIST.AI.600-1>.
3. NIST. (2023). *AI Risk Management Framework FAQs*. National Institute of Standards and Technology.
4. NIST. (2026). *AI Risk Management Framework*. National Institute of Standards and Technology. The current NIST page notes ongoing work on a Trustworthy AI in Critical Infrastructure profile.
5. MITRE. (2026). *MITRE ATT&CK*. MITRE Corporation.



6. Jaffal, N. O., Alkhanafseh, M., & Mohaisen, D. (2025). *Large Language Models in Cybersecurity: Applications, Vulnerabilities, and Defense Techniques*. arXiv.
7. Tian, S., Zhang, T., Liu, J., Wang, J., Wu, X., Zhu, X., Zhang, R., Zhang, W., Yuan, Z., Mao, S., & Kim, D. I. (2025). *Exploring the Role of Large Language Models in Cybersecurity: A Systematic Survey*. arXiv.
8. Habibzadeh, A., Feyzi, F., & Ebrahimi Atani, R. (2025). *Large Language Models for Security Operations Centers: A Comprehensive Survey*. arXiv.
9. *Explainable AI in Cybersecurity: A Comprehensive Framework for Enhancing Transparency, Trust, and Human-AI Collaboration*. (2024). Proceedings of the 2024 International Seminar on Application for Technology of Information and Communication (iSemantic). <https://doi.org/10.1109/iSemantic63362.2024.10762690>.
10. NIST. (2026). *AI Research – Security and Resilience*. National Institute of Standards and Technology.