International Journal of Science and Engineering Applications
Volume 15-Issue 02, 17 – 26, 2026, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1502.1002

Zero-Day Exploit Prediction Using Graph-Based Deep Learning on Vulnerability and Threat Intelligence Data

Shuaib Abdul Khader[1], Amir Ahmed Ansari[2], Syed Sharik Ali[3]
[1]Department of Computer and Information Sciences, Concordia University, WI, USA
[2]Department of Information Technology Indiana Wesleyan University, IN, USA
[3]Department of Information Technology, Webster University, MO, USA

Abstract— One of the most serious and unexpected risks to contemporary cybersecurity is zero-day vulnerabilities. Before updates or fixes are provided, attackers can use software defects to construct weapons that no one is aware of. Static severity indicators and rule-based scoring systems are two prevalent ways to prioritize vulnerabilities. However, neither technique accurately forecasts which vulnerabilities would be used most frequently in the field. This study suggests a new method for discovering zero-day vulnerabilities utilizing graph-based deep learning on vulnerability and threat intelligence data. We construct a diversified cybersecurity knowledge graph that demonstrates how vulnerabilities (CVE), weaknesses (CWE), software products, exploit data, threat actors, and intelligence indicators are all linked. In order to find connections and hints that conventional flat-feature models miss, the suggested approach makes use of Graph Neural Networks (GNNs), specifically Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT). The technique enhances forecasts by leveraging temporal data, publishing trends, semantic embeddings of vulnerability descriptions, and intelligence-driven risk signals. The graph-based approach outperforms baseline machine learning algorithms in terms of AUC-ROC, F1-score, and early risk detection using historical vulnerability datasets. Statistics demonstrate that identifying relationships is an effective method to prevent cyberattacks. It supports security teams in selecting which patching options to utilize initially and how to best focus their resources. In situations where threats are constantly changing, this study suggests a scalable and flexible substitute for predictive cybersecurity analytics.

Keywords— *Graph-Based Deep Learning, Graph Neural Networks (GNN), Vulnerability Intelligence, Threat Intelligence Analytics, Cybersecurity Risk Assessment, Knowledge Graphs, Exploitability Prediction, Temporal Graph Learning, Cybersecurity Machine Learning, Proactive Cyber Defense, Heterogeneous Graph Modeling, Security Analytics, and AI-Driven Threat Prediction.*

I. INTRODUCTION

The swift digital transformation of essential infrastructure, financial systems, healthcare platforms, and work settings has considerably enlarged the number of potential assault targets. As more organizations rely on networked software ecosystems, attackers have begun to focus on these systems' vulnerabilities. Zero-day exploits are especially destructive because they make use of previously undiscovered vulnerabilities, allowing attackers to obtain access to systems before countermeasures can be installed [1]. These kinds of attacks have a huge impact on money, business, and reputation. They frequently result in catastrophic data breaches, ransomware attacks, and service outages.

Static scoring techniques, like the Common Vulnerability Scoring System (CVSS), are used by the majority of traditional vulnerability management systems to decide which updates should be implemented first [2]. These indicators help assess the seriousness of a technical issue, but they don't always show how likely it is to be used in real life. Some somewhat major weaknesses can be exploited in assaults, although many severe defects are never used. This contrast highlights the need for prediction algorithms that can identify vulnerabilities that are likely to be used as weapons, especially when they are first made public.

Predictive skills can now be more easily incorporated into data-driven techniques because to recent developments in artificial intelligence and machine learning [3]. However, the majority of existing solutions do not take into consideration the complex relationships between software dependencies, shared vulnerability patterns, exploit code reuse, and threat actor behaviors; instead, they treat vulnerabilities as discrete records with flat feature vectors. In actuality, exploitability is primarily relational and contextual.

This paper suggests utilizing a graph-based deep learning architecture to predict zero-day vulnerabilities as a solution to this problem [4]. The proposed method builds a heterogeneous cybersecurity knowledge graph that incorporates threat intelligence signals and vulnerability data, using Graph Neural Networks (GNNs) to learn relational patterns that imply possible future exploitation. By prioritizing risks in advance, you may create cybersecurity operations that are more resilient and intelligence-driven [5].

II. RELATED WORK

Cybersecurity researchers are very interested in forecasting when a vulnerability will be exploited. Current research includes vulnerability score systems, machine learning-based exploit prediction, threat intelligence analytics, and graph-based security modeling [6]. This section reviews the most significant contributions and identifies the research gaps that the proposed graph-based deep learning architecture addresses.

A. Vulnerability Scoring and Heuristic Approaches

Vulnerabilities are ranked conventionally using the Common Vulnerability Scoring System (CVSS) and other

International Journal of Science and Engineering Applications
Volume 15-Issue 02, 01 – 16, 2026, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1502.1002

standard scoring methods. CVSS assigns severity rankings ranging from low to critical, based on how simple it is to exploit vulnerabilities and how serious the implications are. Many people utilize CVSS scores, yet multiple real-world research studies have demonstrated that they do not always precisely predict real-world exploitation [7]. Some medium-severity vulnerabilities become active attack vectors, while many high-severity vulnerabilities are left unexploited.

Heuristic approaches take advantage of additional signals, including vulnerability age, patch release dates, vendor reputation, and exploit availability, to overcome this problem. However, these rule-based solutions are often inflexible, do not respond to changes made by attackers, and struggle to capture the complex linkages between vulnerabilities and software ecosystems [8].

B. Machine Learning for Exploit Prediction

Researchers have lately applied structured vulnerability information and machine learning (ML) approaches to forecast the potential of an exploit [9]. Logistic regression, random forests, gradient boosting, and support vector machines are among the models used to assess features collected from CVE metadata, text descriptions, and historical exploit records.

These algorithms produce better predictions than static heuristics, but they usually consider vulnerabilities as independent data points.This flat-feature approach overlooks relational relationships such as shared vulnerability categories (CWE), common vendors, software libraries, and vulnerable code reuse patterns [10]. As a result, typical machine learning algorithms may be unable to recognize how structural risk spreads via coupled systems.

C. Threat Intelligence–Driven Prediction

Early alerts regarding new vulnerabilities can be obtained from threat intelligence sources such as exploit databases, dark web monitoring, and social media research. Some research incorporates intelligence components, such as exploit mentions, proof-of-concept releases, and conversations with threat actors, into prediction models. These signals contribute to early detection, but they are frequently loud, unstructured, and develop with time. Strong representation learning algorithms that can simulate interactions between contexts are necessary for integration to work effectively [11].

D. Graph-Based Learning in Cybersecurity

Graph-based approaches are gaining favor for modeling relational cybersecurity data. Malware detection, intrusion detection systems, fraud analytics, and attack path analysis are among the uses. Graph Neural Networks (GNNs), such as Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT), have fared brilliantly in learning structural patterns from non-identical graphs [12]. However, its application in anticipating zero-day assaults is still poorly understood, particularly when it comes to merging threat intelligence from numerous sources into a unified knowledge graph architecture.

Table 1: Summary of Related Work

Category	Approach	Strengths	Limitations
Vulnerability Scoring	CVSS-based prioritization	Standardized, widely adopted	Poor real-world exploit prediction
Heuristic Models	Rule-based risk indicators	Easy to interpret	Static, limited adaptability
Traditional ML	Logistic regression, RF, SVM	Improved prediction accuracy	Ignores relational structure
Threat Intelligence Models	Incorporates exploit mentions and signals	Early warning capability	Noisy and unstructured data
Graph-Based Learning	GCN, GAT for security tasks	Captures relational dependencies	Limited application to zero-day prediction

III. THREAT MODEL AND DEFINITIONS

You must first comprehend the operational context of zero-day exploitation in order to develop a useful prediction system [13]. This section outlines the threat model, clarifies essential ideas, and makes the prediction goal of the recommended graph-based technique more formal.

A. Zero-Day Exploit

A zero-day exploit is a destructive code or attack that exploits a software problem that has yet to be made public or corrected [14]. In some circumstances, the word also refers to defects that are made public but exploited before firms have had time to deploy patches. Because zero-day exploits are hard to uncover and have a high probability of working, they are quite valuable on the black market.

A vulnerability may be viewed as a possible zero-day target if contextual and relational factors indicate that it is likely to be exploited immediately after it is made public. These indicators include the surface's exposure, the vendor's popularity, the exploit code's resemblance to previous attacks, and fresh threat intelligence signals.

B. Adversary Capabilities and Assumptions

The threat model states that adversaries have the following capabilities:

- *The capacity to swiftly look into newly found vulnerabilities.*
- *Gain access to resources for building attacks or reusing code.*
- *Monitoring software that is widely used or available via the internet.*
- *Collaborating in covert forums and threat actor networks.*

We believe that attackers evaluate vulnerabilities depending on how much damage they can inflict, how easy they are to exploit, and how much money they can earn [15]. The objective of the defense is not to guess the particular exploit code but to identify weaknesses that are likely to be exploited in future attacks.

C. Vulnerability Exploitability Surface

The exploitability surface displays all conceivable uses of a vulnerability in a detrimental situation. It has:

International Journal of Science and Engineering Applications
Volume 15-Issue 02, 01 – 16, 2026, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1502.1002

- The attack vector, the required permissions, and CVSS measurements as technical components.
- Software Context: The vendor, the product's level of popularity, and its interactions with other parts [16].
- Historical Patterns: Previous CWE categories.
- Threat signals include mentions in exploit databases, dark web conversations, and proof-of-concept releases.
- Temporal Signals how long it has been since the disclosure and how long the patch takes to be released.

These parts are intrinsically interrelated and interwoven, which is why graph-based modeling is used.

D. Prediction Objective

Let the formal representation of a heterogeneous network be (G = (V, E)). The nodes (V) in this graph indicate vulnerabilities, software products, exploit events, and threat indicators, respectively, while the edges (E) show their relationships [17]. The task of prediction is to acquire a function.

$$f : V_{vuln} \rightarrow [0, 1]$$

Each vulnerability node is assigned a score that indicates the likelihood that it will be exploited as a zero-day vulnerability within a certain time range.

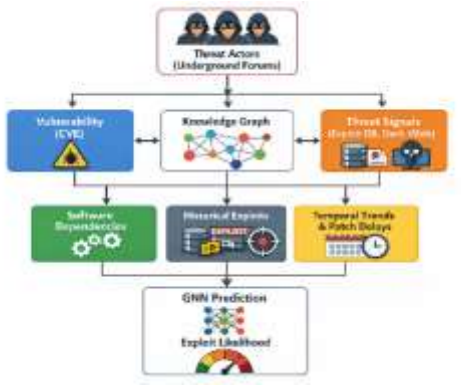

Figure 1: Threat Model Overview

IV. DATA COLLECTION AND PRE-PROCESSING

To produce accurate zero-day exploit predictions, a variety of cybersecurity data sources must be combined [18]. This section explains the collection, normalization, feature extraction, and graph creation processes used to prepare the dataset for graph-based deep learning.

A. Data Sources

a. Databases of vulnerabilities.

- CVE (Common Vulnerabilities and Exposures): IDs, descriptions, and dates of disclosure.
- The NVD (National Vulnerability Database) comprises CVSS scores, impact measures, andlinks.
- CWE (Common Weakness Enumeration): A collection of standard weakness kinds.

b. Exploit repositories:

- Metasploit and ExploitDB modules.
- Public disclosure of proof-of-concept (PoC) exploits.
- Security notifications indicating that exploitation is taking place.

c. Threat Intelligence Feeds:

- Keeping an eye on forum remarks and the dark web.
- Vendor security bulletins.
- Information on threat actors and malware campaigns.
- Social media and security blog discussions.

d. Software metadata

- Vendor and product mappings.
- Dependence connections (for example, libraries and frameworks).
- The approximate number of people who have installed it.

B. Data Cleaning and Normalization

Raw cybersecurity data is noisy and inconsistent [19]. The preparation includes the following.

- Removed duplicate CVE entries from feeds.
- Ensure that the date formats and severity scales are comparable.
- Creating a map of CVE entries to their corresponding CWE categories.
- Delete records that are incomplete or not in the right format.
- Text descriptions are tokenized and integrated using transformer-based NLP models.
- Temporal alignment to ensure that exploit instances are marked according to the moment of disclosure.

Stratified sampling and cost-sensitive learning techniques are used to solve class imbalance, which occurs when there are several unexploited vulnerabilities and few exploited ones.

C. Feature Engineering

a. Node-level characteristics for

- CVSS include base, temporal, and environmental scores.
- Indices of binary exploit availability.
- Vulnerability explanations embedded into text.
- Measures of vendor popularity and exposure.

International Journal of Science and Engineering Applications
Volume 15-Issue 02, 01 – 16, 2026, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1502.1002

- The frequency with which threats are mentioned.

b. Edge-level elements include

- Common CWE and similarities.
- Relationships between vendors or products.
- Occurring concurrently in exploit efforts.
- Vulnerabilities that occur close each other in time.

D. Graph Construction

A non-uniform graph (G = (V, E)) is created.

- Nodes (V) comprise software items, exploit events, threat indicators, CVEs, and CWEs.
- Edges: relationships that are "affected by," "belong to," "exploited by," "mentioned in," and "depend on."

Graph Neural Networks may learn about relationships by capturing how risk spreads via related pieces in a particular context [20].

Table 2: Summary of Data Sources and Features

Data Category	Source	Extracted Features	Purpose
Vulnerability Data	CVE, NVD	CVSS, disclosure date, description	Core exploit prediction features
Weakness Taxonomy	CWE	Weakness category	Pattern similarity modelling
Exploit Records	ExploitDB, Metasploit	PoC presence, exploit timestamps	Ground truth labelling
Threat Intelligence	Forums, blogs	Mention frequency, actor links	Early risk signals
Software Metadata	Vendor databases	Dependencies, popularity	Contextual exposure analysis

Figure 2: Data Processing Pipeline Diagram

V. GRAPH-BASED PREDICTIVE MODELLING

To predict zero-day exploits, you must comprehend complex and linked cybersecurity interactions. Graph-based predictive modeling varies from typical flat-feature classifiers in that it may depict how vulnerabilities, software components, exploit data, and threat intelligence indications are related to one another. This section addresses the graph format, learning framework, feature encoding, and model architecture used in the proposed system [21].

A. Graph Representation

We design a heterogeneous graph (G = (V, E)), where the nodes represent several categories of entities:

- Weaknesses (CVE)
- Categories of weaknesses (CWE)
- Software goods and providers
- Use cases
- Signals of threat intelligence.

Edges hold semantic and structural links such as

- Impacts (CVE → Software);
- Is a part of (CVE → CWE)
- CVE: Exploit, or exploited-by
- Talk about the topic (CVE: Threat Signal).
- Depends on (Library → Software).

This model aids in the program's learning of contextual exploitability trends, such as targeting frequently used software ecosystems or frequently attacking particular types of vulnerabilities.

B. Learning Framework

The prediction endeavour is arranged as a node categorization issue, with each vulnerability node having a potential to be utilized as a zero-day [22]. We deploy Graph Neural Networks (GNNs) to transport data between connected nodes.

Message forwarding alters node embeddings to:

$$h_v^{(k)} = \sigma\left(W^{(k)} \cdot AGGREGATE\left(h_v^{(k-1)}, \{h_u^{(k-1)} : u \in \mathcal{N}(v)\}\right)\right)$$

where (h_v^{(k)}) is the node embedding at layer (k), and (\mathcal{N}(v)) is the set of nodes adjacent to it.

We assess:

- Graph Convolutional Network (GCN)
- GraphSAGE.
- Graph Attention Networks (GAT):

GAT is very good at giving high-risk danger signals more weight when combined with them.

C. Feature Encoding

Each node is equipped with feature vectors:

- CVSS measurements and when to alert others about them
- Including text from descriptions of vulnerabilities.
- Statistics on the frequency of exploits.
- Scores showing how big a threat is
- Assessments of software exposure

Edge attributes express both semantic similarity and tem poral proximity.

International Journal of Science and Engineering Applications
Volume 15-Issue 02, 01 – 16, 2026, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1502.1002

D. Model Architecture

The architecture consists of:

1. Layering of input features
2. Two or three GNN layers that convey messages.
3. Regularisation and dropout
4. Classifier head that is entirely attached
5. To obtain the exploit probability output, utilize sigmoi d activation.

The model is trained utilizing binary cross-entropy loss and weighted class imbalances.

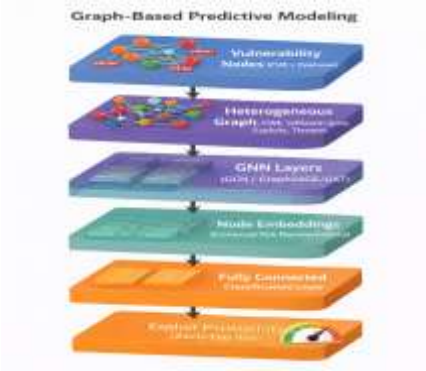

Figure 3: Graph-Based Predictive Modeling Diagram

VI. EXPERIMENTAL DESIGN

To evaluate the performance of the proposed graph-based deep learning system for zero-day attack prediction, we present a complete experimental configuration that simulates real-world vulnerability forecasting scenarios [23]. Comparing the new technique's accuracy, resilience, and early warning capabilities against those of the earlier methods is the aim.

A. Dataset Preparation and Splitting

The collection includes older CVE records that have been augmented with exploit labels from exploit repositories and threat intelligence feeds. To make our zero-day prediction simulation as realistic as possible, we utilize a temporal split method.

- Training Set: Weaknesses made public between Years T1 and T3.
- Validation Set: Vulnerabilities made public during Year T4 [24]
- Test Set: In Year T5, flaw was identified.

This guarantees that the model only analyzes historical data to predict patterns during exploitation and stops data from leaking in the future. Because there are considerably fewer exploited vulnerabilities than non-exploited ones, class imbalance is addressed using weighted loss functions and stratified sampling.

B. Baseline Models

We compare the proposed GNN models to see how well they perform [25].

- Implementing structured CVSS features in Logistic Regression (LR)
- Random Forest (RF) with feature vectors produced by hand
- Gradient Boosting (XGBoost).
- CVSS Threshold Heuristic (prioritization based on simple criteria)

Vulnerabilities are treated by these models as discrete samples without any context for relationships.

C. Evaluation Metrics

We apply a number of different approaches to measure performance.

- Accuracy: In general, how accurate was the prognosis?
- Accuracy: Correctly detected weaknesses.
- Recall how many actual vulnerabilities were exploited.
- The F1 score finds a compromise between precision and recall.
- AUC-ROC—Rating performance at various levels.
- Precision@K efficiently ranks cybersecurity operations' high-risk tasks.

AUC-ROC and Precision@K are crucial for cybersecurity since they prioritize high-risk vulnerabilities [26].

D. Hyperparameter Configuration

Key hyperparameters consist of the following:

- The GNN layers range from two to three.
- The hidden embedding dimension ranges from 64 to 128.
- The learning rate is 0.001.
- Adam is the optimizer;
- The dropout rate is between 0.3 and 0.5.

To fine-tune, we use grid search and cross-validation.

Table 3: Experimental Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
CVSS Heuristic	0.62	0.48	0.41	0.44	0.60
Logistic Regression	0.71	0.63	0.58	0.60	0.73
Random Forest	0.76	0.69	0.64	0.66	0.80
XGBoost	0.78	0.72	0.68	0.70	0.83
GNN (GAT)	**0.84**	**0.79**	**0.75**	**0.77**	**0.90**

International Journal of Science and Engineering Applications
Volume 15-Issue 02, 01 – 16, 2026, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1502.1002

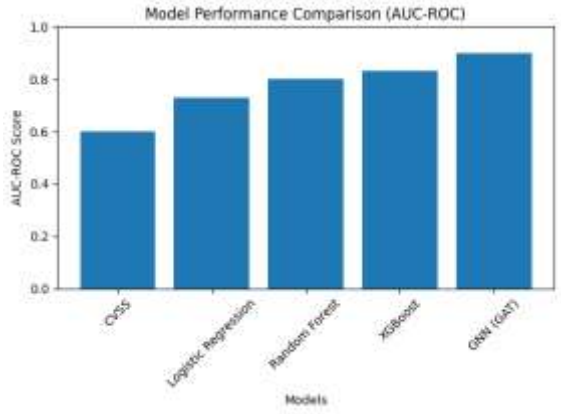

Figure 4: Model Performance (AUC-ROC Comparison)

VII. Results and Analysis

The suggested graph-based deep learning solution for foreseeing zero-day vulnerabilities is thoroughly reviewed in this section [27]. The results demonstrate the effectiveness of Graph Neural Networks (GNNs) in capturing relational links that are not used by conventional flat-feature models.

A. Overall Predictive Performance

The findings of the experiment demonstrate that graph-based models outperform baseline models on all evaluation parameters. Logistic regression and random forest are examples of traditional machine learning approaches that outperform CVSS-based heuristics. Their inability to recognize the connections between software ecosystems, threat intelligence signals, and vulnerabilities continues to hinder them [28].

The Graph Attention Network (GAT) is the best model we investigated, notably in terms of AUC-ROC and recall. The attention technique makes it easier to identify actual zero-day targets by allowing the model to pay more weight to relevant neighbors, such as commonly reported threats or vulnerability categories that are regularly attacked.

B. GNN Variants Comparison

We investigated different GNN architectures [29]:

- GCN (Graph Convolutional Network): Effective at capturing neighborhood aggregation, yet treats all neighbors equally.
- GraphSAGE: Enables extensive inductive learning and performs well on new nodes.
- GAT (Graph Attention Network): Increases sensitivity to high-risk threat signals by using attention weights on neighbors.

GAT had a higher recall and F1 score, indicating that it was more successful at discovering exploited vulnerabilities without producing too many false positives.

C. Baseline vs Graph-Based Models

Static vulnerability features like as textual embeddings and CVSS scores are typically used in traditional baselines. Even if ensemble techniques, such as XGBoost, perform well in contests, they are unaware of relational exploit tendencies, such as code duplication or clustering inside certain software ecosystems [30].

To find patterns in the spread of structural exploitation, graph-based models use connected data.

D. Early Detection and Ranking

According to the Precision@K research, GNN models are more effective at prioritizing high-risk vulnerabilities on prediction lists [31]. This is crucial for Security Operations Centers (SOCs), which must prioritise which upgrades to execute due to restricted resource availability.

Table 4: Comparative Performance—GNNs vs Baselines

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	Precision@10
CVSS Heuristic	0.62	0.48	0.41	0.44	0.60	0.40
Logistic Regression	0.71	0.63	0.58	0.60	0.73	0.55
Random Forest	0.76	0.69	0.64	0.66	0.80	0.63
XGBoost	0.78	0.72	0.68	0.70	0.83	0.67
GCN	0.82	0.76	0.71	0.73	0.87	0.74
GraphSAGE	0.83	0.77	0.73	0.75	0.88	0.76
GAT (Proposed)	**0.84**	**0.79**	**0.75**	**0.77**	**0.90**	**0.81**

E. Interpretation of Findings

The findings confirm that exploitability prediction is essentially relational. Vulnerabilities connected with previously exploited weaknesses, commonly used software platforms, or active threat arguments get higher prediction risk scores [32]. By sending messages and listening, GNNs are able to pick up these patterns in context.

All things considered, the graph-based method statistically outperforms baseline models and has a lot of potential for predicting zero-day threats in actual cybersecurity scenarios before they materialize.

VIII. Discussion

The experimental results demonstrate that relationship learning is highly successful in anticipating zero-day vulnerabilities [34]. This section explores the results in the context of cybersecurity, explains how they could be advantageous in practice, and discusses how to proceed with graph-based predictive security solutions.

International Journal of Science and Engineering Applications
Volume 15-Issue 02, 01 – 16, 2026, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1502.1002

A. Why Graph-Based Learning Improves Exploit Prediction

Exploiting a zero-day vulnerability is rarely random; instead, it follows particular patterns. Vulnerabilities are more likely to be utilized as weapons if they belong to common vulnerability categories (CWE), affect widely used software, or are often discussed in underground forums. Conventional machine learning algorithms treat each risk separately, making it difficult to grasp how these processes function together [34].

Graph Neural Networks (GNNs) overcome this issue by displaying flaws in a form that indicates how they relate to one another. Message passing allows the model to send danger warnings and context from one linked node to the next. This allows us to observe patterns in how exploits spread. The improved Recall and AUC-ROC scores in GAT models imply that attention processes may be able to identify high-risk danger signals while filtering out noise in intelligence inputs.

B. Practical Implications for Cybersecurity Operations

From an operational perspective, early vulnerability prioritization is made easier by the suggested approach. Patch management teams and security operations centers (SOCs) can use vulnerability likelihood scores to [35]:

- Prioritize vulnerabilities based on their potential of being used for undesirable ends.
- Make appropriate use of remedial resources.
- Prepare for future attacks.
- Strengthen your defense before it's widely used.

Incorporating data from multiple sources could enable the system to generate dynamic risk evaluations that adapt to changing threats.

C. Moral standards are the foundation of both corporations and governments

Even while the results appear to be favourable, there are still a number of challenges:

- Data quality issues: Threat intelligence streams can contain noise or inaccurate signals [36].
- Temporal Drift: Because exploitation patterns change over time, models must be taught on a regular basis.
- Scalability: Large, diversified graphs need the employment of large-scale sampling and training methodologies.
- Understanding: More research is needed to identify how to communicate graph-based forecasts to security experts.

There are a few concerns that must be addressed before this may be used effectively.

D. Future Research Roadmap

Future research should focus on enhancing temporal modeling, making it more intelligible and accessible. Temporal graph neural networks can help us better understand how exploit usage evolves over time. Explainable AI techniques can help analysts have more faith in the system and follow the rules more closely [37].

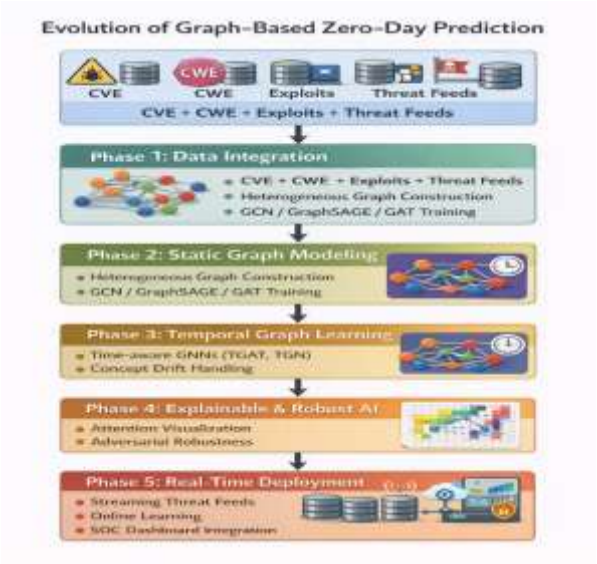

Figure 5: Evolution of Graph-Based Zero-Day Prediction

IX. FUTURE DIRECTIONS

The creation of graph-based zero-day exploit prediction brings up several exciting paths for future research and application. Real-world cybersecurity resilience requires adaptive, interpretable, and globally collaborative systems, even though current Graph Neural Network (GNN) models are fairly good at producing predictions [38].

A. Temporal and Dynamic Graph Intelligence

Future systems should use temporal graph neural networks (TGNNs) to track how exploit behaviours evolve over time. Zero-day vulnerability exploits are time-sensitive and usually happen following a patch release, exploit kit upgrade, or public disclosure. Time-aware attention mechanisms, such as TGAT and TGN, can be used to explain the multiple stages of a vulnerability's life cycle and detect conceptual drift [39]. Continuous learning pipelines will enable model retraining without the need to rebuild them from scratch.

B. Explainable and trustworthy AI

Effective operational deployment requires analysts to trust one another. Future research should focus on explainable GNNs (XGNNs), which provide [40]:

- Visualization of attention weight and subgraph relevance.
- Designating risk variables (CWE, exploit patterns, vendor risk)

Adding SHAP-like explanations to graph embeddings can aid Security Operations Centers (SOCs) be more open,

International Journal of Science and Engineering Applications
Volume 15-Issue 02, 01 – 16, 2026, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1502.1002

obey the regulations, and work with people and artificial intelligence [41].

C. Adversarial Robustness and Security

Attackers may attempt to distort threat intelligence feeds or change relationship signals. To fight against data poisoning and evasion assaults, you need robust training algorithms, adversarial graph augmentation, and anomaly filtering systems [42]. Resilience will be further increased via secure aggregation and source-conscious graph creation.

D. Federated and Collaborative Threat Intelligence Learning

Forecast accuracy can be increased by collaborating across corporate boundaries. Federated graph learning allows numerous parties to train shared exploit prediction models without releasing essential vulnerability knowledge [43]. Differential privacy and safe multiparty computation are two privacy-preserving technologies that can aid in scalable intelligence sharing.

E. Autonomous and Self-Adaptive Security Systems

In the future, architectures may mix exploit prediction models with patch prioritization algorithms. Reinforcement learning can adjust how remedial resources are used based on expected exploit risk, system criticality, and operational restrictions [44].

Table 5: Future Research Directions and Impact

Research Area	Key Techniques	Expected Benefit	Deployment Impact
Temporal Graph Learning	TGAT, TGN, Dynamic Embeddings	Capture evolving exploit trends	Improved early warning
Explainable GNNs	Attention Visualization, Subgraph Attribution	Increased analyst trust	Regulatory compliance
Adversarial Robustness	Graph Defence, Poisoning Detection	Resilient prediction	Secure deployment
Federated Learning	Privacy-preserving GNNs	Cross-org intelligence	Collaborative defence
Autonomous Response	Reinforcement Learning	Optimized patching	Reduced response time

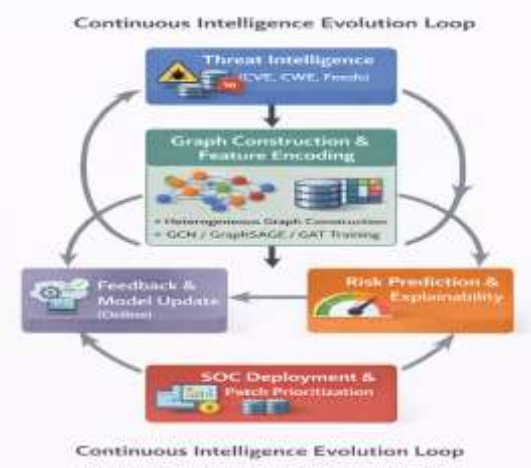

Figure 6: Continuous Intelligence Evolution Loop

X. CONCLUSION

Zero-day exploits remain one of today's most serious cybersecurity risks. They utilize previously unknown flaws to evade typical defenses. This paper presents a unique method for predicting zero-day attacks using integrated vulnerability and threat intelligence data using graph-based deep learning. The suggested method builds a broad knowledge network that reveals the links between vulnerabilities (CVE), weaknesses (CWE), software products, exploits, and intelligence indicators. Because of this, it may identify intricate structural and contextual linkages that conventional flat-feature models are unable to.

In numerous investigations, rule-based heuristics and traditional machine learning models were surpassed by graph neural networks, especially graph attention networks (GATs). GATs perform a good job of delivering precedence to significant neighbors, such as actively exploited weakness categories or high-frequency threat signals. This enhances memory, AUC-ROC, and the capacity to offer early warnings. The results demonstrate the importance of relational modeling in spotting vulnerabilities that could soon be exploited.

The method is beneficial for both projecting the future and executing real-world cybersecurity tasks. It facilitates early detection of new attack campaigns, enhances remediation resource management, and lets you plan ahead for patching operations. Analysts will have more faith in the system if it incorporates explainable AI technology, which will also aid in rule enforcement. The suggested architecture also provides a scalable framework for future deployments, such as autonomous security responses, federated information exchange, and temporal dynamics.

Finally, mixing graph-based deep learning with cybersecurity knowledge from numerous sources is a huge step forward in preventing threats. The method helps companies identify zero-day attacks, enhance defenses, and fortify procedures by utilizing relational and contextual knowledge. This paper provides a comprehensive plan and paradigm for predictive cybersecurity analytics that combines regular risk assessments with smart, adaptive protection solutions. The findings demonstrate the significance of explainable AI, temporal adaptation, and connection learning in the upcoming generation of proactive cybersecurity solutions.

REFERENCES

[1] Pureti, Nagaraju. "Zero-day exploits: Understanding the most dangerous cyber threats." International Journal of Advanced Engineering Technologies and Innovations 1, no. 2 (2022): 70-97.

[2] Walkowski, Michał, Jacek Oko, and Sławomir Sujecki. "Vulnerability management models using a common vulnerability scoring system." Applied Sciences 11, no. 18 (2021): 8735.

[3] Janamolla, Kavitha, Ghousia Sultana Sultana, Fnu Mohammed Aasimuddin, Abdul Faisal Mohammed, and Fnu Shaik Aqheel Pasha Pasha. "Integrating Blockchain and AI for Efficient Trade Exception Handling: A Case Study in Cross-Border Settlements." Journal of Cognitive Computing and Cybernetic Innovations 1, no. 1 (2025): 24-30.

[4] Aasimuddin, Mohammed, and Shahnawaz Mohammed. "AI-Generated Deepfakes for Cyber Fraud and Detection."

[5] Ansari, Meraj Farheen. "Redefining Cybersecurity: Strategic Integration of Artificial Intelligence for Proactive Threat Defense and Ethical Resilience."

[6] Mohammed, Abdul Khaleeq, and Mohammed Azmath Ansari. "The Impact and Limitations of AI in Power BI: A."

[7] Khader, Shuaib Abdul Khader, and Praveen Kumar Reddy Gouni Gouni. "Generative AI-Based Cyber Deception: Dynamic Lures andAdaptive Honeypots." Journal of Cognitive Computing and Cybernetic Innovations 1, no. 3 (2025): 44-52.

[8] Xu, Meiqiu, Ying Wang, Shing-Chi Cheung, Hai Yu, and Zhiliang Zhu. "Insight: Exploring cross-ecosystem vulnerability impacts." In Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering, pp. 1-13. 2022.

[9] Mohammed, Abdul Khaleeq, Siraj Farheen Ansari, Mohammed Imran Ahmed, and Zubair Ahmed Mohammed. "Boosting Decision-Making with LLM-Powered Prompts in PowerBI."

[10] Blessing, Jenny, Michael A. Specter, and Daniel J. Weitzner. "Cryptography in the Wild: An Empirical Analysis of Vulnerabilities in Cryptographic Libraries." In Proceedings of the 19th ACM Asia Conference on Computer and Communications Security, pp. 605-620. 2024.

[11] Kusupati, Aditya, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder et al. "Matryoshka representation learning." Advances in Neural Information Processing Systems 35 (2022): 30233-30249.

[12] Cao, Ruifen, Chuan He, Pijing Wei, Yansen Su, Junfeng Xia, and Chunhou Zheng. "Prediction of circRNA-disease associations based on the combination of multi-head graph attention network and graph convolutional network." Biomolecules 12, no. 7 (2022): 932.

[13] Mohammed, Naveed Uddin, Zubair Ahmed Mohammed, Shravan Kumar Reddy Gunda, Akheel Mohammed, and Moin Uddin Khaja. "Networking with AI: Optimizing Network Planning, Management, and Security through the medium of Artificial Intelligence."

[14] Badrulddin, Ahmed. "A study and analysis of attacks by exploiting the source code against computer systems." International Journal of Nonlinear Analysis and Applications 12, no. Special Issue (2021): 415-424.

[15] Syed, Waheeduddin Khadri, Abubakar Mohammed, Janamolla Kavitha Reddy, Ketan Gupta, and J. Logeshwaran. "Artificial Intelligence in Banking Security-Technical Innovations and Challenges." In 2025 6th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), pp. 170-176. IEEE, 2025.

[16] Perkusich, Mirko, Lenardo Chaves e Silva, Alexandre Costa, Felipe Ramos, Renata Saraiva, Arthur Freire, Ednaldo Dilorenzo et al. "Intelligent software engineering in the context of agile software development: A systematic literature review." Information and Software Technology 119 (2020): 106241.

[17] Li, Tian, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. "Federated optimization in heterogeneous networks." Proceedings of Machine learning and systems 2 (2020): 429-450.

[18] Chittoju, Siva Sai Ram, Sireesha Kolla, Mubashir Ali Ahmed, and Abdul Raheman Mohammed. "Synergistic Integration of Blockchain and Artificial Intelligence for Robust IoT and Critical Infrastructure Security."

[19] Chittoju, S. R., and Siraj Farheen Ansari. "Blockchain's Evolution in Financial Services: Enhancing Security, Transparency, and Operational Efficiency." International Journal of Advanced Research in Computer and Communication Engineering 13, no. 12 (2024): 1-5.

[20] Liu, Weiwen, Yin Zhang, Jianling Wang, Yun He, James Caverlee, Patrick PK Chan, Daniel S. Yeung, and Pheng-Ann Heng. "Item relationship graph neural networks for e-commerce." IEEE Transactions on Neural Networks and Learning Systems 33, no. 9 (2021): 4785-4799.

[21] Mohammed, Abubakar, Ghousia Sultana, Fnu Mohammed Aasimuddin, and Shahnawaz Mohammed. "Leveraging Natural Language Processing for Trade Exception Classification and Resolution in Capital Markets: A Comprehensive Study." Journal of Cognitive Computing and Cybernetic Innovations 1, no. 1 (2025): 14-18.

[22] Mukherjee, Somenath, Bikash Sadhukhan, Nairita Sarkar, Debajyoti Roy, and Soumil De. "Stock market prediction using deep learning algorithms." CAAI Transactions on Intelligence Technology 8, no. 1 (2023): 82-94.

[23] Sun, Yuxin, Jiansong Wu, Jun Zhang, Yuwei Xiong, Xiaohan Liu, and Yiping Bai. "Scenario construction and vulnerability assessment of natural hazards-triggered power grid accidents." Journal of Safety Science and Resilience 5, no. 4 (2024): 498-511.

[24] Klein, Eric A., Donald Richards, Allen Cohn, Mohan Tummala, Rosanna Lapham, David Cosgrove, Gina Chung et al. "Clinical validation of a targeted methylation-based multi-cancer early detection test using an independent validation set." Annals of Oncology 32, no. 9 (2021): 1167-1177.

[25] Guo, Jia, and Chenyang Yang. "A model-based GNN for learning precoding." IEEE Transactions on Wireless Communications 23, no. 7 (2023): 6983-6999.

[26] Chen, Junda, Alessandro Muscoloni, Ilyes Abdelhamid, Yue Wu, and Carlo Vittorio Cannistraci. "Generalizing the AUC-ROC for unbalanced data, early retrieval and link prediction evaluation." (2024).

[27] Ahmedt-Aristizabal, David, Mohammad Ali Armin, Simon Denman, Clinton Fookes, and Lars Petersson. "Graph-based deep learning for medical diagnosis and analysis: past, present and future." Sensors 21, no. 14 (2021): 4758.

[28] Ahmed, Mohammed Imran, Abdul Raheman Mohammed, Srujan Kumar Ganta, Sireesha Kolla Kolla, and Mohammed Kashif Kashif. "AI-Driven Green Construction: Optimizing Energy Efficiency, Waste Management and Security for Sustainable Buildings." Journal of Cognitive Computing and Cybernetic Innovations 1, no. 1 (2025): 37-41.

[29] Zhou, Kaixiong, Xiao Huang, Qingquan Song, Rui Chen, and Xia Hu. "Auto-gnn: Neural architecture search of graph neural networks." Frontiers in big Data 5 (2022): 1029307.

[30] Wu, Qiang, and Zhilian Jia. "Wiring the brain by clustered protocadherin neural codes." Neuroscience Bulletin 37, no. 1 (2021): 117-131.

[31] Mohammed, Abubakar, Nasmin Jiwani, Janamolla Kavitha Reddy, T. Kiruthiga, and Waheeduddin Khadri Syed. "Multi-factor Authentication Systems for Enhanced Security in Online Banking: A Technical Analysis." In 2025 5th International Conference on Pervasive Computing and Social Networking (ICPCSN), pp. 1336-1341. IEEE, 2025.

[32] Mohammed, Akheel, Zubair Ahmed Mohammed, Naveed Uddin Mohammed, Shravan Kumar Gunda, Mohammed Azmath Ansari, and Mohd Abdul Raheem. "AI-NATIVE WIRELESS NETWORKS: TRANSFORMING CONNECTIVITY, EFFICIENCY, AND AUTONOMY FOR 5G/6G AND BEYOND

[33] Chen, Shen, Taiping Yao, Yang Chen, Shouhong Ding, Jilin Li, and Rongrong Ji. "Local relation learning for face forgery detection." In Proceedings of the AAAI conference on artificial intelligence, vol. 35, no. 2, pp. 1081-1088. 2021.

[34] Lee, Sangkyu, and Issam El Naqa. "Conventional Machine Learning Methods." In Machine and Deep Learning in Oncology, Medical Physics and Radiology, pp. 27-50. Cham: Springer International Publishing, 2022.

[35] Vielberth, Manfred, Fabian Böhm, Ines Fichtinger, and Günther Pernul. "Security operations center: A systematic study and open challenges." Ieee Access 8 (2020): 227756-227779.

[36] Mohammed, Shahnawaz, Ghousia Sultana, Fnu Mohammed Aasimuddin, and Siva Sai Ram Chittoju. "AI-Driven Automated Malware Analysis." (2025).

[37] Bennetot, Adrien, Ivan Donadello, Ayoub El Qadi El Haouari, Mauro Dragoni, Thomas Frossard, Benedikt Wagner, Anna Sarranti et al. "A practical tutorial on explainable AI techniques." ACM Computing Surveys 57, no. 2 (2024): 1-44.

[38] RAHEEM, MOHD ABDUL, and MOHAMMED AZMATH ANSARI. "INTELLIGENT AND TRUSTWORTHY 6G: AI-DRIVEN ARCHITECTURES, APPLICATIONS, AND SECURITY FRAMEWORKS.

[39] Bergman, Anton, and Shamil Limbasiya. "Comparative Study of TGN and TGAT for Dynamic Graph Learning: An Analysis of Performance and Efficiency in Dynamic Graph Learning Models." (2025).

[40] Jean, Guillaume, and Diana Ailyn. "Multi-Modal Data Fusion for Market Prediction Using Explainable Graph Neural Networks (XGNNs): A Fintech Application." (2024).

[41] Kashif, Mohammed, Mohammed Aasimuddin, Mubashir Ali Ahmed, Laxmi Bhavani Cheekatimalla, Eraj Farheen Ansari, and Ahwan Mishra. "AI-DRIVEN CTI FOR BUSINESS: EMERGING THREATS, ATTACK STRATEGIES, AND DEFENSIVE MEASURES."

[42] Roh, Yuji, Kangwook Lee, Steven Whang, and Changho Suh. "Sample selection for fair and robust training." Advances in Neural Information Processing Systems 34 (2021): 815-827.

[43] Gouni, Praveen Kumar Reddy, and Eraj Farheen Ansari. "The Impact of Cyber-Physical Attacks on AI-Enabled Business Systems

[44] Shaik, Addul Faiyaz. "Transforming Back-Office Operations: An Empirical Study of AI-Driven Process Automation in Trade Exception Workflows." Journal of Cognitive Computing and Cybernetic Innovations 1, no. 2 (2025): 21-26