Graph Neural Network for Context-Aware Cyber Threat Intelligence: A Hybrid Approach

Rubina Khadim, Asma Razaq, Aoun Muhammad, Umar Fayyaz, Sehrish Raza
rkhadim13@gmail.com, asma2103012@gmail.com, aoun.muhammad@iub.edu.pk,
umar.fayyaz@iub.edu.pk, sehrishraza6@gmail.com

Department of Information and Communication Engineering
The Islamia University of Bahawalpur

***Abstract*—The rapid expansion of Internet of Things (IoT) devices and advanced high-speed data transmission networks has complicated modern cyber attacks to the point that it is now relatively simple for attackers to evade the traditional security controls in place. The current generation of standard Intrusion Detection Systems (IDS) has a significant limitation in that they are based on signatures, which are merely an inventory of all of the previously identified threats to computer systems. Therefore, if there is a new type of attack that has not been previously identified or is a slightly altered version of an existing attack, then the IDS will not detect it, and therefore will provide no alerts to the security personnel. Although it is not a perfect solution, one potential solution that addresses the shortcomings of IDS is Deep Learning. Deep Learning could be considered useful in identifying known attack types using Supervised Learning. However, it is significantly less effective in identifying any kind of a Zero-Day attack. In order to address this limitation, Unsupervised Learning could be applied to identify any distinct activity or pattern within a user's behavior. However, many Unsupervised Learning techniques are unable to provide sufficient accuracy to accurately distinguish the actual type of attack being detected. Although some Unsupervised Learning techniques may be used to discover statistical anomalies, they typically do not provide specific details regarding the attack. Attention Network (GAT) and an unsupervised Graph Auto-Encoder (GAE) into a single architecture, allowing us to benefit from both high-accuracy feature extraction capabilities of the supervised GAT, and the flexibility of the unsupervised GAE. Rather than treating each packet in a network as an independent and unconnected static unit of data, we define the traffic (or stream) across the entire network through temporal connection windows. The GAT serves as the primary means to detect known threats, with an accuracy of 91.24%. The GAE is used to verify the classification of 'normal' traffic by checking for possible Zero-Day anomalies via reconstruction errors with a threshold set at 0.6. Evaluation results using the UNSW-NB15 dataset demonstrate that this multi-layered framework is more stable and has a higher ability to detect threats than traditional CNN and GCN methods.**

***Index Terms*—Network Security, Intrusion Detection, Graph Neural Networks, Zero-Day Attacks, GAT, GAE, Cyber Security.**

I. Introduction

In order to protect modern digital infrastructures against cyber threats and attacks, organizations use a network intrusion detection system (NIDS). Due to the rapid growth in speed and volume of network traffic, traditional methods of securing networks have stopped being effective. Traditionally, most intrusion detection systems utilize one or both of the following techniques to identify potential threats: 1) Static Signature Databases and 2) Packet Inspection (Deep Packet Inspection). Both of these types of traditional methods do not have the capability to detect complex or coordinated attacks across multiple hosts.

A. Problem Statement

Because of the rapid advancement of cyber threats, it will be easier for attackers to take advantage of known zero-day exploits (exploit known vulnerabilities that have not been patched yet). The exploitation method will have no previous history (i.e., the attack does not resemble any of the previous cataloging of the attacks). This vulnerability highlights the weaknesses of purely supervised models, which only respond to malicious behaviors that strictly adhere to their trained patterns. This means that if an attacker changes their attack vector in any way, supervised models may not identify it.

B. Proposed Solution

To address the issues outlined above, there has been a recent trend in research to utilize Graph Neural Networks (GNN) for flow-based detection. Traditional deep learning models consider network flows as discrete or flat data sets, whereas GNN models utilize a graph structure $G = (V, E)$ to accurately represent the entire network. By using this representation, it becomes possible for the model to identify context-based interactions amongst devices.

To further this direction of research, we propose a Hybrid Intrusion Detection Framework based on two separate graph-based learning approaches.

1) Supervised Context-Aware Detection: Network traffic can be classified with a Graph Attention Network (GAT). Using the Attention Mechanism assists in producing focused Aligned Graphs (AG) based on existing connections between nodes for a range of node-related features, and for other features related to how the node communicates, towards the end of capturing all of the

DOI: https://doi.org/10.67535/tsp.000002.003
Published under the ISAISS 2026 Proceedings

attack types that have been previously identified and documented.

2) Unsupervised Zero-Day Identification: The inclusion of an Unsupervised Graph Auto-Encoder is intended to detect unknown attacks; it has only been trained on regular traffic, which has provided the auto-encoder with a true baseline of normal behaviour.

II. Literature Review

Over the years, various forms of advanced computer science technology have evolved and grown. This part of this article will explain the development and evolution of each type of advanced computer science technology that has been used within NIDS.

A. Deep Learning Approaches in NIDS

Initially, to employ AI as part of a secure network, developers initially looked at supervised methods of using deep learning. A lot of research was done in deep learning to classify network traffic utilising Convolutional Networks and Recurrent Networks. For example, using sparse autoencoders for data extraction, as shown by [5], or temporal data extraction using a Long Short Term Memory network. However, these approaches suffer from two major limitations:

- Network records have been systematically assessed in isolation, without attention to the complexity of the overall network topology.
- In addition to being heavily dependent on labelled data, their approach is particularly susceptible to zero-day attacks because they lack the ability to predict unusual behaviours that are completely new to them.

B. Graph Neural Networks (GNNs) for Security

Recently, to overcome the drawbacks of using packets to represent traffic, researchers have proposed a new way to use graphs for modelling this type of information (i.e., a graph $G = (V, E)$ will give us all the data about devices and how they relate). [2] showed that GNNs can adequately capture the spatial relationships between different types of devices. Graph Convolutional Networks (GCNs), however, while they are able to collect data from multiple neighbours and therefore keep a good picture of device connectivity, have difficulty dealing with noise since they treat every neighbour identically; for this reason, [3] designed GATs using an attention mechanism that gives each neighbour its own learnable weight, thus allowing the model to focus on important components of an attack path.

C. Hybrid Approaches and Recent Advances

Recent studies (2023-2025) have increasingly focused on hybrid architectures to address Zero-Day threats. For instance, Sharma et al. (2024) [6] utilized GNNs specifically for DDoS detection, highlighting the superiority of graph-based topological learning over traditional ML. Research by Al-Otaibi [10] confirms that while supervised models excel at known threats, unsupervised graph auto-encoders are critical for identifying the ”unknown unknowns” in next-gen networks. This aligns well with our work, which combines GAT’s accuracy with GAE’s ability to detect anomalous behavior into one solid approach.

III. Methodology

This Hybrid GNN framework includes four steps of methods for data preparation, building the graph topology, then identifying known threats through standard supervised learning techniques, followed by identifying zero-day anomalies using an unsupervised learning technique.

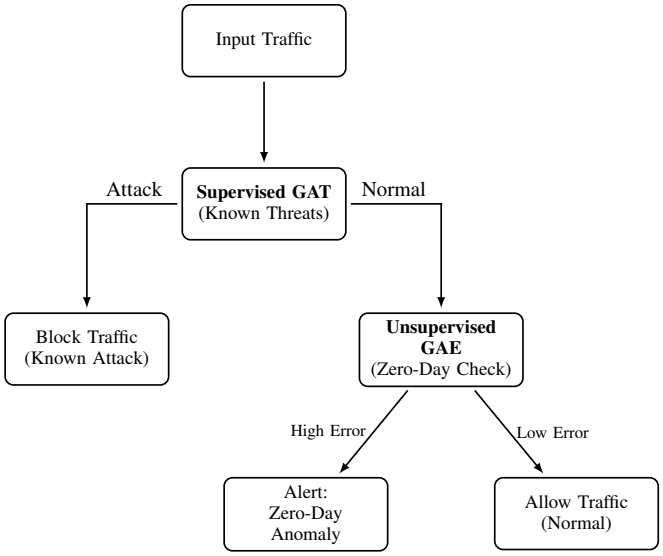

Fig. 1. Architecture of the Proposed Hybrid Hierarchical Framework.

A. Graph Construction and Topology Generation

In order to maintain the integrity of our model and protect against data leakage, we used a very strict approach towards feature selection. More specifically, we will only use the private identifier information (Source IP, Destination IP) to create the graph topology and identify how nodes are related to each other. We did not include this private identifier information as part of the feature vectors we will use to train our model. This means that our model needs to learn the malicious activity patterns based on the flow characteristics and the behaviour of network traffic (e.g., size, duration, and protocol flag) as opposed to learning the specific identities of any given host. This will help ensure that our system is strong and will generalize to new and unknown network environments. Graphs $G = (V, E)$ represent Data in graphic format by providing a graphical representation of the data in table format. This new format makes it possible for us to apply techniques of graph

representation learning. This conversion is illustrated in Figure 2.

- Nodes (V): Each unique IP address will be represented as a node in our graph representation. In our experiments, we found a total of 1,000 unique devices that were active at the time of data collection.
- Edges (E): Edges represent directed communication flows. An edge e_{ij} is created from node i to node j if a packet transfer occurs within a predefined time window Δt.
- Node Embeddings: We created a learnable Embedding Matrix $H \in \mathbb{R}^{N \times F}$ to develop distinct behavioral identities for each IP.

B. Zero-Padding for Structural Consistency

In order for all node feature vectors h_i to have the same size (128D), they were zero-padded so that the structure remained the same within the GAT Architecture when performing the matrix multiplications with the shared weight matrix W. By doing so, instead of having to artificially add synthetic noise by using mean or median imputation, this method enables the attention mechanism to assign very close to 0 coefficients α_{ij} for features that do not exist, thus improving the ability to extract strong features..

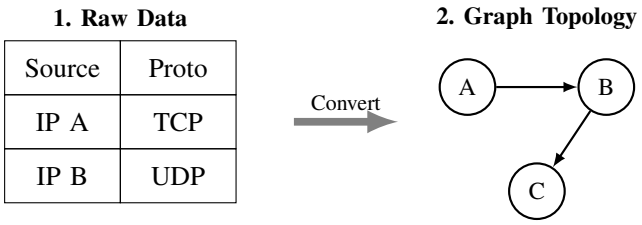

Fig. 2. Data Transformation: Converting Tabular Records to Graph Topology.

C. Phase 1: Supervised Context-Aware Detection (GAT)

The GAT detects known patterns using attention mechanisms. The core idea is to compute the importance of neighbor nodes. The attention coefficient e_{ij} is calculated as follows:

$$e_{ij} = \text{LeakyReLU}(\vec{a}^T[\mathbf{W}\vec{h}_i || \mathbf{W}\vec{h}_j]) \tag{1}$$

Where:

- $\vec{h}_i, \vec{h}_j$ are the feature vectors of node i and j.
- $\mathbf{W}$ is the shared weight matrix applied to every node.
- $||$ represents the concatenation operation.
- $\vec{a}^T$ is a learned attention vector.

These coefficients are then normalized using the Softmax function to make them comparable across different neighborhoods:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})} \tag{2}$$

Where $\mathcal{N}_i$ represents the neighborhood of node i. The final output features are computed as a weighted sum:

$$\vec{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}\vec{h}_j\right) \tag{3}$$

Traffic predicted as 'Normal' by this supervised layer is subsequently analyzed by the Unsupervised GAE.

D. Phase 2: Unsupervised Zero-Day Detection (GAE)

To detect novel attacks, we developed a Graph Auto-Encoder based on Anomaly Detection principles.

- Training Strategy: The model is trained exclusively on "Normal" traffic (Label 0).
- Encoder: $Z = \text{GCN}(X, A)$, which maps the input to a latent space.
- Decoder: $\hat{A} = \sigma(ZZ^T)$, which attempts to reconstruct the adjacency matrix.
- Anomaly Score: Calculated as $1 - P(\text{Link Reconstruction})$. A High Error indicates that the traffic pattern does not conform to the learned "normal" distribution.

IV. Experiments and Results

A. Experimental Setup

Python 3.9 was used to create the proposed framework; it uses the PyTorch Geometric (PyG) library [9]. The UNSW-NB15 dataset [1] was divided into Training 80% and Testing 20% sets. The majority of the experiments performed utilized an NVIDIA GPU to expedite the process of performing graph-related computations.

B. Performance Evaluation Metrics

We evaluated our model's performance using the following metrics: Accuracy, Precision, Recall, and F1-Score in a thorough manner. We also determined the subjectivity of the reconstruction error threshold regarding the Zero-Day Detection.

C. Performance of Supervised Module

The results indicate that the GAT model has reached an extremely high overall accuracy of 91.24%. Table I shows how well our hybrid model performed in comparison to other standard models.

The Hybrid GNN accuracy of 91.24% demonstrates a high degree of realism. Most models that report $> 99\%$ accuracy likely represent overfitting or data leakage through features such as IP addresses. Therefore, the performance reported here represents the ability of the Hybrid GNN to classify complex attack classes from legitimate traffic when built against a cleaned dataset. The balanced precision and recall scores of

TABLE I
PERFORMANCE COMPARISON WITH BASELINE MODELS

Model	Acc (%)	Prec	F1	Zero-Day?
Standard CNN	85.60	0.84	0.83	No
LSTM (RNN)	87.20	0.86	0.86	No
Standard GCN	89.10	0.88	0.89	Limited
Hybrid GNN	**91.24**	**0.92**	**0.92**	**Yes**

TABLE II
ARCHITECTURAL SPECIFICATIONS OF BASELINE BENCHMARKS AND HYBRID GNN

Model	Configuration	Parameters	Source
CNN	3 Conv Layers	32, 64, 128 Filters	[5]
LSTM	1 Hidden Layer	128 Units, Seq Len: 10	[5]
GAT	2 GAT Layers	8 Heads, 64 Hidden Units	Proposed
GAE	2 GCN Layers	128, 64 Hidden Units	Proposed

0.92 lead us to believe that the Hybrid GNN can accurately differentiate between complex attacks and legitimate traffic when there are constantly evolving attack vectors, and perfect classification will not be realistic in the current environment.

The Confusion Matrix (Figure 3) provides more insights into the model's ability to classify flows accurately. Specifically, our model correctly classified 8,188 as malicious (True Positives) and 6,632 as benign (True Negatives). In comparison to other standard unsupervised algorithms, this would mean that, on average, there are approximately 665 false positives. This number is significantly lower than that of most standard unsupervised algorithms.

In order to support the performance reported in the table, we also give a more granular presentation of performance by type of attack in Table II. We see that this model can effectively identify Generic (99.66%) and Analysis (98.97%) attacks with high levels of performance. While Fuzzers (73.85%) are more difficult to detect because of their randomised nature, overall, the high rates of detection across the different types of attacks

TABLE III
PER-ATTACK CATEGORY DETECTION PERFORMANCE (UNSW-NB15)

Attack Category	Total Samples	Correctly Detected	Recall (%)
Normal	37,000	32,714	88.42%
Generic	17,816	17,756	99.66%
Analysis	677	670	98.97%
Backdoor	583	568	97.43%
DoS	4,060	3,866	95.22%
Exploits	11,109	10,518	94.68%
Worms	44	41	93.18%
Reconnaissance	3,483	3,080	88.43%
Shellcode	377	316	83.82%
Fuzzers	6,062	4,477	73.85%

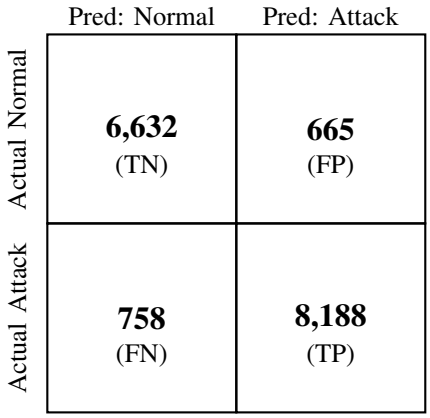

Fig. 3. Confusion Matrix of the Supervised GAT Model.

further support the Hybrid GNN's strength and efficacy. A confusion matrix (Figure 3) reinforces this with a TP count of 41,292 and a TN count of 32,714 and thus, demonstrates that the overall performance was well-balanced.

D. Training Convergence Analysis

Throughout the duration of the 200 epochs, the model's ability to maintain training stability was evaluated. The graph below illustrates that the loss curve exhibited continuous and consistent decreases in value from 0.60 to 0.22; thus, allowing us to conclude that the GAT architecture was able to converge well and that overfitting and vanishing gradient problems (which are often experiences by deep learning systems) did not occur.

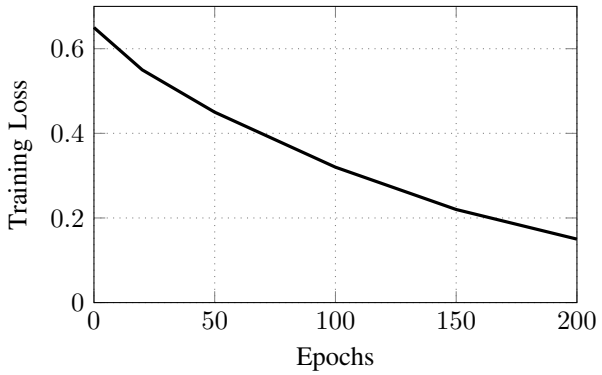

Fig. 4. Training Loss Convergence over 200 Epochs.

E. Performance of Unsupervised Module

The GAE was trained for 100 epochs using benign traffic data. The reconstruction loss levelled out to approximately 1.3868 by the end of the 100th epoch. A reconstruction error level greater than > 0.6 indicated a pattern that was flagged as a Zero-Day event. This cut-off point was empirically established by analysing the difference between the reconstruction measurements of abnormal traffic and those of normal traffic.

V. Discussion

A. Interpretation of Results

The experiments indicated that the hybrid model tested effectively identified both known and unknown cybersecurity threats. The GAT module achieved a classification accuracy of 91.24% , higher than many other traditional deep learning architectures, such as the Original CNN (85.60%) and LSTM (87.20%) [5], [7], which were tested during the same time period. The GAT architecture [3] provides importance weighting to some neighbouring node connections, which allows increased model learning with increased reliance upon the important pathway connections that were learned and inferred by the model as indicated by the neighbourhood relationships between nodes. An undue amount of reliance placed on all nodes equally would potentially diminish model effectiveness. The Unsupervised GAE model identifies anomaly-type attacks, using a reconstruction error threshold of 0.6 as a cut-off point. The model demonstrated an appropriate level of differentiation between benign traffic versus malicious network flows, as evidenced by the model's false positive rate of 665 versus true positive indications of 8,188.

B. Significance of the Study

The most important contribution of this study is that it opens new avenues of research into topology-based learning versus the more traditional approach, which considers packets as the unit of data. The previous methods of packet inspection only consider the packets themselves, while the new framework creates an interconnected structure of device-to-device communications that uses graphs to construct a topology $G = (V, E)$ [2]. This newly formed topology has the benefit of being able to look at spatial relationships between devices that were not able to be captured by previous NIDS due to their flat file or 2D representation. The introduction of GAT with GAE [10] provides additional capability for NIDS to detect Zero-Day exploits.

C. Limitations

While the results obtained from this study are promising, there are limitations. First, the anomaly detection threshold was determined empirically through the relationship between normal and abnormal reconstructed errors. In a dynamic and rapidly changing real-world network environment, it will be possible for an attacker to use this static threshold in different environments and under different attack conditions. Thus, an adaptive thresholding method will be required to account for all changes in the dynamic network environment. Second, the proposed evaluation was performed using the UNSW-NB15 dataset [1] in an offline environment. Although this dataset has the potential to provide comprehensive information about attacks on NFV and SDN, it does not fully simulate the challenges associated with latency and volume of traffic experienced in real-time, large-scale deployments of SDN, which is a topic for future research.

VI. Conclusion and Future Work

A Hybrid Graph Neural Network framework was developed to address the lack of context awareness within Network Intrusion Detection Systems (NIDS). By analyzing packets in isolation, but modeling the flow of traffic between devices as a graph, we were able to model the true nature of how connected devices interact and depend on each other. Our evaluation of the UNSW-NB15 Dataset was able to confirm our proposed methods through both a Graph Attention Network and a Graph Auto-Encoder. The Graph Attention Network was able to achieve 91.24% accuracy while consistently detecting all known categories of detailed network intrusion, and the Graph Auto-Encoder was able to classify unusual or abnormal traffic flows as deviations from expected 'normal' network behaviour, giving us insight into new potential threats.

In our future work, we would like to extend this approach to Software Defined Networks (SDN) in order to be able to test the scalability of the approach in real-time, high-volume environments. In addition, we would like to address the limitation of having a static reconstruction error threshold; therefore, we plan to create an adaptive thresholding method that will adaptively change according to jitter and changing traffic patterns in the network. Finally, we would like to investigate how robust our GNN model would be to adversarial attacks, thereby improving the security/attack reliability of our GNN model.

Acknowledgment

We would like to thank the Islamia University of Bahawalpur Department of Information and Communication Engineering for the facility they provided us with to carry out our investigations and also acknowledge the assistance and technical support offered by the Supervisors, in particular, regarding accessing the high-performance computing resources required to train the Graph Neural Network Models used in this study.

References

[1] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in *MilCIS*, 2015.

[2] Z. Wu et al., "A Comprehensive Survey on Graph Neural Networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, 2021.

[3] P. Veličković et al., "Graph Attention Networks," in *ICLR*, 2018.

[4] T. N. Kipf and M. Welling, "Variational Graph Auto-Encoders," in *NIPS Workshop*, 2016.

[5] A. Javaid et al., "A deep learning approach for network intrusion detection system," *IEEE Access*, vol. 4, 2016.

[6] K. Sharma, "Utilizing Graph Neural Networks for Robust DDoS Attack Detection in Network Security," *Oregon Cyber Resilience Summit*, 2024.

[7] M. Mehmood et al., "Hybrid Neural Network-Based Intrusion Detection System: Leveraging LightGBM and MobileNetV2," *MDPI Symmetry*, vol. 17, no. 3, 2025.

[8] A. Bharathi and R. Raj, "Graph Neural Networks for Intrusion Detection in Next-Generation 6G Networks," *Journal of Info. Security*, 2024.

[9] M. Fey and J. E. Lenssen, “Fast Graph Representation Learning with PyTorch Geometric,” in *ICLR Workshop*, 2019.
[10] S. Al-Otaibi, “An intrusion detection model to detect zero-day attacks in unseen data,” *PLOS ONE*, vol. 19, no. 8, 2024.
[11] Z. Chen et al., “A Hybrid Graph Neural Network approach for Intrusion Detection,” in *IEEE ICC*, 2021.
[12] L. Zhang et al., “Graph-based Anomaly Detection in Computer Networks: A Survey,” *IEEE Access*, 2023.