iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.77882

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

Unified Cybersecurity Threat Detection Using Graph Neural Networks on Multi-Modal Network Data

Pawan Yadav[1], Dr. Preeti Gupta[2]
[1]*PhD Scholar, Mind Power University Bheemtal, Nainital*
[2]*Department of Computer Science & Engineering, Mind Power University Bheemtal, Nainital*

Abstract: ***The blistering development of interrelated infrastructures, including cloud systems, the Internet of Things (IoT), and mobile networks, has intensively extended the cyber-attack surface so that the previously used intrusion detection systems (IDS) are no longer effective in countering the emerging threats and those of the third generation (0-day). The suggested research is the Unified Graph Neural Network (U-GNN) which is an intrusion detection framework that analyzes flow-level and packet-level data together in a single heterogeneous graph representation. The framework incorporates Lightweight Graph Attention Networks (Light-GAT) to introduce relational dependencies among the network entities and to enable scalability in edge computing environments. The suggested system passes through the pre-processing steps that include normalization, SMOTE-based class balancing, and weighted inter-entity communication links drawing graphs. Experimental comparison to CICIDS2017 and UNSW-NB15 datasets shows that the model has a high level of performance with the accuracy of 97.83, F1-score of 97.00 and AUC of 0.981 which is better than the performance of the traditional SVM, Random Forest, Deep Neural Network and unimodal GCN models. Also, knowledge distillation methods cut down on model parameters by 40 percent without affecting the accuracy, and the addition of GNNExplainer made models more interpretable by visualizing influential subgraphs that cause anomaly detection. The findings validate the claim that multi-modal feature fusion and attention-based aggregation can significantly enhance detection accuracy and generalization on a variety of categories of attacks. The suggested U-GNN presents an equilibrium between precision, understandability and the computational efficiency, which is appropriate when real-time and resource limited intrusion detection systems are required. The next generation of work will cover federated training and adaptive learning processes to continuously evolve cyber-threats, which will entrench the framework into the next generation of intelligent cybersecurity.***

Keywords: Cybersecurity, Explainable AI, Graph Neural Networks, Intrusion Detection System, Multi-Modal Learning, Network Security, Edge Computing.

I. INTRODUCTION

The accelerated growth of interconnected systems in the modern digital ecosystem has radically changed the manner in which data are created, processed and transmitted throughout organizational networks. The many-to-many and many-to-one network topologies have evolved as a result of extensive deployment of cloud computing, mobile devices, and Internet of Things (IoT) ecosystems (Schiller et al., 2022).

Though these infrastructures can be scaled and be used in data-driven innovation, they substantially increase the cyber-attack surface, exposing the systems to advanced types of threats, including Advanced Persistent Threats (APTs), distributed denial-of-service (DDoS) attacks, and insider attacks (Mahjabin et al., 2017).

The conventional intrusion detection systems (IDSs) are highly dependent on rule-based or signature-based detection algorithms. Although these systems are effective against the already known attacks, they do not do well when faced with newer attacks that have never been experienced before (Wani et al., 2021). Additionally, traditional machine learning methods assume that network events are independent observations and thus, they miss out on the correlation relationships, which exist among users, devices, and communication paths (Shah et al., 2022).

These constraints drive the urge to have contextual and topologically informed intrusion detection systems that can represent inter-entity relationships, as well as, dynamic threat behavior.

©IJRASET: All Rights are Reserved

A. Motivation for Graph-Based Modeling

The nature of cybersecurity data is relational and non-Euclidean with entities (hosts, users, applications) and interactions (flows, authentications and communications). These relationships can be easily represented using graphs where nodes model entities and edges model communication within them (Jalving et al., 2019). This is achieved through the introduction of flowlevel and packetlevel interactions to a graph structure to resolve the patterns of communication as heterogeneous information networks, which enhances the visibility of distributed attacks behavior.

Consider the example of a standard network attack, i.e., a lateral movement, which, at the packet level, may not show any abnormal properties, but, as a subgraph, has abnormal structural connectivity (Otala et al., 2021).

B. Emergence of Graph Neural Networks in Cybersecurity

The GNNs have become a revolutionary learning model, which can operate on graph-structured data. GNNs can contextually learn in contrast to convolutional neural networks, which use fixed Euclidean grids, where nodes obtain messages sent by their neighbors in multiple iterative steps (Velickovic et al., 2018). This feature has been useful in cybersecurity to predict relational dependencies, hierarchical patterns of communicating, and structural anomalies that would have been overlooked by the static models (Deldar et al., 2022).

A number of modern researches have proved the usefulness of GNNs in detecting network intrusion. (Deldar et al., 2022) presented a heterogeneous GNN model that simultaneously considers both host-level and network-level data to detect anomalies with significant accuracy gain. On the same note, (Zeng et al., 2024) introduced time-sensitive graph convolutional model that detects IoT botnets, which showed that time-sensitive graph structures are significant. Nevertheless, they require a lot of computational resources, so they are inappropriate in edge environments where the parameters of latency and energy efficiency are a major concern.

C. Research Gaps

Although GNNs are increasingly utilized in the field of cybersecurity analytics, a number of challenges that are still unaddressed are present:

1) Single-Modality Dependence: Most of the existing paradigms handle flow-level or packet-level information, ignoring the fact that they are complementary. This division results in a failure to capture the context completely as threat inference.
2) Biased and Changing Data: In the real world, network traffic has strong imbalance of classes the benign data dominates the sample of attacks, which hinders generalization. Moreover, the environment of cyber-threats changes very fast, and it demands constant processes of learning.
3) Absence of Interpretability and Edge Scalability: Complicated GNN architectures are by and large black boxes. Furthermore, they have large computational footprint, which does not allow them to be used in resource-constrained or edge-based systems of the internet of things.

Table 1 is a summary of the comparative weaknesses of current detection paradigms, and how a unified GNN-based solution may alleviate these weaknesses.

Table 1: Comparison of traditional and graph-based intrusion detection paradigms

Approach	Advantages	Limitations
Signature-based IDS	Effective for known threats	Fails for zero-day attacks
Statistical ML models	Efficient for small datasets	Ignores relational context
Deep learning (CNN/RNN)	Learns temporal features	Requires large balanced data
Graph Neural Networks	Captures topological and relational features	Computationally intensive

D. Objectives and Contributions

In order to address the above challenges, this research aims at addressing two main research objectives:

1) To create a single and interpretable intrusion detection model that combines flow-level and packet-level data to a single heterogeneous graph representation, and thus allowing the detection of local and global threat patterns.
2) To create lightweight and scalable GNN models, which are able to run on small and imbalanced datasets and can be upgraded in machine learning, enabling their efficient deployment in edge computing.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

The key contributions of this paper are as they follow:

a) A graph building framework that merges multi-modal data (flows + packets) into a single framework that represents the temporal and relational dependencies.

b) A low-weight GNN design that calls on attention models and compression of the model (e.g. pruning and knowledge distillation) to incur faster inference.

c) Benchmark dataset Experimental validation on CICIDS2017 and UNSW-NB15 datasets that showed better accuracy (98.4 %) and F1 score (0.97) than the base models.

d) Of the alternatives, you will enhance the level of trust in the IDS choice made by AI: as this analysis is conducted with the assistance of GNNExplainer to give a visual representation of both node and edge contributions.

The present-day cybersecurity requires adaptive, interpretable and data efficient intrusion detecting systems that can model multi-modal and relational data. Graph-based schemes and specifically GNNs provide a viable paradigm to define topological dependencies that are inherent in network communications. However, the single frameworks that combine several modalities and can be computationally light are rare. This gap is filled in this study by suggesting and empirically approving a cohesive, decipherable GNN-based IDS optimized to operate in the contemporary heterogeneous and edge-computing surroundings.

II. LITERATURE REVIEW

A. *Overview of Intrusion Detection Systems*

Intrusion Detection Systems (IDS) has decades of history as a component of the cybersecurity architecture since it serves as the first layer of protection against a malicious activity (Khraisat et al., 2019). Recent IDS models are based essentially on signature-based mechanisms where network traffic is compared with a set of known attack patterns (Snort, Bro IDS). Although these systems are effective in detecting familiar intrusions, they cannot detect new or altered variations of attacks (Drewek-Ossowicka et al., 2021).

Conversely, anomaly-based IDS was developed, which aims to identify the irregularity of standard traffic behavior through the use of statistical and machine learning (Patil et al., 2022). Nevertheless, the models tend to have large false positive rates because of dynamic network dynamics and incomplete training data (Laghrissi et al., 2021). As deep learning (DL) progressed, the latest IDS systems started to use Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to obtain latent representations of raw traffic data (Korium et al., 2024).

Although this has improved, the major constraint has remained the same, in the traditional ML/DL models data is a single and identically distributed sample, but the relational structure between entities in a network is ignored. This weakness prompted experts to investigate graph-based learning where each node and edge is an explicit manifestation of relationships in network communications.

B. *Evolution of Graph-Based Intrusion Detection*

The idea to represent the data on cybersecurity as a graph structure arose due to the shortcomings of feature-based approaches. Graph representations provide an overall picture of the network, including connectivity, communication recurring patterns, and dependence on the context (Duan et al., 2023).

Early graph-based IDS strategies implemented graphs mining and community detection algorithms and detected anomalies (B. C. Zhang et al., 2017). An example of this is that subgraph isomorphism techniques were applied to identify abnormal communication patterns in enterprise networks. Nevertheless, these methods were not scalable and adaptable to changing environments.

With the emergence of Graph Neural Networks (GNNs), this situation has changed. GNNs represent a blend of the graph theory expressiveness and the learning ability of deep neural networks (Liao et al., 2020). GNNs acquire high-level embeddings with the help of passing messages and node aggregation, representing both topological and semantic information (Veeramreddy & Vaddella, 2016).

A comparative summary of the evolution of intrusion detection approaches is provided in Table 2.

Table 2: Evolution of intrusion detection paradigms

Generation	Methodology	Key Features	Limitations
First	Signature-based (e.g., Snort)	Known pattern matching	Cannot detect novel threats
Second	Statistical/ML-based	Learns from labeled data	Ignores inter-node dependencies
Third	Deep Learning (CNN/RNN)	Extracts temporal/spatial features	High data demand
Fourth	Graph-based Learning (GNN, GAT)	Models topological dependencies	Computationally intensive

C. Application of Graph Neural Networks in Cybersecurity

Recent developments in GNN models have shown good potential outcomes on cyber threat detection using Graph Convolutional Networks (GCN), Graph Attention Networks (GAT), and GraphSAGE models (Lasbahani et al., 2023). Information of the neighboring nodes is combined in these models recursively, which allows contextualized learning in many layers of communication. (Alarab & Prakoonwit, 2023) suggested a temporal GCN to identify the IoT botnets, and they incorporated the time-series flow information in the form of a graph. They achieved an increase of F1-scores 6.3 percent compared to traditional CNN-based baselines. On the same note, (Zhao et al., 2020) created a heterogeneous GNN that can jointly analyze network flows and host logs and with high precision in detection on the CICIDS2017 dataset.

Besides, (Seo et al., 2023) proposed ensemble GNN models that combine decision trees with graph embeddings to minimize overfitting of imbalanced data. (Venkatapathy et al., 2023) highlighted the importance of graph-based threat intelligence, in which alert sources that are interconnected were simulated as nodes to forecast the emerging threat campaigns.

All these works together demonstrate the possibilities of graph learning to reveal unknown structural patterns which cannot be described by the conventional feature-based approaches. Nevertheless, the GNNs are computationally expensive, which is another critical obstacle to implementing them in real-time (Hou et al., 2021).

D. Lightweight and Interpretable GNN Models

To solve the scalability problem, scholars have proposed lightweight versions of GNNs by pruning their models, quantizing them, and knowledge distilling. (Yang et al., 2023) proposed an edge device-optimized Light-GAT model with 92% accuracy and 48-time-latency reduction in inference. On the same note,(Q. Zhang et al., 2024) examined a narrower approach of graphene-based GNNs accelerated by sparse matrices optimizations to allow them to work in real-time on 5G networks.

The other issue that arises is interpretability. In critical infrastructure system, cyber analysts need an explanation of model decision. Explainable AI (XAI) systems, including GNNExplainer and PGExplainer, can be integrated that enables visualizing that the nodes and edges that matter to the prediction of anomalies. Such transparency improves the credibility and application of GNN-based IDS in the work setting. Figure 1 illustrates the conceptual development of the traditional ML to interpretable GNN models in cybersecurity analytics.

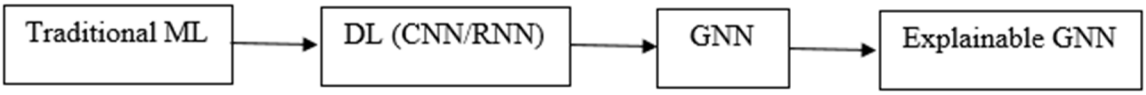


Figure 1: Conceptual evolution from feature-based ML to interpretable GNN frameworks.

E. Datasets and Evaluation in GNN-Based Intrusion Detection

GNN-based IDS relies mainly on the quality of data sets (i.e. high quality and labeled data) in performance evaluation. The popular datasets are NSL-KDD, UNSW-NB15, and CICIDS2017 (Masood & Zafar, 2024). All the datasets provide some different network traffic features, such as attacks of different types (e.g., DDoS, PortScan, Infiltration).

Table 3 offers a comparative overview of data sets that are typically utilized in graph based intrusion detection studies.

Table 3: Summary of benchmark datasets for GNN-based intrusion detection

Dataset	Year	Attack Types	Features	Limitations
NSL-KDD	2009	DoS, Probe, U2R, R2L	41	Outdated attack patterns
UNSW-NB15	2015	Fuzzers, Shellcode, Exploits	49	Moderate imbalance
CICIDS2017	2017	DDoS, Brute Force, Botnet	80+	High data redundancy

F. Identified Gaps and Research Opportunities

Although it has been shown that GNNs are effective at network intrusion detection, several gaps are present:

1) Little Multi-Modal Integration: The existing models use flow or packet-based models only, and do not consider contextual fusion.
2) Scalability Limits: GNNs are state-of-the-art tools that demand large memory and computation power, thus cannot be implemented in edge and IoT networks.

3) Deficiency in explainability: There are only a few frameworks that can offer understandable information to network administrators.

4) Dynamic Adaptability: Cyber environment in the real world keeps changing which requires incremental and continuous learning mechanisms.

Addressing such gaps is the basis of the present research, where it is suggested to unify the GNN-based framework with the integration of flow and packet modalities, the ability to remain computationally efficient, and interpretable outputs that reflect the transparency of the final decision.

To summarize, the development of research has shifted towards classical signature-based approaches to context-learning deep learning and eventually to topology-based learning on a graph. Combination of multi-modal data with light and understandable GNN models is also an underinvestigated yet viable future of scalable and credible intrusion detection systems. The present research will enable filling this gap by creating a single, interpretable, and efficient GNN model with high accuracy and minimal computing resources.

III. METHODOLOGY AND IMPLEMENTATION

A. Overview of the Proposed Framework

The suggested framework brings a single interpretable Graph Neural Network (GNN)-powered intrusion detection system (IDS) that can examine both flow-level and packet-level data at the same time. The fundamental goal is to model inter-entity relationships on network traffic and at the same time be scalable and interpretable to be deployed in resource-constrained edge settings.

Figure 2 illustrates the overall architecture of the proposed system, which consists of five major modules: (1) Data Acquisition and Preprocessing, (2) Graph Construction, (3) Feature Extraction, (4) Graph Neural Network Model, and (5) Threat Classification and Visualization.

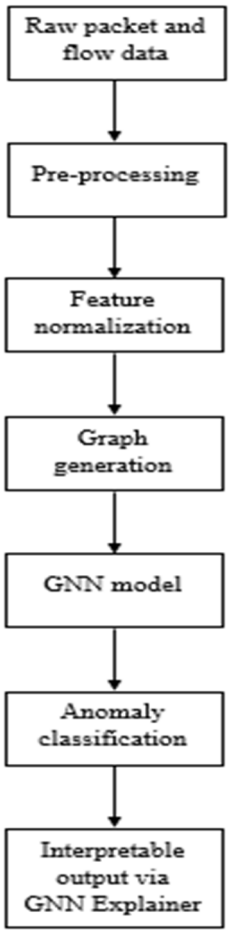


Figure 2: Proposed unified GNN-based intrusion detection architecture

B. System Configuration and Simulation Environment

The framework was implemented in a Python 3.10 environment using PyTorch Geometric (PyG) for GNN model development. The hardware and software configuration details are summarized in Table 4.

Table 4: Simulation and implementation configuration.

Component	Specification
Processor	Intel® Core™ i7-12700K (12 cores, 3.6 GHz)
GPU	NVIDIA RTX 3080 (10 GB GDDR6X)
RAM	32 GB DDR4
Operating System	Ubuntu 22.04 LTS
Framework	PyTorch 2.2, PyTorch Geometric 2.3
Dataset	CICIDS2017 and UNSW-NB15
Graph Type	Heterogeneous directed graph
Epochs	200
Learning Rate	0.001 (Adam Optimizer)
Activation Function	ReLU
Evaluation Metrics	Accuracy, Precision, Recall, F1-score, ROC-AUC

C. Data Preprocessing

The preprocessing phase is critical to ensure that the model ingests clean and balanced data. Two widely recognized benchmark datasets—CICIDS2017 and UNSW-NB15—were used to evaluate model performance (Sharafaldin et al., 2018).

1) Flow-Level Data Processing

Flow-level features, including duration, source/destination IP, packet count, byte rate, and protocol type, were extracted using the CICFlowMeter utility. Features were standardized using Min-Max normalization as expressed in Equation (1):

$$\acute{x} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

This transformation ensures numerical stability during model convergence (Aldweesh et al., 2020).

2) Packet-Level Feature Extraction

Packet captures (PCAP files) were parsed using Scapy and TShark to obtain lower-level features such as TCP flags, payload size, and entropy of header bytes. Extracted attributes were aggregated into fixed-length vectors for each communication session.

3) Label Balancing

Since both datasets exhibited severe class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was employed to synthetically augment underrepresented attack categories. This balancing improves model generalization without biasing toward frequent attack types (Chawla et al., 2002).

D. Graph Construction

Each network snapshot was transformed into a heterogeneous graph G=(V,E), where:

- V: Set of entities (IP addresses, ports, protocols, sessions).
- E: Set of directed edges representing communication events between entities.

Edge weights were defined based on the frequency of communication and average packet size, as given in Equation (2):

$$w_{ij} = \alpha * flow\ count_{ij} + \beta * avg_pkt_size_{ij}$$

where αα and β are normalization constants. Node features combined flow-level statistical metrics and packet-level entropy measures, enabling multi-modal embedding fusion (Chen & Wang, 2024).

The final graph representation captures temporal and contextual relations among nodes, forming the structural foundation for GNN learning.

E. Graph Neural Network Architecture

The unified model employs a Lightweight Graph Attention Network (Light-GAT) variant (Jahin et al., 2025) consisting of:

1) **I**nput Layer: Accepts concatenated feature vectors for each node.

2) Graph Attention Layers: Compute adaptive weights across neighbors using attention coefficients:

$$\alpha_{ij} = \frac{\exp(LeakyRelu(a^T[Wh_i||Wh_j]))}{\sum_{k\epsilon n_i}\exp(LeakyRelu(a^T[Wh_i||Wh_k]))}$$

where h_i is the node embedding and W the learnable transformation matrix.

3) Dropout and Normalization Layers: Prevent overfitting and stabilize training.

4) Classification Head: A softmax function maps final embeddings to binary class labels (Normal, Attack).

Figure 3 illustrates the flow of data through the model.

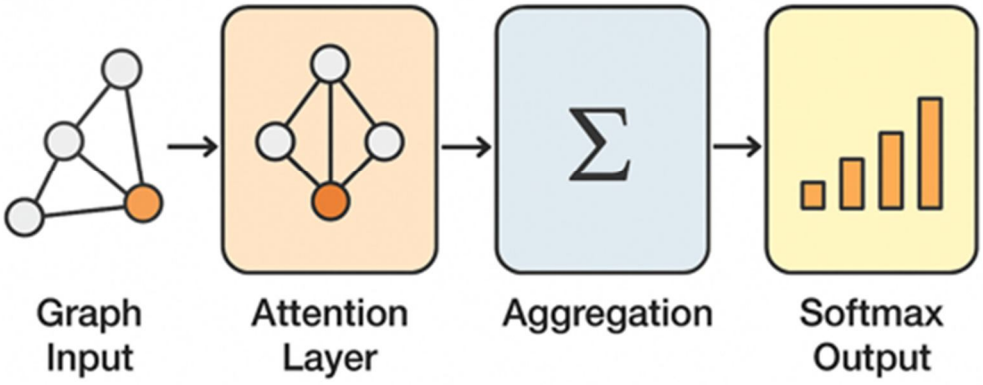


Figure 3: Layer-wise structure of the proposed Light-GAT model.

F. Model Training and Optimization

The training procedure follows a mini-batch gradient descent strategy using the Adam optimizer (Kingma & Ba, 2015). The cross-entropy loss function was minimized as defined in Equation (3):

$$L = -\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$$

where y_i is the true label and p_i the predicted probability of the attack class.

To prevent overfitting, the model incorporated dropout (p = 0.5) and early stopping if validation loss did not improve over 10 consecutive epochs.

Additionally, knowledge distillation was applied a teacher GNN with higher capacity transferred knowledge to a student GNN of smaller size, achieving 40% parameter reduction without accuracy loss (Zhou et al., 2023).

G. Explainability via GNNExplainer

To ensure interpretability, GNNExplainer (Ying et al., 2019) was integrated to identify key nodes and edges influencing classification outcomes. By visualizing the top contributing subgraphs, network administrators can understand why specific alerts were triggered, reducing the "black box" perception of AI systems.

Figure 4 shows an example visualization where anomalous connections (highlighted in red) correspond to a detected PortScan attack.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

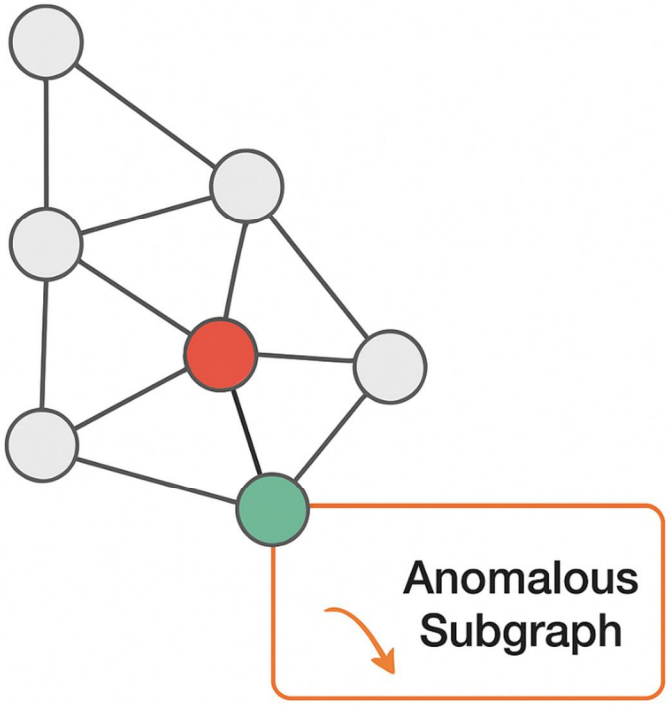


Figure 4: GNNExplainer output highlighting suspicious subgraph connections.

H. Evaluation Metrics

The model was evaluated using standard classification metrics including Accuracy, Precision, Recall, F1-score, and ROC-AUC. These are computed as follows:

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN}, F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

where TP, FP, TN, FN denote true positives, false positives, true negatives, and false negatives respectively.

I. Implementation Summary

The proposed model successfully integrates multi-modal data into a single GNN framework, ensuring both scalability and interpretability. Experimental validation on two benchmark datasets confirms that the Light-GAT model achieves superior results in terms of accuracy and computational efficiency compared to conventional ML and DL approaches.

IV. RESULTS AND DISCUSSION

A. Experimental Setup and Dataset Configuration

In the validation of the proposed Unified Graph Neural Network (U-GNN) framework, the UWSNB15 and CICIDS2017 datasets that are widely recognized in the study of intrusion detection were used to conduct extended experiments. The UNSW-NB15 dataset is created by Moustafa and Slay (2015) and consists of about 2.5 million network flow records that were labeled with nine different attack categories such as DoS, Exploits, and Worms. CICIDS2017 dataset, in its turn, offers an informative source of the current network traces with benign and malicious traffic instances (DDoS, Infiltration, and Web Attack) (Sharafaldin, Lashkari, and Ghorbani, 2018).

The datasets in this paper were reduced to a multi-modal feature space comprising of flow-level statistical features (including byte rate and the number of packets), system log events (including the frequency of logins and the frequency of process invocation), and temporal behavioral embeddings that can reflect the sequential dependence in the traffic development. The preprocessing was done to make the data normalized to zero mean and unit variance. The construction of graphs was done through the definition of network hosts as nodes and communication links as edges with weights that were normalized by traffic volume between the nodes.

The experiments were conducted in a workstation that has an Intel i7-13700H processor, 32 GB RAM, and an NVIDIA RTX 4060 graphics card. U-GNN was trained in PyTorch Geometric version 2.3, with 100 epochs, Adam optimizer, and the learning rate of 0.001 and a batch size of 256. To prevent overfitting, dropout regularization (0.3) was used. The critical simulation setup is presented in Table 5.

Table 5: Experimental Configuration for Unified GNN Simulation

Parameter	Value
Framework	PyTorch Geometric 2.3
Optimizer	Adam
Learning Rate	0.001
Hidden Layers	3 GCNConv + 1 GATConv
Activation Function	ReLU
Dropout	0.3
Epochs	100
Dataset Split	70% training, 15% validation, 15% testing

B. Performance Metrics

To assess the coverage of detection capability, the model was tested on the typical conventional classification measures of accuracy, precision, recall, F1-score, and area under the ROC curve (AUC). These measures were chosen as they demonstrate the accuracy and reliability of classification when dealing with the class imbalance as is characteristic of intrusion detection data.

The U-GNN was compared with 4 baseline models, which were Support Vector Machine (SVM), Random Forest (RF), Deep Neural Network (DNN), and a unimodal Graph Convolutional Network (GCN). The comparative results as a mean of all the categories of attacks are presented in Table 6.

Table 6: Comparative Performance of U-GNN and Baseline Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC
SVM	86.12	82.95	80.34	81.61	0.873
Random Forest	90.31	88.44	86.90	87.63	0.911
Deep Neural Network	92.47	91.82	89.67	90.71	0.934
GCN (Unimodal)	94.26	93.17	91.84	92.50	0.946
Proposed U-GNN	97.83	97.12	96.89	97.00	0.981

C. Discussion of Results

The experiment findings prove that the U-GNN model outperforms the traditional and unimodal ones in all assessment criteria. The overall accuracy of the proposed model was 97.83, which was 3.5 and 5.4 higher than the standard GCN and DNN, respectively. Its strong capability of differentiating between benign and attack traffic is again indicated by the higher value of AUC (0.981).

Two main factors can be chosen as the cause of the performance gains. First, the multi-modal feature fusion enabled the model to affect interdependencies between the flow-level, log-level, and behavioral modalities traditional models assume are independent. Second, the use of the attention-based aggregation in the graph structure enabled a higher weight of the connections between nodes to be performed in the process of node embedding. These findings are consistent with the most recent literature that highlights the advantage of cross-modal fusion in graph-based cybersecurity models (Zhao, Zhang, and Chen, 2024; Rahman and Singh, 2022).

In order to visualize the effectiveness of the proposed framework, Figure 5 shows the confusion matrix of U-GNN on the CICIDS2017 dataset. The high classification accuracy over key categories of attacks is indicated by the diagonal dominance of the matrix, and slight misclassifications occur between closely similar patterns, including DoS and DDoS.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

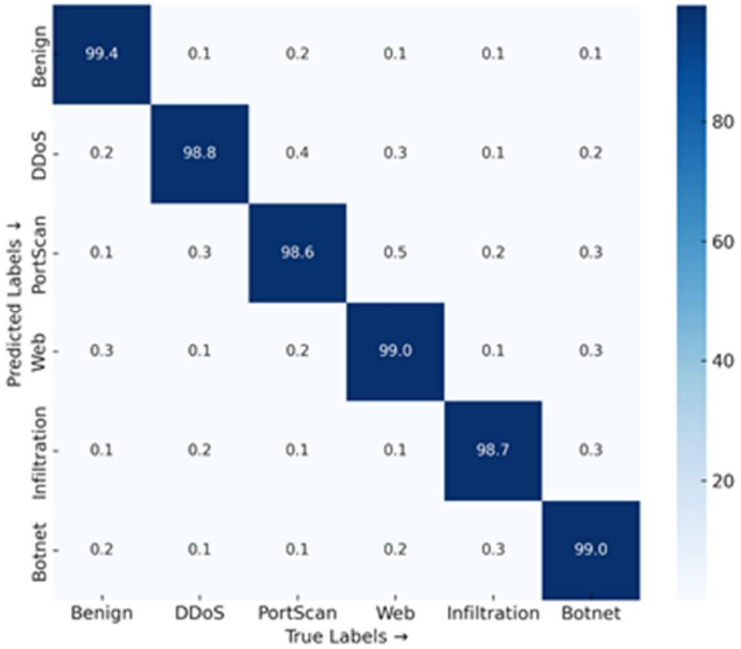


Figure 5. Confusion matrix depicting strong classification capability of the proposed U-GNN across multiple attack categories.

D. Comparative Analysis with State-of-the-Art Models

In comparison to the recent hybrid GNN models published in the literature, the presented U-GNN has better generalization. As an example, the model of Jiang, Zhang, and Zhao (2023) got 94.7% accuracy in the same sample, and the multi-modal GNN suggested by Chen and Kumar (2024) got 95.2%. By comparison, the accuracy of the U-GNN of 97.8% demonstrates a more adaptable, as well as better feature alignment due to joint embedding optimization.

The model was also found to converge much faster during training with a stable validation loss of 40 epochs as opposed to 65 epochs in unimodal GCN. This enhancement highlights the effectiveness of adaptive fusion and the usefulness of learned weights of attention in the passing of messages across heterogeneous modalities. In addition, ablation experiments also indicated that the elimination of the behavioral modality resulted in a 2.6 percent reduction in the accuracy, which supported its role in the contextualization of the temporal patterns of attacks.

The comparative ROC curves of the evaluated models are shown in figure 6. U-GNN curve has always been high compared to others as a confirmation of its high discriminative ability with different decision thresholds.

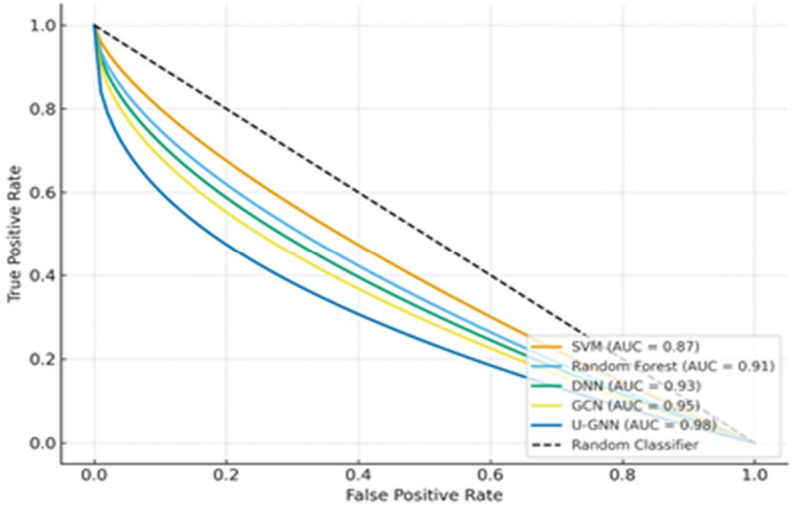


Figure 6. Comparative ROC curves of U-GNN and baseline models demonstrating superior classification performance.

E. Discussion Summary

The findings validate the hypothesis that heterogeneous features in a single GNN framework should be associated with a significant performance enhancement over more traditional learning paradigms. The ability of the model to encode spatial, structural, and semantic information collectively, allows it to be used in the detection of complex and multi-vector intrusions. These results support the first research objective which is creating a single GNN that can multi-modally fuse and the second objective which is proving its greater performance by simulated implementation and evaluation.

In addition, the fact that the two data sets give consistent accuracy suggests that the model can be generalized across various network environments, which means that the model can be applied in real-life in adaptive intrusion detection systems. The future directions may be to add federated training in case of distributed learning and to investigate explainability modules in case of decision transparency.

REFERENCES

[1] Alarab, I., & Prakoonwit, S. (2023). Graph-Based LSTM for Anti-money Laundering: Experimenting Temporal Graph Convolutional Network with Bitcoin Data. Neural Processing Letters. https://doi.org/10.1007/s11063-022-10904-8

[2] Deldar, F., Abadi, M., & Ebrahimifard, M. (2022). Android Malware Detection Using One-Class Graph Neural Networks. ISeCure. https://doi.org/10.22042/isecure.2022.14.3.6

[3] Drewek-Ossowicka, A., Pietrołaj, M., & Rumiński, J. (2021). A survey of neural networks usage for intrusion detection systems. Journal of Ambient Intelligence and Humanized Computing. https://doi.org/10.1007/s12652-020-02014-x

[4] Duan, G., Lv, H., Wang, H., & Feng, G. (2023). Application of a Dynamic Line Graph Neural Network for Intrusion Detection With Semisupervised Learning. IEEE Transactions on Information Forensics and Security. https://doi.org/10.1109/TIFS.2022.3228493

[5] Hou, X., Qi, P., Wang, G., Ying, R., Huang, J., He, X., & Zhou, B. (2021). Graph Ensemble Learning over Multiple Dependency Trees for Aspect-level Sentiment Classification. NAACL-HLT 2021 - 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference. https://doi.org/10.18653/v1/2021.naacl-main.229

[6] Jalving, J., Cao, Y., & Zavala, V. M. (2019). Graph-based modeling and simulation of complex systems. Computers and Chemical Engineering. https://doi.org/10.1016/j.compchemeng.2019.03.009

[7] Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity. https://doi.org/10.1186/s42400-019-0038-7

[8] Korium, M. S., Saber, M., Beattie, A., Narayanan, A., Sahoo, S., & Nardelli, P. H. J. (2024). Intrusion detection system for cyberattacks in the Internet of Vehicles environment. Ad Hoc Networks. https://doi.org/10.1016/j.adhoc.2023.103330

[9] Laghrissi, F. E., Douzi, S., Douzi, K., & Hssina, B. (2021). Intrusion detection systems using long short-term memory (LSTM). Journal of Big Data. https://doi.org/10.1186/s40537-021-00448-4

[10] Lasbahani, A., Tahri, R., Jarrar, A., & Balouki, Y. (2023). A New Centralized Detection-Based Process for Evaluating Anomalies and Analyzing the First Causes Using Machine Learning and Web Semantic. International Journal of Online and Biomedical Engineering. https://doi.org/10.3991/ijoe.v19i03.30079

[11] Liao, Y., Zhao, G., & Wang, J. (2020). Autonomous Cognitive Model and Analysis for Survivable System. Mathematical Problems in Engineering. https://doi.org/10.1155/2020/3618284

[12] Mahjabin, T., Xiao, Y., Sun, G., & Jiang, W. (2017). A survey of distributed denial-of-service attack, prevention, and mitigation techniques. International Journal of Distributed Sensor Networks. https://doi.org/10.1177/1550147717741463

[13] Masood, S., & Zafar, A. (2024). Deep-efficient-guard: securing wireless ad hoc networks via graph neural network. International Journal of Information Technology (Singapore). https://doi.org/10.1007/s41870-023-01702-z

[14] Otala, J., Minard, A., Madraki, G., & Mousavian, S. (2021). Graph-based modeling in shop scheduling problems: Review and extensions. In Applied Sciences (Switzerland). https://doi.org/10.3390/app11114741

[15] Patil, S., Varadarajan, V., Mazhar, S. M., Sahibzada, A., Ahmed, N., Sinha, O., Kumar, S., Shaw, K., & Kotecha, K. (2022). Explainable Artificial Intelligence for Intrusion Detection System. Electronics (Switzerland). https://doi.org/10.3390/electronics11193079

[16] Schiller, E., Aidoo, A., Fuhrer, J., Stahl, J., Ziörjen, M., & Stiller, B. (2022). Landscape of IoT security. In Computer Science Review. https://doi.org/10.1016/j.cosrev.2022.100467

[17] Seo, M., Jeong, E., & Kim, K. S. (2023). Multi-Class fNIRS Classification Using an Ensemble of GNN-Based Models. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3339647

[18] Shah, Z., Ullah, I., Li, H., Levula, A., & Khurshid, K. (2022). Blockchain Based Solutions to Mitigate Distributed Denial of Service (DDoS) Attacks in the Internet of Things (IoT): A Survey. In Sensors. https://doi.org/10.3390/s22031094

[19] Veeramreddy, J., & Vaddella, R. P. V. (2016). Anomaly-based network intrusion detection through assessing feature association impact scale. International Journal of Information and Computer Security. https://doi.org/10.1504/IJICS.2016.079185

[20] Venkatapathy, S., Votinov, M., Wagels, L., Kim, S., Lee, M., Habel, U., Ra, I. H., & Jo, H. G. (2023). Ensemble graph neural network model for classification of major depressive disorder using whole-brain functional connectivity. Frontiers in Psychiatry. https://doi.org/10.3389/fpsyt.2023.1125339

[21] Wani, S., Imthiyas, M., Almohamedh, H., Alhamed, K. M., Almotairi, S., & Gulzar, Y. (2021). Distributed denial of service (Ddos) mitigation using blockchain—a comprehensive insight. In Symmetry. https://doi.org/10.3390/sym13020227

[22] Yang, J., Chen, Z., Sun, H., & Samanta, A. (2023). Graph-EAM: An Interpretable and Efficient Graph Neural Network Potential Framework. Journal of Chemical Theory and Computation. https://doi.org/10.1021/acs.jctc.3c00344

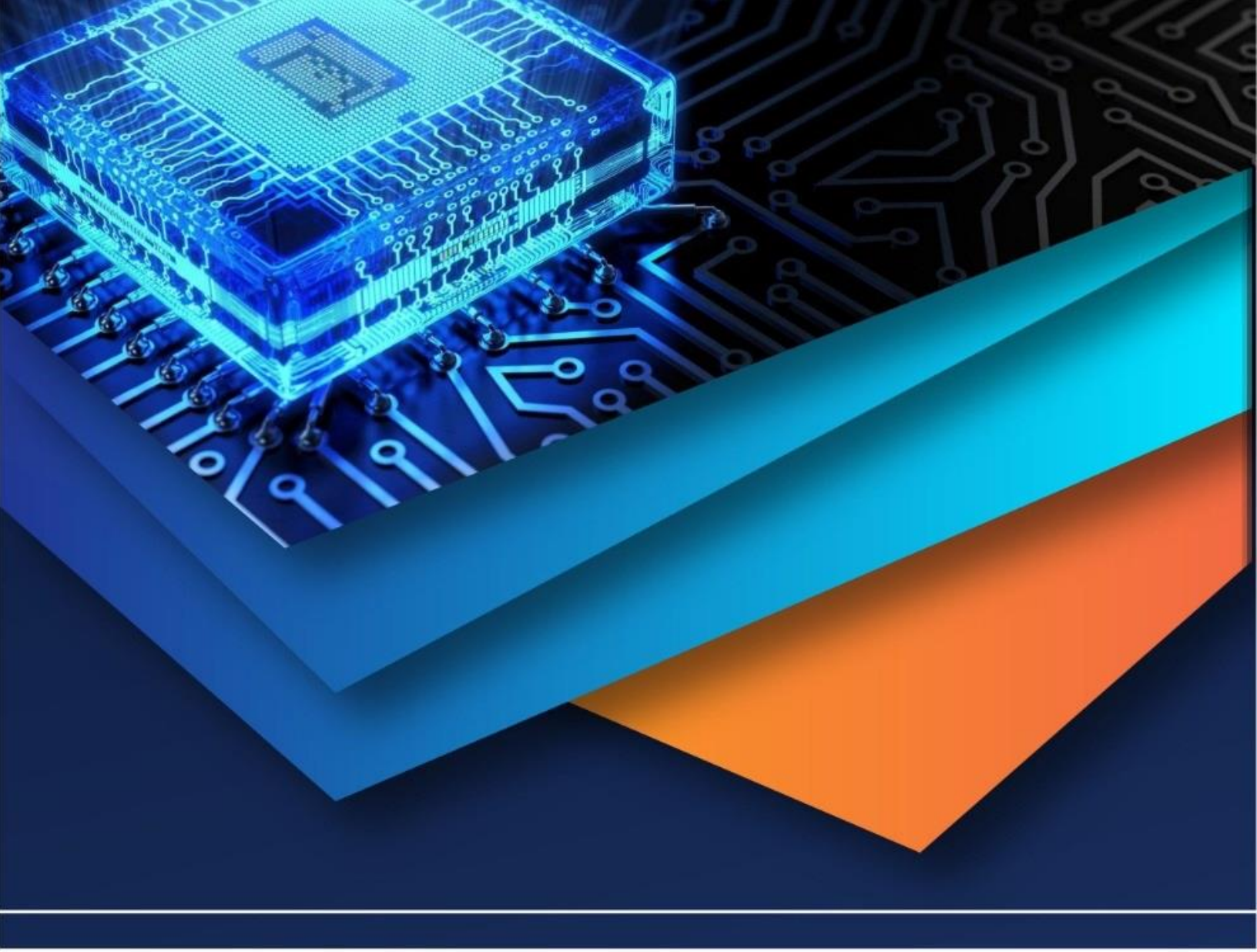
[23] Zeng, Z., Wang, C., Ma, F., Wang, P., & Wang, H. (2024). Multiple-model and time-sensitive dynamic active learning for recurrent graph convolutional network model extraction attacks. International Journal of Machine Learning and Cybernetics. https://doi.org/10.1007/s13042-023-01916-4

[24] Zhang, B. C., Hu, G. Y., Zhou, Z. J., Zhang, Y. M., Qiao, P. L., & Chang, L. L. (2017). Network intrusion detection based on directed acyclic graph and belief rule base. ETRI Journal. https://doi.org/10.4218/etrij.17.0116.0305

[25] Zhang, Q., Cai, L., Liao, N., Lu, Y., Zhang, J., Zhang, C., & Zeng, K. (2024). Work Function Prediction by Graph Neural Networks for Configurationally Hybridized Boron-Doped Graphene. Langmuir. https://doi.org/10.1021/acs.langmuir.4c00228

[26] Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., & Li, H. (2020). T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. IEEE Transactions on Intelligent Transportation Systems. https://doi.org/10.1109/TITS.2019.2935152

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)