International Journal of Computer Techniques – Volume 12 Issue 5, September - October - 2025

A Hub-Centric Graph Neural Network for Scalable Intrusion Detection

Raghul Azhagaiah
Backend Developer, Independent Researcher, India
Email: raghul.a1710@gmail.com

Abstract: Intrusion detection is a critical component of modern cybersecurity, particularly as network traffic continues to increase in scale and complexity. Traditional machine learning methods have achieved promising results but often fail to capture the relational dependencies among network entities. Graph Neural Networks (GNNs) offer an effective framework for learning from graph-structured data, enabling the modeling of such relationships. However, applying GNNs directly to large-scale intrusion datasets poses scalability challenges due to the vast number of nodes and potential interconnections.

In this study, we propose a **hub-centric graph construction** approach where Internet Protocol (IP) addresses act as hub nodes, and all traffic records associated with the same IP are connected to the hub. This strategy reduces graph density and enhances scalability without significant loss of structural information. The proposed GNN model, implemented using a four-layer **SAGEConv** architecture, achieved an accuracy of **80%** on the **UNSW-NB15** dataset [1]. Although traditional models such as Random Forest reached a higher accuracy of **96%**, the GNN demonstrated stronger relational learning and better generalization to complex, non-linear, and previously unseen attack patterns.

Keywords: Graph Neural Network, Intrusion Detection, Cybersecurity, UNSW-NB15, Hub Node

1. Introduction

Over the past few years, Internet applications have become increasingly sophisticated and pervasive, intensifying the demand for robust network security solutions. Intrusion Detection Systems (IDSs), powered by Artificial Intelligence (AI) techniques, play a crucial role in safeguarding network integrity and reliability. Their primary objective is to dynamically detect malicious activities originating from both internal and external sources in real time, thereby maintaining the overall stability of network infrastructures.

With the rapid advancement of global digitization, the frequency and sophistication of cyber-attacks have escalated significantly. Such attacks not only jeopardize the confidentiality and integrity of organizational and individual data but also result in severe financial losses and reputational harm.

Traditional machine learning algorithms, including Random Forests, Support Vector Machines (SVMs) have been widely applied to benchmark intrusion detection datasets with varying degrees of success. However, these models generally assume that all samples are independent, limiting their ability to capture inter-entity relationships such as correlations between network flows sharing the same IP address or session attributes.

Graph Neural Networks (GNNs) overcome this limitation by representing network entities and flows as nodes within a graph, enabling relational learning through structural connectivity. Despite their potential, constructing large-scale graphs for datasets like UNSW-NB15 [1] introduces computational and memory challenges due to dense interconnections. To address these scalability issues, this study introduces a **hub-centric GNN framework**, which optimizes graph construction by designating Internet Protocol (IP) addresses as hub nodes, thereby reducing complexity while preserving critical relational information.

2. Related Work

Intrusion Detection Systems (IDS) have been extensively studied using various machine learning and deep learning techniques to identify abnormal patterns in network traffic. Traditional approaches primarily rely on feature-based models such as **Decision Trees**, **Support Vector Machines (SVMs)**, **Random Forest (RF)**, and **Gradient Boosting Machines (GBM)**. Indira Bharathi *et al.* [2] proposed an ensemble-based NIDS combining Random Forest and Extreme Gradient Boosting, trained on the UNSW-NB15 dataset [1], which effectively improved detection accuracy and reduced overfitting through recursive feature elimination. Such ensemble approaches have demonstrated robustness in handling high-dimensional data and achieving strong generalization performance.

Other studies explored hybrid models to enhance classification performance. Wisanwanichthan *et al.* [3] introduced a double-layered hybrid approach combining Naïve Bayes and SVM to detect rare attack types like R2L and U2R. Ahmim *et al.* [4] used a hybrid deep belief network for distributed denial of service (DDoS) attack detection, while Zou *et al.* [5] integrated hierarchical clustering with twin SVM to improve precision on complex network traffic.

While these classical and deep learning approaches have shown considerable success, they often fail to capture **relational dependencies** between entities in network traffic. Each record is treated independently, which overlooks the **graph-structured relationships** such as shared IP addresses, port numbers, or connection flows. To address this limitation, recent studies have shifted toward **graph-based learning methods**. Hu et al. [6] proposed **GRID**, a graph representation-based intrusion detection framework that models system call sequences as graphs. By leveraging **Graph Random State Embedding (GRSE)**, the method captures **topological dependencies** among system calls using random walks and graph pooling. Their results demonstrated that attack behaviors exhibit **distinct structural patterns**, which can be effectively captured through graph-based representations, outperforming traditional sequence embedding methods.

Graph Neural Networks (GNNs) extend this concept by performing **message passing** between connected entities, enabling the model to learn from both node attributes and network structure. This allows GNN-based IDS to detect coordinated or stealthy attacks that exhibit structural correlations across nodes. Unlike ensemble or statistical learning models, GNNs inherently integrate context-aware learning, making them a promising direction for modern intrusion detection systems.

3. Proposed Methodology

A. Dataset

The **UNSW-NB15** dataset [1] was selected due to its comprehensive coverage of modern cyberattack types and realistic network traffic. It contains 45 features including protocol, service, source IP, destination IP, and various flow-based statistics.

B. Baseline Model: Random Forest

To establish a benchmark for evaluating the proposed graph-based model, a Random Forest (RF) classifier was implemented using the UNSW-NB15 dataset [1]. This step provided a solid reference for comparing the performance of conventional feature-based approaches against graph-based learning methods. The raw dataset, which includes distinct training and testing subsets, was first cleaned by removing all rows containing missing or invalid labels. The two subsets were then merged into a single dataframe to ensure consistent preprocessing across the entire dataset.

The dataset comprises both categorical and numerical features, representing various network attributes such as protocol type, service, source, destination bytes, and connection-related statistics. Categorical columns including proto, service, state, swin, dwin, trans_depth, ct_flw_http_mthd, is_ftp_login, ct_ftp_cmd, and is_sm_ips_ports were encoded using the LabelEncoder to transform non-numeric entries into integer values. This encoding process allowed the model to interpret the categorical variables numerically without introducing ordinal bias. Numerical columns, on the other hand, were scaled using the MinMaxScaler to normalize their values within the range [0, 1]. This normalization ensured that no single feature dominated the model due to its magnitude, allowing all features to contribute proportionally to the training process. The preprocessing pipeline thus maintained feature consistency and supported the model in capturing both packet-level and flow-level variations effectively.

Following preprocessing, the dataset was divided into training and testing sets in an 80:20 ratio using stratified sampling to preserve the class distribution of normal and attack samples. The Random Forest classifier from the scikit-learn library was employed. The model was configured with 200 estimators, allowing the ensemble to capture a diverse set of decision boundaries across multiple trees.

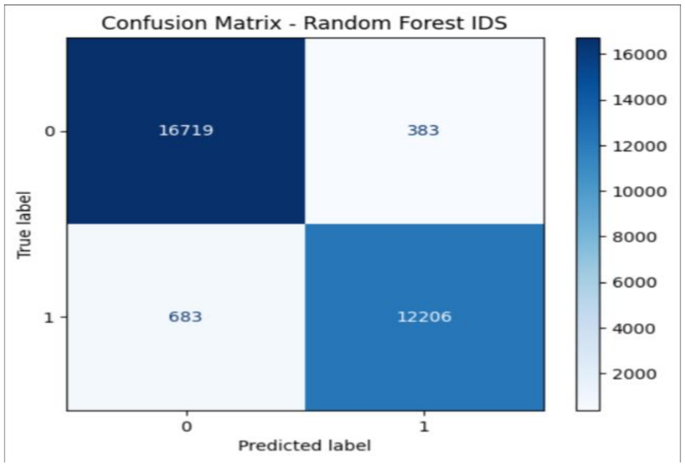

Figure 3.1. Confusion matrix of Random Forest classifier

The model achieved approximately 96% accuracy on the test data, indicating strong baseline performance for network intrusion detection. The classification report further revealed that the model performed consistently across both normal and attack classes, achieving high precision and recall values with minimal class imbalance issues. This demonstrates that the Random Forest model effectively learned discriminative patterns from the dataset's tabular feature representation. The confusion matrix, presented in Figure 3.1, shows that the classifier successfully distinguishes between benign and malicious traffic, with only a small number of false positives and false negatives. Furthermore, the training and testing accuracy plot, shown in Figure 3.2, indicates that the model's training accuracy reaches near-perfect levels, while test accuracy stabilizes around 96–97%.

Despite its strong numerical performance, the Random Forest model inherently assumes that all samples are independent and identically distributed. This assumption limits its ability to capture relational dependencies within the network, such as multiple connections originating from the same IP address or coordinated attack behavior between nodes. These network-level relationships often hold critical contextual information that feature-based models cannot represent. Consequently, while Random Forest serves as a reliable and high-performing baseline, it lacks the structural awareness required to model interdependencies present in modern network traffic. This limitation motivated the development of the proposed Graph Neural Network (GNN) approach, which explicitly incorporates topological relationships among network flows to enhance intrusion detection performance.

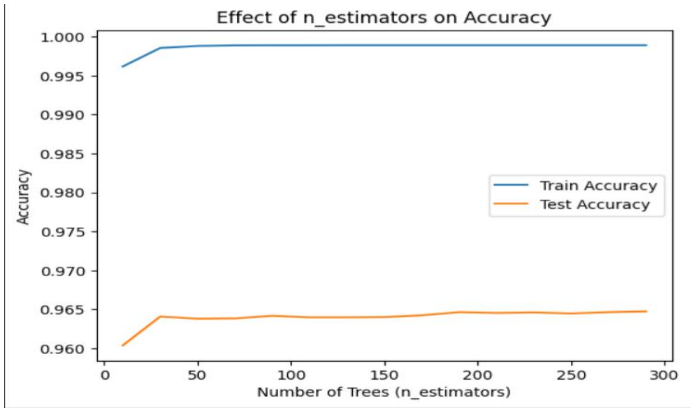

Figure 3.2. Training and testing accuracy of Random Forest model

C. Graph Construction

To incorporate relational dependencies among network flows, same UNSW-NB15 dataset [1] was represented as a graph, allowing the model to learn structural relationships between individual traffic records. Each record in the dataset was initially treated as a node, while connections between nodes were determined by shared IP addresses. In the first attempt, all nodes sharing a common IP address were fully interconnected, resulting in a dense adjacency structure with millions of edges. This caused excessive memory consumption and computational inefficiency, rendering the model impractical for large-scale training. To overcome this limitation, a hub-centric tree structure was introduced. In this design, each unique IP address functions as a central hub node, and all flows (records) associated with that IP are connected exclusively to the hub rather than to one another. This transformation significantly reduced the number of edges while retaining essential relational information between traffic flows. The result is a scalable graph topology that captures communication patterns between hosts without overwhelming system memory.

For each connection, the node features were derived from the preprocessed tabular data used in the baseline Random Forest model. These features include both categorical and numerical. Each record node in the graph inherits a numerical vector representation of these features, while hub nodes, representing unique IP addresses, are initialized as zero vectors to act purely as aggregators of information from their connected record nodes. Bidirectional edges were added between each record and its corresponding hub, enabling message passing in both directions during training. This approach allowed the model to propagate contextual information from multiple flows associated with the same IP, effectively capturing shared behavioral patterns and possible correlations between malicious activities.

D. Feature Engineering for GNN

The feature engineering process for the GNN mirrored that of the Random Forest baseline to maintain consistency. Categorical features such as *proto*, *service*, and *state* were label-encoded into integer indices, while numerical attributes like duration, source bytes, and destination bytes were normalized using MinMaxScaler to the [0, 1] range. This ensured balanced feature scaling and compatibility with PyTorch tensors. The resulting node feature matrix combined categorical and numerical attributes, enabling the GNN to effectively learn both protocol-level and traffic-level patterns. This preprocessing ensured proportional feature contribution and improved relational learning during graph convolution.

E. Graph Neural Network (GNN) Architecture

The graph-based learning model was implemented using GraphSAGE (Graph Sample and Aggregate), a scalable variant of Graph Neural Networks (GNNs) designed for handling large and complex graph data. The proposed architecture comprises four stacked GraphSAGE convolutional layers, sequentially combined with ReLU activation functions and followed by a Softmax output layer. Each GraphSAGE layer aggregates feature information from neighboring nodes and updates node embeddings through non-linear activation using ReLU. The model was developed in PyTorch Geometric, where the input feature matrix included both record and hub nodes, and the edge index captured the bidirectional relationships among them.

The GNN was trained for 500 epochs using the Adam optimizer with a learning rate of 0.01 and a categorical cross-entropy loss function. Training and testing masks were applied to ensure that only record nodes contributed to the classification loss, while hub nodes participated solely in message propagation. Mini-batch training was employed to handle the graph efficiently within available memory resources. During training, the loss decreased steadily with epochs, and the accuracy consistently improved, indicating stable convergence of the model. The training trends are shown in Figure 3.3 and Figure 3.4, where the smooth increase in accuracy and corresponding decrease in loss confirm that the model effectively learned discriminative node representations over time.

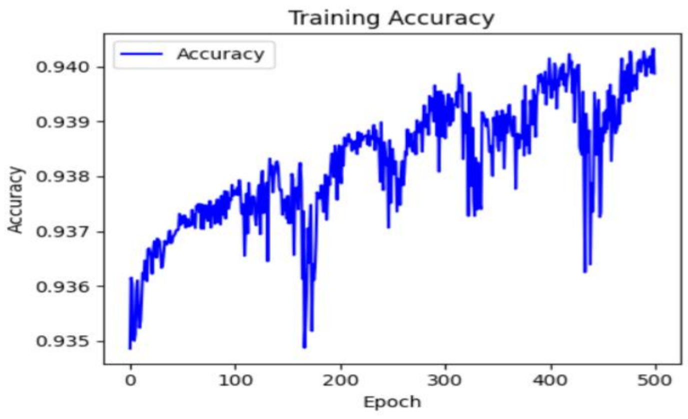

Figure 3.3. Training accuracy of the proposed GNN model

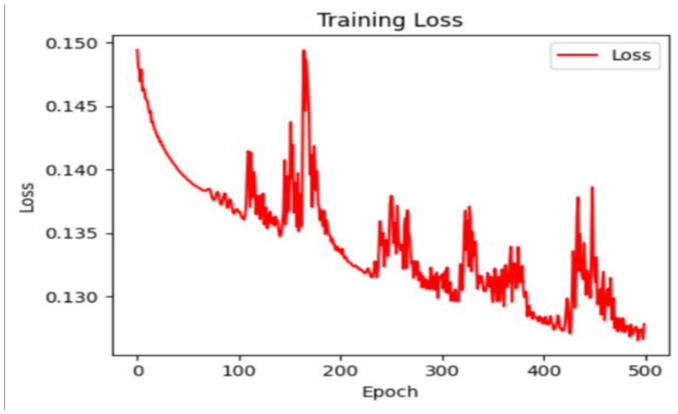

Figure 3.4. Training loss of the proposed GNN model

This multi-layer architecture enabled hierarchical neighborhood aggregation, allowing the model to capture both local dependencies such as connections within the same IP hub and global relationships that span across different hubs. Unlike traditional feature-based classifiers, the graph-based approach captures contextual relationships directly within the network structure, allowing it to detect coordinated or stealthy attacks that emerge as relational patterns rather than isolated anomalies.

4. Results and Discussion

The Random Forest classifier achieved an overall accuracy of approximately 96%, demonstrating strong baseline performance in identifying both normal and attack traffic. The confusion matrix shown in Figure 3.1 illustrates that the model maintained balanced classification between benign and malicious instances, with a relatively small number of false positives and false negatives. However, despite its high accuracy, the Random Forest algorithm inherently assumes that each data instance is independent of others, which limits its capacity to recognize relational patterns between flows that share common network attributes, such as source or destination IP addresses. This independence assumption restricts the model's interpretability in complex network environments where attacks often manifest through correlated or sequential activities.

In contrast, the proposed hub-centric Graph Neural Network (GNN) achieved an accuracy of 83% on the UNSW-NB15 dataset [1]. Although slightly lower than the Random Forest model, the GNN adopts a distinct learning paradigm that emphasizes relational dependencies within the data. By representing each unique IP address as a hub node and linking all associated flows, the model captures structural correlations in network behavior. As shown in Figure 4.1, the confusion matrix indicates effective attack detection, correctly classifying 44,382 malicious instances with low false negatives. These results demonstrate the GNN's ability to model relationships among related connections and identify coordinated attack patterns involving shared IP addresses.

While tree-based ensemble methods like Random Forest and XGBoost often achieve higher raw accuracy on structured data, they lack the contextual awareness to capture topological dependencies. In contrast, the hub-centric GNN leverages graph connectivity to propagate information across related entities, enhancing generalization in relational network environments. Its hub-node design also reduces edge density, mitigating memory overhead and improving scalability. Thus, even with slightly lower accuracy, the proposed GNN offers a more interpretable and scalable solution by learning context-aware representations and identifying coordinated attack patterns often missed by traditional models.

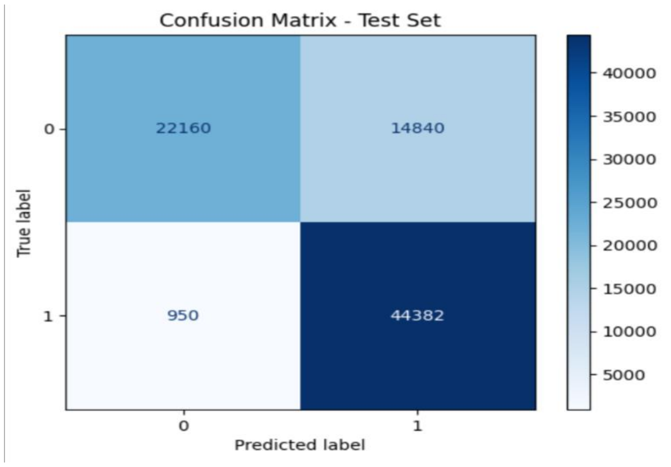

Figure 4.1. Confusion matrix of the proposed GNN model

5. Conclusion

This research presents a **hub-centric GNN** for scalable intrusion detection using the UNSW-NB15 dataset [1]. By leveraging IP addresses as hub nodes, the approach reduces graph complexity while retaining essential structural dependencies. The proposed model achieved an accuracy of **83%**, demonstrating its effectiveness in learning relational patterns across network flows.

Although traditional algorithms such as Random Forest achieved higher accuracy, the GNN's ability to capture **inter-node correlations** offers superior adaptability to non-linear and evolving attack patterns.

References

[1] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)," in 2015 Military Communications and Information Systems Conference (MilCIS), IEEE, 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.

[2] G. Indira Bharathi, Rohan Makhija "Network Intrusion Detection System Using Random Forest and Gradient Boosting Machines," in 2024 4th International Conference on Intelligent Technologies (CONIT), IEEE, 2024, pp. 1–6, doi: 10.1109/CONIT60122.2024.00045.

[3] T. Wisanwanichthan and M. Thammawichai, "A Double-Layered Hybrid Approach for Network Intrusion Detection System Using Combined Naïve Bayes and SVM," IEEE Access, vol. 9, pp. 137818–137830, 2021, doi: 10.1109/ACCESS.2021.3118573.

[4]A.Ahmim, F.Maazouzi, M.Ahmim, and S.Namane, "Distributed Denial of Service Attack Detection for the Internet of Things Using a Hybrid Deep Learning Model," IEEE Access, vol. 10, pp. 69414–69426, 2022, doi: 10.1109/ACCESS.2022.3192665.

[5] Li Zou, Xuemei Luo, Yan Zhang, Xiao Yang, Xiangwen Wang, "Network Intrusion Detection Based on Hierarchical Clustering and Twin SVM," IEEE Access, vol. 8, pp. 68543–68552, 2020, doi: 10.1109/ACCESS.2020.2986957.

[6] Z. Hu, L. Liu, H. Yu, Xiangzhan Yu, "GRID: A Graph Representation-Based Intrusion Detection Framework Using Graph Random State Embedding," IEEE Access, vol. 9, pp. 102834–102847, 2021, doi: 10.1109/ACCESS.2021.3096994.

[7] M. Zhong, M. Lin, C. Zhang, and Z. Xu, "A Survey on Graph Neural Networks for Intrusion Detection Systems: Methods, Trends and Challenges," College of Computer and Cyber Security, Fujian Normal University, Fuzhou, 350117, Fujian, China, 2023.