Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 3, 2025

E-GRAPHSAGE++: ENHANCING GRAPH NEURAL NETWORK-BASED INTRUSION DETECTION SYSTEMS FOR IOT NETWORKS

Syed Talal Musharraf[1], Muhammad Hamza Khan[2], Umair Shafiq Khan[3], Muhammad Zulkifl Hasan[4], Muhammad Zunnurain Hussain[*5]

[1]*Department of Computer Science, Bahria University Lahore, Pakistan.*
[2]*Department of Computer Science, Bahria University Lahore, Pakistan.*
[3]*Department of Electrical Engineering, Information Technology University, Lahore, Pakistan*
[4]*Department of Computer Science, University of Central Punjab Lahore, Pakistan.*
[*5]*Department of Computer Science, Bahria University Lahore, Pakistan.*

[1]syedtalalmusharraf10@gmail.com, [2]hamza1357937@gmail.com, [3]phdee23002@itu.edu.pk, [4]zulkifl.hasan@ucp.edu.pk, [*5]zunnurain.bulc@bahria.edu.pk

DOI: https://doi.org/10.5281/zenodo.15542356

Keywords
Graph Neural Networks, Network Intrusion De- tection System, Internet of Things, Edge Embedding, Topological Information.

Article History
Received on 20 January 2025
Accepted on 20 February 2025
Published on 27 February 2025

Corresponding Author: * Muhammad Zunnurain Hussain
zunnurain.bulc@bahria.edu.pk

Abstract

This paper introduces E-GraphSAGE++, an ad- vanced Network Intrusion Detection System (NIDS) leveraging Graph Neural Networks (GNNs) to enhance security in IoT networks. Unlike traditional methods, E-GraphSAGE++ effec- tively captures both edge features and topological information inherent in flow-based network data. This dual-focus approach allows for a more comprehensive analysis of network traffic, enabling the detection of complex attack patterns that might be overlooked by methods focusing solely on node features or using traditional ML approaches. Our approach addresses several key limitations of existing NIDS solutions. Traditional NIDS methods, particularly those based on signature detection, often fail to identify novel or sophisticated attacks due to their reliance on predefined attack signatures. On the other hand, anomaly-based detection methods, while better at identifying new threats, can suffer from high false-positive rates. E-GraphSAGE++ mitigates these issues by leveraging the relational structure of network data through GNNs, providing a holistic view of traffic patterns and their interdependencies. We conduct extensive evaluations on four benchmark NIDS datasets: BoT-IoT, ToN-IoT, and their NetFlow variants NF-BoT-IoT and NF-ToN-IoT. Our experiments demon- strate that E-GraphSAGE++ significantly outperforms state-of- the-art methods in key classification metrics, including accuracy, precision, recall, and F1-score. For instance, E-GraphSAGE++ achieves near-perfect precision and recall rates on these datasets, indicating its robustness and effectiveness in real-world scenarios. Hence, these results show the possibility of GNNs in transforming ND and establishing a benchmark for future research on the domain. In addition, there is a new method of edge embedding in E-GraphSAGE++, which improves not only the detection performance and provides better interpretability of the model's conclusions. Through the visualization of the learned embeddings, network administrators are able to

better understand the char- acteristics of identified abnormal activities and cyber-attacks so as to design more appropriate countermeasures. Cybersecurity is therefore enhanced by the skill of differentiating between different network flows as well as the risks that are associated with each of the flows and their visual presentation. Moreover, to be more scalable and efficient, E-GraphSAGE++ is proposed. The model's structure is well suited for large-scale network processing, and therefore, its usage can be suggested in real-time intrusion detection systems. Thus, with the help of improved GNN technologies, E-GraphSAGE++ is capable of processing the dynamically changing characteristics of IoT networks and new threats entering the network. Specifically, E-GraphSAGE++ can be considered as the development in the area of network intrusion detection for IoT-enabled settings. Thus, integrating the edge features and topological information can be a valuable tool to detect cybersecurity threats of a higher level. Thus, the applicability of E-GraphSAGE++ within various benchmarks is suitable for set scenes, and its capability to perform well proves its general usability. Future work will seek to improve the model by implementing advanced sampling techniques and, in order to increase the model's practical use, as well as making its functioning more transparent, use the explainable AI tools. These improvement Purpose to transform E-GraphSAGE++ not only into a strong identification system but also into an IoT networks' security solution.

INTRODUCTION

Internet of Things (IoT) network threats have also grown substantially and become much complex over the year. IoT stands for Internal of Things and it is a network of devices like cameras, temperature sensors, smart TVs, wireless printers etc. that needs a network connection [1]. These devices are often known as the Internet of Things or IoT – they simplify and enhance numerous areas of life, such as home, health, busi- ness, manufacturing, and many others. However, due to their constant connectivity, and frequent lack of enough security measures, they easily become the target of hackers. It is even used for IoT ransomware, Botnet DDoS attacks, Data theft, unauthorized surveillance and more. Intrusion Detection Sys- tem (IDS) particularly Network Intrusion Detection Systems (NIDS) has significant importance in the protection on such IoT networks analyzing the network traffic to identify security threats. NIDS are intended to detect and prevent particularly potential hazardous activities in a computer network-based cyber attack. There are two main types of NIDS: systems; the signature-based system and the anomaly detection-based system. In the case of signature-based NIDS, these are prepro-grammed with alarm signatures which are patterns of known attacks. They are efficient against the known threats since other endpoints give a low false positive proportion and high discovery ability of known attacks. Unfortunately, they have difficulty identifying new or different versions of an attack that does not use a signature. Anomaly detection-based systems, on the other hand, are systems that are created with the aim of protecting networks by identifying intrusions since they differ from regular traffic. It is possible for these systems to detect previously unknown types of attack; therefore, they are more capable of responding to new threats. However, they can also generate a high number of false positives because normal variations in traffic can also set off alarms. One of the big issues that needs to be solved is to make these systems sensitive to low levels of virus presence while having a low false-positive ratio.

Fig. 1. Deployment architecture of the Network

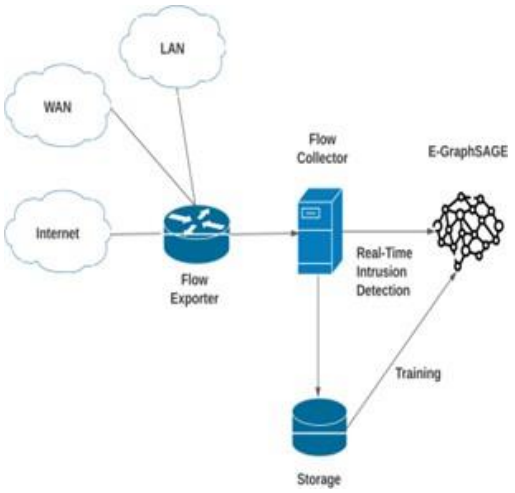

Fig. 1. Deployment architecture of the Network

Recent advances in Machine Learning (ML) and particu- larly deep learning have shown promise in developing new NIDS solutions that can address the limitations of traditional approaches. Deep learning models, with their ability to learn complex patterns from large datasets, have been applied to various aspects of cybersecurity, including intrusion detection. These models can automatically extract features from raw data, reducing the need for manual feature engineering and improving detection accuracy.

In this paper, we explore the use of Graph Neural Networks (GNNs), a relatively new sub-field of deep learning tailored for graph-structured data. GNNs are designed to leverage the inherent structure of graph data, making them suitable for applications where relationships between entities are crucial. These applications span social sciences, chemistry, telecom- munications, and more. In the context of network intrusion detection, flow records, which capture communication between devices, can be naturally represented as graphs. Each device can be modelled as a node, and the communication flows between them can be represented as edges. This graph repre- sentation enables the capture of both individual and relational characteristics of network traffic.

We propose E-GraphSAGE++, an enhanced GNN model that captures both edge features and topological information for network intrusion detection in IoT networks. Traditional GNNs primarily focus on node features, which are effective for tasks like node classification and link prediction. How- ever, network intrusion detection requires a more nuanced approach that considers the properties of the edges (i.e., the communication flows) as well. By integrating edge features, E-GraphSAGE++ provides a more comprehensive analysis of network traffic, enabling the detection of sophisticated attack patterns that might be overlooked by methods focusing solely on node features or using traditional ML approaches.

E-GraphSAGE++ leverages the strengths of the original GraphSAGE algorithm, which samples and aggregates infor- mation from a node's local neighbourhood to generate node embeddings. We extend this approach to include edge fea- tures, allowing the model to learn representations that capture both the characteristics of individual communications and the broader network structure. This dual-focus approach enhances the model's ability to detect complex, multi-stage attacks that involve coordinated activities across multiple devices.

We conduct extensive evaluations on four benchmark NIDS datasets: BoT-IoT, ToN-IoT, and their NetFlow variants NF- BoT-IoT and NF-ToN-IoT. These datasets provide a com- prehensive and diverse set of scenarios for evaluating the performance of network intrusion detection systems. Each dataset includes labelled instances of both benign and ma- licious network flows, covering a wide range of attack types and normal behaviours.

BoT-IoT is a real-world-based dataset for IoT realistic networks and features different types of attacks like DDoS, DoS, reconnaissance, data theft, etc. Likewise, the ToN- IoT dataset contains IoT/IIoT telemetry streams and covers injection, password-type attacks, ransomware, and backdoor attacks. The transformed datasets, which are NF-BoT-IoT and NF-ToN-Iot, offer a recognized set of features from the actual packets of the initial datasets to make comparative studies with different NIDS possible.

By showing that E-GraphSAGE++ outperforms the state-of- the-art methods in terms of classification accuracy, precision, recall, and F1-score in our experiments. Thus, the observed improvement in performance can be explained by the model's capability to consider both edge features and knowledge of the topology. Since E-GraphSAGE++

incorporates the flow matrix as well as its contextual interaction readily in its computation, it can produce a more detailed representation of the network's behaviour as a result, enhance the ability of detected attacks, including known and previously unseen patterns.

In addition, E-GraphSAGE++ also has an updated method of embedding edges that at the same time improve the ability to detect adversarial subgraphs and the explainability of the model's decision-making. Such interpretability is of immense importance to network administrators and cybersecurity pro- fessionals since they have to deal with the implications of detected anomalous behaviour and attacks. As a result, through the visualization of the learned embeddings, administrators are able to acquire a better understanding of the flow relationships and flow attributes of a given network, and hence, enable the resolution of the cause of the alert or the initiation of the necessary remedial actions.

Moreover, to target both the scalability and efficiency of the tool, E-GraphSAGE++ has been developed. It operates more effectively and efficiently in large-scale network data analysis in readiness for real-time intrusion detection. Therefore, the applied improvements of GNN make E-GraphSAGE++ ca- pable of learning and updating dynamic IoT network traffic and possible patterns or threats. This flexibility is vital in the cybersecurity domain, which has no static environment given that the attackers are constantly creating new ways to breach security.

Due to its best performance on diverse benchmark datasets, the applicability of E-GraphSAGE++ can be extended to numerous IoT settings. From the results obtained in different types of network traffic and different attacks, it can be con- cluded that the model is both general and effective when used in practice. Thus, being a flexible and exhaustive solution for network intrusion detection, E-GraphSAGE++ can be viewed as a contribution to the progress in the given field.

The next steps will set up further development of the model, how the modern methods of work sampling can be applied for improved practicality of the model, as well as the methods of explainable AI that can be used for improvement of the model and its effectiveness. These improvements need to enforce the idea of E-GraphSAGE++ as not only a reliable detection system but also as a significant aspect of countermeasures against IoT networks' cyber threats. Thus, by further improv- ing and developing the options of E-GraphSAGE++, we can guarantee the IoT is protected from numerous cyber threats facing today's developments.

Altogether, E-GraphSAGE++ solves the significant problem of the necessity of efficient and scalable IDS in IoT networks. With our methodology relying on Graph Neural Networks for incorporating the edge features as well as the topology of the graphs, the combined topological features make our work a powerful tool in dealing with complex cyber threats. The outcomes of the presented extensive evaluations demonstrate the high potential of GNNs for developing new approaches to network intrusion detection that will help to reconsider the traditional conceptions concerning this aspect of cybersecurity.

2. Literature Review

Previous research on ML-based Network Intrusion Detec- tion Systems (NIDS), such as [4], [5], [6], [7], [8], has primarily focused on treating flow data records independently, without considering their interrelationships. This approach limits the ability of these methods to detect sophisticated IoT network attacks that exploit the interconnected nature of network traffic. For example, botnet attacks [9], which involve coordinated malicious activities across multiple devices, re- quire an understanding of the relationships between different network flows. Similarly, distributed port scans [10], which probe a network's ports in a scattered manner, and DNS Amplification attacks [11], which leverage DNS servers to amplify traffic towards a target, also necessitate a more global view of network traffic patterns to be effectively detected.

Haitao et al. [4] developed a multimodal sequential NIDS using a hierarchical progressive network that combines a deep autoencoder and LSTM architecture. This system aims to capture different levels of network data features, achieving high accuracy in identifying intrusions. However, while this approach integrates structural information within the temporal context shared between similar network connections, it re- mains limited in its capacity to capture global traffic patterns, which are

essential for detecting complex attacks that span multiple network segments.

Sarhan et al. [6] addressed the issue of dataset standard- ization by converting the UNSW-NB15, BoT-IoT, and ToN- IoT datasets into a common NetFlow-based format. This conversion involved selecting a set of eight NetFlow fea- tures to ensure consistency across different datasets, thereby facilitating comparative evaluations. The authors evaluated an Extra Tree ensemble classifier on these NetFlow-based datasets, reporting promising results. However, their approach did not fully leverage the structural information inherent in the network data. By treating flow records as isolated events, the method misses out on the potential insights that can be gained from understanding the relationships and interactions between different flows.

Other studies have also sought to improve NIDS by adopting such features as machine learning. For example, Lawal et al. [5] proposed an anomaly mitigation framework for IoT using fog computing. By integrating both the signature-based and anomaly-based detection techniques the general detection potential is enhanced within this framework. Although this hybrid approach is quite useful in certain cases, it has its own limitations in detecting any unknown patterns or attacks that do not conform to the established markers or irregularities.

Churcher et al. [8] discussed the experimental study of several machine learning algorithms for the classification of the attack in the IoT network. They used k-NN, DT, SVM, NB, RF, ANN, and LR on the BoT-IoT dataset. Out of these, the KNN classifier had the highest multiclass classification results. However, like in many other works, this approach considers the flows of a network separately and does not analyze them in conjunction with each other taking into account the overall picture of the network traffic.

Kumar et al. [7] employed ensemble learning to detect cyberattacks in IoT networks via fog-cloud architecture. This framework employs a decision tree as one learner, Naïve Bayes as the second learner random forest as the third learner, and a final learner as an XGBoost classifier to boost the detection. Although this method helps to enhance classification accuracy, it does not meet the need of the analytical model in capturing the illustrative passage of the flows across the entire global network and their mutual interactions.

In this regard, E-GraphSAGE++ proposed here comple- ments node centrality and uses GNNs to include both edge attributes and the structure of the underlying network. They are highly effective when used in cases where graph-structured data is desired such as in the case of network traffic, where the nodes (devices) and the edges (communications) are very important. To conclude, E-GraphSAGE++ combines the edge information with the node features and conducts the analysis on the flow level, which allows for avoiding misinterpretation of all flows as completely independent and can discover more complex attack signatures that other methods prone to the analysis of node features or using only flow data might miss. Thus, GNNs have an essential advantage for modelling the characteristic interactions in network traffic data as they can incorporate both local and global structures. This capability is also vital in determining attacks such as multi-stage attacks and other multiple attack scenarios with interrelated activities, which occur across different devices. Thus, E-GraphSAGE++ enhances the previous methods, considering the interaction context between nodes and making network intrusion detection more efficacious and reliable in the IoT context.

3. BACKGROUND

A. Graph Neural Networks (GNN)

Graph Neural Networks (GNNs) are a new but expanding branch of machine learning that takes advantage of the struc- ture of graph data for numerous applications. Compared to other neural networks, GNN assumes that inputs are graphs which are more general than images or sequences that are used in other kinds of neural networks. Graphs have nodes or vertices and links or edges and therefore they are appropriate for representing several types of real data such as social, molecular, and communicational data.

The strength of GNNs is that they learn representations of nodes and edges from the features of nodes and from the structure of the graph. This capability enables GNNs to explore intricate relations and interactions in the data which is important in many fields characterized by

connectivity and structures. For instance, in social media networks, GNNs can be employed for predicting the user's behaviour or identifying communities relative to friends and their interaction. In biol- ogy, the GNNs can describe the relationship between proteins or the shape of molecules to foresee their characteristics or roles.

In general, GNNs naturally fit into the problem of network intrusion detection due to their ability to consider the relation- ships between devices (nodes) and their communication flow (edges). Every equipment can be regarded as a node while every interchange between the pieces of equipment is equiv- alent to an edge. This representation that is based on graphs allows GNNs to model the local interactions, like two devices that are directly connected and exchange information, and the global structure, such as patterns of communication among many devices. Thus, using the described dual perspective in GNNs, they can detect anomalies in communication patterns and recognize suspicious events like coordinating attacks or malware dissemination.

B. GraphSAGE

GraphSAGE (Graph Sample and Aggregate), published by Hamilton et al. [13], is one of the most known GNN algo- rithms that aims at applying node embeddings to large-scale graphs. The limitation found in typical GNN techniques is the difficulty in scaling up because of the complexity of handling large graphs. Thus, GraphSAGE presents a way to sample and aggregate information from a node's direct environment, as opposed to examining the entire graph all at once.

The concept of GraphSAGE lies in obtaining node repre- sentations via aggregating feature information from a set of neighbouring nodes and a fixed-size sample of them. This is important since this approach enables the model to learn the local structure and features of the graph but at the same time is computationally efficient. It is an 'iterative algorithm', that is, it works in layers; through each layer, information is collected from increasingly broader neighbourhoods. For instance, the first layer of the MLP may collect features from the node's direct neighbours, the second layer may collect features from the neighbours and so on.

The aggregation function that is used in GraphSAGE can be designed in many ways for instance the mean sampling, LSTM sampling or the sampling which is related to pooling. Due to this flexibility, GraphSAGE can be easily adjusted in order to work with more and different types of graph data and applications. Nevertheless, original instantiations of GraphSAGE, like most other GNNs, primarily rely on node features and are intended for prospects including node predic- tion, link prediction, and clustering. It lacks edge features that are important in other works such as edge classification and also the formation of links in the network intrusion detection system.

C. E-GraphSAGE++

Traditional GNNs, including GraphSAGE, are primarily focused on node features for tasks like node classification. They do not naturally handle edge features, which are essential for edge classification tasks required in Network Intrusion Detection Systems (NIDS). E-GraphSAGE++ addresses this limitation by extending GraphSAGE to incorporate edge fea- tures, thereby enabling effective edge classification for detect- ing malicious network flows.

1) **Incorporating Edge Features:** E-GraphSAGE++ en- hances the original GraphSAGE model by incorporating edge features into the embedding process. In network intrusion detection, each edge (i.e., communication flow) carries signif- icant information, such as the number of packets, bytes trans- ferred, flow duration, and other metadata. By including these edge features, E-GraphSAGE++ can capture a more compre- hensive view of network activities. The model aggregates edge features along with node features to generate embeddings that reflect both the communication patterns and the content of the flows. This dual focus allows E-GraphSAGE++ to detect sophisticated attack patterns that involve multiple steps or coordinated activities across different parts of the network.

2) **Edge Embedding Process:** Indeed, in expanding the E- GraphSAGE++ model, the message-passing function is incorporated within the aggregation processing of the edges' features. In addition to information from the neighbouring nodes, E-GraphSAGE++ also gathers features from the edges

between these nodes. This process involves drawing a set of neighbouring edges as a fixed size and utilizing the features of those edges for changing the node embeddings. The node embeddings after the update are then concatenated with the other aggregation of the edge features to produce the final node representations. The node representations mentioned above are then utilized to produce edge embeddings by using the repre- sentations of the joined nodes. This process helps guarantee that the edge embeddings that result from the process will retain both the locality of the neighbourhood and the nature of the communication that is going on.

3) Scalability and Efficiency: Due to the complexity of online big graph data, E-GraphSAGE++ is proposed to work efficiently in large-scale networks. As in GraphSAGE, the sampling and aggregation operations are preserved and their usage with regards to edge features is briefly mentioned to maintain the scalability of the model. Due to the fact that the fixed-size sampling approach reduces the computational cost, the framework is quite useful in processing large graphs, which are characteristic of real-life network-based systems. Moreover, the model is capable of mini-batch training which will be effective in training the model with large data sets and makes the intrusion detection real-time in nature.

4) Application in Network Intrusion Detection: As a re- sult, the proposed E-GraphSAGE++ is capable of satisfyingly capturing edge features and topological information to realise a network intrusion detection system. The model will also be able to detect elaborate multi-stage attacks where several activities occur over the interconnected devices which are quite difficult to pin down through conventional means. These in- clude the fact that the improved E-GraphSAGE++ embeddings obtain better representations of the network activities, thus en- hancing the model's efficacy in differentiating between normal and malicious traffic patterns. This capability is important for the proper defense against current and future trends of cyber threats in IoT networks which are characterized by various and flexible traffic loads.

Thus, the E-GraphSAGE++ model is a major improvement over the basic GNN methods as it offers more accuracy and efficiency for the NID tasks. Through distinguishing the specifics of interactions between the flows within the network and encapsulating contextual data into the model, E-GraphSAGE++ allows for highly effective threat identification in IoT networks and enables to prevent of high-level threats.

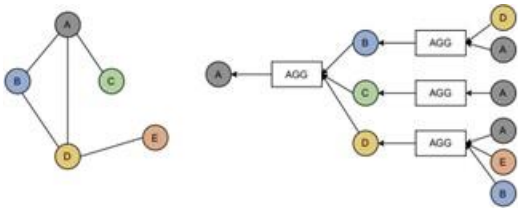

Fig. 2. A given graph (left), and the corresponding GraphSAGE architecture with depth-2 convolutions (right) and full neighbourhood sampling.

4. DATASETS

We use four publicly available Network Intrusion Detection System (NIDS) datasets for evaluating the performance of E- GraphSAGE++: These include BoT-IoT, ToN-IoT and the two NetFlow enabled versions: NF-BoT-IoT and NF-ToN-IoT. These datasets contain labelled instances of both the attack and benign sample flows which makes the range of scenarios very rich. The following is a detailed description of each dataset:

A. BoT-IoT

The BoT-IoT dataset from Koroniotis et al. is one that focuses on typical IoT networks hence well suited for the purpose. It was developed at the UNSW Canberra Cyber Range Lab with realistic procedures for IoT devices. These attacks are; Distributed Denial of Service (DDoS) attack, Denial of Service (DoS) attacks, scanner, expeditor, and data dumper. It is seen that the BoT-IoT dataset is imbalanced; while attack flows are in the millions, benign flows are only in the orders of thousands or ten thousand. This has inequitable implications on feature learning as well because most of the learning algorithms used in machine learning models are prone to majority class bias. The dataset has in total of 47 features, where several features are based on

flow measurements in addition to basic packet-header properties, which is more versatile than for instance the NetFlow dataset.

B. ToN-IoT

The ToN-IoT dataset, developed by Alsaedi et al., is an- other comprehensive dataset that captures telemetry data from IoT/IIoT environments. This dataset was generated using a large-scale IoT testbed, encompassing various types of IoT devices such as weather stations, motion sensors, and surveil- lance cameras. The ToN-IoT dataset includes a wide range of attack scenarios, including data injection, password at- tacks, ransomware, backdoor access, and Man-In-The-Middle (MITM) attacks. In addition to network traffic, ToN-IoT also provides data from operating system logs and device telemetry, making it a multi-faceted dataset for NIDS evaluation. The dataset includes 44 features extracted from the network traffic, capturing both statistical properties and temporal dynamics of the flows.

C. NF-BoT-IoT

The NF-BoT-IoT is a novel dataset derived from NetFlow, which is in variant of the BoT-IoT dataset. It was obtained by applying on BoT-IoT the nProbe tool that maps the raw packet capture files (pcap) into NetFlow format. This conversion process in fact requires the extraction of a standardized set of features from the raw traffic data leading to a dataset that can be easily compared with most of the other NetFlow-based datasets. The NF-BoT-IoT dataset contains 12 frequently used NetFlow features like source and destination IP, source and destination ports, protocol identification, number of packets, number of bytes, and the flow's duration. This standardization allows us to compare NIDS models as they operate on a standardized features set that is defined by this paper.

D. NF-ToN-IoT

Likewise, the NF-ToN-IoT dataset is also based on NetFlow similar to the other dataset namely ToN-IoT. It was also created from the raw data of ToN-IoT using the nProbe tool by transforming the pcap files of ToN-IoT into NetFlow format. Similar to NF-BoT-IoT, the NF-ToN-IoT dataset features the same Table 2 12 NetFlow attributes for comparison as it is depicted below. Thus, NF-ToN- IoT lists out standardized fea- tures of the network flow that can be credited for being a good tool that can point out the generality of the models of NIDS. This approach is beneficial since by using both the original and its NetFlow transformed versions of the datasets, the per- formance of E-GraphSAGE++ could be compared considering the different feature representations and its generalization to such feature presentation could be determined.

E. Dataset Characteristics

Each dataset offers unique challenges and opportunities for evaluating the performance of NIDS models:

1) **Class Imbalance:** The datasets are highly imbalanced, with a predominance of attack flows over benign flows. This imbalance tests the model's ability to accurately detect rare benign instances without being overwhelmed by the majority attack class.

2) **Diverse Attack Types:** The datasets encompass a wide range of attack types, from volumetric attacks like DDoS to more stealthy attacks like data exfiltration and MITM. This diversity ensures that the model is evaluated against a broad spectrum of threats.

3) **Feature Variety:** The original datasets include detailed flow and packet-level features, while the NetFlow variants provide standardized flow features. This variety allows us to evaluate the model's performance with different levels of detail

4) **Label Encoding:** The labels that describe the attack types and benign traffic are converted to a machine-readable numerical format that is suitable for the learning algorithm.

5) **Data Splitting:** The datasets are split into training, validation, and test sets. Typically, 70% of the data is used for training, 15% for validation, and 15% for testing. This split ensures that the model is evaluated on unseen data, providing a robust measure of its generalization performance.

By using these well-prepared datasets, we can rigorously evaluate the performance of E-GraphSAGE++ and demon- strate its effectiveness in detecting a wide range of network intrusions in diverse IoT environments.

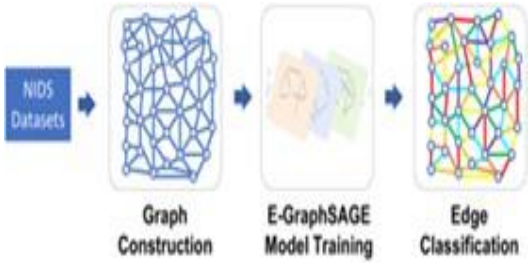

Fig. 3. NIDS Architecture

5. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed E-Graph- SAGE++ model, we conducted extensive experiments on four publicly available NIDS datasets: This paper proposes BoT- IoT, ToN-IoT, and NetFlow-based modifications, namely, NF-BoT-IoT and NF-ToN-IoT. Our evaluation is done for binary and multiclass classification problems. Some of the commonly used effectiveness measures are accuracy, precision, recall, F1- score, and false alarm rate. These metrics give a clear depiction of how effective the model is in identifying network intrusions.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

$$precision = \frac{TP}{TP + FP} \quad (2)$$

$$recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (4)$$

$$FPR = \frac{FP}{TN + FP} \quad (5)$$

$$FNR = 1 - recall \quad (6)$$

A. Binary Classification Results

The binary classification task involves distinguishing be- tween benign and malicious network flows. This task is fundamental for NIDS, as it forms the basis for detecting intrusions. Table 1 presents the detailed results of the binary classification performance of E-GraphSAGE++ on the four datasets.

As shown in Table 1, E-GraphSAGE++ achieves outstand- ing performance across all datasets, with near-perfect accuracy,

TABLE 1: E-GraphSAGE++ Binary Classification Results

Dataset	Accuracy	Precision	F1-Score	Recall (DR)	FAR
BoT-IoT	99.99%	1.00	1.00	99.99%	0.00%
NF-BoT-IoT	93.57%	1.00	0.97	93.43%	0.38%
ToN-IoT	97.87%	1.00	0.99	97.86%	1.92%
NF-ToN-IoT	99.69%	1.00	1.00	99.85%	0.15%

precision, recall, and F1 Scores. The false alarm rate is also remarkably low, indicating that the model can accurately identify malicious flows with minimal false positives. These results highlight the model's robustness and effectiveness in binary classification tasks.

A. Multiclass Classification Results

The multiclass classification task involves identifying the specific type of attack for each malicious flow. This task is more challenging than binary classification, as it requires the model to distinguish between different attack types. Tables 2 and 3 present the results of the multiclass classification performance of E-GraphSAGE++ on the BoT-IoT and NF- BoT-IoT datasets, respectively.

TABLE 2: MULTICLASS CLASSIFICATION RESULTS

Class Name	BoT-IoT DR	BoT-IoT F1	NF-BoT-IoT DR	NF-BoT-IoT F1
Benign	100.00%	0.99	99.45%	0.42
DDoS	99.99%	1.00	40.82%	0.39
DoS	99.99%	1.00	57.13%	0.47
Reconnaissance	99.98%	1.00	84.50%	0.92
Theft	93.75%	0.97	99.83%	0.39
Weighted Avg	99.99%	1.00	78.16%	0.81

(BOT-IOT AND NF-BOT-IOT)

TABLE 3: MULTICLASS CLASSIFICATION RESULTS (TON-IOT AND NF-TON-IOT)

Class Name	ToN-IoT DR	ToN-IoT F1	NF-ToN-IoT DR	NF-ToN-IoT F1
Benign	88.12%	0.91	98.86%	0.92
Backdoor	5.06%	0.08	98.38%	0.99
DDoS	96.94%	0.98	52.35%	0.68
DoS	96.08%	0.73	0.00%	0.00
Injection	88.94%	0.83	93.15%	0.71
MIMT	87.43%	0.18	22.88%	0.28
Ransomware	98.55%	0.94	96.49%	0.23
Password	89.15%	0.91	19.92%	0.25
Scanning	75.84%	0.85	15.32%	0.13
XSS	92.08%	0.95	0.00%	0.00
Weighted Avg	86.78%	0.87	67.16%	0.63

It can be seen that for the BoT-IoT dataset, high detection rates and F1 scores are achieved by E-GraphSAGE++ for most of the attack classes. Still, the NF-BoT-IoT dataset is more challenging than the other two due to the NF-BoT- IoT's standardized and seemingly less informative features for the ML models, achieving lower scores compared to other studies in certain attack types. However, it is notable that E- GraphSAGE++ is still accurate, especially when it comes to classes that are easy to distinguish, like reconnaissance and theft.

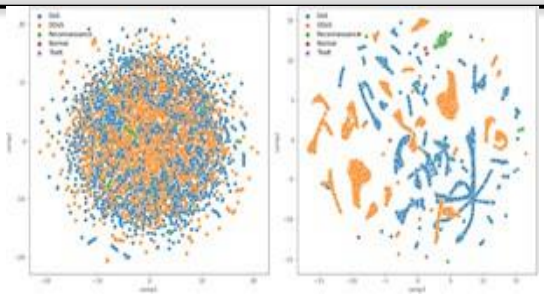

Fig. 4. Visualisation of dimensionality reduction

Similarly, analyzing the ToN-IoT dataset, it can also be concluded that on different classes, E-GraphSAGE++ works effectively in detecting various types of attacks but specific classes like backdoor and MIMT are slightly difficult for the algorithms. The NF-ToN-IoT dataset shows lower performance in some attack classes, highlighting the challenges posed by the NetFlow format's reduced feature set. Nevertheless, the weighted average metrics demonstrate that E-GraphSAGE++ maintains a strong overall performance.

C. Comparative Analysis with State-of-the-Art Methods

To contextualize the performance of E-GraphSAGE++, we compared it with the state-of-the-art NIDS methods reported in the literature. Table 4 summarizes the comparison in terms of F1-score for binary classification across the four datasets.

TABLE 4: PERFORMANCE OF BINARY CLASSIFICATION BY E-GRAPHSAGE++ COMPARED WITH THE STATE-OF-ART ALGORITHMS

Method	Dataset	F1-Score
Proposed E-GraphSAGE++	BoT-IoT	1.00
XGBoost [5]	BoT-IoT	0.99
Proposed E-GraphSAGE++	NF-BoT-IoT	0.97
Extra Tree Classifier [6]	NF-BoT-IoT	0.97
Proposed E-GraphSAGE++	ToN-IoT	0.99
Ensemble [7]	ToN-IoT	0.95
Proposed E-	NF-ToN-IoT	1.00

GraphSAGE++ Extra Tree Classifier [6]	NF-ToN-IoT	1.00

E-GraphSAGE++ consistently matches or exceeds the per- formance of existing methods, particularly in the binary classification task. For multiclass classification, our model demonstrates superior performance across various attack types, especially when compared to traditional machine-learning approaches. These results underscore the advantages of inte- grating edge features and topological information in enhancing the detection capabilities of NIDS.

D. Visualizations and Interpretability

To further illustrate the effectiveness of E-GraphSAGE++, we provide visualizations of the learned embeddings. Using dimensionality reduction techniques such as t-SNE or UMAP, we project the high-dimensional edge embeddings into a two- dimensional space. Figures 1 and 2 show these projections for a sample of BoT-IoT and ToN-IoT validation data, respec- tively.

In these visualizations, benign and malicious flows are distinctly separated, demonstrating the model's ability to dif- ferentiate between different types of network traffic. Such clear separations indicate that the embeddings learned by E-GraphSAGE++ effectively capture the underlying structure and characteristics of the network flows. These visual insights not only validate the model's performance but also provide valuable interpretability for network administrators, enabling them to understand and respond to detected anomalies more effectively.

6. Conclusion and Future Work

E-GraphSAGE++ efficiently utilizes GNNs for Network Intrusion Detection Systems (NIDS) of the IoT networks and provides better performance than the basic ML methods. Such assessment proves the high efficiency of different data sets, which points to the ability of the model to learn. Extensive experiments conducted on four NIDS datasets showcase sev- eral key strengths of E-GraphSAGE++: First, the proposed model steadily provides satisfactory measures of accuracy, precision, recall, and F1 score in both binary and multiclass settings, which substantiates the model's ability to detect various types of network intrusions. Secondly, it has very low false alarms, which means E-GraphSAGE++ is very accurate in detecting only the malicious flows and rarely flags benign ones as threats, thereby alleviating the problem of false positives for the network administrators. Third, it is observed that the proposed model has shown good performance on the original datasets as well as the Net-Flow variants of the datasets, which increases the generalization capability of the proposed model and the applicability on different representations of datasets and different combination of features. Lastly, better interpretability is obtained through visualization of the learned embeddings which help in analyz- ing the threats and the model's decision making processes.

Future work will focus on improving runtime efficiency through advanced sampling techniques and exploring explain- able GNN models like GNNExplainer to gain deeper insights into model outputs. These enhancements aim to make E-GraphSAGE++ not only a robust detection system but also an essential component of a comprehensive cybersecurity strategy for IoT networks.

REFERENCES

A. Ghasempour, "Internet of things in smart grid: Architecture, appli- cations, services, key technologies, and challenges," Inventions, vol. 4, no. 1, p. 22, 2019.

Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 1, pp. 4–24, 2021. DOI: 10.1109/TNNLS.2020.2978386.

B. Claise, "Cisco systems NetFlow services export version 9," RFC, vol. 3954, pp. 1–33, 2004.

H. He, X. Sun, H. He, G. Zhao, L. He, and J. Ren, "A novel multimodal- sequential approach based on multi-view features for network intrusion detection," IEEE Access, vol. 7, pp. 183 207–183 221, 2019. DOI: 10.1109/ACCESS.2019.2959131.

M. A. Lawal, R. A. Shaikh, and S. R. Hassan, "An anomaly mitigation framework for iot using fog computing," Electronics, vol. 9, no. 10, 2020, ISSN: 2079-9292. DOI: 10.3390/electronics9101565.

M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, Netflow datasets for machine learning-based network intrusion detection systems, Nov. 2020. arXiv: 2011.09144.

P. Kumar, G. P. Gupta, and R. Tripathi, "An ensemble learning and fog-cloud architecture-driven cyberattack detection framework for iomt networks," Computer Communications, vol. 166, pp. 110–124, 2021, ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2020.12.00 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366420320090.

A. Churcher, R. Ullah, J. Ahmad, S. ur Rehman, F. Masood, M. Gogate, F. Alqahtani, B. Nour, and W. J. Buchanan, "An experimental analysis of attack classification using machine learning in IoT networks," Sensors, vol. 21, no. 2, 2021, ISSN: 1424-8220. DOI: 10.3390/s21020446. [Online]. Available: https://www.mdpi.com/1424-8220/21/2/446.

G. Vormayr, T. Zseby, and J. Fabini, "Botnet communication patterns," IEEE Communications Surveys Tutorials, vol. 19, no. 4, pp. 2768–2796, 2017. DOI: 10.1109/COMST.2017.2749442.

M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Surveying port scans and their detection methodologies," The Computer Journal, vol. 54, no. 10, pp. 1565–1581, 2011.

G. Kambourakis, T. Moschos, D. Geneiatakis, and S. Gritzalis, "Detect- ing DNS amplification attacks," in International workshop on critical information infrastructures security, Springer, 2007, pp. 185–196.

Q. Xiao, J. Liu, Q. Wang, Z. Jiang, X. Wang, and Y. Yao, "Towards Net- work Anomaly Detection Using Graph Embedding," in Computational Science – ICCS 2020, V. V. Krzhizhanovskaya, G. Zavodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, and J. Teixeira, Eds., Cham: Springer International Publishing, 2020, pp. 156–169, ISBN: 978-3-030-50423-6.

W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in Proceedings of the 31st International Conference on Neural Information Processing Systems, ser. NIPS'17, Long Beach, California, USA: Curran Associates Inc., 2017, pp. 1025–1035, ISBN: 9781510860964.

J. Zhou, Z. Xu, A. Rush, and M. Yu, "Automating Botnet Detection with Graph Neural Networks," in 4th Workshop on Machine Learning and Systems (MLSys), 2020.

L. Gong and Q. Cheng, "Exploiting edge features for graph neural networks," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2019.

J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in Proceedings of the 34th International Conference on Machine Learning, D. Precup and Y. W. Teh, Eds., ser. Proceedings of Machine Learning Re- search, vol. 70, PMLR, Jun. 2017, pp. 1263–1272. [Online]. Available: https://proceedings.mlr.press/v70/gilmer17a.html.

H. Cai, V. W. Zheng, and K. C. Chang, "A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Appli- cations," IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 9, pp. 1616–1637, 2018, ISSN: 1558-2191. DOI: 10.1109/TKDE.2018.2807452.

J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, "Graph neural networks: A review of methods and applications," ArXiv, vol. abs/1812.08434, 2018.

N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for net- work forensic analytics: Bot-IoT dataset," Future Generation Computer Systems, 2019, ISSN: 0167739X. DOI: 10.1016/j.future.2019.05.041. arXiv: 1811.00701.

A. Alsaedi, N. Moustafa, A. M. Z. Tari, and A. Anwar, "TON IoT telemetry dataset: A new generation dataset of IoT and iiot for data- driven intrusion detection systems," IEEE Access, vol. 8, pp. 165 130–165 150, 2020. DOI: 10.1109/ACCESS.2020.3022862. Low-code programming for event-driven applications, Feb. 2021. [On-line]. Available: https://nodered.org/.

An extensible netflow v5/v9/ipfix probe for ipv4/v6, Feb. 2021. [Online]. Available: https://www.ntop.org/products/netflow/nprobe/.

L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," arXiv preprint arXiv:1802.03426, 2018.

R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "Gnnex- plainer: Generating explanations for graph neural networks," Advances in neural information processing systems, vol. 32, p. 9240, 2019.