# Enhanced GraphSAGE for Multi-Class Intrusion Detection

Hong-Dang Le[1], Minho Park[2]
[1]Department of Information Communication Convergence Technology,
Soongsil University, Seoul 156-743, South Korea
[2]School of Electronic Engineering, Soongsil University, Seoul 156-743, South Korea
lehongdang.vnuhc@gmail.com, mhp@ssu.ac.kr

***Abstract*—Nowadays, the number of devices connecting to the Internet is increasing significantly, expanding the size of networks. Along with this development, cyberattacks are rising, targeting sensitive information or even causing critical infrastructure disruption. Hence, it is crucial to detect and thwart attacks before their execution, presenting a significant challenge in protecting businesses from latent threats. In response, various methods have been introduced to enhance network security, with Network Intrusion Detection Systems (NIDS) emerging as a promising solution to monitor traffic and flow within the network. Many years ago, the Graph Neural Network (GNN) was proposed as an effective Deep Learning (DL) algorithm for application in NIDS, demonstrating the ability to capture complex structural data. While NIDS aims to detect attacks based on traffic and flows, traditional GNN studies often concentrate on node features for node classification tasks, without considering edge features that represent the flow information in the network. To overcome this limitation, we propose a method to capture edge features and leverage them to enhance GraphSAGE, a variant of GNN, for the edge classification task. In this paper, we use a two-layer GraphSAGE network to extract edge features. Finally, we use the CICIDS2017 dataset to evaluate the performance of the proposed method. The experimental results show that our proposed model can improve the performance in the detection process of NIDS.**

***Index Terms*—Network Intrusion Detection System (NIDS), Graph Neural Network (GNN), Machine Learning, Edge Classification, Intrusion Detection**

## I. Introduction

Cyberattacks are not only increasing in frequency but also growing in complexity. The challenge for businesses is to effectively mitigate these attacks and prevent information loss. Detecting attacks before execution has become a crucial task. Initially, Network Intrusion Detection Systems (NIDS) were introduced as a promising approach to monitor network traffic and flows, offering two types: signature-based and anomaly-based. Signature-based NIDS [1] can identify known attacks based on predefined rules in its database. However, it has limitations, such as the inability to recognize unknown attacks or different variants of known attacks that do not match any signature in the rules. Additionally, the high false-positive rate in signature-based NIDS makes it inefficient in detection.

Currently, deep learning (DL) has been successfully applied in many fields, such as image processing [2], storage systems [3], speech processing [4], and cybersecurity [5]. In the realm of NIDS, numerous deep learning methods have been suggested for detecting anomalies, including convolutional neural networks (CNN) [6], recurrent neural networks (RNN) [7], and conventional multi-layer perceptron (MLP) [8]. While these methods have demonstrated impressive performance, a limitation exists in that these approaches were trained on a flat data structure (data in the form of grid or vector), which fails to capture complex structural data related to flow data in a network. Besides, there is a correlation between flows; nonetheless, prior methods analyze these flows separately. This could result in an information deficiency issue during training [9].

To address this issue, graph neural network (GNN) is emerged as a promising solution. Since network intrusion detection typically operates on flow-based network data, this flow data can be depicted in a graph format, where the nodes represent flow endpoints and the edges are mapped with network traffic flows (Fig. 1). GNN possesses the capability to capture unstructured patterns as well as the correlation between flows as it can preserve the network topology when transformed into a graph.

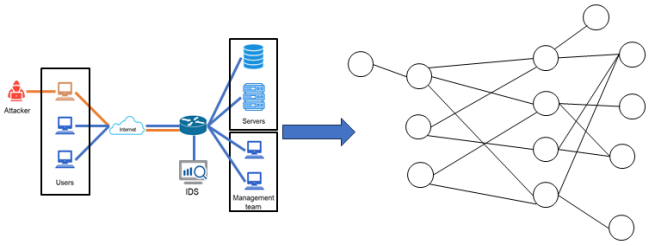


Fig. 1: Graph transformation.

The goal of NIDS is to identify and detect attacks on traffic and flows. Moreover, relying on information from NIDS benchmark network datasets [10], which offer more information as edge features rather than node features, enables effective edge classification. Consequently, this leads to the challenge of edge classification on flow datasets, where crucial information is provided through edges. However, traditional GNN primarily focuses on node features for node classification tasks without considering edge features [11] [12].

In [11], they proposed using a graph convolutional network (GCN) and node features to detect botnet through node classification task. While their proposal reached a positive result, excelling in binary detection by considering whether there

was an attack or not, it falls short in handling various types of attacks that may occur in an intrusion network, requiring detection based on a multi-class approach. In addition, their graph representation does not include any flow information from the dataset.

In [13], they suggested incorporating edge features into their graph aimed to enhance node representation to achieve improved performance. Nevertheless, these edge features were not directly employed for network intrusion detection in edge classification, which is the primary objective of NIDS.

In [14], the author introduced the E-GraphSAGE approach for IoT intrusion detection. The model incorporates edge features in its input and necessitates both sampling and aggregating edge information for effective operation. However, in their graph, all-one vectors (V =1,1,..,1) are embedded as node features which do not include any flow information. The inclusion of all-one vectors not only has the potential to lead the model to incorrect representations during aggregation but also results in an increase in the dimensions of both node and edge representations due to the concatenation function. Furthermore, for aggregating information from neighbors, the authors only utilize two layers of multi-layer perceptron (MLP), including the input and output which makes the aggregation step too simple to exploit. These limitations could result in inefficient and high resource costs while training.

To overcome this issue, we propose a method which leverages edge features to enhance GraphSAGE [15]. In our proposal, important information related to flow data is embedded as edge features such as flow duration, packets per second, length of flow, and so on. Less important information related to endpoints, such as IP addresses and ports, is embedded as node features. This design allows our model to avoid using an all-one vector and preserves the dimension of flow data during aggregation. In addition, we add two layers of GraphSAGE for the implementation step. The experimental results show that the accuracy of the proposed model achieved 96% for the CICIDS2017 dataset

We summarize our contributions as follows:

- We propose an approach to rearrange features embedded in nodes and edges when creating a graph.
- We propose the GraphSAGE model, which is a novel GNN-based model, and leverage edge features to effectively improve NIDS in multi-class attack detection
- The proposed model is applied to the CICIDS2017 benchmark datasets, and we provide simulation results as proof of the effectiveness of the proposed model.

The remainder of this paper is organized as follows. Section II presents our proposed method. The experimental results are discussed in Section III. Finally, we provide conclusions in Section IV.

## II. Proposed Method

In previous studies, the authors successfully applied GNN to various fields. However, the way they extract features for nodes and edges in the graph is not accurate, making the model inefficient in classifying flow for multi-class detection. In our paper, we propose a method to leverage edge features from the traffic flows to enhance our GraphSAGE model. We introduce our conceptual model, depicted in Fig. 2, which contains four steps: graph creation, nodes and edges embedding graph data, and classification.

In the graph creation step, the CICIDS2017 dataset includes numerous flows, representing sets of packets transmitted or received through communication between endpoints. We concatenate the Source IP with the Source Port and the Destination IP with the Destination Port. Then, the graph is constructed where nodes symbolize the IP addresses and ports, and edges represent the flows between two IP addresses. The flow information is embedded as the edge features, while less important information such as IP addresses and ports is embedded as node features.

In the nodes and edges representation step, we employ Message Passing Neural Networks (MPNN). This process involves two key functions: aggregation and update. Within each embedding context, the aggregation function gathers information from neighboring nodes and edges. In our approach, we utilize the CONCATENATE aggregation function to consolidate data for each embedding layer, employing three multi-perception layers (MLP) including input, hidden, and output per embedding step to exploit deeper information from the neighbors. Following each iteration, a linear activation function is applied to each node to produce its updated version.

In the final iteration, we update the information of all nodes and edges in the graph, serving as the input for the model training step.

Finally, we construct two GraphSAGE layers to train the model, using ReLU as a non-linear activation function for the classifier to categorize the type of each edge in the model.

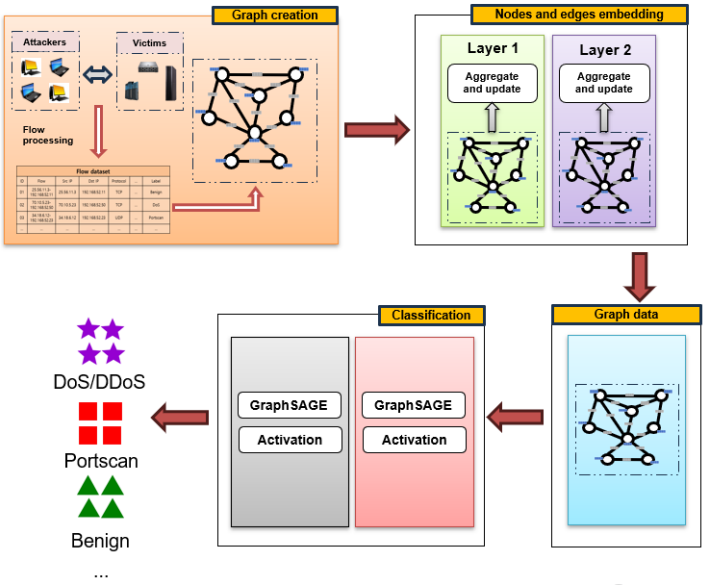


Fig. 2: Model Framework

As our goal in training the model is multi-class detection, the output is designed to classify various types of attacks,

including DoS, DDoS, PortScan, and so on. The results will be explained and evaluated in the next section.

## III. Experiments and evaluation

We used the model in Fig. 2 to simulate the network intrusion datasets as the CICIDS 2017. The CICIDS 2017 contains benign and the latest prevalent attacks, with each flow incorporating 84 features. In the model, we split the dataset into 60% for the training set and 40% for the testing set. By training 200 epochs with 2 layers of GraphSAGE, Adam optimizer, and the binary cross-entropy loss function, Fig. 3 shows the accuracy of the training step in our model converges after 150 epochs, reaching a peak of 96.68%. In comparison, our model outperforms recent works of GraphSAGE for edge classification in NIDS.

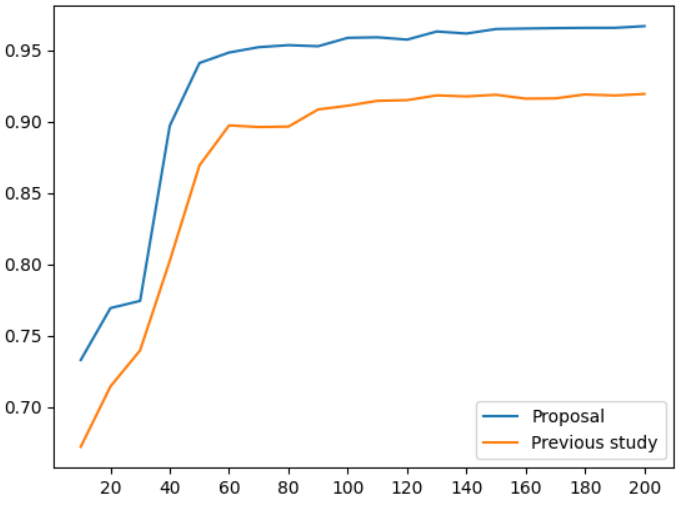


Fig. 3: Training accuracy of multi-class classification

Table I displays the results of the comparison between our proposal and previous studies of GraphSAGE for edge classification on the test set. As illustrated, although our model takes more time to test due to an additional MLP layer aimed at extracting deeper information from neighbors, we have achieved higher accuracy performance compared to previous studies.

TABLE I
Testing result of multi-class classification

| | Test accuracy (%) | Testing time (s) |
|---|---|---|
| Proposal | 93.32 | 0.327 |
| Previous study | 87.23 | 0.278 |

## IV. Conclusion

We have proposed a method to rearrange features embedded in nodes and edges in GraphSAGE when transforming flow data into a graph. In the model, the graph is created with nodes representing both IP addresses while the edges signify the connections or flows between two IP addresses. To enhance the GraphSAGE model, we embed flow information as edge features and endpoint information, which is less critical in flow classification based on NIDS, into node features. Additionally, we use three layers of MLP to aggregate deeper information from neighbors. The results of the model achieved high accuracy on both the training set and the test set.

## Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2023R1A2C1005461).

## References

[1] W. Lee, S. J. Stolfo and K. W. Mok, “A data mining framework for building intrusion detection models,” in Proceedings of the 1999 IEEE Symposium on Security and Privacy (Cat. No.99CB36344), Oakland, CA, USA, 1999, pp. 120-132

[2] M. -T. Duong, S. Lee and M. -C. Hong, “DMT-Net: Deep Multiple Networks for Low-Light Image Enhancement Based on Retinex Model,” in IEEE Access, vol. 11, pp. 132147-132161, 2023.

[3] T. A. Nguyen and J. Lee, “A Nonlinear Convolutional Neural Network-Based Equalizer for Holographic Data Storage Systems,” Applied Sciences, vol. 13, pp. 13029, 2023.

[4] L. Nguyen-Vu, T. -P. Doan, M. Bui, K. Hong and S. Jung, “On the Defense of Spoofing Countermeasures Against Adversarial Attacks,” IEEE Access, vol. 11, pp. 94563-94574, 2023.

[5] C. -N. Nhu and M. Park, “Dynamic Network Slice Scaling Assisted by Attention-Based Prediction in 5G Core Network,” IEEE Access, vol. 10, pp. 72955-72972, 2022.

[6] S. Haider et al., “A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks,” IEEE Access, vol. 8, pp. 53972-53983, 2020.

[7] S. Sivamohan, S. S. Sridhar and S. Krishnaveni, “An Effective Recurrent Neural Network (RNN) based Intrusion Detection via Bi-directional Long Short-Term Memory,” in 2021 International Conference on Intelligent Technologies (CONIT), Hubli, India, 2021, pp. 1-5.

[8] H. Liu and B. Lang, “Machine learning and deep learning methods for intrusion detection systems: A survey,” Applied Sciences, vol. 9, p. 4396, Oct. 2019.

[9] T. Bilot, N. E. Madhoun, K. A. Agha and A. Zouaoui, “Graph Neural Networks for Intrusion Detection: A Survey,” IEEE Access, vol. 11, pp. 49114-49139, 2023

[10] I. Sharafaldin, A. H. Lashkari and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in ICISSp, 2018, pp. 108-116.

[11] J. Zhou, Z. Xu, A. M. Rush and M. Yu, “Automating Botnet Detection With Graph Neural Networks,” in 4th Workshop on Machine Learning and Systems (MLSys), 2020

[12] J. Suárez-Varela et al., “Graph Neural Networks for Communication Networks: Context, Use Cases and Opportunities,” IEEE Network, vol. 37, pp. 146-153, May/June 2023

[13] L. Gong and Q. Cheng, “Exploiting edge features in graph neural networks”, in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 9211-9219.

[14] W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher and M. Portmann, “E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT,” NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2022, pp. 1-9.

[15] W. L. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in Proc. 31st Int. Conf. Neural Inf.Process. Syst., 2017, pp. 1024–1034.