Graph Density Mutation Study to Improve Cyber-Physical Systems Intrusion Detection with Heterogeneous GNN and Express Edges

Hongwei Li
Department of Electrical and Computer Engineering
Villanova University
Email: hli8@villanova.edu

Danai Chasaki
Department of Electrical and Computer Engineering
Villanova University
Email: danai.chasaki@villanova.edu

***Abstract*—The growing reliance on Cyber-Physical Systems (CPS) in critical infrastructure heightens the risk of cyber threats impacting our daily lives. One recent example is the "FrostyGoop" malware, which targeted district heating systems, leaving citizens in 600 Ukrainian apartment buildings exposed to sub-zero temperatures in January 2024. This study advances CPS threat detection by examining the relationship between graph density and the performance of Graph Neural Networks (GNNs) through the addition of virtual Express Edges to a heterogeneous graph representation of network flow data. Our findings indicate that the Heterogeneous GNN method with Express Edges outperforms state-of-the-art detection techniques, particularly in real-world scenarios characterized by sparse network flow graphs. This research provides valuable insights for cybersecurity practitioners seeking to deploy advanced machine learning methods to protect CPS against threats such as botnets, reconnaissance, denial of service, information theft, ransomware, backdoor cracking, and Man-in-the-Middle attacks.**

***Index Terms*—Cyber-Physical Systems, machine learning, intrusion detection, graph, neural networks, GNN.**

I. Introduction

Imagine in the coldest month of winter, the central heating system of your entire city district of 600 apartment buildings were taken over by adversaries from the cyber space. The attackers would first cripple the activity monitoring capabilities of your heating system controllers via a forced firmware downgrade. Then, they would compromise the temperature monitoring function of your controllers. This would result in hundreds of thousands of your citizens unable to keep themselves warm in sub-zero temperatures for days.

The scenario described above was what a Ukrainian community experienced in January 2024. The malware was later identified as "FrostyGoop" by Dragos, Inc. [1] Dragos claims that FrostyGoop is the first Cyber-Physical System (CPS) malware that uses Modbus [2] communications to achieve an impact on Operational Technology! To further highlight the threat facing the broader community, a survey conducted by the authors in 2023 using the Shodan platform (www.shodan.io) revealed the presence of over 300,000 MODBUS devices accessible on the public internet [3].

Building on existing machine learning approaches for detecting CPS attacks [4], Graph Neural Networks (GNNs) [5] have demonstrated promising results by combining the underlying system topology (graph relationship structure) with other relevant attributes of CPS components and their interactions, including lateral movement attack patterns [6], [7], [3].

In this study, we introduce a novel approach using Flow-based Heterogeneous Graph Neural Networks (HGNNs) as a superior NIDS technique. By applying this method to a range of CPS cyber attack datasets, we demonstrate its effectiveness against various attack tactics and its resilience to different degrees of data imbalance.

Additionally, we assess this novel technique across varying levels of graph density mutations to illurstrate when Express Edges are most effective in further enhancing HGNN model performance.

The remainder of this paper is organized as follows: Section II reviews related work. Section III outlines the proposed methodology. Section IV describes the experimental design and data. Section V presents the experimental results and the analysis. Finally, Section VI provides the conclusion and discusses future work.

II. Related Work

Extensive research has been conducted on utilizing tabular data and machine learning techniques for Network Intrusion Detection Systems (NIDS) to detect CPS attacks [4]. A selection of published results for the Ton-IoT dataset [8] is presented in Table III, and for the NF-BoT-IoT dataset [9], [10] in Table IV.

Lo et al. [11] generated significant interest in the CPS NIDS research community by leveraging Graph Neural Network (GNN) algorithms. Their approach involved constructing homogeneous graphs where network device IP addresses were represented as nodes and network flows as edges. Typical tabular flow features were associated with the graph edges, as illustrated in Figure 1a. Using this E-GraphSAGE modification of the GraphSAGE algorithm [12], Lo et al. reported generally improved performance over ensemble tree-based machine learning models.

A different method for constructing the network flow graph was proposed by Chang and Branco [13] and Friji et al. [14].

979-8-3315-2096-0/25/$31.00 ©2025 IEEE

To leverage existing techniques in node classification, their approach involved representing the graph as a homogeneous line graph, where nodes correspond to network flows and edges represent IP addresses, as shown in Figure 1b.

The ability of GNNs to leverage both spatial features and typical flow features in resisting adversarial attacks was demonstrated by Pujol-Perich et al. [15]. Their experiments showed that GNNs produced more robust results against simulated adversarial conditions, such as variations in packet size or inter-arrival times, when compared to ensemble tree-based models and traditional Neural Networks. For graph data modeling, they represented both network devices and network flows as nodes within a heterogeneous graph, as illustrated in Figure 1c.

Wang et al. [16] utilized a Bi-LSTM with residual connections in their model to capture both long-term and short-term dependencies in network traffic temporal sequences. This approach enabled the effective extraction of temporal features, thereby complementing the graph-based spatial features and enhancing the model's overall detection capabilities. Homogeneous graphs were used to model the data.

In real-world CPS environments, networks consist of a diverse array of device types and roles, including servers, mobile devices, IoT devices, and CPS devices. Communication patterns, especially for CPS devices in critical infrastructures such as power or transportation systems, exhibit significant variability. This diversity renders homogeneous graphs for data modeling less expressive due to their inability to represent different node types and edge relations. Consequently, there is a need for a heterogeneous graph approach that can more accurately capture the complexities of real-world networks, providing a robust foundation for GNN-based graph representation learning.

III. Proposed Methodology

A. Heterogeneous Graph Data Modeling with Express Edges

We propose a heterogeneous graph data model to effectively capture the complexity of real-world CPS networks, which involve various device types with distinct network behaviors that homogeneous graphs struggle to represent accurately due to their inability to differentiate between multiple node and edge types.

A heterogeneous graph, denoted as $G = (V, E)$, consists of nodes V and edges E, with node types A and edge types R. Each node $v \in V$ has a mapping $\phi(v) : V \rightarrow A$, and each edge $e \in E$ has a mapping $\phi(e) : E \rightarrow R$. A heterogeneous graph reduces to a homogeneous form when $|A| = 1$ and $|R| = 1$.

In CPS networks, nodes can represent device roles such as MTUs, RTUs, HMIs, and more. Given that specific device roles are unknown in our data, we use Source (src) and Destination (dst) IP addresses as proxies. Network flows are included as node types, with undirected edges connecting src/dst to flow nodes (Figure 1d).

To improve message passing in the Heterogeneous Graph Neural Network (HGNN), we introduce **Express Edges** that directly connect src and dst nodes (Figure 1d). The effectiveness of **Express Edges** is demonstrated in Section V.

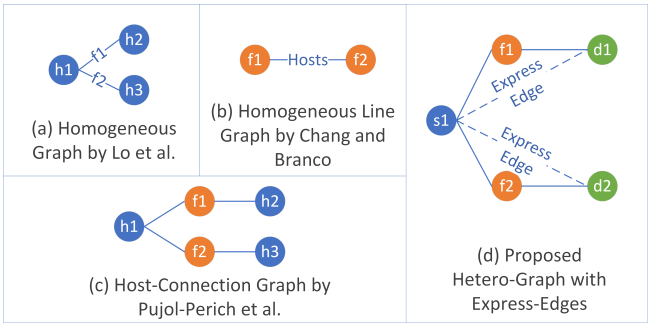

Fig. 1: Different graph data modeling techniques: (a) homogeneous graph used by Lo et al. in [11]; (b) homogeneous line-graph used by Chang and Branco [13]; (c) heterogeneous graph used by Pujol-Perich et al. [15]; (d) Proposed heterogeneous graph with Express Edges.

B. Graph Neural Network Algorithms

The introduction of Graph Convolutional Networks (GCN) by Kipf and Welling [5] was a key development in Graph Neural Networks (GNNs), applying spectral convolution methods to non-Euclidean graph data, inspired by Convolutional Neural Networks (CNNs).

GNNs generally operate under a neural message-passing framework where nodes exchange vector messages and update their states using neural networks, as depicted in Figure 2 adapted from [17]. Given a graph $G = (V, E)$, where V are nodes and E are edges, and a set of node features $X \in \mathbb{R}^{d \times |V|}$, the message-passing operation from the hidden state h_u^k at layer k for node u to the next layer h_u^{k+1} is formulated as:

$$h_u^{k+1} = \mathbf{UPDATE}^k\left(h_u^k, \mathbf{AGG}^k(h_v^k, v \in N(u))\right), \quad (1)$$

$$= \mathbf{UPDATE}^k\left(h_u^k, \mathbf{message}_{N(u)}^k\right), \quad (2)$$

where $\mathbf{UPDATE}$ and $\mathbf{AGG}$ are differentiable functions, and $\mathbf{message}_{N(u)}$ represents the aggregated messages from node u's neighbors.

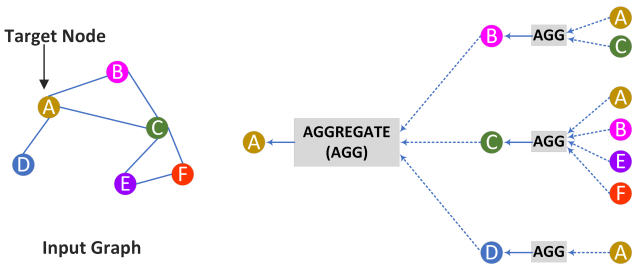

Fig. 2: GNN message passing framework [17].

For the GCN, Kipf and Welling applied symmetric normalization with self-loops in the aggregation function:

$$h_u^{k+1} = \sigma\left(\mathbf{W}^k \sum_{v \in N(u) \cup \{u\}} \frac{h_v^k}{\sqrt{|N(u)||N(v)|}}\right). \quad (3)$$

To efficiently train on large graphs, Hamilton et al. [12] proposed the GraphSAGE algorithm, which incorporates node neighbor sampling $N_s(u)$ and modifies the **UPDATE** function to include a concatenation step:

$$h_u^{k+1} = \sigma\left(W^k \cdot \mathbf{CONCAT}(h_u^k, \mathbf{message}_{N_s(u)}^k)\right). \quad (4)$$

C. Extensions of Graph Neural Networks to Heterogeneous Graphs (HGNN)

To extend the Graph Convolutional Network (GCN) framework of Kipf and Welling [5] to heterogeneous graphs, Schlichtkrull et al. [18] incorporated edge type relationships R into the GCN equation. This adaptation, known as the Relational GCN (RGCN), accounts for multiple types of edges, resulting in the following formulation:

$$h_u^{k+1} = \sigma\left(\sum_{r \in R} \sum_{v \in N_u^r} \frac{1}{C_{u,r}} W_r^k h_v^k + W_0^k h_u^k\right), \quad (5)$$

where N_u^r represents the set of neighbors of node u under relation $r \in R$, and $C_{u,r}$ is a normalization constant, which can either be pre-defined (e.g., $C_{u,r} = |N_u^r|$) or learned during training.

Similarly, the GraphSAGE algorithm [12] has been extended to heterogeneous graphs, known as Rsage, by accounting for edge type relationships in the GraphSAGE Equation (4).

Graph Attention Networks (GAT) [19] introduced an attention mechanism initially developed for Natural Language Processing [20], enabling the model to learn unique attention coefficients for each neighboring node, thus enhancing information aggregation. The attention coefficient α_{uv} between nodes u and v is defined as:

$$\alpha_{uv} = \frac{\exp\left(\sigma\left(a^T[Wh_u \parallel Wh_v]\right)\right)}{\sum_{k \in \mathcal{N}_u} \exp\left(\sigma\left(a^T[Wh_u \parallel Wh_k]\right)\right)}, \quad (6)$$

where a is a learnable weight vector, W is a shared weight matrix, σ is a non-linear activation function, and $\parallel$ denotes concatenation. The updated node representation h'_u is computed by aggregating the embeddings of its neighbors, weighted by the attention coefficients:

$$h'_u = \sigma\left(\sum_{v \in N(u)} \alpha_{uv} W h_v\right). \quad (7)$$

Following the approach used in RGCN (Equation (5)), the extension to include edge type relations results in the Relational GAT (RGAT) model.

Wang et al. [21] further extended this concept by proposing the Heterogeneous Graph Attention Network (HAN), which adds an additional attention layer to learn different weights across various meta-paths. Meta-paths represent human-curated sequences of meaningful node types within the graph, such as "Paper-Author-Paper" or "Paper-Term-Paper" in an academic bibliographic network.

IV. Evaluation Data and Experimental Design

A. CPS Evaluation Data: NF-BoT-IoT dataset

The NF-BoT-IoT dataset [9], [22] contains a rich simulation of botnet attacks targeting cyber-physical systems, specifically simulated Internet of Things (IoT) devices such as weather stations, smart fridges, smart lighting, garage doors, and thermostats. In addition to benign traffic, the dataset includes various simulated cyber-attacks, such as reconnaissance (including service scanning and OS fingerprinting), denial of service (DoS), distributed denial of service (DDoS), and information theft attacks (e.g., keylogging and data theft).

Due to the limitations of the simulation lab environment, the NF-BoT-IoT dataset significantly differs from real-world scenarios typically with much bigger number of devices; it includes only a small number of source IP addresses. Among the 15 distinct source IP addresses in the dataset, only 8 are associated with attack traffic, see Figure 3. Furthermore, the dataset is highly imbalanced, with only 2.3% of the traffic flows being normal and 97.7% representing attack traffic. A detailed analysis of the dataset is presented in Figure 4.

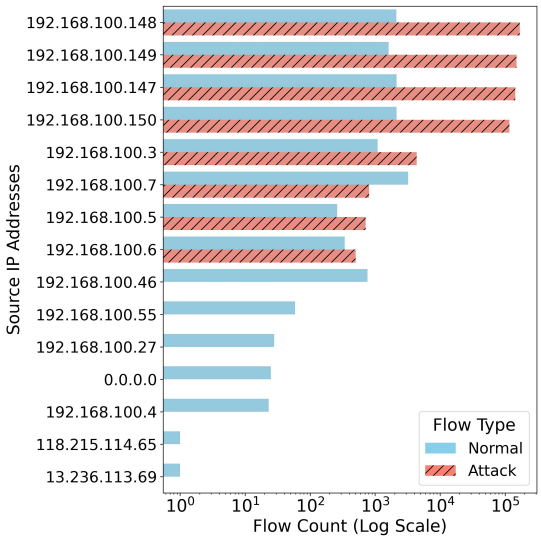

Fig. 3: Distribution of normal and attack flows by source IP addresses.

The ToN-IoT dataset, developed by Alsaedi et al. [8], is also widely used in studies focused on Graph Neural Networks (GNN) for Network Intrusion Detection Systems (NIDS). The ToN-IoT dataset captures a comprehensive representation of CPS and Internet of Things (IoT) devices within the Cyber Range and IoT Labs at UNSW Canberra, Australia.

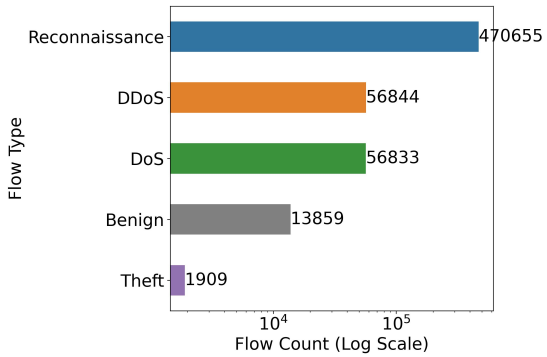

Fig. 4: Distribution of network flow types.

It comprises both real devices, such as smartphones and smart TVs, and simulated CPS devices, including fridges, GPS units, and thermostat sensors. The dataset includes various types of cyber-attacks, such as scanning, Denial of Service (DoS), Distributed Denial of Service (DDoS), ransomware, backdoor attacks, data injection, cross-site scripting (XSS), password cracking, and Man-in-The-Middle (MITM) attacks. In the Train_Test version, the dataset contains 161,043 attack flows alongside 300,000 normal flows, providing a more balanced basis for assessing intrusion detection performance than the BoT-IoT dataest.

B. Additional Evaluation Data

To evaluate the broader applicability of the proposed methodology, we extended our analysis to include additional NIDS datasets: CIC-IDS2017 [23] and CIC-Darknet [24]. A summary of the key statistics for these datasets is provided in Table I. Most datasets exhibit a typical class imbalance, reflecting real-world scenarios where the majority of data points belong to the "Normal" class. However, the NF-BoT-IoT dataset shows a significant skew towards the "Attack" class, with only 2.3% of the data representing the "Normal" class.

TABLE I: Statistics of Dataset Used in This Study

Dataset	Normal Flows	Attack Flows	Normal:Attack Ratio
ToN-IoT	300,000	161,043	65.1% : 34.9%
CIC-IDS2017	1,657,693	443,121	78.9% : 21.9%
CIC-Darknet	117,219	24,311	82.8% : 17.2%
NF-BoT-IoT	13,859	586,241	2.3% : 97.7%

C. Data Processing

To address the limitations of small lab environments, which typically generate datasets with limited device diversity (as illustrated in Figure 3), a method was employed to enhance device diversity simulation by appending a timestamp (accurate to the second) to the source IP addresses. This approach enables the emulation of a larger number of devices than those actually present in the dataset. For datasets lacking explicit timestamps, such as NF-BoT-IoT [22], the source port is utilized as a substitute, as most new TCP flows select a different port number at the source.

To manage the bursty nature of the simulation data, a stratified temporal data split is applied within each day and for each attack type, maintaining a 70:30 train:test ratio. In cases where datasets do not include explicit timestamps, the chronological order of the data, along with sub-experiment types, is used to achieve the same split.

For feature engineering, common port numbers are treated as categorical features, while uncommon port numbers are binned by 1024 to reduce cardinality. One-hot encoding is used for features with fewer categories, such as protocol and service, whereas binary encoding is applied to categorical features with more than 10 categories. For numerical features, a standard scaler is applied to those with smaller variance, and a Yeo-Johnson transformation is applied to features with variance greater than 10, such as those representing flow duration, total bytes, and the number of packets.

D. Performance Metrics and Hyper-parameters

Due to the imbalanced nature of the dataset, the F1 score is selected as the performance metric for comparisons across all four dataset instead of accuracy alone. The F1 score offers a balanced assessment between predicted and actual positives, regardless of a potentially high number of true negatives (TN) or normal samples. The F1 score is computed using the formula in Equation (8):

$$F1Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{8}$$

To ensure a fair comparison of model results, the hyper-parameters for most GNN algorithms in this study are set as follows: The number of GNN layers is 2, the mini-batch size is 2048, and the hidden dimension is 64 for the ToN-IoT and NF-BoT-IoT datasets, and 128 for the CIC-IDS2017 and CIC-Darknet datasets to accommodate their greater number of features. The models are trained for 2 or 3 epochs, with a dropout ratio of 0.3 and a learning rate of 5e-3.

Each performance data point is an average of five independent runs ($n = 5$), except for the Rsage algorithm, where ($n = 10$) compensates for the inherent randomness in the neighbor sampling process. The primary software libraries used were PyTorch, DGL, and OpenHGNN [25].

V. Results and Discussions

A. Graph Density Mutation Study and performance Lift of Express Edges

In a bipartite graph, the density of the graph $G = (U, V, E)$ is given by:

$$\text{Density} = \frac{|E|}{|U| \cdot |V|}$$

where:

- $|E|$ is the number of edges in the graph.
- $|U|$ is the number of nodes in the first set.
- $|V|$ is the number of nodes in the second set.

Table II presents the densities of the heterogeneous graphs constructed from each dataset. To prevent target leakage, these graphs have been diversified by source IP address, as outlined in Section III, except in cases explicitly noted as non-diversified in the table. The NF-BoT-IoT dataset, in particular, stands out not only for its unusual data imbalance but also for its exceptionally high graph density, nearly 150 times denser than the other datasets, even after source diversification.

To evaluate the impact of the proposed modeling technique across graphs with different densities, we provide the average results of 10 random mutations with varying seeds for the NF-BoT-IoT dataset in Table II and Figure 5, alongside results from the other three datasets. The Express Edge technique demonstrated performance improvements in most of the sparser graphs, including ToN-IoT, CIC-IDS2017, CIC-Darknet, BoT-IoT 10X reduction, and BoT-IoT 30X reduction, with an average performance lift of 2.3% in these cases.

In contrast, at higher graph densities, such as those in the NF-BoT-IoT dataset and the 3X edge reduction scenario, the introduction of Express Edges did not lead to improvement in model performance. Moreover, in scenarios with extreme edge reductions of 100X or more, over 90% of the 64,488 diversified source nodes became isolated, having no connections to the remaining 6,001 edges. This severe sparsity likely explains the minimal impact observed in the 100X and 300X edge reduction scenarios.

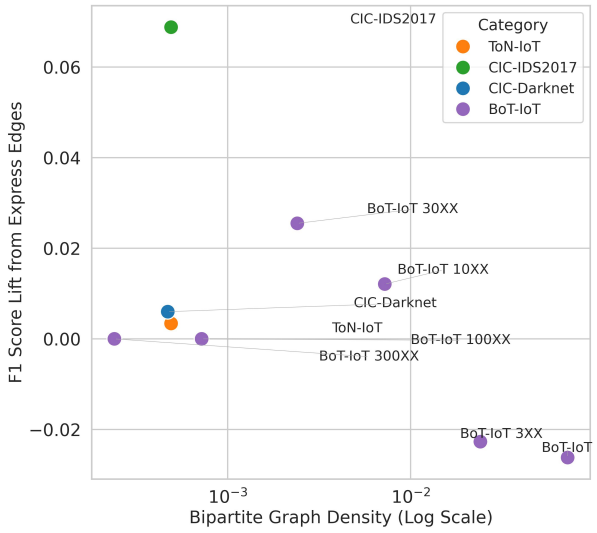

Fig. 5: Performance lifts from Express Edges across multiple datasets. Notations such as "BoT-IoT 30XX" represent average performance lifts of 10 random-seeded 30X edge reduction of NF-BoT-IoT datasets.

Another notable observation from Table II is that there was no single HGNN algorithm that dominates all other tested algorithms. While the RSAGE and RGCN models with the **Express Edge** technique achieved the best results for most of the datasets, other models, such as RGAT and HAN, occasionally outperformed their counterparts in different scenarios.

B. Comparison of Model Results with State-of-the-Art Studies

To facilitate direct comparison with other state-of-the-art (SOTA) studies on machine learning-based NIDS, we report binary classification results. Tables III and IV present findings from recent publications.

For the ToN-IoT dataset, our proposed methods, the RSAGE and RGCN algorithms with the **Express Edge** technique, achieved the highest performance among models with comparable experimental setups. One can observe that GNN-based models outperform other models for this representative dataset, while HGNN models demonstrate superior performance compared to homogeneous graph models.

For the NF-BoT-IoT dataset, two of our methods ranked among the top-performing models, placing within a group tied for 4th to 8th positions. Due to the unique data imbalance, characterized by a predominance of attack data and a very limited number of unique device IP addresses, there are more modeling methods clustered at the top. Notably, 4 of the top 8 results are achieved by GNN-based models.

VI. Conclusion

With the ever-increasing role of Cyber-Physical Systems (CPS) in our daily lives, the likelihood of threats originating from cyberspace affecting physical systems is also rising. Consequently, the impact of cyber threats is becoming more severe, as demonstrated by incidents like the "FrostyGoop" attack on Urkranian power facility. In this study, we expanded upon our previous research on Heterogeneous Graph Neural Networks (HGNN) with Express Edges, providing valuable guidance to cybersecurity researchers on the effective use of this technique for detecting cyber threats.

Using four cybersecurity datasets, NF-BoT-IoT, ToN-IoT, CIC-IDS2017, and CIC-Darknet, we demonstrated that the proposed NIDS detection method is effective against a wide range of cyber threats, including reconnaissance, Denial of Service (DoS), Distributed Denial of Service (DDoS), ransomware, cross-site scripting (XSS), password cracking, Man-in-the-Middle (MITM) attacks, keylogging, data theft, and botnet attacks. Our results also show that the method performs well under various conditions of data imbalance, whether benign or attack traffic predominates.

Additionally, our results demonstrate that the proposed method, leveraging HGNN with Express Edges, outperforms other state-of-the-art detection techniques, particularly in real-world scenarios where network traffic is distributed across a wide variety of source and destination devices, resulting in relatively sparse network flow graphs.

Building on this work, future research will focus on incorporating temporal signals to enhance NIDS capabilities, aiming to improve the detection of both known and unknown threats, thereby providing stronger protection for Cyber-Physical Systems (CPS).

References

[1] M. M. Graham, C. Ahlers, and K. O'Meara, "Impact of FrostyGoop ICS Malware on Connected OT Systems," Dragos, Inc, Tech. Rep.,

TABLE II: Performance Benchmark across Multiple Datasets

Dateset	Winning Algorithm	Graphs without Express Edges	Graphs with Express Edges	Lift from Express Edges	Bipartite Graph Density * 1000
ToN-IoT	RSAGE	0.9752	**0.9786**	0.0034	0.49
CIC-IDS2017	RSAGE	0.8308	**0.8996**	0.0688	0.49
CIC-Darknet	RGAT	0.9367	**0.9427**	0.0060	0.47
NF-BOT-IOT Non-diversified Source IP	RGCN	**0.9769**	**0.9769**	-	310000
NF-BOT-IOT	RGAT	**0.9566**	0.9304	-0.0262	72.14
NF-BOT-IOT 3X Edge Reduction	RGCN	**0.9401**	0.9174	-0.0227	24.05
NF-BOT-IOT 10X Edge Reduction	RGCN	0.9543	**0.9664**	0.0121	7.22
NF-BOT-IOT 30X Edge Reduction	RGCN	0.9513	**0.9767**	0.0255	2.4
NF-BOT-IOT 100X Edge Reduction	HAN	**0.9785**	**0.9785**	-	0.72
NF-BOT-IOT 300X Edge Reduction	RGCN	**0.9743**	**0.9743**	-	0.24

TABLE III: Performance Comparison with State-of-the-Art Studies of ToN-IoT Dataset

ML Algorithm	Study	F1-Score
DNN	Friji et al.[14]	0.3344
XG-Boost	Friji et al.[14]	0.4807
E-GraphSAGE	Lo et al. [11] Friji et al. [14]	0.88
Dual-Relation GNN	Friji et al. [14] Pujol-Perich et al. [15]	0.902
GNN with Residuals	Friji et al.[14]	0.937
RGCN with Express Edges	This study	**0.9749**
RSAGE with Express Edges	This study	**0.9786**

TABLE IV: Performance Comparison with State-of-the-Art Studies of NF-BoT-IoT Dataset

ML Algorithm	Study	F1-Score
XG-Boost	Fernando et al.[26]	0.8953
RBFNN Ensemble	Mohy-eddine et al.[27]	0.9031
Random Forest	Fernando et al. [26] Siddiqi et al. [28]	0.9098
DNN	Siddiqi et al. [28]	0.9259
Spatial-Temporal Graph	Wang et al.[16]	0.9314
E-GraphSAGE	Lo et al.[11]	0.9660
RNN	Koroniotis et al.[9]	0.9769
LSTM	Koroniotis et al. [9] Zeeshan et al. [29]	0.9769
BiLSTM	Kumar et al.[30]	0.9769
RGCN without ExpressEdges	This study	**0.9769**
RGCN with ExpressEdges	This study	**0.9769**
NEGAT+NEGSC	Xu et al.[31]	0.9859
GCN+GAT+GCN	Altaf et al.[32]	0.9874
SVM	Koroniotis et al.[9]	0.9883

Jul. 2024. [Online]. Available: https://hub.dragos.com/hubfs/Reports/Dragos-FrostyGoop-ICS-Malware-Intel-Brief-0724_r2.pdf

[2] Modbus.org, "Modbus Application Protocol V1.1b3," 2012. [Online]. Available: http://www.modbus.org/docs/

[3] H. Li and D. Chasaki, "Heterogeneous GNN with Express Edges for Intrusion Detection in Cyber-Physical Systems," *In Review*, 2023.

[4] M. A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, I. Ali, and M. Guizani, "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2020.

[5] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," Feb. 2017, arXiv:1609.02907 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1609.02907

[6] G. Dong, M. Tang, Z. Wang, J. Gao, S. Guo, L. Cai, R. Gutierrez, B. Campbel, L. E. Barnes, and M. Boukhechba, "Graph Neural Networks in IoT: A Survey," *ACM Transactions on Sensor Networks*, vol. 19, no. 2, pp. 47:1–47:50, Apr. 2023.

[7] T. Bilot, N. E. Madhoun, K. A. Agha, and A. Zouaoui, "Graph Neural Networks for Intrusion Detection: A Survey," *IEEE Access*, vol. 11, pp. 49 114–49 139, 2023.

[8] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, 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, conference Name: IEEE Access.

[9] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, Nov. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X18327687

[10] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a Standard Feature Set for Network Intrusion Detection System Datasets," *Mobile Networks and Applications*, vol. 27, no. 1, pp. 357–370, Feb. 2022.

[11] W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, and M. Portmann, "E-GraphSAGE: A Graph Neural Network based Intrusion Detection System," *arXiv:2103.16329 [cs]*, Jul. 2021, arXiv: 2103.16329. [Online]. Available: http://arxiv.org/abs/2103.16329

[12] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," Sep. 2018, arXiv:1706.02216 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1706.02216

[13] L. Chang and P. Branco, "Graph-based Solutions with Residuals for Intrusion Detection: the Modified E-GraphSAGE and E-ResGAT Algorithms," Nov. 2021, arXiv:2111.13597 [cs]. [Online]. Available: http://arxiv.org/abs/2111.13597

[14] H. Friji, A. Olivereau, and M. Sarkiss, "Efficient Network Representation for GNN-Based Intrusion Detection," in *Applied Cryptography and Network Security*, ser. Lecture Notes in Computer Science, M. Tibouchi and X. Wang, Eds. Cham: Springer Nature Switzerland, 2023, pp. 532–554.

[15] D. Pujol-Perich, J. Suárez-Varela, A. Cabellos-Aparicio, and P. Barlet-Ros, "Unveiling the potential of Graph Neural Networks for robust Intrusion Detection," *arXiv:2107.14756 [cs]*, Jul. 2021, arXiv: 2107.14756. [Online]. Available: http://arxiv.org/abs/2107.14756

[16] X. Wang, X. Wang, M. He, M. Zhang, and Z. Lu, "Spatial-Temporal Graph Model Based on Attention Mechanism for Anomalous IoT Intrusion Detection," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 3, pp. 3497–3509, Mar. 2024.

[17] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation Learning on Graphs: Methods and Applications," Apr. 2018, arXiv:1709.05584 [cs]. [Online]. Available: http://arxiv.org/abs/1709.05584

[18] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling Relational Data with Graph Convolutional Networks," in *The Semantic Web*, ser. Lecture Notes in Computer Science, A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy,

L. Hollink, A. Tordai, and M. Alam, Eds. Cham: Springer International Publishing, 2018, pp. 593–607.
[19] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," Feb. 2018, arXiv:1710.10903 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1710.10903
[20] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," May 2016, arXiv:1409.0473 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1409.0473
[21] X. Wang, H. Ji, C. Shi, B. Wang, P. Cui, P. Yu, and Y. Ye, "Heterogeneous Graph Attention Network," Jan. 2021, arXiv:1903.07293 [cs]. [Online]. Available: http://arxiv.org/abs/1903.07293
[22] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow Datasets for Machine Learning-based Network Intrusion Detection Systems," *arXiv:2011.09144 [cs]*, vol. 371, pp. 117–135, 2021, arXiv: 2011.09144. [Online]. Available: http://arxiv.org/abs/2011.09144
[23] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, pp. 108–116, 2018.
[24] A. Habibi Lashkari, G. Kaur, and A. Rahali, "DIDarknet: A Contemporary Approach to Detect and Characterize the Darknet Traffic using Deep Image Learning," in *Proceedings of the 2020 10th International Conference on Communication and Network Security*, ser. ICCNS '20. New York, NY, USA: Association for Computing Machinery, Mar. 2021, pp. 1–13.
[25] H. Han, T. Zhao, C. Yang, H. Zhang, Y. Liu, X. Wang, and C. Shi, "OpenHGNN: An Open Source Toolkit for Heterogeneous Graph Neural Network," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, ser. CIKM '22. New York, NY, USA: Association for Computing Machinery, Oct. 2022, pp. 3993–3997.
[26] G.-P. Fernando, A.-A. H. Brayan, A. M. Florina, C.-B. Liliana, A.-M. Héctor-Gabriel, and T.-S. Reinel, "Enhancing Intrusion Detection in IoT Communications Through ML Model Generalization With a New Dataset (IDSAI)," *IEEE Access*, vol. 11, pp. 70 542–70 559, 2023.
[27] M. Mohy-Eddine, A. Guezzaz, S. Benkirane, M. Azrour, and Y. Farhaoui, "An Ensemble Learning Based Intrusion Detection Model for Industrial IoT Security," *Big Data Mining and Analytics*, vol. 6, no. 3, pp. 273–287, Sep. 2023.
[28] M. A. Siddiqi and W. Pak, "An Agile Approach to Identify Single and Hybrid Normalization for Enhancing Machine Learning-Based Network Intrusion Detection," *IEEE Access*, vol. 9, pp. 137 494–137 513, 2021.
[29] M. Zeeshan, Q. Riaz, M. A. Bilal, M. K. Shahzad, H. Jabeen, S. A. Haider, and A. Rahim, "Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and Bot-IoT Data-Sets," *IEEE Access*, vol. 10, pp. 2269–2283, 2022.
[30] P. Jagdish Kumar, S. Nedunchelliyan, M. Mundher Adnan, S. K, and A. Sudhakar, "Anomaly-Based Intrusion Detection System Using Bidirectional Long Short-Term Memory for Internet of Things," in *2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, Apr. 2024, pp. 01–04. [Online]. Available: https://ieeexplore.ieee.org/document/10549281
[31] R. Xu, G. Wu, W. Wang, X. Gao, A. He, and Z. Zhang, "Applying Self-supervised Learning to Network Intrusion Detection for Network Flows with Graph Neural Network," Mar. 2024, arXiv:2403.01501 [cs]. [Online]. Available: http://arxiv.org/abs/2403.01501
[32] T. Altaf, X. Wang, W. Ni, G. Yu, R. P. Liu, and R. Braun, "A new concatenated Multigraph Neural Network for IoT intrusion detection," *Internet of Things*, vol. 22, p. 100818, Jul. 2023.