Performance evaluation of ML and GNN algorithms in attack detection in various IoT environments

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*—Cyber attacks on critical infrastructure, especially IoT-based systems like the electrical grid, are growing in scale and complexity. Traditional IT security models fall short in addressing the unique challenges of Internet of Things (IoT) environments. This paper explores the use of Machine Learning (ML) and Graph Neural Networks (GNNs) for intrusion detection in IoT environments. We analyze high-profile attack case studies and evaluate our models on both balanced and imbalanced datasets. Our results demonstrate the effectiveness of graph-based approaches and offer practical guidance for securing IoT systems.**

***Index Terms*—Internet of Things, machine learning, intrusion detection, graph, neural networks, GNN.**

I. Introduction

"Never, never, never believe that any war will be smooth and easy," Winston Churchill famously characterized warfare. As we delve into the subject of artificial intelligence (AI) and machine learning (ML) methods for cybersecurity, this quote takes on new meaning. The ongoing battle to safeguard IoT environments, including the electrical grid, is nothing but smooth and easy. Recent reports from reputable sources like The New York Times and The Wall Street Journal [1] have exposed state-sponsored cyber attacks targeting the North American electrical grid. These incidents not only reveal the complexity of the challenges we face but also emphasize the urgent need to fortify cybersecurity defenses for vital infrastructure and industrial control systems.

While extensive research has examined cyber intrusions and defenses in general IT systems, including the seminal publication by Lockheed Martin researchers Hutchins, Cloppert, and Amin on the cyber kill chain [2], which provides a comprehensive overview of the field, a separate body of knowledge focuses on the protection of IoT systems. The distinctive impact dimensions, specialized knowledge requirements, often bespoke designs, and the efforts to air-gap IoT networks make cyber attacks on CPS systems markedly different from those on general IT systems. In-depth studies of IoT and Operational Technology intrusions remain a burgeoning field, partly due to the relatively small attack sample size. A notable analysis by Assante and Lee [3] offers a reference model for the CPS cyber kill chain, highlighting its variations from the traditional IT kill chain. In our researrch, we examined several high-profile IoT attack case studies through the lens of the CPS Cyber Kill Chain. A summary of these attack examples is outlined in Table I.

AI and ML techniques have been hailed for their potential to make a profound impact on datadriven business applications [4]. This potential has certainly captured the attention of many cybersecurity professionals, including the authors, as a promising means to defend against constantly advancing adversaries. Although the characteristics of these studies differ significantly from real-life network traffic, their significance lies in their trailblazing efforts. They have demonstrated the applicability of ML methodologies in cybersecurity, providing a foundational platform for new generations of cybersecurity professionals to learn and practice using ML as a valuable tool.

This paper explores the intersection of ML and Graph Neural Network (GNN) algorithms in the context of protecting critical infrastructure. We compare our work on machine learning models and hetergeneous graph neural networks to other state-of-the-art works and make recommendations on different popular IoT datasets, both balanced and imbalanced.

The remainder of this paper is organized as follows: Section II reviews related work. Section III describes popular IoT datasets used to evaluate popular AI/ML models for IoT intrusion detection. Section IV compares our results to state-of-the-art works and makes recommendations on the best approach for the different types of datasets. Finally, Section V provides the conclusion and discusses future work.

II. Related Work

A. ML-Based IoT Intrusion Detection Techniques

ML-based techniques naturally encompass the capabilities of knowledge-based and simple statistical-based detection techniques. As described in more in-depth reviews [5], [6], [7], [8], [9], [10], [11], clustering-based techniques like KNN can capture most of what knowledge-based techniques offer, while Naive Bayes accomplishes much of the statistical modeling task and can learn the underlying thresholds automatically. ML provides a broader spectrum of techniques, including SVM, Decision Trees, ensemble methods, and advanced neural networks. Perhaps not surprisingly, the majority (61%) of recent IoT intrusion detection publications [8] have employed some version of ML techniques due to the inherent limitations

TABLE I: Example Cyber Attacks on Critical Infrastructure

IoT Attack Name	Who, When and Where	How, Stage 1 (IT System)	How, Stage 2 (OT System)	Impact
Stuxnet	Strategic Adversaries of Iran, 2009–2010 in Iran	Used Physical Delivery and Zero-Days to Exploit, with Years of Dwell Time	CPS Payload Took Control of PLC and MITM Attack Control Systems	Destroyed Nuclear Enrichment Centrifuges
Dragonfly aka Havex	DRAGONFLY Group Energetic Bear, 2011–2015 in Europe and US	Spear-phishing Emails, Watering-Hole Attacks, and Custom RATs, with Uncertain Dwell Time	Reconn. with Legitimate Functionality in the OPC Protocol	Cyber-espionage
BlackEnergy2/3	SANDWORM, 2015 in Ukraine	Target Internet Connected HMI, Phishing, Credential Capturing, with Many Months of Dwell Time	Manual Attacks to Disconnect Substations and Disable Hardware	Resulted in 200,000+ Customers Without Power for 6 Hours
Crashoverride aka Industroyer	ELECTRUM, 2016 in Ukraine	Phishing, Credential Capturing, with Many Months of Dwell Time	Modularized Multi-Stage Attack to Disable Electric Grid	Took Down a Transmission-Level 330KW Sub-Station for 1 Hour
Triton aka Trisis, Hatman	XENOTIME, 2017 in Saudi Arabia	Preferred Commodity Tools, with up to One Year of Dwell Time	DOS Attack on Safety System	Repeated Plant Shutdowns and May Lead to Human Casualties
Electric Utility Reconn.	ALLANITE, 2017–2019 in US and UK	Phishing, Credential Capturing, and Native Tools, with Uncertain Dwell Time	Recon with Screen Capture of HMIs	Cyber-espionage
Ransomware Attacks	Multiple Actors	Phishing, Credential Capturing, and Native Tools	Uncertain	Ransom Payment, Denial of Service

of other intrusion detection methods when dealing with constantly evolving attacker tactics, techniques, and procedures (TTPs).

This section includes a few recent representative published works, primarily selected because they compared the performance of multiple ML methods in the same study, so that we can offer practitioner a more comprehensive view of the efficacy of each selected ML method. In 2016, Ponomarev and Atkison [12] constructed a dataset using traffic directed to a honeypot system that simulated the Modbus protocol. They compared multiple ML algorithms using the same evaluation criteria and concluded that tree-based algorithms, particularly bagged and boosted trees, exhibited superior detection performance compared to Logistic Regression or Naive Bayes when using Accuracy as the metric. It is noteworthy that the dataset used in their study was not made public.

In 2019, Zolanvari et al. [10] established a Modbus-based water tank monitoring testbed system and made the resulting data partially available to the public. The dataset was intentionally created to simulate the real-life imbalance between normal and attack traffic, with a 500:1 ratio. However, the report did not mention any specific data re-balancing techniques. Multiple ML algorithms were employed for comparative studies, with Random Forest outperforming Naive Bayes, Decision Tree, K-Nearest Neighbor (KNN), Neural Network, Support Vector Machine (SVM), and Logistic Regression across various metrics. The authors used multiple evaluation metrics for the comparative study, including Accuracy, False Alarm Rate, Undetected Rate, Matthews Correlation Coefficient, Sensitivity, and ROC Curve.

In 2019, Perez et al. [11] utilized an existing gas pipeline dataset [13] to investigate the effectiveness of various ML methods. The selected ML algorithms included SVM, RF, and Bidirectional Long Short Term Memory (LSTM), a type of Recurrent Neural Network (RNN). The authors reported that Random Forest was the overall superior method based on F1 score, Precision, and Recall metrics.

In 2019, Gomez et al. [14] developed a Modbus and S7Comm based testbed. They subsequently published the resulting datasets. Random Forest, SVM, Neural Network, OCSVM, and Isolation Forest algorithms were tested on a balanced subset of data, with Random Forest and SVM emerging as two of the best-performing methods based on F1 score, Precision, and Recall metrics.

In 2017, Feng et al. [15] presented the results of a hybrid multi-layer detection approach that combined statistical-based and ML-based detection methods. They utilized an existing gas pipeline dataset [13]. The first step involved applying statistical-based anomaly detection using a Bloom Filter. Only the normal data was then forwarded to the second detector, which was built using the LSTM algorithm to detect additional anomalous events. The detection results were evaluated using metrics such as F1 score, Accuracy, Precision, and Recall.

The proposed multi-layer detection method outperformed Bloom Filter only, Naive Bayes, SVM, Isolation Forest, and Gaussian Mixture Model (GMM), which is a type of clustering algorithm. In 2019, Khan et al. [16] introduced a slightly different hybrid detection technique, also combining statistical-based and ML-based methods. They used Bloom Filter as the initial step and employed K-Nearest Neighbor (KNN) as the second detector. In addition to comparing their results to the method presented by Feng et al. [15], Khan et al. extended their study to include Random Forest and Neural Network. They reported that the proposed hybrid detection approach outperformed the other methods, using a similar set of evaluation metrics. The key differentiating factor appears to be the data re-balancing pre-processing step applied before the Bloom Filter is employed.

In 2020, Bovenzi et al. [17] introduced a method that leverages a stacking technique. The first-stage detection utilized an AutoEncoder for anomaly detection. Subsequently, the results of the anomaly detection and a subset of the data were passed to a classifier in the second stage to perform additional intrusion detection functions.

B. GNN Studies on IoT Network Intrusion Detection

Lo et al. [18] generated significant interest in the IoT 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. Using this E-GraphSAGE modification of the GraphSAGE algorithm [19], 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 [20] and Friji et al. [21]. 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.

The ability of GNNs to leverage both spatial features and typical flow features in resisting adversarial attacks was demonstrated by Pujol-Perich et al. [22]. 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.

Wang et al. [23] 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.

Another, potentially more fundamental, limitation of traditional machine learning-based NIDS that employs classical ML and Deep Learning (DL) algorithms is the isolation of each packet or network flow from the broader context of the IoT. This shortcoming is particularly evident when dealing with multi-flow formats of attacks, shown in Figure **??**, such as those observed in the reconnaissance, Distributed Denial-of-Service (DDoS), and lateral movement stages described in the Mitre ATT&CK [24] framework.

In real-world IoT 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.

The seminal work on Graph Convolutional Networks (GCN) by Kipf and Welling [25] marked a significant advancement in the field of Graph Neural Network (GNN) research. In their work, the authors introduced the concept of applying spectral convolution to non-Euclidean graph data, drawing inspiration from Convolutional Neural Networks (CNNs).

III. Popular IoT Datasets

The NF-BoT-IoT dataset [26], [27] 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. Furthermore, the dataset is highly imbalanced, with only 2.3% of the traffic flows being normal and 97.7% representing attack traffic.

The ToN-IoT dataset, developed by Alsaedi et al. [28], 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. 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.

To evaluate the broader applicability of the proposed methodology, we extended our analysis to include additional NIDS datasets: CIC-IDS2017 [29] and CIC-Darknet [30]. A summary of the key statistics for these datasets is provided in Table II. 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 II: 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%

IV. Performance Evaluation of ML and GNN based NIDS

A. Data Processing

To address the limitations of small lab environments, which typically generate datasets with limited device diversity, 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 [27], 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.

B. 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 (1):

$$F1Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (1)$$

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 [31]. All experiments were conducted on a machine with an Intel i7-11700 CPU (16 cores), 64 GB of memory, and an NVIDIA GeForce RTX 3070 GPU with 8 GB of video memory. Numerous experiments were run with selected parameters similar to the code snippet as shown in the following Listing leveraging the Experiment API from OpenHGNN package. (Note: custom "experiment conditions" input data was added to the Experiment API by the author in order to track the numerous experiments).

```
experiment = Experiment(
    model=MODEL,
    dataset='demo_graph',
    task='node_classification',
    gpu=0,
    hpo_search_space=search_space,
    lr=0.005,
    verbose=False,
    use_self_loop=True,
    hidden_dim=HIDDEN_DIM,
    in_dim=IN_DIM,
    max_epoch=MAX_EPOCH,
    mini_batch_flag=True,
    fanout=FANOUT,
    num_heads=num_heads,
    experiment_conditions=
        experiment_conditions,
)
```

Listing 1: Experiment Setup in Python

C. Results

Extensive research has been conducted on utilizing tabular data and machine learning techniques for Network Intrusion Detection Systems (NIDS) to detect CPS attacks [32]. A selection of published results for the Ton-IoT dataset [28] is presented in Table III, and for the NF-BoT-IoT dataset [26], [33] in Table IV.

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

ML Algorithm	Study	F1-Score
DNN	Friji et al.[21]	0.3344
XG-Boost	Friji et al.[21]	0.4807
E-GraphSAGE	Lo et al. [18] Friji et al. [21]	0.88
Dual-Relation GNN	Friji et al. [21] Pujol-Perich et al. [22]	0.902
GNN with Residuals	Friji et al.[21]	0.937
RGAT with Undirected Edges	Our study	0.9448
HAN with Undirected Edges	Our study	0.9699
RGCN with Undirected Edges	Our study	0.9747
Rsage with Undirected Edges	Our study	0.9752
RGCN with Express Edges	Our study	**0.9749**
RSAGE with Express Edges	Our study	**0.9786**

1) ToN-IoT Dataset: Binary classification results are employed in this study to enable comparisons with other published State-of-the-Art (SOTA) studies on ML-based NIDS.

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

ML Algorithm (and graph)	Study	F1-Score
XG-Boost	Fernando et al. [34]	0.8953
RBFNN Ensemble	Mohy-eddine et al. [35]	0.9031
Random Forest	Fernando et al. [146], Siddiqi et al. [148]	0.9098
DNN	Siddiqi et al. [36]	0.9259
RGCN with Undirected Edges	Our study	0.9294
RGAT with ExpressEdges	Our study	0.9304
RGCN with ExpressEdges	Our study	0.9307
Spatial-Temporal Graph, Non-diversified Source IP	Wang et al. [23]	0.9314
RGAT with Undirected Edges	Our study	0.9566
E-GraphSAGE	Lo et al. [18]	0.9660
RNN	Koroniotis et al. [26]	0.9769
LSTM	Koroniotis et al. [26], Zeeshan et al. [37]	0.9769
BiLSTM	Kumar et al. [38]	0.9769
RGCN with Undirected Edges, Non-diversified Source IP	Our study	0.9769
RGCN with Directional Edges	Our study	0.9769
NEGAT+NEGSC	Xu et al. [39]	0.9859
GCN+GAT+GCN, Non-diversified Source IP	Altaf et al. [40]	0.9874
SVM	Koroniotis et al. [26]	0.9883

Table III presents results from publications that utilized data-splitting methods aimed at mitigating the target leakage effects from random data splitting, a common issue in network intrusion detection system (NIDS) research. By ensuring that the training and test sets are split with an awareness of time dependencies and session grouping, the target leakage problem is reduced, leading to more reliable and generalizable results.

Among the methods with similar experimental setups, all four HGNN algorithms selected for testing from our work, when applied to heterogeneous graphs with undirected edges, outperformed the rest of the reported results, including Dual-Relation GNN (0.902) and GNN with Residuals (0.937) reported by Friji et al. [21]. The Rsage model with Undirected Edges achieved the highest performance with an F1-score of 0.9752. This suggests that undirected edges may offer a more flexible framework for modeling certain types of network behaviors, enabling the GNN algorithm to better capture the attack behaviors inherent in the ToN-IoT dataset.

When comparing these results to other studies, it is notable that conventional machine learning models such as MLP Neural Networks and XGBoost, as demonstrated by Friji et al. [21], perform significantly worse. The MLP NN achieves an F1-score of 0.33438, and XGBoost achieves 0.4807, both of which are considerably lower than the GNN-based models. This significant difference reinforces the idea that GNNs are particularly well-suited to network-based intrusion detection tasks due to their ability to model the relational structure of the data, which is inherently graph-like in network traffic scenarios.

Furthermore, the E-GraphSAGE algorithm, reported by Lo et al. [18] and Friji et al. [21], achieves a F1-score of 0.88. While this is a strong result, it is still notably lower than the performance of the Rsage algorithm used in this study, which was evaluated using undirected edges. This indicates that incorporating Heterogeneous GNN and undirected edge modeling, can provide significant improvements over conventional graph algorithms.

2) NF-BoT-IoT dataset: 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.

3) Multiple datasets: 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 V, 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.

Another notable observation from Table V 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. Given the absence of a consistently superior HGNN algorithm, it is recommended to compare the performance of multiple representative HGNN algorithms to identify the most suitable one for a specific problem, as demonstrated in the comparative analysis in Table V.

TABLE V: Performance Benchmark across Multiple Datasets

Dateset	Algorithm	Graphs without Express Edges	Graphs with Express Edges
ToN-IoT	RGCN	0.9747	0.9749
	RSAGE	0.9752	**0.9786**
	RGAT	0.9448	0.9340
	HAN	0.9699	0.9699
CIC-IDS2017	RGCN	0.7891	0.7892
	RSAGE	0.8308	**0.8996**
	RGAT	0.8546	0.8433
	HAN	0.7087	0.6911
CIC-Darknet	RGCN	0.9375	0.9328
	RSAGE	0.9391	0.9356
	RGAT	0.9367	**0.9427**
	HAN	0.9390	0.9393
NF-BOT-IOT Non-diversified	RGCN	0.9769	0.9769
NF-BOT-IOT	RGAT	**0.9566**	0.9304
NF-BOT-IOT 3X Edge Reduction	RGCN	**0.9401**	0.9174
NF-BOT-IOT 10X Edge Reduction	RGCN	0.9543	**0.9664**
NF-BOT-IOT 30X Edge Reduction	RGCN	0.9513	**0.9767**
NF-BOT-IOT 100X Edge Reduction	HAN	0.9785	0.9785
NF-BOT-IOT 300X Edge Reduction	RGCN	0.9743	0.9743

V. Conclusion

This study presents a rigorous evaluation of machine learning (ML) and graph neural network (GNN) algorithms for intrusion detection in Internet of Things (IoT) environments, with a particular emphasis on critical infrastructure protection. By leveraging both balanced and imbalanced datasets—including ToN-IoT, NF-BoT-IoT, CIC-IDS2017, and CIC-Darknet—we demonstrate that GNN-based models, especially those employing heterogeneous graph structures and undirected or express edge configurations, consistently outperform traditional ML approaches in capturing the complex relational patterns of network traffic. Our findings underscore the importance of graph-based representation learning in enhancing the robustness and generalizability of intrusion detection systems. Notably, the Rsage and RGCN models achieved state-of-the-art performance across multiple datasets, validating the efficacy of heterogeneous GNNs in modeling diverse and dynamic IoT environments. Furthermore, the Express Edge technique showed measurable improvements in sparse graph scenarios, offering a promising direction for future research. While no single model dominated across all conditions, the adaptability of GNN architectures to varying graph densities and attack formats highlights their potential as a foundational tool for next-generation cybersecurity solutions. Future work will focus on real-time detection, dynamic graph modeling, and integration with broader cyber-physical system (CPS) security frameworks to further strengthen the resilience of IoT infrastructures against evolving threats.

References

[1] D. E. Sanger, "Russian Hackers Appear to Shift Focus to U.S. Power Grid," *The New York Times*, Jul. 2018. [Online]. Available: https://www.nytimes.com/2018/07/27/us/politics/russian-hackers-electric-grid-elections-.html

[2] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains," *6th International Conference on Information Warfare and Security 2011 (ICIW 2011)*, vol. 2011, pp. 113–125, 2011.

[3] M. J. Assante and R. M. Lee, "The Industrial Control System Cyber Kill Chain," *SANS Institute InfoSec Reading Room*, vol. 1, pp. 1–24, 2015.

[4] A. Kronz, J. Richardson, and R. Sallam, "Critical Capabilities for Analytics and Business Intelligence Platforms," 2019. [Online]. Available: https://www.gartner.com/en/documents/3913552/critical-capabilities-for-analytics-and-business-intelli

[5] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[6] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine Learning and Deep Learning Methods for Cybersecurity," *IEEE Access*, vol. 6, pp. 35 365–35 381, 2018.

[7] E. Hossain, I. Khan, F. Un-Noor, S. S. Sikander, and M. S. H. Sunny, "Application of Big Data and Machine Learning in Smart Grid, and Associated Security Concerns: A Review," *IEEE Access*, vol. 7, pp. 13 960–13 988, 2019.

[8] S. V. B. Rakas, M. D. Stojanović, and J. D. Marković-Petrović, "A Review of Research Work on Network-Based SCADA Intrusion Detection Systems," *IEEE Access*, vol. 8, pp. 93 083–93 108, 2020, conference Name: IEEE Access.

[9] H. Li and S. Qin, "Optimization and implementation of industrial control system network intrusion detection by telemetry analysis," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, Dec. 2017, pp. 1251–1254.

[10] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine Learning-Based Network Vulnerability Analysis of Industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6822–6834, Aug. 2019.

[11] R. Lopez Perez, F. Adamsky, R. Soua, and T. Engel, "Forget the Myth of the Air Gap: Machine Learning for Reliable Intrusion Detection in SCADA Systems," *ICST Transactions on Security and Safety*, vol. 6, no. 19, p. 159348, Jan. 2019. [Online]. Available: http://eudl.eu/doi/10.4108/eai.25-1-2019.159348

[12] S. Ponomarev and T. Atkison, "Industrial Control System Network Intrusion Detection by Telemetry Analysis," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 2, pp. 252–260, Mar. 2016.

[13] T. H. Morris, Z. Thornton, and I. Turnipseed, "Industrial Control System Simulation and Data Logging for Intrusion Detection System Research," 2015, p. 6, event-place: Huntsvile, AL.

[14] L. Perales Gómez, L. Fernández Maimó, A. Huertas Celdrán, F. J. García Clemente, C. Cadenas Sarmiento, C. J. Del Canto Masa, and R. Méndez Nistal, "On the Generation of Anomaly Detection Datasets in Industrial Control Systems," *IEEE Access*, vol. 7, pp. 177 460–177 473, 2019.

[15] C. Feng, T. Li, and D. Chana, "Multi-level Anomaly Detection in Industrial Control Systems via Package Signatures and LSTM Networks," in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Jun. 2017, pp. 261–272, iSSN: 2158-3927.

[16] I. A. Khan, D. Pi, Z. U. Khan, Y. Hussain, and A. Nawaz, "HML-IDS: A Hybrid-Multilevel Anomaly Prediction Approach for Intrusion Detection in SCADA Systems," *IEEE Access*, vol. 7, pp. 89 507–89 521, 2019, conference Name: IEEE Access.

[17] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, "A Hierarchical Hybrid Intrusion Detection Approach in IoT Scenarios," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Dec. 2020, pp. 1–7, iSSN: 2576-6813.

[18] 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

[19] 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

[20] 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

[21] 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.

[22] 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

[23] 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.

[24] "MITRE ATT&CK®." [Online]. Available: https://attack.mitre.org/

[25] 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

[26] 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

[27] 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

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] P. Jagdish Kumar, S. Neduncheliyan, 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

[39] 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

[40] 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.