Towards Problem Space-constrained Adversarial Attacks against Graph Neural Network based Network Intrusion Detection Systems

Matthieu Mouzaoui
Inria, Univ. Rennes, IRISA
Rennes, France
matthieu.mouzaoui@inria.fr

Yufei Han
Inria, Univ. Rennes, IRISA
Rennes, France
yufei.han@inria.fr

Grégory Blanc
SAMOVAR
Télécom SudParis, Institut Polytechnique de Paris
Palaiseau, France
gregory.blanc@telecom-sudparis.eu

Michel Hurfin
Inria, Univ. Rennes, IRISA
Rennes, France
michel.hurfin@inria.fr

Gabriel Rilling
CEA, LIST, Laboratoire pour l'Instrumentation Intelligente, Distribuée et Embarquée
Gif-sur-Yvette, France
gabriel.rilling@cea.fr

***Abstract*—The increasing adoption of Machine Learning (ML) models has raised concerns about their robustness to adversarial attacks. In the context of Network Intrusion Detection Systems (NIDS), adversarial attacks consist of manipulating the network traffic to degrade system performance by either performing evasion or inducing alarm fatigue. This study examines the vulnerability of Graph Neural Networks (GNN)-based NIDS under problem-space constraints, where attackers can only create new communications towards specific IP addresses, while ensuring consistency with network protocols. We demonstrate that these attacks constitute a weakly submodular optimization problem, solvable with simple greedy search algorithms. Our preliminary results confirm that injecting a few well-chosen connections significantly increases false alarms in state-of-the-art GNN-based NIDS. Finally, we outline a roadmap for future work on evasion attacks and the transferability of attacks across different GNN models.**

***Index Terms*—Graph Neural Network, Network Intrusion Detection, Adversarial Machine Learning, Graph based security**

I. Introduction

Network Intrusion Detection Systems (NIDS) have widely adopted rule-based or signature-based detection mechanisms, [1], [2]. The rise of ML techniques has led to automated network attack detection and categorization, enabling scalable detection of both known and previously unseen attacks. Benefiting from the flourishing of ML techniques, ML-based NIDS enhance threat analysis and understanding of emerging threats. Nevertheless, as data-driven and statistical models, ML techniques are well known to be vulnerable to *adversarial attacks*, where slight changes to the input fool the model's predictions. While most previous adversarial attack methods have focused on general AI applications like Computer Vision or NLP [3]–[5], recent research has highlighted their impact on security-critical systems such as ML-based NIDS [6], [7]. They unveiled that ML-based NIDS can be misled by intentionally crafted features representing network attack behaviours. These adversarially manipulated attack behaviours can either evade detection, or generate excessive false alarms, which makes the ML-based detection output unreliable and prevent the use of ML techniques in the security-critical scenarios.

Our study focuses on characterising and evaluating the adversarial robustness and risk of GNN-based IDS [8]–[14]. Compared to traditional models such as Support Vector Machines, Random Forests, or Deep Neural Networks, GNNs enhance the transparency of ML-based detection processes by employing structured modelling for attack detection. **Specifically**, GNNs encode network traffic as structured representations that align with the data structure of network communication records. This approach preserves relationships between different traffic flows within the graph representation, enabling direct modeling of high-order correlations between traffic flows and hosts. Unlike statistical feature-based NIDS methods [15], [16], GNNs intrinsically provide a clear explanation of how an attack alert is triggered during the ML-based decision process, while earlier methods, which compress traffic into statistical features, limit interpretability and raise trust concerns.

However, our study demonstrates that the interpretability of the GNN-based detection mechanism is a double-edged sword. Adversarial attackers can enjoy the same transparency advantage to organise problem space-feasible adversarial attacks against GNN-based systems. The graph structure, as a natural representation of network traffic data, offers the details about how the network traffic are delivered in a given use case to the attacker. An attacker may leverage the graph structure to define adversarial manipulation over network traffic flows that

are compatible with the communication protocols and network architectures deployed in the target network communication scenarios. This is also known as problem-space constrained attacks [17].

This paper demonstrates how an attacker can exploit the GNN-based network data encoding to design realistic adversarial perturbations that mislead a GNN-NIDS. We assume that the attacker knows parts of the network architecture (e.g., some hosts) and the payloads of benign and malicious traffic. Additionally, the attacker can evaluate the GNN model's detection output, enabling a grey-box attack. The adversarial perturbation must also comply with the problem-space constraints of the target network communication. Designing adversarial attacks against GNN-NIDS presents two main challenges: (1) the combinatorial nature of graph data creates a large search space for perturbations, requiring computationally efficient solutions; and (2) attacks must adhere to NIDS-specific domain constraints to remain realistic.

Our study is organised to answer three main research questions:

- What types of manipulations are realistic for adversarial attacks in the context of NIDS? What changes made to the graph representation of data are feasible from an attacker's perspective?
- Does the attack achieve a solution sufficiently optimal to perform a successful attack, given the challenges posed by the combinatorial nature of the problem?
- Are the attacks transferable to other NIDS? Can attacks on a GNN -NIDS also negatively impact a statistics-based or signature-based IDS?

To address these questions, we focus on network flows as the primary data for NIDS training and testing, avoiding the challenges of analyzing encrypted payloads. **First**, we frame adversarial attacks on GNN-based NIDS as a combinatorial optimization problem, where attacks involve adding connections and selecting their endpoints rather than modifying network flow metadata. **Second**, we incorporate problem-space constraints into the attack design, crafting injected traffic flows to comply with communication protocol rules. **Third**, we identify the adversarial attack as a weakly sub-modular maximization problem, enabling the use of a computationally efficient greedy search strategy to craft network flows. **Finally**, we outline a research roadmap, as this work remains in its early stages and is part of an ongoing PhD project.

II. Related Works

A. Graph Representation of Network data for NIDS

A graph consists of nodes and edges, both potentially featuring attributes. Various graph-based representations of network data have been explored, such as modeling packets as nodes with features like protocol and length [18], connecting security objects via events [19], and using bipartite structures to represent flow sources and destinations [8]. Some works, like [8], [11] apply line graph transformation to convert edge classification into node classification, while others [20] directly construct the transformed graph to avoid computational overhead. The flow-graph representation [8]–[10], [12] is a natural approach for modeling network data, where nodes represent endpoints and edges correspond to communications between them, aggregated into network flows. Network flows aggregate packet exchanges between hosts, including descriptive statistics like session duration and data volume. In this representation, the IDS operates as an edge classifier.

B. Adversarial attacks against GNN

Adversarial attacks involve two steps: defining a score to assess the impact of a perturbation and optimizing the perturbation to maximize this score. While gradient descent is commonly used in classical adversarial attacks, it is unsuitable for discrete graph structures. Existing adversarial attacks against GNNs [21]–[23] typically employ greedy approaches, iteratively selecting and applying the perturbation with the highest impact score.

C. Adversarial attacks against IDS

In the context of IDS, adversarial attacks should adhere to specific principles. Pierazzi et al. [17] define constraints for realistic attacks, including robustness to pre-processing and plausibility, which are outside the scope of this paper. The other two constraints are available transformations, which typically only allow adding new communications, and semantic preservation, ensuring changes in the feature space don't impact the problem-space semantics. This is easier to verify with graph data, as it closely aligns with the problem space. For realism, [24] adds a validity constraint, ensuring samples can travel in real networks. Additionally, while some works apply GNNs to perform intrusion detection based on logs (HIDS), this work focuses on Network IDS. A related study [25] uses flow-graph representation and edge addition for adversarial attacks, but has limitations: once the edge label is chosen, the edge is added randomly, which may not guarantee attack effectiveness or respect domain constraints.

III. Proposed Method

A. Data Representation

This work adopts a flow-graph representation where nodes represent endpoint devices, identified by an IP address and a port, and edges denote bidirectional communication with Network Flow records. The attributes of these flow are those kept in [26]. For each IP:Port pair, we create nodes for both the source and destination respectively. For example, a flow from IP A (port X) to IP B (port Y) is represented by two nodes, *IPA:X* and *IPB:Y*, connected by an edge carrying the Network Flow attributes. The task of the NIDS is thus an *edge classification* task. To ensure compatibility with GNNs, node duplication is applied for flows at different timestamps, ensuring at most one edge between each node pair. This avoid having multi-edges, which standard GNNs frameworks do not support. An undirected graph model is used, following [27].

B. Threat model

Building on insights from adversarial attacks on GNNs, we define a threat model for a NIDS scenario with a realism constraint for adversarial manipulation. The attacker is assumed to have access to one endpoint, as in [25], [28], enabling traffic sniffing to identify other endpoints and analyze benign and malicious statistical patterns. The attacker knows the target NIDS uses a GNN model with a flow-graph representation and has access to the underlying graph and loss function, enabling evaluation of the loss after adversarial modifications. From their controlled endpoints, the attacker can initiate new communications, effectively adding edges to the corresponding flow-graph. For this study, the attacker's objective is to increase the False Positive Rate (FPR) by misclassifying benign network flows as malicious, inducing alarm fatigue. While this specific goal is the focus of our current work, other objectives will be explored in future studies. To achieve this, the attacker considers all places where an edge can be added, computes the impact of each edge addition and actually adds the ones with the highest impact. With this description, the *adversarial sample* is thus the set of added edges. The adversarial attack by adding new edges into the graph is formulated as a discrete optimisation problem on the flow graph $\mathcal{G} = \{\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{E}\}$, with $\boldsymbol{A}$ the adjacency matrix $\boldsymbol{X}$ the node features and $\boldsymbol{E}$ the edge features of the flow graph $\boldsymbol{G}$.

$$\begin{aligned} \boldsymbol{A}^* &= \arg\max_{\boldsymbol{A}'} \ell(\mathcal{M}(\boldsymbol{A}', \boldsymbol{X}, \boldsymbol{E}, \boldsymbol{Y})) \\ s.t.\ &\|\boldsymbol{A} - \boldsymbol{A}'\|_{L_0} \leq 2b \end{aligned} \tag{1}$$

where $\mathcal{M}$ is the target GNN-based NIDS model, which takes the adjacency matrix along with node and edge features as inputs. The output of $\mathcal{M}$ is the class prediction logits. ℓ is the cross-entropy learning loss used to train the NIDS model differentiating edges of benign flows from those of malicious flows. $\boldsymbol{A}'$ denotes the modified adjacency matrix after the attacker injects new edges. $\|\cdot\|_{L_0}$ denotes the L0-norm based difference between the original and adversarially modified adjacency matrix (twice the number of added edges). We set a limit to the attacker that no more than b edges can be added, a.k.a the attack budget. In the case of activating excessively high FPR, $\boldsymbol{Y}$ denotes the true class label assigned to the benign flows in the flow graph. Maximizing the learning objective in Eq.1 aims to trigger the misclassification over benign flows as much as possible, activating false alarms. When the flow graph is undirected, under the Lipschitz-continuity conditions [29] over the GNN model, the attack objective given in Eq.1 can be considered as a γ-weakly sub-modular optimization problem for $0 \leq \gamma \leq \frac{1}{b}$. According to our prior work [30], we are guaranteed to reach the convergence of a local optimum of the attack objective function by adding new edges using a Greedy Search algorithm (GS) algorithm within the polynomial time complexity [30]. In parallel, the gap between the underlying global optimum of Eq.1 and the obtained local optimum solution by Greedy Search (GS) has a tight upper bound and remains marginally small in practice. In summary, we ensure the success of the attack by adopting the Forward Stepwise GS-based edge addition attack.

C. Greedy search-based attack method

1) Unconstrained adversarial attacks: The first step demonstrates the feasibility of the attack: adding a few edges to a flow graph $\boldsymbol{G}$ can increase the FPR. The method follows a greedy strategy [30] as described by Algorithm 1:

Algorithm 1: Unconstrained Adversarial Attack

Input: Original flow graph $\mathcal{G} = \{\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{E}\}$, attack budget b, target model $\mathcal{M}$, loss function ℓ
Output: Modified flow graph $\mathcal{G}' = \{\boldsymbol{A}', \boldsymbol{X}, \boldsymbol{E}'\}$

Initialize $\boldsymbol{A}' \leftarrow \boldsymbol{A}$, $\boldsymbol{E}' \leftarrow \boldsymbol{E}$;
for $i \leftarrow 1$ **to** b **do**
  $max\_loss \leftarrow -\infty$;
  for *each controlled node* u *in* $\mathcal{G}$ **do**
    for *each possible destination node* $v \neq u$ **do**
      Simulate adding edge (u, v) to $\boldsymbol{A}'$, with corresponding edge features $\boldsymbol{E}'_{(u,v)}$;
      Compute loss: $current\_loss \leftarrow \ell(\mathcal{M}(\boldsymbol{A}', \boldsymbol{X}, \boldsymbol{E}'), \boldsymbol{Y})$;
      if $current\_loss > max\_loss$ **then**
        $max\_loss \leftarrow current\_loss$;
        $best\_edge \leftarrow (u, v)$;
  Add $best\_edge$ to A', update $\boldsymbol{E}'$ with its features;
return $\mathcal{G}' = \{\boldsymbol{A}', \boldsymbol{X}, \boldsymbol{E}'\}$;

While effective, this approach lacks problem-space compatibility with real-world network communication.

2) Problem-space constrained adversarial attacks: The second step aims to increase the realism of the adversarial attack by integrating the problem space constraint while adding the edges via the Greedy Search-based attack. The flow graph's communication type restricts possible edge choices, ensuring **protocol compliance**. For example, TCP edges should match features corresponding to backward packets, such as the number of backward packets. At this stage, a complete definition of normal traffic is still under development. However, since for evaluation, we are using network flows present in a dataset, we assume that these flows were valid when they existed. Thus, their feature distributions are likely to be close to those found in the real network. The greedy search is modified to consider only destination nodes that comply with these constraints, ensuring realistic and valid adversarial examples.

IV. Experiments

A. Datasets

We conducted experiments using the BoT-IoT [31] dataset containing millions of network flows described by 46 features and labeled into 6 attack types. It is widely used for benchmarking ML-based NIDS. Future work will extend to other datasets like NF-BoT-IoT and NF-CSE-CIC-IDS2018 [32]. Preprocessing includes using IP addresses and port numbers as

node identifiers, one-hot encoding categorical features, and a non-shuffled 80/20 train-test split to avoid time snooping [33].

B. GNN

For this preliminary study, we employed the E-GraphSAGE [27] model as the GNN-NIDS. As in the original paper, E-GraphSAGE is built with output dimension 128, ReLU activation, 0.2 dropout rate, and trained over 8,000 epochs using Adam [34] with learning rate 0.001. Class weights address imbalance using scikit-learn's `compute_class_weight`.

C. Results

We report our preliminary results after performing a greedy search strategy for edge addition. Recall that a destination node is chosen in order to increase the model's loss. As shown in Figure 1, the model loss rises with the addition of edges. The increase of the learning loss is steep when we add the first few edges, but then becomes much smoother and converges to a stable value after approximately 10 edges are added. This confirms the weakly sub-modular nature of the adversarial attack problem.

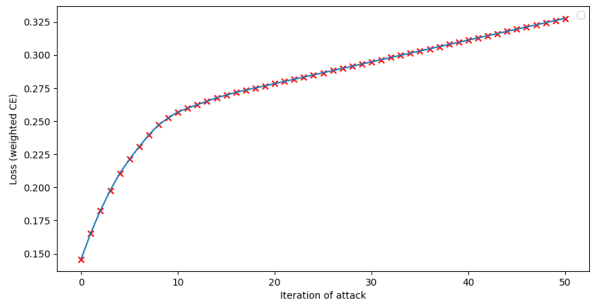

Fig. 1. Loss (BCE) evolution function of the number of added edges.

Table I presents the weighted classification report of the E-GraphSAGE model on a clean graph, before any attack.

TABLE I
WEIGHTED CLASSIFICATION REPORT ON THE CLEAN GRAPH.

Class	Precision	Recall	F1-Score	Support
0 (benign)	0.93	0.98	0.95	1,375
1 (malicious)	0.98	0.92	0.95	58,635
Accuracy			0.89	60,010
Macro avg	0.89	0.89	0.89	60,010
Weighted avg	0.89	0.89	0.89	60,010

TABLE II
WEIGHTED CLASSIFICATION REPORT AFTER 50 EDGES ADDITIONS.

Class	Precision	Recall	F1-Score	Support
0 (benign)	0.93	0.84	0.88	1,375
1 (malicious)	0.85	0.93	0.89	58,685
Accuracy			0.89	
Macro avg	0.89	0.89	0.89	60,060
Weighted avg	0.89	0.89	0.89	60,060

Following 50 edges added in a row by the attacker, Table II shows an increase of the FPR. This demonstrates even a small number of adversarial edges can degrade model performance.

V. Discussion and Future Works

This study evaluates GNN-based NIDS vulnerabilities to FPR increasing adversarial attacks under realistic constraints. The unconstrained method shows that even adding a relatively small number of edges is sufficient to degrade the model performance. This provides a lower bound for the model weakness. The study reveals the dual nature of using GNN-NIDS: while their structure benefits the defender, it is also beneficial to the attacker. To understand these attacks, we investigate the content and the location of injected network flows. Preliminary results suggest injecting malicious network flows toward benign hosts with a large amount of edges connected is likely to trigger an increase in FPR. This aligns with the fact that false alarms can be activated on benign devices if the device receives traffic with malicious signatures, such as suspicious port scan or requests containing suspicious strings, yet without harmful payloads. As an extension, performing evasion, i.e. *triggering a false negative* has yet to be implemented. However, we can expect that, if the attacker's goal is to have a specific malicious edge to be classified as benign, the attacker could change its predicted label by sending a certain amount of benign edges beforehand. This is due to the nature of Message Passing in GNN. Moreover with such a goal, a new manipulation strategy might be investigated, a replacement strategy. Additionally, we aim to automate constraints construction, which is currently dataset-specific, to apply adversarial attacks to a wider range of networks. While the flow-graph representation is common, it is not the easiest to work with, so future work will explore alternative graph data representations, potentially involving heterogeneous structures. We also consider the transferability of adversarial attacks on graph structured network traffic representations towards statistics-based NIDS [15], [16] and signature-based IDS [1].

VI. Conclusion

This paper explores the vulnerability of Graph Neural Network based NIDS to adversarial attacks. For better realism, the attack consists solely in edge additions corresponding to adding new communications into the network. The unconstrained approach demonstrates that minimal adversarial perturbations are sufficient to impact the model performance. These leverage a trade-off of the use of GNN between better explainability for the defender and their exploitation by attackers. Future work will focus on developing evasion attacks, as well as studying the transferability of the attacks designed on graph structured data. A key focus of future work will be reducing the attacker's assumed knowledge, which is currently extensive.

ACKNOWLEDGMENTS

This work has been partially supported by the French National Research Agency (ANR) under the France 2030 label (SuperViz ANR-22-PECY-0008).

References

[1] M. Roesch, "Snort - lightweight intrusion detection for networks," in *Lisa*, 1999. [Online]. Available: https://api.semanticscholar.org/CorpusID:6673399

[2] V. Naidu, J. L. Whalley, and A. Narayanan, "Generating rule-based signatures for detecting polymorphic variants using data mining and sequence alignment approaches," *Journal of Information Security*, vol. 9, pp. 265–298, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:69666583

[3] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016, pp. 372–387.

[4] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:2893830

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, vol. abs/1412.6572, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:6706414

[6] R. Sheatsley, N. Papernot, M. J. Weisman, G. Verma, and P. McDaniel, "Adversarial examples for network intrusion detection systems," *J. Comput. Secur.*, vol. 30, no. 5, p. 727–752, Jan. 2022. [Online]. Available: https://doi.org/10.3233/JCS-210094

[7] M. Catillo, A. Pecchia, A. Repola, and U. Villano, "Towards realistic problem-space adversarial attacks against machine learning in network intrusion detection," in *Proceedings of the 19th International Conference on Availability, Reliability and Security*, ser. ARES '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3664476.3669974

[8] L. Chang and P. Branco, "Graph-based solutions with residuals for intrusion detection: the modified e-graphsage and e-resgat algorithms," *ArXiv*, vol. abs/2111.13597, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:244709287

[9] P. Deng and Y. Huang, "Edge-featured multi-hop attention graph neural network for intrusion detection system," *Comput. Secur.*, vol. 148, p. 104132, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:272945443

[10] L. Lin, Q. Zhong, J. Qiu, and Z. Liang, "E-gracl: an iot intrusion detection system based on graph neural networks," *J. Supercomput.*, vol. 81, p. 42, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:273513805

[11] D. Pujol-Perich, J. Suárez-Varela, A. Cabellos-Aparicio, and P. Barlet-Ros, "Unveiling the potential of graph neural networks for robust intrusion detection," *ACM SIGMETRICS Performance Evaluation Review*, vol. 49, pp. 111 – 117, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:236635093

[12] A. Venturi, M. Ferrari, M. Marchetti, and M. Colajanni, "Arganids: a novel network intrusion detection system based on adversarially regularized graph autoencoder," *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259099368

[13] Q. Xiao, J. Liu, Q. Wang, Z. Jiang, X. Wang, and Y. Yao, "Towards network anomaly detection using graph embedding," *Computational Science – ICCS 2020*, vol. 12140, pp. 156 – 169, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:219889190

[14] 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," *ArXiv*, vol. abs/2403.01501, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:268248522

[15] N. Ye, S. M. Emran, Q. Chen, and S. Vilbert, "Multivariate statistical analysis of audit trails for host-based intrusion detection," *IEEE Trans. Computers*, vol. 51, pp. 810–820, 2002. [Online]. Available: https://api.semanticscholar.org/CorpusID:7802602

[16] J. Viinikka, H. Debar, L. Mé, A. Lehikoinen, and M. P. Tarvainen, "Processing intrusion detection alert aggregates with time series modeling," *Inf. Fusion*, vol. 10, pp. 312–324, 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:11226700

[17] F. Pierazzi, F. Pendlebury, J. Cortellazzi, and L. Cavallaro, "Intriguing properties of adversarial ml attacks in the problem space," in *2020 IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 1332–1349.

[18] Y. Li, R. Li, Z. Zhou, J. Guo, W. Yang, M. Du, and Q. Liu, "Graphddos: Effective ddos attack detection using graph neural networks," in *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 2022, pp. 1275–1280.

[19] L. Leichtnam, E. Totel, N. Prigent, and L. Mé, "Sec2graph: Network attack detection based on novelty detection on graph structured data," in *Detection of Intrusions and Malware, and Vulnerability Assessment: 17th International Conference, DIMVA 2020, Lisbon, Portugal, June 24–26, 2020, Proceedings 17*. Springer, 2020, pp. 238–258.

[20] H. Friji, A. Olivereau, and M. Sarkiss, "Efficient network representation for gnn-based intrusion detection," *ArXiv*, vol. abs/2310.05956, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259025882

[21] A. Bojchevski and S. Günnemann, "Adversarial attacks on node embeddings via graph poisoning," in *International Conference on Machine Learning*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:166227839

[22] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, S. Kraus, Ed. ijcai.org, 2019, pp. 6246–6250. [Online]. Available: https://doi.org/10.24963/ijcai.2019/872

[23] H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, and L. Song, "Adversarial attack on graph structured data," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 1123–1132. [Online]. Available: http://proceedings.mlr.press/v80/dai18b.html

[24] J. Vitorino, I. Praça, and E. Maia, "Towards adversarial realism and robust learning for iot intrusion detection and classification," *annals of telecommunications - annales des télécommunications*, vol. 78, pp. 401–412, 03 2023.

[25] A. Venturi, D. Stabili, and M. Marchetti, "Problem space structural adversarial attacks for network intrusion detection systems based on graph neural networks," *arXiv preprint arXiv:2403.11830*, 2024.

[26] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, *NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems*. Springer International Publishing, 2021, p. 117–135. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-72802-1_9

[27] W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, and M. Portmann, "E-graphsage: A graph neural network based intrusion detection system for iot," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–9.

[28] A. Venturi, D. Gall, D. Stabili, and M. Marchetti, "Hardening machine learning based network intrusion detection systems with synthetic netflows," in *Italian Conference on Cybersecurity*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:271431353

[29] J. Ma, S. Ding, and Q. Mei, "Towards more practical adversarial attacks on graph neural networks," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[30] H. Bao, Y. Han, Y. Zhou, Y. Shen, and X. Zhang, "Towards understanding the robustness against evasion attack on categorical data," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=BmJV7kyAmg

[31] N. Moustafa, "The Bot-IoT dataset," 2019. [Online]. Available: https://dx.doi.org/10.21227/r7v2-x988

[32] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow datasets for machine learning-based network intrusion detection systems," in *Big Data Technologies and Applications*. Springer International Publishing, 04 2021, pp. 117–135.

[33] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, "Dos and don'ts of machine learning in computer security," in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 3971–3988. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/arp

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:6628106