Constrained Network Adversarial Attacks: Validity, Robustness, and Transferability

Anass Grini*, **Oumaima Taheri***, **Btissam El Khamlichi***, **Amal El Fallah-Seghrouchni***§

*Ai movement, The International Artificial Intelligence Center of Morocco, UM6P, Rabat, Morocco
§Sorbonne University, LIP6 - UMR 7606 CNRS, Paris, France
{anass.grini, btissam.elkhamlichi, amal.ElFallah-Seghrouchni}@um6p.ma

Abstract—While machine learning has significantly advanced Network Intrusion Detection Systems (NIDS), particularly within IoT environments where devices generate large volumes of data and are increasingly susceptible to cyber threats, these models remain vulnerable to adversarial attacks. Our research reveals a critical flaw in existing adversarial attack methodologies: the frequent violation of domain-specific constraints, such as numerical and categorical limits, inherent to IoT and network traffic. This leads to up to 80.3% of adversarial examples being invalid, significantly overstating real-world vulnerabilities. These invalid examples, though effective in fooling models, do not represent feasible attacks within practical IoT deployments. Consequently, relying on these results can mislead resource allocation for defense, inflating the perceived susceptibility of IoT-enabled NIDS models to adversarial manipulation. Furthermore, we demonstrate that simpler surrogate models like Multi-Layer Perceptron (MLP) generate more valid adversarial examples compared to complex architectures such as CNNs and LSTMs. Using the MLP as a surrogate, we analyze the transferability of adversarial severity to other ML/DL models commonly used in IoT contexts. This work underscores the importance of considering both domain constraints and model architecture when evaluating and designing robust ML/DL models for security-critical IoT and network applications.

Index Terms—Network Intrusion Detection, Internet of Things, Adversarial attacks, Cybersecurity, Machine Learning

I. INTRODUCTION

ML has become a transformative technology across numerous domains, including the rapidly growing area of the Internet of Things (IoT). In cybersecurity, particularly within IoT environments, ML demonstrates significant proficiency in accurately classifying network traffic as legitimate or malicious [1]. However, the integration of ML into IoT introduces vulnerabilities, as research indicates ML-based Network Intrusion Detection Systems (NIDS) in IoT are susceptible to adversarial examples—inputs deliberately crafted to induce misclassification [2].

The field of Adversarial Machine Learning (AML) originated in computer vision, where minor, often imperceptible alterations to images can deceive classifiers [3]. These attacks exploit vulnerabilities in pre-trained models by subtly shifting data points across decision boundaries to produce incorrect outputs. ML algorithms' inherent pattern-recognition capabilities make them particularly vulnerable to evasion by adversaries seeking to bypass detection, especially in IoT

networks characterized by diverse and resource-constrained devices.

Traditionally, AML research emphasizes unconstrained domains, offering attackers full control over the feature space. This scenario is unrealistic for IoT and cybersecurity applications, where adversaries face strict domain-specific constraints and limited control over feature manipulation [4]. Unlike image data, IoT-generated network data is substantially harder to alter without breaking functional dependencies intrinsic to network protocols and IoT device operations.

Although considerable research has been conducted on improving ML model robustness [5], [6], insufficient attention has been paid to the fundamental constraints inherent in IoT and network security domains. These domain-specific constraints present unique challenges, such as:

- **Perturbation Imperceptibility**: IoT network data does not involve human perception, making human-imperceptible perturbations irrelevant.
- **Feature Space Control**: IoT features reflect specific network behaviors, severely limiting an adversary's capacity to freely alter them.

These challenges underscore the necessity for targeted AML approaches tailored to IoT contexts. This paper provides a comprehensive evaluation of AML techniques, explicitly accounting for IoT and network security constraints. We investigate state-of-the-art AML attacks' effectiveness on NIDS under realistic constraints and evaluate adversarial attacks' transferability across various IoT models, particularly under limited attacker knowledge (black-box conditions).

Our research further identifies a critical relationship between model complexity and robustness against adversarial attacks within IoT and network security domains. This finding emphasizes that choosing the right surrogate model significantly impacts the generation and effectiveness of adversarial examples within constrained environments.

Specifically, this work contributes by:

- Formalizing and implementing domain constraints in an IoT network traffic scenario.
- Proposing a novel validation process for adversarial examples that incorporates *Numerical Dependencies* and *Categorical Dependencies* to ensure compliance with IoT and network constraints.

- Measuring the validity of adversarial examples generated by existing algorithms by projecting them onto a feasible input space.
- Analyzing how model complexity influences the generation of valid adversarial examples, highlighting increased robustness in sophisticated IoT model architectures.
- Evaluating the effectiveness and transferability of feasible adversarial examples within constrained IoT network environments.

The remainder of this paper is structured as follows: Section II reviews relevant literature on adversarial example generation in constrained IoT domains. Section III details our methodology for modeling constraints, specifically defining the feasible adversarial example space. Section IV presents our experimental setup, results, and discusses their implications. Finally, Section V concludes the paper and suggests directions for future research.

II. RELATED WORK

Since their formal introduction in 2013 [3], AML techniques have posed a growing risk across various domains. While early research on AML emphasized image classification, cybersecurity presents unique challenges due to its use of categorical, continuous, and discrete feature types.

Although research interest in this area is growing, some studies overlook real-world constraints, which can limit the practical applicability of their AE generation techniques. The current literature exhibits several limitations:

- *Lack of realistic threat modeling:* Works by [7], [8], and [9] showcase the potential of AML for cyberattacks, yet they do not explicitly address the real-world constraints present in actual network environments.
- *Arbitrary network traffic generation:* Yang et al. [10] utilize arbitrary network traffic without establishing realistic perturbation limits, while studies like [11] and [12] often overlook or bypass important domain-specific constraints.

Recent works addressing IoT-specific challenges include Kumar et al. [13], who proposed NIDS-CBAD, a constraint-based adversarial detection method tailored to IoT networks, significantly enhancing detection efficiency. Additionally, Sharma and Chen [14] systematically analyzed adversarial attacks on ML-based NIDS in IoT environments, highlighting varying vulnerabilities across ML models.

While recent works in AML for cybersecurity acknowledge the importance of network-specific constraints, they often lack detailed consideration of these constraints. Venturi [15] discusses adversarial solutions without addressing specific network constraints. Hashemi et al. [16] focus on feature dependencies and limited attacker control in flow-based NIDS but do not fully explore all network constraints. Teuffenbach et al. [17] enhance the Carlini & Wagner attack by incorporating some feature limitations and weightings, yet they do not cover the complete range of network-specific constraints. These studies highlight significant progress but also reveal gaps in comprehensively addressing all network-specific constraints in AML.

Sheatsley et al. [18] build upon investigations into adversarial examples in constrained domains. While it provides a valuable exploration of categorical constraints, such as protocol and service dependencies, it gives less attention to numerical dependencies. Features like duration, packet sizes, and byte counts, which can carry important continuous relationships in network data, are not as prominently addressed.

III. METHODOLOGY

This work is focused on the realistic feasibility of the proposed threats against ML and DL-based Network Intrusion Detection Systems (NIDS). It aims to validate adversarial examples generated by existing algorithms within the context of domain constraints. Unlike previous studies, we provide a comprehensive model and analysis of these constraints, including numerical and categorical dependencies in feature space derived from domain knowledge and data observations.

A. Categorical Dependencies

The adversary must follow the TCP/IP protocol to carry out network attacks. Any feature vector that breaks the TCP/IP protocol is not valid. These dependencies are defined using *primary features* identified through understanding the domain and data observation [18]. Primary features are, by definition, substantially correlated with the majority of features, meaning that changing one of them will affect a subset of the others, which must be updated correspondingly. Because most ML-based NIDS features are related to protocols, it is intuitive to have transport layer protocols as primary features.

Messages traveling a network without set rules or procedures would be unformatted and may not be intelligible to the receiving device. Protocols will be the conditionals on the values of other secondary features, such as flags and services. The following is a description of this relationship [18]:

$$\forall \mathbf{x} \in \mathbb{X} : \mathbf{x}_p \Rightarrow (\mathbf{x}_1 \in \mathbb{Y}_1) \wedge (\mathbf{x}_2 \in \mathbb{Y}_2) \wedge ... \wedge (\mathbf{x}_n \in \mathbb{Y}_n)$$

Where $\mathbf{x}$ is an input, $\mathbb{X}$ is the dataset, $\mathbf{x}_p$ is the key feature, and $\mathbb{Y}_i$ indicates the range of values that the semantics of feature $\mathbf{x}_{i \in [1,n]}$ allow (where n is the number of features). A network limitation between TCP/UDP and service type can be represented as follows.

$$\forall \mathbf{x} \in \mathbb{X} : \mathbf{x}_{TCP} \Rightarrow \mathbf{x}_{service} \in [nntp, ssh, ftp\_data, smtp, ...]$$

$$\forall \mathbf{x} \in \mathbb{X} : \mathbf{x}_{UDP} \Rightarrow \mathbf{x}_{service} \in [bootps, tftp, ntp, snmp, ...]$$

B. Numerical Dependencies

While adversarial attacks often consider categorical dependencies, numerical dependencies remain underexplored. In this paper, we address these numerical dependencies, which can naturally emerge in raw data or through feature engineering. Examples include one-hot encoded features such as service type, connection type, and flags, as well as continuous values like duration (the time between the first and last packet), original bytes, received bytes, and packet counts. By

incorporating both categorical and numerical dependencies, we aim to provide a more comprehensive approach to adversarial attack generation within constrained domains.

To ensure the validity and realism of the adversarial examples, we apply a filtering process that enforces both categorical and numerical constraints, as described in Algorithm **??**. Specifically, binary features in the dataset are rounded to their nearest integer values; if a perturbation results in a binary feature taking on a value like $1+\epsilon$, it is rounded back to 1, and similarly for values near 0. This ensures that binary features remain within their valid set of $\{0, 1\}$. Continuous (floating-point) features are where adversarial perturbations have an effect, enhancing the realism of the generated examples.

In addition to the numerical dependencies we've addressed, the connection type exhibits one-hot encoded (OHE) categorical dependencies that must be respected. Specifically, for a set of binary variables $\{x_{i=1}^N\}$, where each $\mathbf{x} \in \{0, 1\}$ and N is the total number of unique categories, the following condition holds:

$$\sum_{i=1}^{N} x_i = 1$$

Here, x_i represents a categorical feature. For example, consider the feature *"protocol_type"*, which has categorical values "tcp", "udp", and "icmp". Using one-hot encoding, we create three new binary features: *"protocol_type_tcp"*, *"protocol_type_udp"*, and *"protocol_type_icmp"*. If the protocol type is TCP, then *"protocol_type_tcp"* is set to 1, and both *"protocol_type_udp"* and *"protocol_type_icmp"* are set to 0. This same process applies to other categorical features like services and flags. To ensure data integrity, any adversarial example that violates this OHE restriction is filtered out. Additionally, binary fields impose further limits on data modification; when modifying binary values, only 0 or 1 can be used as the changed value.

Algorithm **??** acts as a filter for adversarial examples, retaining only valid samples that ensure both numerical and categorical dependencies for the TCP case.

IV. EXPERIMENTAL RESULTS

A. Evaluation Setting

1) Dataset: We utilize the established NSL-KDD dataset [19] to evaluate our approach, as it offers a balanced and diverse range of network traffic features, including categorical, binary, discrete, and continuous types. To prepare data, we employ standard preprocessing: one-hot encoding for categorical features and min-max normalization for overall stability and bias reduction. Importantly, we generate adversarial examples solely from the malicious traffic within NSL-KDD. This isolates the impact of adversarial techniques on misclassification, allowing us to directly assess how ML/DL-based NIDS models are affected within the specific context of intrusion detection.

Algorithm 1: Filter Adversarial Examples for Categorical and Numerical Dependencies in TCP Case

Input: - x_{adv}: Adversarial Examples (structured as a dataframe),
    - Service_tcp, Service_udp, Service_icmp: Lists of protocol-specific services,
    - Flag_tcp, Flag_udp, Flag_icmp: Lists of protocol-specific flags,
    - Binary_features: List of features expected to be binary

Output: Valid adversarial examples x_{valid}

1 **Convert** x_{adv} to dataframe $adversarial\_data$;
2 **Initialize** $x_{valid} \leftarrow []$;
3 **for** $i = 1$ **to** *size(x_{adv})[0]* **do**
4   Cast binary features and protocol type fields of $adversarial\_data_i$ to integer;
5   **if** $adversarial\_data_i[protocol\_type\_tcp] == 1$ **then**
6   Ensure $adversarial\_data_i[protocol\_type\_udp] = 0$ and $adversarial\_data_i[protocol\_type\_icmp] = 0$;
7   **for** *each feature in Binary_features* **do**
8   Ensure $adversarial\_data_i[feature] \in \{0, 1\}$;
9   **end**
10   **if** *any($adversarial\_data_i[service] == 1$) for $service \in Service\_tcp$* **then**
11   **if** *any($adversarial\_data_i[flag] == 1$) for $flag \in Flag\_tcp$* **then**
12   Append $adversarial\_data_i$ to x_{valid};
13   **end**
14   **end**
15   **end**
16 **end**
17 **return** x_{valid};

---

2) Attacks: We use a suite of seven well-established adversarial attack algorithms for comprehensive evaluation: Fast Gradient Sign Method (FGSM) [20], Carlini & Wanger (C&W) [21], Jacobian-Based Saliency Map Attack (JSMA) [22], DeepFool [23], Projected Gradient Descent (PGD) [24], Zeroth Order Optimization (ZOO) [25], and Basic Iterative Method (BIM) [26]. These techniques represent a diverse range of strategies for crafting adversarial perturbations.

First, we consider a multi-layer perception (MLP) pretrained on NSL-KDD dataset as a target model for adversarial attacks, encompassing both benign and malicious traffic samples. The MLP architecture consists of three hidden layers with 512, 256, and 64 units respectively, utilizing ReLU activations. To mitigate overfitting, dropout regularization is incorporated after each hidden layer (rate of 0.01). The output layer employs a Sigmoid activation, with Adam optimization and cross-entropy loss for the training process.

B. Attacks feasibility

Upon implementing domain constraints, the feasibility of existing adversarial instances is rigorously assessed through their projection onto the space of valid inputs. Adversarial instances deemed *invalid* are represented by samples prior to the application of any filtering mechanism. Figure 1 shows the proportion of valid and invalid adversarial examples produced by each attack method, highlighting notable differences.

The assessment of adversarial example validity reveals a critical weakness in existing attack methodologies: a disregard for domain-specific constraints. In particular, numerical constraints (e.g., bounded feature values) and categorical constraints (e.g., inter-feature relationships) are frequently violated. This leads to a significant proportion of generated adversarial instances being *invalid*, with our findings indicating rates as high as 80.3%. Consequently, evaluations based on such attacks may substantially overstate real-world vulnerabilities. The discrepancy stems from the fact that these adversarial examples, while effective in deceiving the model, do not constitute plausible inputs that an attacker could realistically generate in practice. As such, relying on these results could lead to an inflated perception of the model's susceptibility to adversarial manipulation, potentially misguiding resource allocation for defensive measures.

Among the evaluated attacks, JSMA exhibits the highest real-world practicality due to its targeted modification of a limited feature set. Conversely, PGD, FGSM, and BIM are less suitable for attacking ML-based NIDS given their propensity to modify all traffic attributes. While C&W yields the lowest rate of invalid adversarial instances, its effectiveness hinges on crafting minimally distorted data points closely resembling valid samples, leading to successfully engineered misclassifications.
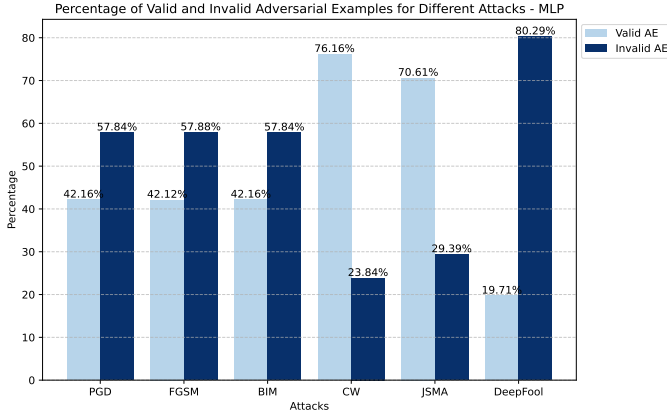

Fig. 1: Validity Percentage of AEs for different attacks.

For a good integration of ML in IDS, it's crucial to determine the resilience or susceptibility of constrained environments to adversarial attacks. By restricting adversarial examples to the domain of permissible inputs, we conduct a more rigorous and realistic evaluation of attack effectiveness.

Imposing domain constraints on adversarial examples significantly reduces classifier vulnerability, with severity rates decreasing by 52.49% to 99.84% across most attacks (Figure 2). This substantial reduction indicates that many adversarial perturbations become less effective when restricted to feasible inputs.

However, some attack methods remain relatively potent even under these constraints, likely due to their ability to make
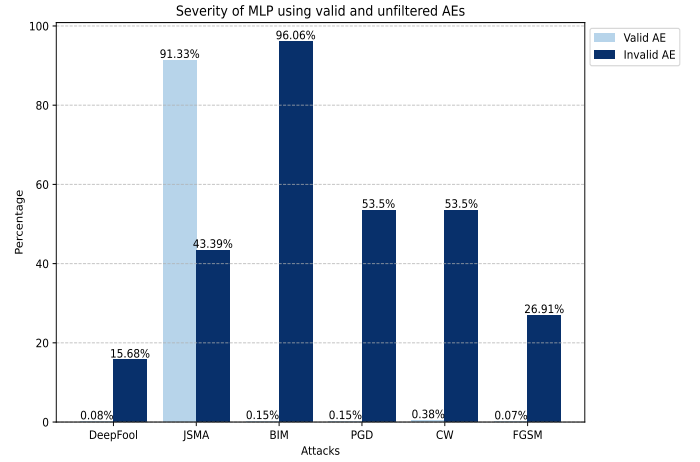

Fig. 2: Vulnerability assessment results with Valid and Invalid Adversarial Instances.

subtle, targeted changes that respect domain boundaries while still pushing inputs towards decision thresholds.

C. Effect of selecting a surrogate model in generating valid adversarial examples

The experimental results reveal significant variations in the generation of valid AEs across different model architectures and attack methods, as illustrated in Figure 3. The figure highlights how model complexity influences robustness against adversarial attacks within network domain constraints.

A clear pattern emerges: simpler models like the MLP exhibit a higher proportion of valid AEs across various attacks, including PGD, FGSM, and BIM. In contrast, more complex models, such as CNNs and those incorporating LSTM components, demonstrate increased robustness, with a noticeable reduction in the rate of valid AEs. For certain attacks (e.g., PGD, FGSM, BIM, and DeepFool), the CNN architecture did not yield any valid adversarial examples that complied with network domain constraints. This suggests that more intricate decision boundaries in complex models are more resistant to constrained perturbations, although some attack methods deviate from this trend, underscoring the need to consider both model architecture and specific attack characteristics in adversarial robustness assessments.

The results underscore the importance of selecting appropriate surrogate models in AML, especially when working within domain-specific constraints. More complex architectures could offer enhanced robustness in constrained environments. As the MLP consistently produced more valid AEs across most attacks, it was chosen for the transferability assessment in the subsequent section to evaluate the worst-case scenario regarding vulnerability to adversarial examples.
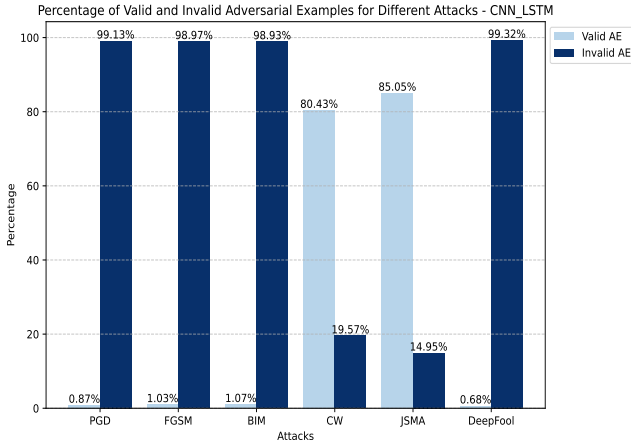
D. Attacks Transferability

Two critical factors influence the success of adversarial attack transferability: the inherent vulnerability of the target model and the sophistication of the surrogate model used for
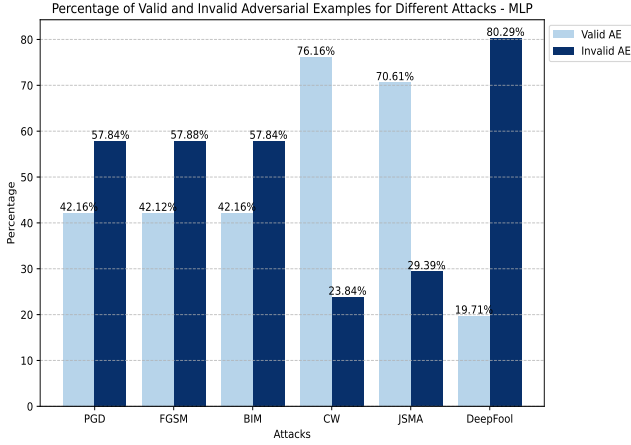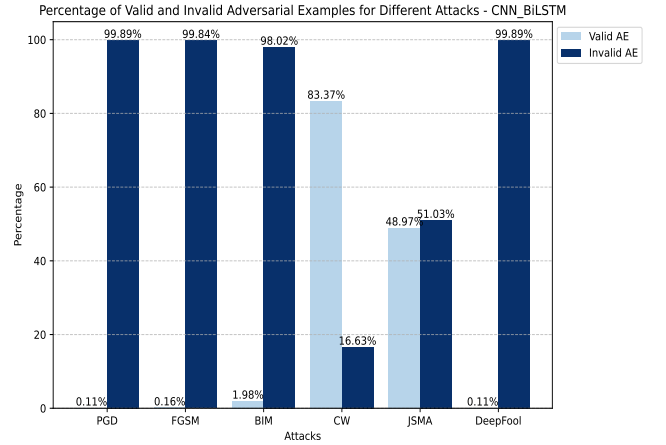
(a) MLP as surrogate model
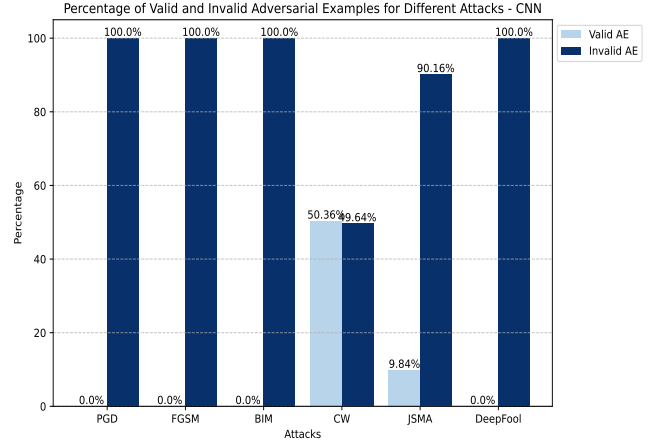
(b) CNN as surrogate model

(c) CNN-LSTM as surrogate model

(d) CNN-BiLSTM as surrogate model

Fig. 3: Effect of different surrogate models on generating valid adversarial examples

attack generation. We assess this transferability by quantifying the variation of accuracy and severity on target classifiers (SVM, Decision Tree, Random Forest, KNN, CNN variants) when exposed to adversarial examples crafted using an MLP surrogate model. Our findings, presented in Table I, reveal several insights.

The results indicate a sharp decline in attack severity after filtering, with traditional ML models showing reductions as high as 52.10% (SVM, PGD) and 44.18% (RF, Deep-Fool). However, KNN's increase in JSMA severity (-8.74%) suggests this attack can still exploit certain vulnerabilities post-filtering. DL models consistently exhibit high resilience, with significant drops, especially in FGSM and BIM, where reductions exceed 50% across CNN variants. These findings highlight the filtering process's effectiveness, especially for DL models, which appear more capable of neutralizing constrained adversarial attacks compared to traditional models.

V. CONCLUSION AND PERSPECTIVES

Crucial insights emerge from our investigation into the interplay between domain constraints and adversarial machine learning (AML) within cybersecurity and IoT contexts. Our findings demonstrate that domain constraints significantly reduce the space of potential adversarial examples, enhancing model robustness by hindering attackers' ability to craft deceptive perturbations within valid input ranges. This effect may be partially attributed to deep learning models' inherent capacity to distinguish realistic data points from anomalous inputs, possibly due to implicit regularization during training.

However, our research also exposes limitations in current AML algorithms when applied to cybersecurity, particularly in IoT environments where resource-constrained devices generate distinctive network traffic patterns. Rigorous evaluation within realistic IoT network constraints shows that most existing algorithms struggle to generate feasible and effective real-world adversarial examples. Paradoxically, constraining adversarial examples to valid input ranges often amplifies their impact on model vulnerability, particularly highlighting the effectiveness of black-box transferability attacks in tightly controlled IoT environments.

These results suggest a paradigm shift in AML research for cybersecurity, including IoT-specific applications. Future

TABLE I: Results of transferability

| | Severity | | | | | | | | | | | |
| | Before Filter | | | | | | After Filter | | | | | |
	JSMA	FGSM	DeepFool	C&W	PGD	BIM	JSMA	FGSM	DeepFool	C&W	PGD	BIM
SVM	36.32	36.09	26.36	35.84	53.80	51.84	25.55	1.61	1.70	25.55	1.70	1.70
DT	36.99	32.56	34.16	32.83	37.96	45.33	23.29	2.26	2.35	23.29	2.35	2.35
RF	41.61	42.11	48.45	39.20	48.03	48.03	4.27	4.18	4.27	31.39	4.27	4.27
KNN	35.39	35.77	38.65	35.29	38.42	37.65	44.13	0.76	0.85	25.75	0.85	0.85
CNN	23.83	23.19	41.32	8.53	54.38	54.69	9.98	0.07	0.08	1.92	0.16	0.16
CNN+LSTM	25.84	24.05	39.81	10.14	53.80	53.51	7.84	0.05	0.04	0.53	0.16	0.15
CNN+BiLSTM	23.05	13.73	17.42	10.53	37.92	31.47	5.14	0.04	0.00	0.44	0.16	0.15

work must prioritize developing and evaluating defense mechanisms specifically tailored to counter feasible adversarial examples within domain-constrained IoT environments. Additionally, the resilience of targeted, smaller-scale perturbation strategies like JSMA to domain constraints warrants further investigation. To comprehensively address these challenges, researchers could explore diverse datasets and implement proof-of-concept defenses within live IoT network traffic simulations. Such efforts are crucial for building more robust and resilient ML-based intrusion detection systems capable of withstanding sophisticated adversarial attacks while maintaining operational validity within real-world IoT constraints.

REFERENCES

[1] G. Andresini, F. Pendlebury, F. Pierazzi, C. Loglisci, A. Appice, and L. Cavallaro, "Insomnia: Towards concept-drift robustness in network intrusion detection," in *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, 2021, pp. 111–122.

[2] E. Alhajjar, P. Maxwell, and N. Bastian, "Adversarial machine learning in network intrusion detection systems," *Expert Systems with Applications*, vol. 186, p. 115782, 2021.

[3] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[4] E. Anthi, L. Williams, M. Rhode, P. Burnap, and A. Wedgbury, "Adversarial attacks on machine learning cybersecurity defences in industrial control systems," *Journal of Information Security and Applications*, vol. 58, p. 102717, 2021.

[5] X. Xiao, W. Xiao, R. Li, X. Luo, H. Zheng, and S. Xia, "Ebsnn: Extended byte segment neural network for network traffic classification," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 5, pp. 3521–3538, 2021.

[6] A. Grini, B. El Khamlichi, A. El Afia, and A. El Fallah-Seghrouchni, "Hpac-ids: A hierarchical packet attention convolution for intrusion detection system," in *2024 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2024, pp. 1–6.

[7] G. Apruzzese, M. Colajanni, L. Ferretti, and M. Marchetti, "Addressing adversarial attacks against security systems based on machine learning," in *2019 11th international conference on cyber conflict (CyCon)*, vol. 900. IEEE, 2019, pp. 1–18.

[8] N. Martins, J. M. Cruz, T. Cruz, and P. H. Abreu, "Adversarial machine learning applied to intrusion and malware scenarios: a systematic review," *IEEE Access*, vol. 8, pp. 35 403–35 419, 2020.

[9] M. J. De Lucia and C. Cotton, "Adversarial machine learning for cyber security," *Journal of Information Systems Applied Research*, vol. 12, no. 1, p. 26, 2019.

[10] K. Yang, J. Liu, C. Zhang, and Y. Fang, "Adversarial examples against the deep learning based network intrusion detection systems," in *MILCOM 2018-2018 ieee military communications conference (MILCOM)*. IEEE, 2018, pp. 559–564.

[11] M. Rigaki, "Adversarial deep learning against intrusion detection classifiers," 2017.

[12] Z. Lin, Y. Shi, and Z. Xue, "Idsgan: Generative adversarial networks for attack generation against intrusion detection," *arXiv preprint arXiv:1809.02077*, 2018.

[13] A. Kumar *et al.*, "Nids-cbad: Constraint-based adversarial detection in network intrusion detection systems," *Journal of Network and Systems Management*, 2024.

[14] R. Sharma and Y. Chen, "A systematic study of adversarial attacks against ml-based network intrusion detection systems in iot environments," *Electronics*, vol. 13, no. 24, p. 5030, 2024.

[15] A. Venturi and C. Zanasi, "On the feasibility of adversarial machine learning in malware and network intrusion detection," in *2021 IEEE 20th International Symposium on Network Computing and Applications (NCA)*. IEEE, 2021, pp. 1–8.

[16] M. J. Hashemi, G. Cusack, and E. Keller, "Towards evaluation of nidss in adversarial setting," in *Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks*, 2019, pp. 14–21.

[17] M. Teuffenbach, E. Piatkowska, and P. Smith, "Subverting network intrusion detection: Crafting adversarial examples accounting for domain-specific constraints," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, 2020, pp. 301–320.

[18] R. Sheatsley, N. Papernot, M. Weisman, G. Verma, and P. McDaniel, "Adversarial examples in constrained domains," *arXiv preprint arXiv:2011.01183*, 2020.

[19] L. Dhanabal and S. Shantharajah, "A study on nsl-kdd dataset for intrusion detection system based on classification algorithms," *International journal of advanced research in computer and communication engineering*, vol. 4, no. 6, pp. 446–452, 2015.

[20] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[21] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. IEEE, 2017, pp. 39–57.

[22] 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)*. IEEE, 2016, pp. 372–387.

[23] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[24] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[25] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 15–26.

[26] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.