Realism versus Performance for Adversarial Examples Against DL-based NIDS

Huda Ali Alatwi*
Newcastle University
Newcastle Upon Tyne, United Kingdom
h.a.o.alatwi2@newcastle.ac.uk

Charles Morisset
Newcastle University
Newcastle Upon Tyne, United Kingdom
charles.morisset@newcastle.ac.uk

ABSTRACT

The application of deep learning-based (DL) network intrusion detection systems (NIDS) enables effective automated detection of cyberattacks. Such models can extract valuable features from high-dimensional and heterogeneous network traffic with minimal feature engineering and provide high accuracy detection rates. However, it has been shown that DL can be vulnerable to adversarial examples (AEs), which mislead classification decisions at inference time, and several works have shown that AEs are indeed a threat against DL-based NIDS. In this work, we argue that these threats are not necessarily realistic. Indeed, some general techniques used to generate AE manipulate features in a way that would be inconsistent with actual network traffic. In this paper, we first implement the main AE attacks selected from the literature (FGSM, BIM, PGD, NewtonFool, CW, DeepFool, EN, Boundary, HSJ, ZOO) for two different datasets (WSN-DS and BoT-IoT) and we compare their relative performance. We then analyze the perturbation generated by these attacks and use the metrics to establish a notion of "attack unrealism". We conclude that, for these datasets, some of these attacks are performant but not realistic.

CCS CONCEPTS

• **Security and privacy** → *Intrusion detection systems*; • **Computing methodologies** → *Neural networks*;

KEYWORDS

Network Intrusion Detection, Deep Neural Networks, Adversarial Examples, Evasion Attacks, Adversarial Machine Learning, Network Security

ACM Reference Format:
Huda Ali Alatwi and Charles Morisset. 2023. Realism versus Performance for Adversarial Examples Against DL-based NIDS. In *Proceedings of ACM SAC Conference (SAC'23).* ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/https://doi.org/10.1145/3555776.3577671

*Also with Tabuk University, Tabuk, Saudi Arabia (Email: *haladheedi@ut.edu.sa*).

SAC'23, March 27 – March 31, 2023, Tallinn, Estonia

ACM ISBN 978-1-4503-9517-5/23/03…$15.00
https://doi.org/https://doi.org/10.1145/3555776.3577671

1 INTRODUCTION

Machine learning-based (ML) applications have been utilized to provide different security solutions, such as malware detection [6], spam detection [14], network intrusion detection system (NIDS) [11]. The recent integration of Deep Learning (DL) with automated feature engineering has improved both the effectiveness and efficiency of NIDS models [5]. However, DL-based models can be vulnerable to so-called Adversarial Examples (AEs), inputs crafted to induce misclassification at training or inference times [39]. This vulnerability can be exposed by injecting malicious inputs to subvert the learning process (poisoning attack), adding slight perturbations to the original inputs in order to cause misclassification (evasion attack), or probing the model to obtain confidential information (oracle attack) [3].

The current literature employs generic adversarial examples generation approaches to assess the resilience of DL-based NIDSs against adversarial traffic [1, 2]. However, these approaches were mainly designed for unconstrained domains (i.e., image processing). For image recognition, the adversary can perturb any arbitrary amount of pixels, and the features can be amended independently. For NIDS, the traffic data must preserve some domain constraints which restrict how features can be perturbed, e.g., interdependence between the values of several features, features with fixed values, features with a limited range of values [5].

We assess the performance of these attacks along with their realism in terms of the compliance of their outputs with the traffic domain constraints. The assessment was conducted for the attacks in two setups, targeted and untargeted, against multi-classification DL-based NIDSs. We use the *Unrealism Index* metric, which is an average of percentages of void AEs and features perturbation to measure the realism of the attack. Our contributions are:

(1) We establish a list of 11 AEs attacks against NIDS (7 white-box and 4 black-box) based on the critical analysis of 15 academic papers (Section 3)
(2) We introduce a new notion of Unrealism Index (Section 4.4).
(3) We implement these attacks against a multi-classifier DL-based NIDS for two different datasets, demonstrating and comparing their relative performances without constraints (Section 5.1).
(4) We finally analyse the realism of these attacks network by taking traffic constraints into account (Section 5.2).

2 BACKGROUND AND RELATED WORK

In this section, we first outline the main DL-based schemes for NIDS. Then, we present the fundamental principles of adversarial machine learning. Lastly, we describe the network traffic constraints that must be maintained to generate valid adversarial flow.

2.1 Deep-Learning based Network Intrusion Detection Systems

Machine learning approaches have been used for Anomaly-based NIDS to detect anomalies and novel attacks by comparing the network packets to a profile of benign traffic to discover any deviation from the norm. Shallow ML approaches demand labeling training datasets and manual feature engineering by domain experts, which are difficult and time-consuming tasks [4]. Therefore, they are impractical in the real world due to the high dynamicity and large scale of modern networks. On the other hand, DL techniques can automatically extract different representations for feature learning at different processing layers. They provide end-to-end problem anomaly detection solutions because they can discover complex correlations and map raw input directly to the output [4]. DL-based architectures for NIDS are classified into three major groups discriminative, generative, or hybrid [4]. Discriminative or supervised models learn the decision boundaries between the classes from labeled training datasets and are used for traffic classification tasks. These models include Deep Neural Network (DNN) [15, 18, 19, 27, 28, 33, 38, 40–42] and Convolutional Neural Network (CNN) [17, 18, 20, 27, 38]. Generative or unsupervised models learn the probability distribution of each class from unlabeled training datasets, and they can be used for clustering and dimensionality reduction. These models include Auto-Encoder (AE) [20], Restricted Boltzmann Machine (RBM) [25], Deep belief Network (DBN) [36], and Recurrent Neural Network (RNN) [17, 18, 25, 38]. Hybrid architectures combine generative and discriminative models, such as Generative Adversarial Networks (GAN) [35].

2.2 Adversarial Machine Learning Fundamentals

DL models can be misled towards incorrect decisions with high confidence because of intently crafted perturbations added to the original inputs, so-called **Adversarial examples (AEs)** [12, 23]. In other words, an adversarial example is a data instance with tiny intentional feature perturbations that deceive the machine learning models and cause them to make false classification decisions. Most of adversarial examples generation approaches insert a calculated perturbation (γ) to the original input (x) to produce a new version (x^*) (i.e., adversarial input) while reducing the distance between the original input (x) and the adversarial one (x^*), and shifting the classification decision to the aimed adversarial outcome [10, 12, 23]. The robustness of AEs generation techniques depends on their ability to produce AEs as close as possible to the original examples. Adversarial attacks are mainly categorized into poisoning attacks, evasion attacks, and inference attacks [12]. In **poisoning attacks**, the adversary inserts adversarial examples into the training data to degrade the model performance after deployment. In **evasion attacks**, the adversary manipulates the inputs to deceive the model and induce misclassification decisions. In **oracle attacks**, the adversary crafts adversarial inputs to observe model outputs. The collected pairs of inputs and correspond outputs are used to build a substitute model that maintains most of the targeted model functionality. The adversary then can design costumed attacks over the substitute model that can transfer to the targeted model.

2.3 Network Traffic Constraints

Adversarial examples generation techniques were initially intended for unconstrained domains (e.g., image recognition). In such domains, the features are independent and can be perturbed arbitrarily. However, network traffic features are constrained by some characteristics such as [5]:

- Every feature can have a continuous, categorical, or binary value.
- The values of some features can be highly interdependent and correlated.
- The values of some features can be constant and unmodifiable.

The binary feature can take either 1 or 0, the categorical feature takes a value that belongs to one category at once, and the numeric feature can only take a value within the allowed range. For instance, some features are linearly related, and others are immutable, such as protocol type or connection flag. The adversarial perturbations must maintain the above constraints to generate valid and functional flow.

3 MAJOR AES ATTACKS AGAINST NIDS

This section outlines and discusses previous studies that employed adversarial evasion techniques to prove their effectiveness in evading and degrading the performance of DL-based NIDS models. The current literature states the vulnerability of detection models to generic evasion adversarial attacks and regards them as significant threats. However, these studies did not verify the practicality of the generated adversarial traffic for real-world attacks.

Yang et al. [42] employed three black-box attacks: a substitute model, Wasserstein Generative Adversarial Networks (WGANs), and ZOO, to induce a DNN classifier to misclassify the attack traces as normal traffic. Despite the realism of the attack outputs not being investigated, the employed attacks suffer from practicality issues. The substitute model attack fails to manifest the functionality and structure of the targeted model fully. On the other hand, ZOO is computationally extensive as it requires querying the targeted model for estimating the gradients. In a real-world scenario, the NIDS limits the number of queries the adversary can make to prevent potential suspicious probing attempts. Finally, GANs still suffer from unstable training, model collapse, vanishing gradients, and convergence failure.

Warzyński and Kołaczek [41] showed that the FGSM attack completely compromised a DNN binary classifier over the NSL-KDD dataset. Accordingly, they confirmed that the FGSM attack, designed for image recognition, can be applied to the network traffic domain. Clements et al. [13] claimed the vulnerability of DL-based NIDS to AEs through an assessment of the robustness of Kitsune, a lightweight DL-NIDS for IoT networks, to FGSM, CW, and ENM attacks using the Mirai dataset. Wang [40] indicated that different levels of effectiveness were achieved by FGSM, DeepFool, and CW attacks and identified their feature pattern usages. The author claimed that the most selected features to be perturbed by these techniques could contribute more to the vulnerability of DL-based NIDS to adversarial traffic. However, the study did not analyze how these features were being manipulated to verify whether the perturbations resulted in consistent traffic instances.

The attacks were carried out against an MLP classifier over the NSL-KDD dataset.

Peng et al. [33] demonstrated a drop in the performance of DNN, SVM, RF, and LR classifiers against PGD, MI-FGSM, L-BFGS, and SPSA attacks over the NSL-KDD dataset. Ibitoye et al. [19] compared the performance of Self-Normalizing Neural Networks (SNN) and DNNs under the FGSM, BIM, and PGD attacks using the BoT-IoT dataset. The authors concluded that while DNNs outperformed SNNs in the accuracy rate, the SNNs were more resilient to AEs. Jeong et al. [20] analyzed the efficiency of Autoencoder and CNN under FGSM attack over the NSL-KDD dataset.

Huang et al. [18] assessed the efficiency of three port-scan attack detecting models for SDN environments: MLP, CNN, and LSTM under the FGSM attack. Martins et al. [28] showed a deterioration in the mean performance of DT, RF, SVM, NB, NN, and DAE classifiers under FGSM, DeepFool, and CW attacks. Sriram et al. [38] analyzed the performance of LSTM, SMR, CNN, AB, DNN, RF, SVM, RF, NB, DT, KNN, and LR classifiers against FGSM attack using the NSL-KDD dataset.

Piplai et al. [35] leveraged GANs as a defensive mechanism against AEs, and to solve the class imbalance that is common in network traffic datasets. GANs involve training two neural networks competing each other, where the Generator craft AEs to deceive the Discriminator. However, their experimental results demonstrated that the FGSM attack defeated the GANs classifier. Debicha et al. [15] concluded that the FGSM, BIM, and PGD attacks significantly deteriorated the performance of a DNN detection model. Maarouf et al. [27] compared the resilience of C4.5, KNN, ANN, CNN, and RNN traffic classifiers against ZOO, PGD, and DeepFool attacks using the SCX VPN-NonVPN and NIMS datasets. The authors concluded that DL models are more robust to AEs than conventional ML ones.

Pacheco and Sun [32] claimed the feasibility of FGSM and CW attacks to reduce the performance of DT, SVM, and RF classifiers over the BoT-IoT and UNSW-NB15 datasets. Fu et al. [17] assessed the robustness of CNN, LSTM, and Gated (GRU) to the FGSM attack over the CICIDS2018 dataset. Merzouk et al. [30] analyzed the outputs of FGSM, BIM, DeepFool, and CW attacks. The experimental evaluation was carried over a binary MLP classifier using the NSL-KDD dataset. The scope of this study was relatively narrow, being primarily concerned with a few white-box attacks implemented for a binary classifier and in the untargeted setup only.

The previous studies focused on compromising DL-based NIDS using generic evasion adversarial attacks. Although these attacks might result in AEs with high Evasion Rates, the realism of these attacks was not considered.

Furthermore, most studies focused on the impact of adversarial attacks in traditional IP networks [15, 17, 20, 28, 30, 33, 35, 38, 40–42]. On the other hand, security risks in other networking environments such as WSN, SDN, and IoT must be assessed as they are emerging and expanding over the upcoming years. Less research considered these different contexts [13, 18, 19, 32]. Moreover, the outdated dataset NSL-KDD was used by most of the studies [15, 20, 30, 33, 38, 40–42]. This dataset is ideal and outdated and does not reflect modern network complexity. Additionally, the studies did not verify their experimental results by utilizing an additional network dataset in order to prove the effectiveness of the adversarial evasion attacks among different datasets [13, 15, 18–20, 20, 30, 33, 35, 38, 40–42].

Recent research in malware detection has proposed adversarial problem-space attacks that preserve the malicious functionality [16, 26, 34, 37]. Similarly, in the network anomaly detection domain, techniques such as Generative Adversarial Networks and Reinforcement Learning have been leveraged to design functionality and domain constraints preserving adversarial attacks for network traffic [29]. However, the scope of this study is to assess the validity of the generic adversarial attacks as they are still being used to assess the robustness of DL-based NIDS to adversarial attacks.

Based on the previous related works, no research has verified the realism of adversarial flow and its compliance with network domain constraints for a multiclass DL-based NIDS. Furthermore, we validated the outputs of widely used white-box and black-box evasion attacks in targeted and untargeted setups over two recent traffic datasets representing different contemporary networking contexts. Table 1 demonstrates the attacks used by the literature to assess the resilience of a wide spectrum of conventional and DL-based NIDS and whether domain constraints verification was conducted for produced adversarial network traffic or not.

Ref. \ Attacks	White-Box							Black-Box			C. V.?
	BIM	CW	DeepFool	FGSM	NewtonFool	PGD	ZOO	HSJ	EN	Boundary	
[41]				✓							
[40]			✓	✓							
[42]							✓				
[13]		✓		✓					✓		
[18]				✓							
[19]	✓			✓		✓					
[28]		✓	✓	✓							
[33]						✓					
[30]	✓	✓	✓	✓							✓
[20]				✓							
[38]				✓							
[32]		✓		✓							
[15]	✓			✓		✓					
[17]				✓							
[27]			✓			✓	✓				
Ours	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison with Related Works

4 EXPERIMENTAL SETUP

This section provides detailed information about the implementation of the experiments. It describes the used datasets, dataset preprocessing procedure, attack implementation, and targeted model architecture. Then, we explain the used metrics to evaluate the performance of the detection model, and realism of the crafted adversarial traffic by measuring its compliance with network domain constraints.

4.1 Datasets

We utilized two recent datasets that represent advanced networking environments a wireless sensor network (WSN-DS) and an IoT network (BoT-IoT). **WSN-DS** contains 374,661 records representing normal traffic and four different types of DoS attacks, namely: flooding, TDMA, grayhole, and blackhole [7]. **BoT-IoT** compromises a approximately 3.6 million records in its proposed scaled version of the best 10 features [22]. It represents normal IoT network traffic

and various attacks that include DDoS, DoS, Keylogging, Data exfiltration, OS and Service Scan. Table 2 shows the features we used in both datasets for the experiments.

Features	WSN-DS	BoT-IoT
Binary	is_CH, JOIN_S, Join_R	N/A
Numeric	Who_CH, Dist To CH, Consumed_Energy, ADV_S, ADV_R, SCH_S, SCH_R, Rank, DATA_S, DATA_R, Data_Sent_To_BS, dist_CH_To_BS, send_code,	Seq, state_number, Stddev, Min, Max, Srate, Drate, N_IN_Conn_P_SrcIP, N_IN_Conn_P_DstIP
Categorical	N/A	Proto

Table 2: Used Features in the Datasets

4.2 Dataset Pre-processing

Data pre-processing is a crucial stage to convert raw inputs into an understandable format by the ML algorithm. The first step in this stage is **One-Hot Encoding** which converts categorical features into numeric ones. We applied this step to the BoT-IoT dataset. However, there was no need for one-hot encoding for the WSN-Ds as the dataset does not contain categorical features. The second step is standardization, in which all numeric features are converted into a standard scale. **Min-Max Normalization** was used to transform the features into a scale between 0 and 1. This step is essential as the dataset has feature values with different scales drawn from different distributions or tainted by outliers. Furthermore, normalization prevents the features with large values from dominating others which causes imbalanced results. This method converts the maximum value into 1, the minimum value into 0, and the other values into decimals between 0 and 1. It is calculated via the following equation:

$$X(norm) = \frac{X - X(min)}{X(max) - X(min)} \tag{1}$$

where X denotes the feature value, X_{min} the minimum feature value, X_{max} the maximum feature value.

4.3 Adversarial Attacks & Target Model

The Adversarial Robustness Toolbox (ART) library[31] was used to generate the AEs using the examined approaches and with the default parameters. The target model was a Feed-Forward Deep Neural Network (FF-DNN) implemented using the Keras library with a TensorFlow backend. The architecture of the model and the training parameters are demonstrated in Table 3. The implementation of the experiments was conducted on a Google Colab Notebook, and it is publicly available on Google Colab[1].

4.4 Evaluation Metrics

We selected Evasion Rate (ER) as the primary metric to evaluate the performance of the attack. ER refers to the proportion of perturbed attack instances misclassified as benign by the detection model. The higher achieved ER by the approach indicates a more performant attack.

$$ER = \frac{Misclassified\ Attacks\ Records}{Total\ Attacks\ Records} \times 100 \tag{2}$$

[1]https://colab.research.google.com/drive/1sRty5Is-iFazdgOuuafg6v$_D$oOI9NNHc?usp=sharing

Parameter	Value
No. of hidden layers	3
Layer 1	128 neurons
Layer 2	64 neurons
Layer 3	32 neurons
Dropout	0.25
Optimizer	ADAM
Activation function	ReLU and Sigmoid
Learning rate	0.01
Epoch	100
Batch Size	64

Table 3: Feed-Forward DNN Model Parameters

To measure the realism of the attack, we consider three metrics.

- Approaches producing adversarial examples by manipulating all the features are unlikely to lead to realistic attacks; the adversary cannot have control over all of the traffic features to change them in a fine-grained manner. Furthermore, such massive manipulation breaks the semantic links between the correlated features. We introduce the metric *PF* measuring feature perturbation:

$$PF = \frac{Average\ of\ Perturbed\ Features}{Total\ Features} \times 100 \tag{3}$$

- Adversarial examples that do not comply with the network domain constraints given in Section 2.3, e.g. by introducing out-range values to the continuous features, assigning non-binary values to the binary features, and triggering multiple categories at once for categorical features, are unlikely to correspond to realistic traffic. We introduce a generic metric VAE_b:

$$VAE_c = \frac{Attacks\ Records\ violating\ b}{Total\ Attack\ Records} \times 100 \tag{4}$$

  We consider in the following VAE_{oor}, VAE_{nb} and VAE_{mc} for the constraints out-range, non-binary and multi-categories, respectively.
- The Unrealism Index (UI) is calculated by averaging the metrics above that are relevant to a particular dataset.

$$UI_{WSN-DS} = \frac{PF + VAE_{or} + VAE_{nb}}{3} \times 100 \tag{5}$$

$$UI_{BoT-IoT} = \frac{PF + VAE_{or} + VAE_{mc}}{3} \times 100 \tag{6}$$

5 EXPERIMENTAL RESULTS & ANALYSIS

In this section, we report and analyze the outcomes of executing the attacks in targeted (T) and untargeted (U) setups over the two datasets. Table 4 shows assessment results of attacks performance and unrealism over the two datasets, using the metrics introduced in the previous section, presented in decreasing order based on Evasion Rate. For further investigation, the attacks outputs are available on Google Colab[2].

Figures 1 and 2 demonstrate the correlation between Evasion Rate and Unrealism Index of the attacks over the WSN-DS and BoT-IoT datasets, respectively. Figures 3 and 4 demonstrate the average percentages of void adversarial examples for each validation metric

[2]https://colab.research.google.com/drive/1sRty5Is-iFazdgOuuafg6v_DoOI9NNHc?usp=sharing

	Attack	Setup	WSN-DS					BoT-IoT					Avg.	
			ER	PF	VAE		UI_{WSN-DS}	ER	PF	VAE		$UI_{BoT-IoT}$	ER	UI
					VAE_{or}	VAE_{nb}				VAE_{or}	VAE_{mc}			
-	Clean	-	2	0	0	0	0	0.06	0	0	0	0	1.03	0
White-box	BIM	T	84.59	92.69	100	98.79	97.16	45.43	85.27	98.38	93.73	92.46	65.01	94.81
	PGD	T	84.59	92.69	100	98.79	97.16	45.43	85.27	98.38	93.73	92.46	65.01	94.81
	CW2	T	36.36	44.94	0	34.36	26.43	0.89	33.53	0.47	0.83	11.61	18.63	19.02
	FGSM	T	12.51	100	100	100	100	3.85	100	100	100	100	8.18	100
	CW	T	2	98.5	0	97.92	65.47	0.06	99.8	4.03	99.94	67.92	1.03	66.7
	NewtonFool	U	22.2	90.63	85.04	89.59	88.42	4.76	90.33	90.47	90.43	90.41	13.48	89.42
	BIM	U	19.77	91.38	92.9	92.9	92.39	3.83	89.93	94.05	94.05	92.68	11.8	92.54
	PGD	U	19.77	91.38	92.9	92.9	92.39	3.83	89.93	94.05	94.05	92.68	11.8	92.54
	CW2	U	20.26	70.44	0	63.3	44.58	0.01	55.13	1.11	32.45	29.56	10.14	37.07
	DeepFool	U	7.02	93.13	87.61	92.88	91.21	0.35	94.33	94.04	94.05	94.14	3.69	92.68
	FGSM	U	4.81	94.25	92.9	92.88	93.34	0.5	95.8	94.05	94.05	94.63	2.66	93.99
	CW	U	4.22	22	0	3.94	8.65	0.03	43.27	1.85	15.98	20.37	2.13	14.51
Black-box	EN	T	85.02	50.06	0	42.59	30.88	61.65	49.93	0	43.49	31.14	73.34	31.01
	HSJ	T	100	92.69	0	98	63.56	23.32	47.2	4.14	23.31	24.88	61.66	44.22
	Boundary	T	91.79	95.31	0	98	64.44	20.38	48.73	1.3	24	24.68	56.09	44.56
	ZOO	T	1.41	1.81	0.59	0.59	1	0.01	1.07	0.04	0.04	0.38	0.71	0.69
	HSJ	U	28.1	85.88	0	99.16	61.68	0.48	86.07	58.72	99.74	81.51	14.29	71.6
	Boundary	U	26.89	91.25	0	99.99	63.75	0.98	93.93	53.4	100	82.44	13.94	73.1
	EN	U	10.36	34.44	0	5.77	13.4	1.01	41.2	0	14.33	18.51	5.69	15.96
	ZOO	U	6.86	13.06	13.89	13.96	13.64	0.15	50.07	55.1	52.94	52.7	3.51	33.17

Table 4: Attacks Assessment Results over WSN-DS & BoT-IoT Dataset

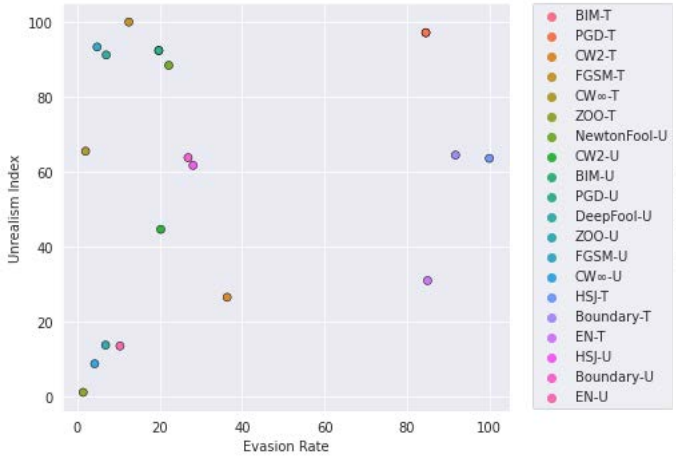

Figure 1: Evasion Rate vs. Unrealism Index over WSN-DS

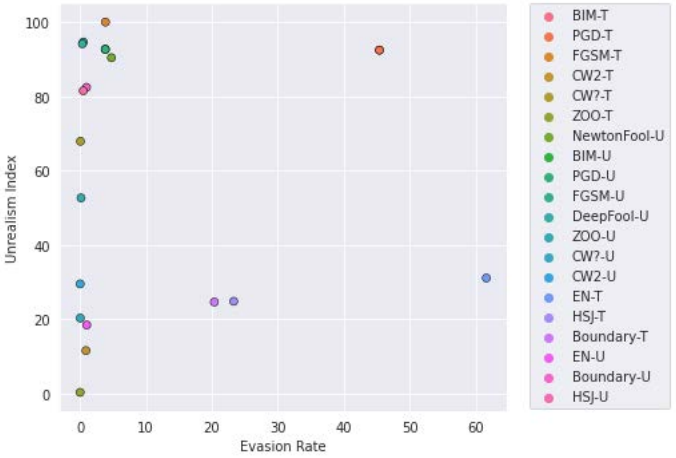

Figure 2: Evasion Rate vs. Unrealism Index over BoT-IoT

over the two datasets for white-box and black-box attacks, respectively. The suffixes (-T) and (-U) were added to the attack names to refer to the attacks in targeted and untargeted setups, respectively. Lastly, Table 5 displays the validation metrics that were violated by the attacks over the two datasets. The results were presented in descending order based on the number of violated validation metrics scored by the attacks over the two datasets.

5.1 Performance

As reported in Table 4 the model over the BoT-IoT dataset recorded less ER of 0.06 on the clean attack instances compared to the WSN-DS model which scored an ER of 2 as shown in Table 4. This difference can be justified by the imbalance of normal and attacks data distribution between the two datasets. The attack instances are the majority of the BoT-IoT dataset records with a percentage of 99%, while they are the minority in the WSN-DS dataset with a percentage of 9.2%. However, both models were able to detect the clean attack traces with high accuracy.

From Table 4, we can see the variation in the attack effectiveness in terms of ER over the two datasets. Overall, we can observe that the performance of each attack depend on the dataset type. What stands out in Figures 1 and 2 is that the attacks overall achieved higher evasion rates over the WSN-DS compared to the BoT-IoT. The attacks over WSN-DS achieved ERs between 100-1.44 and 61.65-0.01 over BoT-IoT. The white-box and the black-box attacks performed better in both setups, targeted and untargeted, over the WSN-DS dataset compared to the BoT-IoT. Two reasons can justify that; first, the proportion of benign traffic instances constitutes about 91% of the WSN-DS dataset, which enriches learning the characteristics of normal flow behavior by the targeted attacks. The second reason can be attributed to the number and datatype in a dataset. The WSN-DS dataset consists of binary and continuous

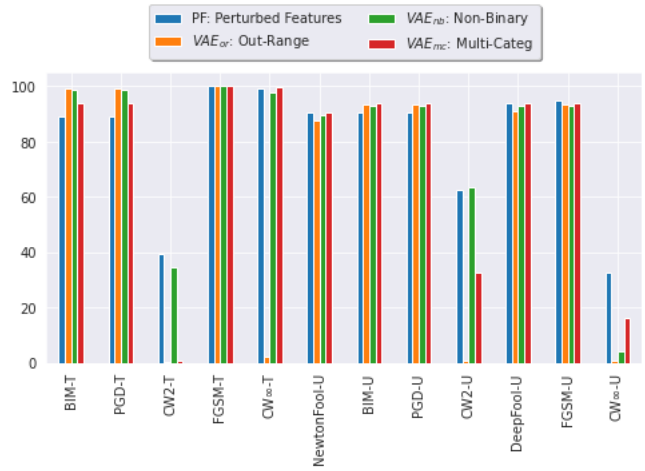

Figure 3: White-box Attack Unrealism Metrics Percentages

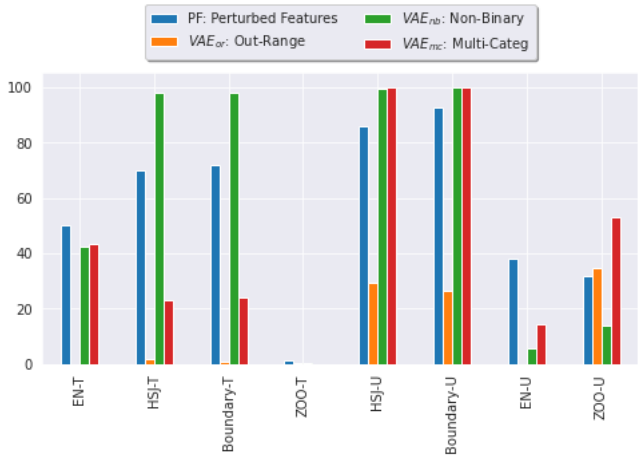

Figure 4: Black-box Attack Unrealism Metrics Percentages

features represented in numbers. The attacks could introduce any arbitrary numbers to these fields with the possibility of generating successful evasive examples. However, the BoT-IoT dataset contains continuous features and a categorical feature (*proto*) as shown in Table 2. A categorical feature can take one value from a finite set of possible values. The proto feature can be a value from a set of five values i.e., *arp, tcp, udp, icmp, and ipv6-icmp*. After one-hot encoding, this feature is represented in a binary vector in which only the corresponding category is assigned to 1, and the others are zeros. The attacks spread their perturbations to all features and introduce arbitrary numbers to fields belonging to a categorical feature that must be zeros, and only one of them can be 1. Such massive perturbation results in corrupted examples that cannot evade the detection model and are easily detected. This explains the terrible performance of the attacks over the BoT-IoT dataset.

It is apparent from Table 4 that the BIM and PGD attacks are the top performing white-box attacks over the two datasets with ERs of 65 and 12 in the targeted and untargeted setups, respectively. The results of our experiments support the findings of previous research that has demonstrated that multi-step (iterative) perturbation strategies such as PGD and BIM are among the strongest attacks compared to the single-step attack (e.g., FGSM) [24]. The multi-step adversarial perturbation generation is an extension of the single-step method in which it iteratively adds a perturbation that follows the sign of the gradient with respect to the current adversarial example of the original input [24]. The PGD attack is similar to BIM. The differences are that PGD adds more iterations and uses random initialization. Because of that, they had the same effect on the detection model as shown in Table 4. Although other studies support the same finding of us [15, 21], the BIM attack was reported as performing better than the PGG in [19].

What is striking in Table 4 is that the black-box attacks performed better than the white-box, with the EN being the best. Although the EN is optimized to limit total perturbation across feature-space inputs, it minimizes the number of perturbed features. Therefore, the high effectiveness of this attack can be attributed to its ability to produce AEs with minimal perturbation. As a consequence, the resulting examples become very close to the original examples and can successfully fool the detection model. The Boundary and HSJ reported closer ERs. That can be justified by the fact that they are both from the same family of decision-based attacks, with HSJ is an extension of the Boundary attack.

5.2 Unrealism

5.2.1 *Percentage of Features Perturbation*.

As shown in Figure 3 the majority of white-box attacks manipulated on average above 85% of the features over the two datasets including BIM, PGD, CW, and NewtoonFool which were the best performing. However, CW2-T and CW∞-U altered around 35% of the features. Though, as shown in Table 4 they reported low ERs of 18.63 and 2.13, respectively. Most of the black-box attacks manipulated around 62% of the feature, as can be observed from Figure 4. Although the ZOO-T, ZOO-U, and EN-U attacks were the least, they scored low ERs between 0.71-5.69 (Table 4).

From these results, it is clear that these attacks altered a vast amount of the traffic features by over 50%. The unrealism of the attack can be attributed to the infeasibility of accessing and controlling such amount of features by the adversary in real-world scenarios. Furthermore, massive features modification will break the semantic links among the features, invalidating traffic traces realism.

5.2.2 *Out-Range Values*.

It is apparent from Figure 3 that the white-box attacks except CW2 and CW∞ generate above 90% AEs that hold out-range values. The CW2 and CW∞ produce between 0.24-2.02 void AEs. However, their ERs range between 18.63-1.03, as can be seen in Table 4. The targeted black-box attacks produced on average less than 2.07% AEs with out-range values, and 23% in the untargeted setup, as shown in Figure 4.

Each feature in traffic can take a value within a limited range of possible values. For instance, the *Rank* feature in WSN-DS has originally a range of values between [0,99] which scaled to [0,1] using min-max normalization for model training. However, the

BIM-T attack, best performing, introduced values [-0.3,0.57] for that feature which do not comply with the [0,1] range.

5.2.3 *Non-Binary Values*.

As can be seen in Figure 3, the majority of white-box attacks introduced non-binary values to the binary features with percentages of void AEs above 90%, except CW2 and CW∞-U. In the back-box attacks, Boundary and HSJ generated above 98% void AEs. The ZOO attack produced the lowest percentage of void AEs. However, it achieved averaged ERs between 0.71 and 3.51 in targeted and untargeted setups, respectively.

The WSN-DS dataset contains three binary features: *Is_CH, JOIN_S, JOIN_R* which can take only 0 or 1, as shown in Table 2. However, the BIM-T attack introduced values between [-0.1,1.3], [-0.3,1.1], [-0.3,1.09] for those features, respectively. For instance, the *Join_S* feature in the WSN-DS dataset denotes whether the join request was sent from the sensor node to the head of the cluster, which can be True or False, i.e., a flag value of 1 or 0. Assigning a decimal or negative value to this feature makes no sense.

5.2.4 *Multi-Category Belonging Values*.

It is apparent from Figure 3, that the majority of white-box attacks triggered multiple categories at once for the categorical feature for above 90% of AEs, except CW2 and CW∞-U. In the untargeted back-box attacks, Boundary and HSJ generated almost 100% void AEs. The other attacks produced less than 50%.

The BoT-IoT dataset includes a categorical feature *proto* as shown in Table 2. A categorical feature contains a limited number of possible values. The *proto* feature has five values i.e., *arp, tcp, udp, icmp, and ipv6-icmp*. After one-hot encoding, this feature is mapped into a binary vector containing either 0 or 1. Here, only the associated category is assigned to 1 and the others to 0. However, the attacks spread their perturbations overall of the fields that belong to the encoded categorical feature, which triggers multiple categories at once.

6 DISCUSSION

In this section, we first summarize the key findings. We then place our findings in the context of the literature that has employed the examined attacks to assess the robustness of ML-based NIDS to adversarial evasive examples. We discuss the literature from two points of view: First, the compatibility of the generated adversarial traffic with domain constraints of network traffic; Second, how likely the adversary can utilize these attacks for real-world scenarios. Finally, we state the study limitations and provide directions for future work.

6.1 Attack Unrealism

As can be seen in Table 4, the top performant techniques violated the validation metrics for realistic adversarial attacks. The attacks vary in the percentages of unrealistic AEs they produce. Some approaches generated less unrealistic AEs. However, they were the least performing attacks. On the other hand, the highest effective attacks created the highest percentages of unrealistic AEs.

We found that all of the attacks introduce non-binary values to binary features and trigger multiple categories at once to categorical features, as demonstrated in Table 5. Most of the attacks violate all of the metrics explained in sections 4.4 and 2.3 over the two datasets as shown in Tables 5. Although some of the AEs generation methods can be theoretically successful, no attack maintains all of the domain constraints. These techniques can not lead to practical and realistic attacks as they violate the network domain constraints. They break the semantic links among the features due to the high percentage of perturbed features, as shown in Table 4. These findings indicate that these attacks result in void data that cannot represent practical and realistic packets that can be delivered over the network. Therefore, they cannot be used to prove the resilience of DL-based NIDS to adversarial evasive flow in a real-world setup.

	Attack	Setup	WSN-DS			BoT-IoT		
			A	B	C	A	B	D
White-box	BIM	T	✓	✓	✓	✓	✓	✓
	FGSM	T	✓	✓	✓	✓	✓	✓
	PGD	T	✓	✓	✓	✓	✓	✓
	CW∞	T	✓		✓	✓	✓	✓
	CW2	T			✓		✓	✓
	BIM	U	✓	✓	✓	✓	✓	✓
	DeepFool	U	✓	✓	✓	✓	✓	✓
	FGSM	U	✓	✓	✓	✓	✓	✓
	NewtonFool	U	✓	✓	✓	✓	✓	✓
	PGD	U	✓	✓	✓	✓	✓	✓
	CW2	U	✓		✓	✓	✓	✓
	CW∞	U			✓		✓	✓
Black-box	Boundary	T	✓		✓		✓	✓
	ZOO	T		✓	✓		✓	✓
	HSJ	T	✓		✓		✓	✓
	EN	T	✓		✓			✓
	ZOO	U		✓	✓	✓	✓	✓
	Boundary	U	✓		✓	✓	✓	✓
	HSJ	U	✓		✓	✓	✓	✓
	EN	U			✓			✓

A=%of Perturbed Features over 50% B=Out-Range Values
C=Non-Binary Values D=Multi-Class Values

Table 5: Attacks Unrealism Metrics Over WSN-DS & BoT-IoT

6.2 Attack Infeasibility

Similar to the literature, the implemented attacks work with feature vectors extracted from pre-processed raw network traffic in the form of tabular CSV files. Such attacks are known as feature-space attacks, in which perturbations are applied directly to the inputs of the detection model. The data processor component in the NIDS parses raw packets to extract important features analyzed by the detection engine to classify the passing traffic using pre-constructed ML models. To implement such attacks, the adversary either has to know what features are parsed on the other side or control the channel of transforming the raw traffic to the pre-processed ML model inputs. Acquiring capabilities over the feature set or the pre-processing pipeline by the adversary is unlikely feasible for real-world scenarios. Differently, the problem-space attacks involve manipulating the actual packets and producing new adversarial ones. The difficulty of these attacks lies in perturbing the original raw input that corresponds to the adversarial feature vector. Although they are challenging to implement, they are feasibly realistic as the adversary can have the capability to craft the packet contents compared to knowing the feature set or controlling the pre-processing procedure.

White-box attacks denote scenarios where the adversary can access everything related to the target system and have complete knowledge of the model architecture, parameters, hyperparameters, weights, and configurations. Hence, the adversary can directly craft

adversarial examples by computing or approximating the model gradients [9]. To gain such knowledge, the adversary must access the model source code. However, the source code can be unobservable for a commercial NIDS or securely stored on a different machine for in-house NIDS [8]; hence, these attacks are unlikely feasible.

6.3 Literature Drawbacks

Our findings note that the outputs of these attacks are void and unrealistic as they do not obey traffic data restrictions. Previous studies on assessing DL-based NIDS robustness to adversarial attacks ignored the compliance of generated examples with network domain constraints [13, 15, 17–20, 27, 28, 32, 33, 38, 40–42]. Furthermore, they did not consider that the required perturbations for generating the AEs do not directly correlate to modifying the actual network packets; hence, their incapability for end-to-end attacks was not taken into account. The authors applied the generic adversarial examples generation methods to the statistical features collected from the packet metadata. Moreover, they did not demonstrate how the adversarial raw packets can be generated. These attacks are known as feature-space attacks that transform the original feature vector as an into a new perturbed feature vector. Although such attacks can be theoretically successful, they operate at the pre-processed traffic data level, not at the packet level; therefore, the adversary cannot use them for real-world scenarios. On the other hand, the problem-space attacks perturb the raw packets to produce new functional adversarial ones that can result in realistic end-to-end attacks.

A common drawback in the previous studies is that they assume the adversary knows everything about the targeted model and training data, allowing him to directly apply the perturbations to the model inputs [13, 15, 17–20, 28, 30, 32, 33, 35, 38, 40, 41]. In real-world scenarios, this assumption is unlikely common as the adversary in most cases an outsider.

To conclude our discussion of the literature work, the previous studies have three significant flaws. First, they did not consider the necessity of maintaining traffic domain constraints in generating the adversarial flow for preserving the validity and functionality of attack traces. Second, they assume the adversary can freely perturb any amount of features that can break the semantic links among the interdependent features. Lastly, they assume a white-box threat model, where the adversary has access to the parameters of the targeted model, which is not commonly feasible in real-world scenarios.

6.4 Study Limitations

The findings of this study have to be seen in the light of some limitations. Although the metrics we adopted to assess attack realism help exclude unrealistic attacks, compliance with these metrics does not ensure their realism. Additionally, the imbalanced distribution of attacks and normal instances in the selected datasets made it hard to compare attack performance over the two datasets, so we could not obtain a generalized measure of their impacts. Therefore, it is not evident how the performance results would generalize to other architectures of DL models or other datasets.

6.5 Future work Directions

Future research should work on proposing comprehensive validation metrics that define rules of realistic adversarial attacks against ML-based NIDS. Research on designing adversarial attacks for ML-based NIDS must consider end-to-end real-world attack scenarios. The attacks must be implemented at the packet level to prove their feasibility for real-world evasion of ML-based NIDS. Overall, for realistic attacks, the generated adversarial traffic needs to maintain the domain constraints and semantic links among the traffic features without knowledge of the detection model. Furthermore, future work should incorporate contemporary network traffic datasets that represent different networking environments and employ more adversarial attacks to acquire a general measure of their impact.

7 CONCLUSION

The existing literature utilizes the generic adversarial examples generation approaches to generate adversarial traffic and assess the robustness of DL-based NIDSs. This study validated the compliance of the generated adversarial examples with network domain constraints of network traffic and discussed the feasibility of utilizing these examples for real-world attacks. It assessed the outputs of seven white-box and four black-box attacks widely used in the literature, and they were implemented in different settings: targeted and untargeted. Furthermore, we incorporated a wireless sensor network traffic representing a different networking environment that has not been investigated and an IoT network traffic.

We demonstrated the effect of adversarial evasion attacks on the performance of a DL-based NIDS. Overall, the attacks vary in their performance, and some attacks achieved remarkable Evasion Rates. However, they result in void adversarial examples that do not comply with the network traffic domain constraints. The examined attacks introduce arbitrary and unrealistic perturbations such as non-binary values to the binary features that only accept 0 or 1, out-range values to the numeric features that have a fixed range of possible values, or trigger multiple categories at once to the categorical features. Furthermore, some of the attacks manipulate more than half of the traffic features; controlling such amount of features in a fine-grained manner is unlikely feasible to the adversary and eventually breaks the semantic links between the features. Based on these Unrealism metrics, we concluded that although these attacks can be performant, they are impractical and unrealistic for DL/ML-based NIDSs.

The literature focus on adversarial attacks that perturb the aggregated and pre-processed traffic features directly. These feature-space attacks are impractical as they modify features that are already parsed from the raw packets or aggregated statistically from the flow connections. They cannot be replayed directly in the network for end-to-end attacks. Thus, practical adversarial attacks should revolve around problem-space perturbations that directly amend the raw packets.

Furthermore, the current study assumes that the adversary can arbitrarily perturb traffic features using the generic approaches, have complete knowledge about the target model elements, or freely probe the model for an oracle attack. Such assumptions are unrealistic. Therefore, future research should be conducted in more realistic setups that involve: crafting valid examples that comply

with network traffic domain constraints, black-box threat models, and minimal knowledge and capabilities over the target model. These considerations will be addressed in our future studies.

REFERENCES

[1] Huda Ali Alatwi and Amjad Aldweesh. 2021. Adversarial Black-Box Attacks Against Network Intrusion Detection Systems: A Survey. In *2021 IEEE World AI IoT Congress (AIIoT)*. IEEE, 0034–0040.

[2] Huda Ali Alatwi and Charles Morisset. 2021. Adversarial Machine Learning In Network Intrusion Detection Domain: A Systematic Review. *arXiv preprint arXiv:2112.03315* (2021).

[3] Huda Ali Alatwi and Charles Morisset. 2022. Threat Modeling for Machine Learning-Based Network Intrusion Detection Systems. In *2022 IEEE International Conference on Big Data (Big Data)*. IEEE.

[4] Arwa Aldweesh, Abdelouahid Derhab, and Ahmed Z Emam. 2020. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowledge-Based Systems* 189 (2020), 105124.

[5] Elie Alhajjar, Paul Maxwell, and Nathaniel Bastian. 2021. Adversarial machine learning in network intrusion detection systems. *Expert Systems with Applications* 186 (2021), 115782.

[6] Huda Ali Alatwi, Tae Oh, Ernest Fokoue, and Bill Stackpole. 2016. Android malware detection using category-based machine learning classifiers. In *Proceedings of the 17th Annual Conference on Information Technology Education*. 54–59.

[7] Iman Almomani, Bassam Al-Kasasbeh, and Mousa Al-Akhras. 2016. WSN-DS: a dataset for intrusion detection systems in wireless sensor networks. *Journal of Sensors* 2016 (2016).

[8] Giovanni Apruzzese, Mauro Andreolini, Luca Ferretti, Mirco Marchetti, and Michele Colajanni. 2021. Modeling Realistic Adversarial Attacks against Network Intrusion Detection Systems. *arXiv preprint arXiv:2106.09380* (2021).

[9] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*. PMLR, 274–283.

[10] Battista Biggio and Fabio Roli. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition* 84 (2018), 317–331.

[11] Anna L Buczak and Erhan Guven. 2015. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications surveys & tutorials* 18, 2 (2015), 1153–1176.

[12] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2021. A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology* 6, 1 (2021), 25–45.

[13] Joseph Clements, Yuzhe Yang, Ankur Sharma, Hongxin Hu, and Yingjie Lao. 2019. Rallying adversarial techniques against deep learning for network security. *arXiv preprint arXiv:1903.11688* (2019).

[14] Michael Crawford, Taghi M Khoshgoftaar, Joseph D Prusa, Aaron N Richter, and Hamzah Al Najada. 2015. Survey of review spam detection using machine learning techniques. *Journal of Big Data* 2, 1 (2015), 1–24.

[15] Islam Debicha, Thibault Debatty, Jean-Michel Dricot, and Wim Mees. 2021. Adversarial Training for Deep Learning-based Intrusion Detection Systems. *arXiv preprint arXiv:2104.09852* (2021).

[16] Luca Demetrio, Scott E Coull, Battista Biggio, Giovanni Lagorio, Alessandro Armando, and Fabio Roli. 2021. Adversarial exemples: A survey and experimental evaluation of practical attacks on machine learning for windows malware detection. *ACM Transactions on Privacy and Security (TOPS)* 24, 4 (2021), 1–31.

[17] Xingbing Fu, Nan Zhou, Libin Jiao, Haifeng Li, and Jianwu Zhang. 2021. The robust deep learning–based schemes for intrusion detection in Internet of Things environments. *Annals of Telecommunications* (2021), 1–13.

[18] Chi Hsuan Huang, Tsung Han Lee, Lin huang Chang, Jhih Ren Lin, and Gwoboa Horng. 2019. Adversarial attacks on SDN-based deep learning IDS system. *Lecture Notes in Electrical Engineering* 513 (2019), 181–191. https://doi.org/10.1007/978-981-13-1059-1_17

[19] Olakunle Ibitoye, Omair Shafiq, and Ashraf Matrawy. 2019. Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks. *arXiv* (2019). arXiv:1905.05137

[20] JaeHan Jeong, Sungmoon Kwon, Man-Pyo Hong, Jin Kwak, and Taeshik Shon. 2019. Adversarial attack-based security vulnerability verification using deep learning library for multimedia video surveillance. *Multimedia Tools and Applications* (2019), 1–15.

[21] Houda Jmila and Mohamed Ibn Khedher. 2022. Adversarial machine learning for network intrusion detection: A comparative study. *Computer Networks* (2022), 109073.

[22] Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. 2019. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems* 100 (2019), 779–796.

[23] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236* (2016).

[24] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*. Chapman and Hall/CRC, 99–112.

[25] Chaopeng Li, Jinlin Wang, and Xiaozhou Ye. 2018. Using a recurrent neural network and restricted Boltzmann machines for malicious traffic detection. *NeuroQuantology* 16, 5 (2018).

[26] Keane Lucas, Mahmood Sharif, Lujo Bauer, Michael K Reiter, and Saurabh Shintre. 2021. Malware Makeover: breaking ML-based static analysis by modifying executable bytes. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*. 744–758.

[27] Ramy Maarouf, Danish Sattar, and Ashraf Matrawy. 2021. Evaluating Resilience of Encrypted Traffic Classification Against Adversarial Evasion Attacks. *arXiv preprint arXiv:2105.14564* (2021).

[28] Nuno Martins, José Magalhães Cruz, Tiago Cruz, and Pedro Henriques Abreu. 2019. Analyzing the footprint of classifiers in adversarial denial of service contexts. In *EPIA Conference on Artificial Intelligence*. Springer, 256–267.

[29] Nuno Martins, José Magalhães Cruz, Tiago Cruz, and Pedro Henriques Abreu. 2020. Adversarial machine learning applied to intrusion and malware scenarios: a systematic review. *IEEE Access* 8 (2020), 35403–35419.

[30] Mohamed Amine Merzouk, Frédéric Cuppens, Nora Boulahia-Cuppens, and Reda Yaich. 2020. A Deeper Analysis of Adversarial Examples in Intrusion Detection. In *International Conference on Risks and Security of Internet and Systems*. Springer, 67–84.

[31] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, et al. 2018. Adversarial Robustness Toolbox v1. 0.0. *arXiv preprint arXiv:1807.01069* (2018).

[32] Yulexis Pacheco and Weiqing Sun. 2021. Adversarial Machine Learning: A Comparative Study on Contemporary Intrusion Detection Datasets.. In *ICISSP*. 160–171.

[33] Ye Peng, Jinshu Su, Xiangquan Shi, and Baokang Zhao. 2019. Evaluating deep learning based network intrusion detection system in adversarial environment. *ICEIEC 2019 - Proceedings of 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication* (2019), 61–66. https://doi.org/10.1109/ICEIEC.2019.8784514

[34] Fabio Pierazzi, Feargus Pendlebury, Jacopo Cortellazzi, and Lorenzo Cavallaro. 2020. Intriguing properties of adversarial ml attacks in the problem space. In *2020 IEEE symposium on security and privacy (SP)*. IEEE, 1332–1349.

[35] Aritran Piplai, Sai Sree Laya Chukkapalli, and Anupam Joshi. 2020. NAttack! Adversarial Attacks to bypass a GAN based classifier trained to detect Network intrusion. In *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*. IEEE, 49–54.

[36] Insoo Sohn. 2021. Deep belief network based intrusion detection techniques: A survey. *Expert Systems with Applications* 167 (2021), 114170.

[37] Wei Song, Xuezixiang Li, Sadia Afroz, Deepali Garg, Dmitry Kuznetsov, and Heng Yin. 2020. Mab-malware: A reinforcement learning framework for attacking static malware classifiers. *arXiv preprint arXiv:2003.03100* (2020).

[38] S. Sriram, K. Simran, R. Vinayakumar, S. Akarsh, and K. P. Soman. 2020. Towards Evaluating the Robustness of Deep Intrusion Detection Models in Adversarial Environment. *Communications in Computer and Information Science* 1208 CCIS, January (2020), 111–120. https://doi.org/10.1007/978-981-15-4825-3_9

[39] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[40] Zheng Wang. 2018. Deep Learning-Based Intrusion Detection with Adversaries. *IEEE Access* 6 (2018), 38367–38384. https://doi.org/10.1109/ACCESS.2018.2854599

[41] Arkadiusz Warzyński and Grzegorz Kołaczek. 2018. Intrusion detection systems vulnerability on adversarial examples. In *2018 Innovations in Intelligent Systems and Applications (INISTA)*. IEEE, 1–4.

[42] Kaichen Yang, Jianqing Liu, Chi Zhang, and Yuguang Fang. 2019. Adversarial Examples Against the Deep Learning Based Network Intrusion Detection Systems. *Proceedings - IEEE Military Communications Conference MILCOM* 2019-October (2019), 559–564. https://doi.org/10.1109/MILCOM.2018.8599759