SN Computer Science (2022) 3: 412
https://doi.org/10.1007/s42979-022-01321-8

ORIGINAL RESEARCH

# A Comparative Study on the Impact of Adversarial Machine Learning Attacks on Contemporary Intrusion Detection Datasets

**Medha Pujari[1] · Yulexis Pacheco[1] · Bhanu Cherukuri[1] · Weiqing Sun[1]**

Received: 1 October 2021 / Accepted: 9 July 2022 / Published online: 3 August 2022


## Abstract

Adversarial attack techniques have taken a firm stand against the capabilities of deep neural networks, rendering them less efficient in performing their functions. Various kind of attacks have been studied and appropriate defense mechanisms have been proposed in the Computer Vision and Image Processing domains. The progress in Intrusion Detection System (IDS) domain is relatively less although it is gaining momentum lately. One of the concerns in the IDS domain is that most of the research work has been carried out using old datasets. There is a need to study the properties of newer benchmark datasets and analyze their characteristics under adversarial settings. Contemporary datasets include modern network behaviors and attack scenarios, which help IDSs perform well in modern networks. The more realistic a dataset is, the more efficient it can make an IDS model in a real environment. This paper addresses the said concern by conducting a study on recent datasets in the light of adversarial perturbations. We analyze how various adversarial attack algorithms, under white box settings, impact contemporary IDS datasets, namely, UNSW-NB15, Bot-IoT, and CSE-CIC-IDS2018. This paper summarizes the study and discusses how various classification algorithms perform when an IDS model is trained with each of the chosen datasets. The results included in the paper indicate that the adversarial examples are successful in decreasing the detection capabilities of the IDS models covered in the study. We provide a conclusion based on the evaluation results and share thoughts on the direction in which we are headed for future work.

**Keywords** Intrusion detection systems · Intrusion detection datasets · Adversarial machine learning · Deep learning · Deep neural networks

## Introduction

The Intrusion Detection Systems (IDSs), introduced in the year 1980 [1], became one of the most essential defenses in network security and cybersecurity. They were designed to proactively monitor the traffic and raise alerts when something malign or intrusive is detected [2]. The IDS technology evolved in many stages since it was introduced [3]. However, despite several developments made, the detection rates were not improving as expected, and there has not been a significant decrease in the number of false alarms. To overcome such performance issues and widen the capabilities of the IDSs, research began in the late 1990s to incorporate Machine Learning (ML) techniques in IDS development [4]. With the power of ML, IDSs gain the ability to detect unknown attacks. Attack behaviors change rapidly with time, and an IDS should be able to correctly recognize the malign activities in a network. When traditional IDSs encounter new or sophisticated signatures, they may take relatively longer time to analyze the packets and respond [5].

As early as in 2004, a study by N. Dalvi et al. [6] revealed a concerning vulnerability that machine learning algorithms possess against adversarial inputs. Later, it was shown that such a vulnerability profoundly exists in deep learning and neural networks when presented with adversarial perturbations [7–13]. Various adversarial attack scenarios were developed, and their impacts on classifiers were analyzed. Mechanisms have also been proposed to defend the models

This article is part of the topical collection "Information Systems Security and Privacy" guest edited by Steven Furnell and Paolo Mori.

✉ Weiqing Sun
weiqing.sun@utoledo.edu

Medha Pujari
medharani.pujari@rockets.utoledo.edu

[1] University of Toledo, Toledo, OH, USA

from adversarial perturbations and minimize their impacts [14]. However, much of this progress was made in the image-based areas, like computer vision, image processing, etcetera. A relatively lesser progress has been made in the IDS domain [15]. One of the major concerns about training IDSs is datasets. The performance of an IDS hugely depends on the quality of the data it learns from.

The availability of good quality IDS datasets is a challenge. A major portion of research work in this domain is being conducted and/or evaluated using old datasets [16]. Unlike in image domain, the data in IDS domain quickly becomes outdated, as data patterns rapidly change in networks and attack behaviors turn sophisticated. A dataset should reflect the contemporary network behaviors and cover sufficient attack scenarios so that an IDS model learns a wide variety of traffic characteristics. On the bright side, there are some datasets that are relatively newer and can serve better than older benchmark datasets like NSL-KDD, DARPA, etcetera [12]. It is important to study the characteristics of modern datasets and analyze how they are affected by adversarial algorithms, so that the analysis makes it easier for the research community to choose which dataset might fit better into a project's requirements.

The objective behind choosing recently published IDS datasets for this study is to understand how an IDS model, trained with such a dataset, behaves in adversarial environments. An IDS deployed in a modern network needs to have sufficient knowledge of modern traffic behaviors to properly analyze and correctly identify undesired data patterns in its network. To achieve this, the IDS needs to learn from a dataset that covers a fair amount of traffic scenarios that are common to occur in a typical real-time network.

The novelty of this work lies in the combination of elements such as the contemporary IDS datasets, the adversarial white-box attack algorithms, and more significantly, the domain in which we want to evaluate the impacts of adversarial machine learning. The motive behind choosing the CSE-CIC-IDS2018 dataset is its characteristics, as highlighted in "CSE-CIC-IDS2018 Dataset", which are close to a real-world environment. Network data that is far from reality might make a model behave as expected in an experimental/research setup, but cannot guarantee the model's performance in a real-time network. The lesser the gap is between a research IDS dataset and the traffic observed in a real-time network, the greater the chance is for an experimental model to be capable of doing well in a real-world environment.

This work contributes to evaluate the impacts of adversarial algorithms on contemporary datasets that represent modern traffic behaviors and attack scenarios. The datasets covered in this study are UNSW-NB15, published in 2015; Bot-IoT, published in 2018; and CSE-CIC-IDS2018, published in 2018. The adversarial attack algorithms studied are Jacobian-based Saliency Map Attack (JSMA), Fast Gradient Sign Method (FGSM), and Carlini Wagner (CW). Metrics such as Accuracy, Area Under the Curve (AUC), *F*1 Score, and Recall were used to evaluate the results and analyze the impact of the adversarial algorithms.

The remaining portion of this paper is organized as follows: "Background" presents an overview of adversarial machine learning, the adversarial methods used in this study, and briefly summarizes the datasets studied. "Related Work" presents related work on adversarial sample generation and adversarial machine learning."Experimental Evaluation" discusses the experimental evaluation process implemented for the study. "Experimentation Results" presents the evaluation results. "Analysis and Discussion" provides an analysis of the adversarial attacks on the datasets. "Conclusions and Future Work" concludes the paper and presents our thoughts for future work.

## Background

### Adversarial Machine Learning: A Bird-eye View

Adversarial Machine Learning (AML) is the process of deceiving an ML model by providing a perturbed input that makes the model render incorrect prediction. The perturbed input is imperceptible to humans but makes a considerable difference to a neural network. Neural networks are vulnerable to adversarial attacks during training as well as testing/validation phases. Variations in attack techniques can be introduced based on factors like phase (training, testing, etc.), the knowledge of the model that the attacker has, the target of the attack, influence of the attacker, etc. The attacks carried out in the training phase are termed as Poisoning attacks and those launched during the testing phase are called Evasion attacks. Barreno et al. [17] highlights three properties of an attack - influence, focus of violation (confidentiality, integrity, availability), and specificity of the target. For example, based on some of the factors stated above, an evasion attack can be classified as either a *white-box* attack, where the attacker has complete knowledge of the model (including details like training dataset, parameters, etcetera), or a *black-box* attack, where the attacker has almost no knowledge of the model, or a *gray-box* attack, where the attacker has partial knowledge of the same.

### Methods used for Generation of Adversarial Samples

The adversarial algorithms chosen for this study are all white-box evasion attacks. Although black-box and gray-box attacks are more common in practice (i.e., in real-time environments), most of these techniques aim at collecting information about their target models in a variety of ways,

implying that they gradually progress towards becoming white-box attacks, which tend to be more powerful than the other two categories. This thought process motivated us to choose white-box attacks for our study. The current section briefly explains the algorithms we chose for the experiment.

### Jacobian-based Saliency Map Attack

The Jacobian-based Saliency Map Attack (JSMA), introduced by [11], is one of the attack techniques evaluated in this study. It is an evasion attack that works by minimizing the L0 norm by iteratively generating a saliency map which is used to choose a feature that will have a maximum error in prediction when added with perturbation [18]. The attack aims to perturb least possible number of features to cause misclassification. The process consists of obtaining the Jacobian matrix where the component i is the input and j is a derivative of the class for input i [11]:

$$J_F(X) = \frac{\partial F(x)}{\partial x} = \left[\frac{\partial j(x)}{\partial x_i}\right]_{ixj} \tag{1}$$

In the above equation,$F$ represents the second to last layer [19]. For each feature selected, the perturbation is adjusted and the iterations are continued until misclassification in the target class is achieved or the limit for a maximum number of perturbed features is met [11]. If it fails to achieve this, the algorithm selects the next feature and repeats the process with it [12]. The authors were successful in modifying as less as 4.02% of the features per sample and achieved a success rate of 97% [19]. It is a white-box attack algorithm, therefore, requires a complete knowledge of the architecture and parameters of the model targeted [11].

Although the success rates achieved by JSMA and FGSM are almost similar, the number of features modified are relatively lesser and the computational costs higher with JSMA, than with FGSM [18].

### Fast Gradient Sign Method

The FGSM attack was a technique proposed by [9] for adversarial data generation. As per this technique, a perturbation can be defined as follows:

$$\eta = \epsilon * \text{sign}(\nabla x J(\theta, x, y)) \tag{2}$$

In the above equation, $\theta$ represents the parameters of a model, where $x$ is the input, $y$ is/are the corresponding target(s), and J($\theta$, $x$, $y$) is the cost to train the neural network [9]. $\epsilon$ represents the magnitude of the attack, and the gradient can be obtained by back propagation.

The attack algorithm has a loss function, and works by aiming to minimize it [15]. Unlike the JSMA attack, the FGSM attack does not aim at generating minimal adversarial perturbations. However, it tries to speed up the adversarial data generation process [8], and this is why it saves computation time when compared to JSMA.

### Carlini Wagner

This attack, proposed by [8], is considered to be one of the powerful attacks in defeating neural network models. It is often used as a benchmark algorithm to evaluate the vulnerability of a model, and also to assess the strength of an adversarial data generation technique. An L2 attack norm is used to generate adversarial samples, and can be defined as follows:

$$\text{minimize}||\frac{1}{2}(\tanh(w)+1) - x||_2^2 + e.f(\frac{1}{2}(\tanh(w))+1) \tag{3}$$

The main goal of the algorithm is to minimize the distortion in the L2 metric. The evaluations conducted by the authors show that the CW attack fails defensive distillation mechanism, which is another potential reason for its robustness. The L2 attack, implemented in this work, is available in Cleverharns library [20].

## Overview of the Datasets

Data is a fundamental and an essential ingredient to conduct research in any field of science. In the modern era, the research community has a greater advantage because of the publicly available datasets, a good number of which are used as benchmark datasets for research and development. In an IDS dataset, the records represent the network traffic, and each data point is either categorized as normal or as malicious, and this categorization is used for the evaluation [21]. Generating a realistic dataset is not only tedious, but also involves complications to make it publicly available because of the sensitive information present in it related to the network, its environment, and the users in it [22]. Despite the hurdles, fortunately, there have been a considerable number of datasets recently made available, that cover relatively modernized network traffic scenarios [23]. They have been generated in a way to overcome the shortcomings of the older benchmark datasets like NSL-KDD [24], and make data more useful for research activities. There is a need to study their characteristics and properties, to understand how useful they can be in various forms of research. This study uses three recently published datasets, UNSW-NB15, Bot-IoT, and CSE-CIC-IDS2018.

### UNSW-NB15 Dataset

Developed in the Cyber Range Lab, at UNSW (University of New South Wales) Canberra, the UNSW-NB15 is one

of the benchmark datasets that has a hybrid of realistically generated normal traffic behaviors and synthetically generated contemporary attack behaviors. The IXIA PerfectStorm tool was used for the generation of the data [16, 25]. The tcpdump tool was used to capture 100 GB of traffic in the raw form. The dataset covers nine types of attacks, and has a total of 49 features including the label attribute. A total of up to 12 algorithms are developed using tools like Argus and Bro-IDS, to generate the features of the dataset [26–29]. This dataset is well-balanced when compared with the other two datasets used in this study. This is because there is relatively much lesser difference between the number of benign and malign traffic instances in this dataset.

#### Bot-IoT Dataset

The Bot-IoT dataset was also developed in the Cyber Range Lab of UNSW Canberra, in the year 2018. A realistic network environment was created to generate this dataset. As it is clear from the name of the dataset, it consists of IoT-based traffic, both benign and botnet. The total raw data captured is 69.3 GB in size, and has over 72 million records. For easier handling of the dataset, the authors also published a smaller version of this dataset, extracting 5% of its data through specific MySQL queries [23, 30–34]. This smaller version, split into training and testing sets, with about 3 million records and around 1 GB in size, has been used in this study.

#### CSE-CIC-IDS2018 Dataset

The CSE-CIC-IDS2018 dataset hereafter referred to as the CIC-IDS2018 dataset, was developed as a collaborative project between the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC). The dataset covers seven different attack scenarios, and was generated in an environment that is close to reality because of the massive resources used. The attack-generating network had up to 50 devices and the victim network was divided into 5 departments, with a total of 450 devices including servers and other machines. The CICFlowMeter-V3 was used to generate a bidirectional network traffic, and for feature extraction as well [35–37]. The traffic data was collected for 10 days, and was saved in 10 different files. There are 79 features in 9 of those files, and 83 features in the remaining file. This dataset is huge to be handled in full, therefore, we have used about 20% of the dataset, making sure we have all the classes included, and a balanced amount of instances in all of them. A brief summary of the datasets is presented in Table 1.

**Table 1** Overview of the datasets

| Dataset | Total attributes | Total instances |
|---|---|---|
| UNSW-NB15 | 49 | 2,540,044 |
| Bot-IoT | 46 | 73,370,443 |
| CSE-CIC-IDS2018 | 79 (in 9 files), 83 (in 1 file) | 16,233,002 |

## Related Work

This section discusses various works that revolve around adversarial machine learning, including works that propose adversarial attack techniques, layout taxonomies for approaches to generate adversarial data put forth mechanisms for defending the adversarial techniques, etcetera.

One of the early studies on adversarial attack techniques and defenses was published in 2006, by [17]. The authors discussed how the learning algorithm can be corrupted when detailed information about the model and its properties is provided.

The authors of [38] propose a strategy to make linear classifiers more robust against adversarial settings, and in particular, investigate two methods, namely, random subspacing and bagging, for the construction of ensemble-classifier models.

In [39], the authors propose an adaptive adversarial technique for embedding a backdoor in a model's training data and/or its parameters, and can bypass the currently existing mechanisms that detect the presence of backdoors.

The authors of [40] studied the vulnerability of the NSL-KDD dataset against the FGSM technique. They conducted experiments to investigate the presence of attack vector in the data samples that can be used to let the adversarial inputs bypass the detection mechanism.

In [15], the authors used the NSL-KDD dataset to study the impacts of adversarial learning algorithms on deep neural networks, with a Multi-Layer Perceptron (MLP) model. They also examined the uses of feature selection in adversarial sample generation. The attack techniques used in their work are FGSM, Deepfool, JSMA, and CW. Their evaluation results indicate that it is not so beneficial for an adversary to modify a large number of features in the adversarial sample generation.

The authors of [41] propose a GAN-based black-box adversarial technique and analyze how practical its impacts are on a network-based IDS (NIDS). Their results suggest that a black-box adversarial attack can also have a considerable impact on the performance of a deep neural network (DNN). The NSL-KDD dataset was used for their study.

In [12], the author studied the performance of IDS model when trained with each of NSL-KDD and KDD-99 datasets under two attacks, JSMA and FGSM. The classifiers used for the analysis include Random Forest (RF), MLP, Support

Vector Machine (SVM), and Decision Tree (DT). Although the attacks used in this study were proposed for image domain-based classifiers, the results in the study showed that these attack methods affect IDS models, too.

In [18], the authors evaluated the perfomance of IDS models by training them with NSL-KDD and CIC-IDS2017 datasets separately. The adversarial techiques they used were DeepFool, JSMA, FGSM, and CW. The study was performed only based on Denial-of-Service (DoS) attack instances. The evaluation results show that the overall performance of the model when trained with CIC-IDS2017 dataset decreased by up to 40%, and by 13% when trained with NSL-KDD.

The authors of [42] conducted a survey on the commonly used IDS datasets for the AML research in the IDS domain, and the attacks implemented. Their study suggests that up to 60% of the works use NSL-KDD dataset, upto 30% use CTU-13, and upto 10% use CIC-IDS2017 dataset. Additionally, it suggests that more commonly used attack algorithms are JSMA, DeepFool, FGSM, and WGAN. Most affected classfiers include SVM, DT, Naive Bayes (NB), while RF and SVM with Radial Basis Function (RBF) kernel are relatively more robust than others.

Aayush Arora and Shantanu [43] present a review of GAN applications in the cybersecurity domain on currently stable datasets. In this paper, they review the extensions of GAN frameworks relevant to the cybersecurity domain such as Deep Convolutional Generative Adversarial Networks (DCGANs), Bidirectional Generative Adversarial Networks (BiGANs), Cycle-Consistent Adversarial Networks (CycleGANs) and commonly used stable datasets. They also discuss applications of GAN like Steganography, Password Guessing, and Intrusion Detection Systems. Additionally, they provide a case study to evaluate the performance of the BiGANs for Anomaly Detection.

A survey by Kusha Sadeghi et al. [44] on attacks and defenses in adversarial ML provides system-driven taxonomies for the following aspects - datasets; the architectures of ML models; adversary's utilities (knowledge, capability, and goal); strategies followed by the adversaries; results of the defense mechanisms. The authors' idea behind a system-oriented classification is that a system model is necessary to conduct and repeat experiments launching adversarial attacks and to implement their corresponding defenses. In the author's view, a race between the attacks and defenses carried out using such a model can help enhancing the robustness of the model, and of the ML applications.

## Experimental Evaluation

The study summarized in this paper is oriented around multi-class classification, as all the datasets used in this study have multiple classes. To suit the nature of the datasets, four efficient classification algorithms have been chosen, namely, MLP, DT, RF, and SVM. Table 2 presents the hyperparameters chosen for the evaluations. To handle multi-class classification, the OneVsRestClassifier function is used, to fit one classifier per class.

### Software Specifications

The entire programming set-up is based on Python 3.6.5, Scikit-learn V.0.19.1 library [45], Tensorflow V.1.13.2 [46], and Keras V.2.1.5 [47]. For the implementation of the attack algorithms, Cleverhans V.3.0.1 library [20] has been used.

### Data Pre-Processing

The data oftentimes needs processing before a learning algorithm is subjected to training with the dataset. There are two steps of pre-processing implemented in this work - the One-Hot Encoding, and the Min-Max Normalization.

#### One-Hot Encoding

This technique was opted for to convert the entire data to a numerical format. There are some features in each dataset that do have non-numerical values, for example, they may have categorical data. The One-Hot encoding method helps address this scenario.

#### Min-Max Normalization

This technique was applied to all the datasets to scale the values in each of them between 0 and 1. Since different

**Table 2** Hyperparameters for the classifiers [48]

| Classifier | Parameters |
|---|---|
| MLP | Dropout = 0.4, Layer 1 = 256, Layer 2 = 128, Activation = Relu, Loss = categorical crossentropy, Optimizer = Adam, Output Layer Activation = Softmax |
| DT | criterion = gini, max_depth = 12 |
| RF | n_estimators = 200, random_state = 4, min_samples_split = 10 |
| SVM | C = 1, random_state = 42, loss = hinge |

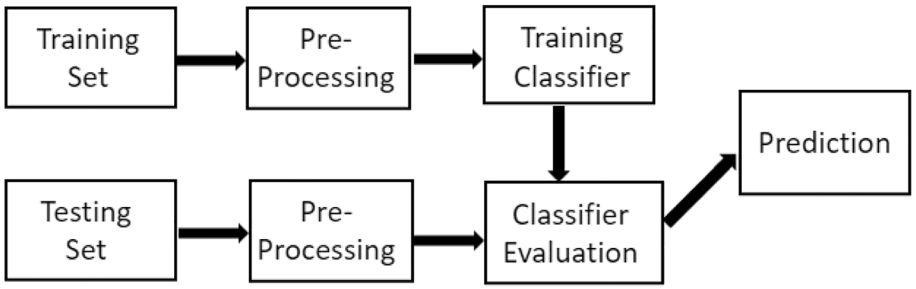


**Fig. 1** Sequence of steps involved using the original data

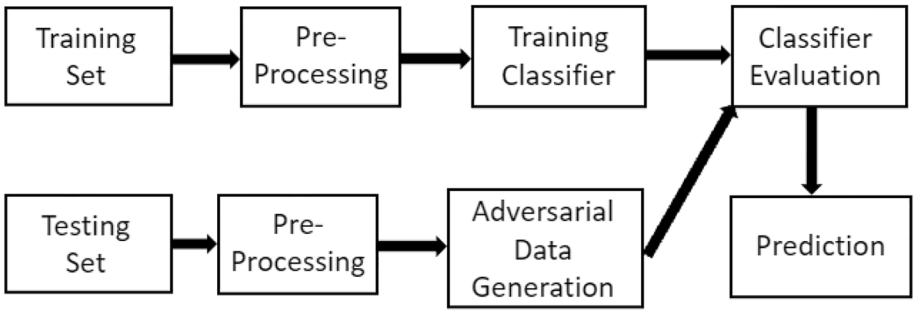


**Fig. 2** Sequence of steps involved using the adversarial data

features in a dataset might have values distributed on different scales, this technique helps convert all the values to a common scale and eliminate outliers, if any. Additionally, the attack methods require that all the features are within a common range, to be effective [18].

## Steps Involved in the Experiment

There are two stages implemented in the experiment: 1) training a learning algorithm with original data; 2) generating adversarial samples from the original data. In first stage, training and testing phases are carried out, as shown in Fig. 1. In both phases, the original data is pre-processed. MLP has been used as the baseline learning algorithm. Therefore, baseline results are obtained when MLP is tested with the data (original or adversarial), and for the evaluation purpose, each of the other algorithms (DT, RF, and SVM) are implemented over the baseline algorithm.

Figure 2 outlines the steps involved in the second stage, the adversarial sample generation. There are training and testing phases in this stage, too. The main difference here is that, in the testing phase, after the test-data is pre-processed, it is fed to the MLP, and each of the attack algorithms are invoked to introduce adversarial perturbations into the test-data. The obtained adversarial test-set is forwarded to the classifier for final predictions. The attacks have been performed targeting the normal class in the chosen datasets, with white-box settings. Table 3 presents the parameters set for each of the attacks.

**Table 3** Parameters set for the attacks on all datasets [48]

| Attacks | Parameters |
|---|---|
| JSMA | Theta = 1, Gama = 0.1, clip_min = 0, clip_max = 1 |
| FGSM | Eps = 0.3 |
| CW | binary_search_steps = 2, max_iterations = 100, learning_rate = 0.2, batch_size = 1, initial_cost = 10 |

The evaluation was initially conducted 10 times on a machine with UNSW-NB15 and Bot-Iot datasets, and the average values were noted as the experimental results. Later, the evaluation with the UNSW-NB15 dataset was carried out for an additional 3 times and with the CIC-IDS2018 dataset for 3 times, on a different machine (a server), whose configuration is as follows: 128 GB RAM, dual-core processor, and 3.17 TB secondary storage. The results included in this paper are the averages of the corresponding runs.

## Evaluation Metrics

The last step in the evaluation with each attack algorithm is to test every classifier with the original test-set and then with the poisoned set. The same process is applied in case of every dataset. The metrics used for evaluation are Accuracy, Area Under the Curve (AUC), *F*1-score, and Recall.

**Table 4** Accuracy results for UNSW-NB15 dataset

| Classifier | Accuracy | | | |
|---|---|---|---|---|
| | Baseline | JSMA | FGSM | CW |
| MLP | 0.72 | 0.39 | 0.38 | 0.21 |
| SVM | 0.59 | 0.22 | 0.26 | 0.23 |
| DT | 0.64 | 0.57 | 0.19 | 0.15 |
| RF | 0.64 | 0.64 | 0.25 | 0.28 |

**Table 5** AUC results for UNSW-NB15 dataset

| Classifier | AUC | | | |
|---|---|---|---|---|
| | Baseline | JSMA | FGSM | CW |
| MLP | 0.90 | 0.58 | 0.62 | 0.55 |
| SVM | 0.89 | 0.31 | 0.63 | 0.61 |
| DT | 0.84 | 0.80 | 0.53 | 0.51 |
| RF | 0.92 | 0.92 | 0.83 | 0.78 |

**Table 6** *F*1 score results for UNSW-NB15 dataset

| Classifier | *F*1 Score | | | |
|---|---|---|---|---|
| | Baseline | JSMA | FGSM | CW |
| MLP | 0.73 | 0.45 | 0.35 | 0.33 |
| SVM | 0.68 | 0.30 | 0.29 | 0.31 |
| DT | 0.69 | 0.62 | 0.32 | 0.24 |
| RF | 0.73 | 0.68 | 0.38 | 0.43 |

**Table 7** Recall results for UNSW-NB15 dataset

| Classifier | Recall | | | |
|---|---|---|---|---|
| | Baseline | JSMA | FGSM | CW |
| MLP | 0.72 | 0.42 | 0.40 | 0.25 |
| SVM | 0.60 | 0.23 | 0.31 | 0.26 |
| DT | 0.66 | 0.62 | 0.38 | 0.22 |
| RF | 0.65 | 0.56 | 0.24 | 0.28 |

## Experimentation Results

This section presents the results obtained, ordered by the datasets, and discusses the impact of each attack algorithm on each of the datasets.

### UNSW-NB15 Dataset

Tables 4, 5, 6, 7 summarize the results in terms of the various metrics used. The highest accuracy with normal data is obtained from the baseline algorithm, MLP, with the least from SVM. Considering the overall adversarial accuracy scores, the results indicate that CW attack has the highest impact, and JSMA has the least.

#### Jacobian-based Saliency Map Attack

A total of 95 distinct features are altered by JSMA attack in this dataset, with average of 22 per data point. The total percentage of altered features is 11%. The average time taken to generate adversarial samples is 8 min. With UNSW-NB15, the overall results show that this attack has the highest impact on the SVM classifier and the lowest impact on the RF classifier. This makes SVM the most vulnerable to JSMA among the chosen clsasifiers, and RF the least vulnerable.

#### Fast Gradient Sign Method

A total of 192 features are altered by this attack, with an average of 162 features per data point. The total percentage of altered features is 78%. The time taken for adversarial sample generation is less than 5 seconds. The results suggest that this attack has more impact on DT classifier than on the others, and the least impact on RF. Therefore, RF and SVM are almost equally robust against the FGSM attack, and are better than the DT.

#### Carlini Wagner

A total of 196 features are altered by this attack, with an average of 133 features per data point. The total percentage of altered features is 65%. The time taken for adversarial sample generation is almost 50 min, the longest among all the selected attack algorithms. The results suggest that this attack has the highest impact on DT classifier and the least impact on RF. Therefore, RF is more robust against the CW attack than the other two algorithms and DT is the most vulnerable to CW.

**Table 8** Accuracy results for Bot-IoT dataset [48]

| Classifier | Accuracy | | | |
|---|---|---|---|---|
| | Baseline | JSMA | FGSM | CW |
| MLP | 0.91 | 0.39 | 0.36 | 0.34 |
| SVM | 0.94 | 0.48 | 0.40 | 0.48 |
| DT | 0.99 | 0.45 | 0.48 | 0.65 |
| RF | 0.99 | 0.86 | 0.47 | 0.60 |

### Bot-IoT Dataset

Tables 8, 9, 10, 11 summarize the results for Bot-IoT in terms of the various metrics used. The highest accuracy with normal data is obtained from both DT and RF classifiers,

**Table 9** AUC results for Bot-IoT dataset [48]

| Classifier | AUC | | | |
|---|---|---|---|---|
| | Baseline | JSMA | FGSM | CW |
| MLP | 0.98 | 0.48 | 0.97 | 0.96 |
| SVM | 0.99 | 0.50 | 0.98 | 0.95 |
| DT | 0.99 | 1.0 | 0.97 | 0.97 |
| RF | 0.99 | 0.50 | 0.98 | 0.95 |

**Table 10** *F*1 score results for Bot-IoT dataset [48]

| Classifier | *F*1 Score | | | |
|---|---|---|---|---|
| | Baseline | JSMA | FGSM | CW |
| MLP | 0.99 | 0.76 | 0.40 | 0.55 |
| SVM | 1.0 | 0.77 | 0.33 | 0.58 |
| DT | 0.99 | 0.61 | 0.46 | 0.67 |
| RF | 0.99 | 0.96 | 0.42 | 0.57 |

**Table 11** Recall results for Bot-IoT dataset [48]

| Classifier | Recall | | | |
|---|---|---|---|---|
| | Baseline | JSMA | FGSM | CW |
| MLP | 0.99 | 0.93 | 0.39 | 0.57 |
| SVM | 1.0 | 0.93 | 0.41 | 0.59 |
| DT | 1.0 | 0.45 | 0.40 | 0.60 |
| RF | 0.99 | 0.95 | 0.41 | 0.57 |

**Table 12** Accuracy results for CSE-CIC-IDS2018 dataset

| Classifier | Accuracy | | | |
|---|---|---|---|---|
| | Baseline | JSMA | FGSM | CW |
| MLP | 0.62 | 0.39 | 0.38 | 0.35 |
| SVM | 0.61 | 0.58 | 0.61 | 0.20 |
| DT | 0.88 | 0.47 | 0.85 | 0.38 |
| RF | 0.92 | 0.84 | 0.91 | 0.81 |

**Table 13** AUC results for CSE-CIC-IDS2018 dataset

| Classifier | AUC | | | |
|---|---|---|---|---|
| | Baseline | JSMA | FGSM | CW |
| MLP | 1.0 | 0.42 | 0.97 | 0.99 |
| SVM | 1.0 | 0.44 | 0.91 | 0.99 |
| DT | 1.0 | 0.44 | 1.0 | 0.99 |
| RF | 1.0 | 0.44 | 1.0 | 0.99 |

**Table 14** *F*1 score results for CSE-CIC-IDS2018 dataset

| Classifier | *F*1 Score | | | |
|---|---|---|---|---|
| | Baseline | JSMA | FGSM | CW |
| MLP | 0.89 | 0.43 | 0.85 | 0.51 |
| SVM | 0.66 | 0.39 | 0.64 | 0.47 |
| DT | 0.91 | 0.11 | 0.89 | 0.69 |
| RF | 0.94 | 0.59 | 0.92 | 0.83 |

with the least from MLP. Considering the overall adversarial accuracy scores, the results indicate that FGSM attack is degrading the accuracy by a greater magnitude than the other two, and JSMA has the least impact.

### Jacobian-Based Saliency Map Attack

A total of 57 features are altered, with an average of 28 per data point, making the total perentage of altered features 43%. The time taken for adversarial data generation is close to 14 min. The DT classifier is the most vulnerable to this attack, and RF is the least.

### Fast Gradient Sign Method

A total of 60 distinct features are altered using this attack, with average of 34 per data point. The percentage of altered features is 52%. The attack takes around 20 seconds to generate adversarial data with Bot-IoT dataset. DT and RF classifiers are almost equally robust against this attack, and are better than the SVM.

### Carlini Wagner

A total of 59 distinct features are altered, with an average of 42 per data point, and 52% as the total percentage of altered features. The attack takes close to 2 h to generate adversarial samples. The impact is almost the same on all the classifiers, with DT showing relatively lesser vulnerability than the other two, and SVM being more vulnerable than the other two.

## CIC-IDS2018 Dataset

Tables 12, 13, 14, 15 summarize the results for the CIC-IDS2018 dataset in terms of the various metrics used. The highest accuracy with normal data is obtained from the RF classifier, with the least from MLP. Considering the overall adversarial accuracy scores, the results indicate that CW attack is degrading the accuracy by a greater magnitude than the other two, while FGSM has the least impact.

**Table 15** Recall results for CSE-CIC-IDS2018 dataset

| Classifier | Recall | | | |
|---|---|---|---|---|
| | Baseline | JSMA | FGSM | CW |
| MLP | 0.90 | 0.58 | 0.85 | 0.81 |
| SVM | 0.97 | 0.64 | 0.92 | 0.95 |
| DT | 0.94 | 0.12 | 0.93 | 0.82 |
| RF | 0.91 | 0.57 | 0.91 | 0.81 |

#### Jacobian-based Saliency Map Attack

A total of 93 features are altered, with an average of 72 per data point. The percentage of altered features is 42%. The time taken for this attack to generate adversarial samples is close to 10 h. The SVM classifier has been affected the least of all, and the DT has been affected the most.

#### Fast Gradient Sign Method

A total of 187 features are altered, with an average of 136 per data point. The percentage of altered features is 85%. The time taken for this attack to generate adversarial samples is around 6 h. The RF classifier has been affected the least of all, and the SVM has been affected the most.

#### Carlini Wagner

A total of 189 features are altered, with an average of 157 per data point. The percentage of altered features is about 86%. The time taken for this attack to generate adversarial samples is around 14 h. The RF classifier has been affected the least of all, and the DT has been affected the most.

## Analysis and Discussion

Considering datasets, classifiers, and attacks as three entities, the results obtained from the evaluation indicate that the influence of an entity varies with the other two. This section analyzes the results further and notes appropriate implications.

### Implications of this Study

Although all three attack algorithms affected the performances of the classifiers, the variations in their impacts can help investigate deep into the characteristics of the datasets used. Based on the results, the overall impact on the CIC-IDS2018 dataset is relatively lesser, which is followed by the UNSW-NB15 dataset, and then the Bot-IoT. One possible reason behind this pattern is the number of features in the datasets. With lesser number of features, the vulnerabilties may increase. If the entire volume of each of the datasets was considered for the study, the scale of imbalance (being well-balanced or imbalanced) in the datasets would also become a factor for the variations in performance.

Looking at the overall results from classifiers end, the RF classifier stood almost steadily robust against all three attacks, with all three datasets. Another significant behavior is that the impact patterns are not uniform among different evaluation metrics. It means, an adversary needs to decide on a performance metric as target and design the attack accordingly.

Although the CW attack is considered one of the most sophisticated and powerful algorithms, its result patterns on the IDS datasets chosen for this study are similar to the other two attack techniques, and are not exceptional, per se.

### Contribution to the Literature

Data is a precious entity, driving ML-based research in nearly every area of science. The quality and characteristics of a dataset are crucial in tuning the efficiency of a model. This work contributes to the literature by analyzing the behaviors of ML-based IDSs in adversarial environments using datasets that consist of realistic network patterns.

A consequential avenue for investigation is the extent of validity of these adversarial white-box attacks in the context of IDS datasets. Although the adversarial samples generated by the attack algorithms succeed in dropping the performance of an IDS model, there is a need to examine their efficiency in generating valid adversarial data. The goal of an adversarial algorithm targeting an IDS model is to modify an attack data instance in a way that it should look like a benign instance to the target while retaining the properties that make it the attack it is supposed to be. In other words, an adversarial data instance, X', generated from an original (non-adversarial) attack instance, X, is valid only if X' can achieve exactly what X can, in the network guarded by the target IDS. The real success of an adversarial attack lies in generating valid deceptive samples that can bypass detection and launch the attacks they are meant for. Pujari et al. [49] lists some factors that indicate the validity of adversarial samples. We want to continue this research by analyzing how successful various white-box attacks can be on IDS research datasets.

### Limitations

A substantial limitation is the resources to process the huge volumes of datasets utilized in the experiments. Datasets like Bot-IoT and CIC-IDS2018 are big data and need efficient frameworks to handle them. We used smaller portions of these datasets to accommodate the resource constraints. One of the extensions to this work would be to evaluate

the experiments with full datasets using a framework like Hadoop.

### Insights into Mitigation Strategies

Improving the resistance of IDS models towards adversarial inputs has been a substantial stream of research ever since the vulnerabilities were discovered. The insights on how to enhance the resistance of a model, drawn based on our experiment are presented here. The datasets chosen for this work have many features, but not all the features in a dataset have a significant contribution to the outputs. One approach to reducing the impact of adversarial inputs is to extensively train a model on the features that decide the output. Techniques such as feature selection, feature reduction, etcetera, can help filter the features bearing less to no weightage in predicting the output. Such a training process enables a model to focus more on the deciding attributes and ignore the adversarial perturbations in the remaining features. Furthermore, some features in a dataset may allow values only within a specific range, in which case, an extra step can be added to validate the values in those features before prediction. Another strategy for filtration can be to validate the values that non-changeable features of an input hold. The approaches mentioned here are superficial, as it requires a much more thorough defensive mechanism to effectively make an IDS model robust.

## Conclusions and Future Work

There is a need to study the properties of the available modern IDS datasets and switch from the old and outdated datasets to the contemporary ones. As important as it is to analyze how useful the modern datasets are in machine learning-based research, it is essential to know how useful they are under adversarial settings. This work studies three recently published IDS datasets, namely, UNSW-NB15, Bot-IoT, and CIC-IDS2018 under the light of three adversarial attack algorithms, namely, JSMA, FGSM, and CW. The performance is evaluated using multiple classifiers - SVM, DT, and RF - while using MLP as the baseline classifier. The experimental results have shown that RF is relatively more robust in adversarial environments, and in terms of the datasets, CIC-IDS2018 has offered more resilience to the classifiers. The impacts of the attacks have been varying with the datasets and classifiers.

We would like to extend this study in multiple directions. One of them is to analyze the impacts of the white-box attacks on recent datasets using other powerful algorithms, especially, deep learning algorithms. Another direction is to study black-box and gray-box attack techniques and develop defense mechanisms to tackle them.

**Funding** This study was not funded by any grant.

### Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

1. Anderson JP. Computer security threat monitoring and surveillance. Fort Washington: Anderson Co.; 1980.
2. Kemmerer RA, Vigna G. Intrusion detection: a brief history and overview. Computer. 2002;35(4):supl27–30.
3. Innella P, et al. The evolution of intrusion detection systems. Tetrad Digit Integr. 2001;1–15
4. Li AZ, Barton D. A brief history of machine learning in cybersecurity. 2022. https://www.securityinfowatch.com/cybersecurity/article/21114214/a-brief-history-of-machine-learning-in-cybersecurity. Accessed 14 Nov 2019.
5. Othman SM, Ba-Alwi FM, Alsohybe NT, Al-Hashida AY. Intrusion detection model using machine learning algorithm on big data environment. J Big Data. 2018. https://doi.org/10.1186/s40537-018-0145-4.
6. Dalvi N, Domingos P, Sanghai S, Verma D. Adversarial classification. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004, pp. 99–108.
7. Biggio B, Corona I, Maiorca D, Nelson B, Šrndić N, Laskov P, Giacinto G, Roli F. Evasion attacks against machine learning at test time. In: Advanced information systems engineering. Berlin: Springer; 2013. p. 387–402. https://doi.org/10.1007/978-3-642-40994-3_25.
8. Carlini N, Wagner D. Towards evaluating the robustness of neural networks. IEEE Symp Secur Priv (SP). 2017. https://doi.org/10.1109/sp.2017.49.
9. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. In: 3rd International Conference on Learning Representations (ICLR), ICLR2015. http://arxiv.org/abs/1412.6572
10. Papernot N, McDaniel P, Goodfellow I. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. 2016. arXiv preprint arXiv:1605.07277
11. Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A. The limitations of deep learning in adversarial settings. IEEE Eur Symp Secur Priv. 2016. https://doi.org/10.1109/eurosp.2016.36.
12. Rigaki M. Adversarial deep learning against intrusion detection classifiers. MS Thesis, Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Sweden, 2017. [Online]. https://www.diva-portal.org/smash/get/diva2:1116037/FULLTEXT01.pdf
13. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R. Intriguing properties of neural networks. 2013. arXiv preprint arXiv:1312.6199
14. Huang L, Joseph AD, Nelson B, Rubinstein BI, Tygar JD. Adversarial machine learning. In: Proceedings of the 4th ACM workshop on Security and artificial intelligence, 2011, pp. 43–58.
15. Wang Z. Deep learning-based intrusion detection with adversaries. IEEE Access. 2018;6:38367–84. https://doi.org/10.1109/access.2018.2854599.

16. Dwibedi S, Pujari M, Sun W. A comparative study on contemporary intrusion detection datasets for machine learning research. IEEE Int Conf Intell Secur Inf (ISI). 2020. https://doi.org/10.1109/isi49825.2020.9280519.
17. Barreno M, Nelson B, Sears R, Joseph AD, Tygar JD. Can machine learning be secure? In: Proceedings of the 2006 ACM Symposium on Information, computer and communications security, 2006, pp. 16–25.
18. Martins N, Cruz JM, Cruz T, Abreu PH. Analyzing the footprint of classifiers in adversarial denial of service contexts. In: Progress in artificial intelligence. Berlin: Springer International Publishing; 2019. p. 256–67. https://doi.org/10.1007/978-3-030-30244-3_2210.1007/978-3-030-30244-3_22.
19. Yuan X, He P, Zhu Q, Li X. Adversarial examples: attacks and defenses for deep learning. IEEE Trans Neural Netw Learn Syst. 2019;30(9):2805–24. https://doi.org/10.1109/tnnls.2018.2886017.
20. Papernot N, Goodfellow I, Sheatsley R, Feinman R, McDaniel P, et al. cleverhans v2. 0.0: an adversarial machine learning library. 2016. arXiv preprint arXiv:1610.00768
21. Ring M, Wunderlich S, Scheuring D, Landes D, Hotho A. A survey of network-based intrusion detection data sets. Comput Secur. 2019;86:147–67. https://doi.org/10.1016/j.cose.2019.06.005.
22. Javaid A, Niyaz Q, Sun W, Alam M. A deep learning approach for network intrusion detection system. Proc EAI Int Conf Bio-inspired Inf Commun Technol. 2016. https://doi.org/10.4108/eai.3-12-2015.2262516.
23. Koroniotis N, Moustafa N, Sitnikova E, Turnbull B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. Futur Gener Comput Syst. 2019;100:779–96. https://doi.org/10.1016/j.future.2019.05.041.
24. Moustafa N, Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). Mil Commun Inf Syst Conf (MilCIS). 2015. https://doi.org/10.1109/milcis.2015.7348942.
25. Zoghi Z, Serpen G. Unsw-nb15 computer security dataset: analysis through visualization. 2021. arXiv preprint arXiv:2101.05067
26. Moustafa N, Creech G, Slay J. Big data analytics for intrusion detection system: Statistical decision-making using finite dirichlet mixture models. In: Data analytics and decision support for cybersecurity. Berlin: Springer International Publishing; 2017. p. 127–56. https://doi.org/10.1007/978-3-319-59439-2_5.
27. Moustafa N, Slay J. The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Inf Secur J. 2016;25(1–3):18–31. https://doi.org/10.1080/19393555.2015.1125974.
28. Moustafa N, Slay J, Creech G. Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. IEEE Trans Big Data. 2019;5(4):481–94. https://doi.org/10.1109/tbdata.2017.2715166.
29. Sarhan M, Layeghy S, Moustafa N, Portmann M. Netflow datasets for machine learning-based network intrusion detection systems. 2020. arXiv preprint arXiv:2011.09144
30. Koroniotis N. Designing an effective network forensic framework for the investigation of botnets in the internet of things. Ph.D. thesis, University of New South Wales, Sydney, Australia, 2020.
31. Koroniotis N, Moustafa N. Enhancing network forensics with particle swarm and deep learning: The particle deep framework. Int Conf Artif Intell Appl. 2020. https://doi.org/10.5121/csit.2020.100304.
32. Koroniotis N, Moustafa N, Schiliro F, Gauravaram P, Janicke H. A holistic review of cybersecurity and reliability perspectives in smart airports. IEEE Access. 2020;8:209802–34. https://doi.org/10.1109/access.2020.3036728.
33. Koroniotis N, Moustafa N, Sitnikova E. A new network forensic framework based on deep learning for internet of things networks: a particle deep framework. Futur Gener Comput Syst. 2020;110:91–106. https://doi.org/10.1016/j.future.2020.03.042.
34. Koroniotis N, Moustafa N, Sitnikova E, Slay J. Towards developing network forensic mechanism for botnet activities in the IoT based on machine learning techniques. In: Lecture notes of the institute for computer sciences, social informatics and telecommunications engineering. Berlin: Springer International Publishing; 2018. https://doi.org/10.1007/978-3-319-90775-8_3.
35. AWS: a realistic cyber defense dataset (cse-cic-ids2018). 2018. https://registry.opendata.aws/cse-cic-ids2018/
36. Kanimozhi V, Jacob TP. Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. Int Conf Commun Signal Process (ICCSP). 2019. https://doi.org/10.1109/iccsp.2019.8698029.
37. Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. Proc Int Conf Inf Syst Secur Priv. 2018. https://doi.org/10.5220/0006639801080116.
38. Biggio B, Fumera G, Roli F. Multiple classifier systems for robust classifier design in adversarial environments. Int J Mach Learn Cybern. 2010;1(1–4):27–41. https://doi.org/10.1007/s13042-010-0007-7.
39. Tan TJL, Shokri R. Bypassing backdoor detection algorithms in deep learning. IEEE Eur Symp Secur Priv. 2020. https://doi.org/10.1109/eurosp48549.2020.00019.
40. Warzynski A, Kolaczek G. Intrusion detection systems vulnerability on adversarial examples. Innov Intell Syst Appl (INISTA). 2018. https://doi.org/10.1109/inista.2018.8466271.
41. Yang K, Liu J, Zhang C, Fang Y. Adversarial examples against the deep learning based network intrusion detection systems. IEEE Mil Commun Conf (MILCOM). 2018. https://doi.org/10.1109/milcom.2018.8599759.
42. Martins N, Cruz JM, Cruz T, Abreu PH. Adversarial machine learning applied to intrusion and malware scenarios: a systematic review. IEEE Access. 2020;8:35403–19. https://doi.org/10.1109/access.2020.2974752.
43. Arora A. Shantanu: a review on application of GANs in cybersecurity domain. IETE Tech Rev. 2020. https://doi.org/10.1080/02564602.2020.1854058.
44. Sadeghi K, Banerjee A, Gupta SKS. A system-driven taxonomy of attacks and defenses in adversarial machine learning. IEEE Trans Emerg Top Comput Intell. 2020;4(4):450–67. https://doi.org/10.1109/tetci.2020.2968933.
45. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. Scikit-learn: machine learning in python. J Mach Learn Res. 2011;12:2825–30.
46. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. 2016. arXiv preprint arXiv:1603.04467
47. Chollet F, et al. Keras. 2021. https://github.com/keras-team/keras
48. Pacheco Y, Sun W. Adversarial machine learning: a comparative study on contemporary intrusion detection datasets. Proc Int Conf Inf Syst Secur Priv. 2021. https://doi.org/10.5220/0010253501600171.
49. Pujari M, Cherukuri BP, Javaid AY, Sun W. An approach to improve the robustness of machine learning based intrusion detection system models against the carlini-wagner attack. 2022 IEEE International Conference on Cyber Security and Resilience (IEEE CSR). IEEE (2022). (in press)