Adversarial Machine Learning: A Comparative Study on Contemporary Intrusion Detection Datasets

Yulexis Pacheco and Weiqing Sun
College of Engineering, University of Toledo, Toledo, Ohio, U.S.A.

Keywords: Adversarial Machine Learning, Deep Learning, Deep Neural Networks, Intrusion Detection Datasets.

Abstract: Studies have shown the vulnerability of machine learning algorithms against adversarial samples in image classification problems in deep neural networks. However, there is a need for performing comprehensive studies of adversarial machine learning in the intrusion detection domain, where current research has been mainly conducted on the widely available KDD'99 and NSL-KDD datasets. In this study, we evaluate the vulnerability of contemporary datasets (in particular, UNSW-NB15 and Bot-IoT datasets) that represent the modern network environment against popular adversarial deep learning attack methods, and assess various machine learning classifiers' robustness against the generated adversarial samples. Our study shows the feasibility of the attacks for both datasets where adversarial samples successfully decreased the overall detection performance.

1 INTRODUCTION

Machine learning has become one of the most important techniques for image classification, voice recognition, intrusion detection, and many other systems. It has a significant impact on everyday tasks and it shaped how we process information and data. Machine learning methods have shown a great improvement in terms of processing time, scalability, and reliability. These techniques and methods are being used by many areas from health care, retail, government, transportation, and many others. Intrusion Detection Systems (IDS) has adopted machine learning to monitor network traffic to detect specific types of attacks against the systems. Traditional IDS could take relatively long time to analyse the complex data and provide results, which can result in the vulnerability of the system for missing alerts (Othman et al., 2018). To address the issue and other challenges, deep learning techniques have been the focus of the IDS area due to its efficiency at processing data and providing great analysis results.

The challenge of a machine learning algorithm is that it can be vulnerable against an adversary who tries to inject malicious data into the learning algorithm, with the main goal of making the algorithm fail to detect the attack (Papernot et al., 2016). This process is called adversarial machine learning which involves designing "machine learning algorithms that can resist sophisticated attacks and also the study of capabilities and limitations of attackers" (Huang et al., 2011). An attacker can adopt different techniques depending on the end goal such as seeking to launch targeted attacks and evade the detection systems. Other goals could be causing misclassification or lower accuracy metrics.

Many studies have showed that machine learning is vulnerable to adversarial data (Biggio et al., 2010; Carlini & Wagner, 2017; Goodfellow et al., 2015; Papernot et al., 2016; Rigaki & Elragal, 2017; Szegedy et al., 2014). Although there is a great deal of research based on image datasets, the issue with adversarial machine learning in the intrusion detection domain has been popular in the last few years (Wang, 2018). New challenges have arisen with machine learning in the intrusion detection field since an attacker can evolve and improve attack techniques by adopting available technology being developed, when the main focus was to find patterns and vulnerabilities in particular applications and systems (Biggio et al., 2010).

The main contribution of this work is to evaluate the effectiveness of adversarial deep learning attacks against contemporary datasets which represent the new networking and computing environment. This

Pacheco, Y. and Sun, W.
Adversarial Machine Learning: A Comparative Study on Contemporary Intrusion Detection Datasets.
DOI: 10.5220/0010253501600171
In *Proceedings of the 7th International Conference on Information Systems Security and Privacy (ICISSP 2021)*, pages 160-171
ISBN: 978-989-758-491-6

study provides the evaluation of different intrusion detection datasets such as UNSW-NB15 and Bot-IoT to demonstrate their vulnerability against popular adversarial attack methods including Jacobian Saliency Map Attack (JSMA), Fast Gradient Method (FGSM), and Carlini Wagner (CW). The study was performed by evaluating different metrics such as accuracy, AUC (area under the curve), F1 score, and Recall with different classifiers to compare and analyze the impact of the attacks with different datasets.

The rest of the paper is organized as follows: Section 2 presents an overview of adversarial machine learning and the methods used in this study. Section 3 presents related work in adversarial samples generation and adversarial machine learning. Section 4 discusses the experimental evaluation process for the study. Section 5 provides experimentation results. Section 6 provides a discussion of the adversarial attacks on the datasets. Finally, we conclude in Section 7.

2 BACKGROUND

2.1 Adversarial Machine Learning

The training and testing phases for machine learning algorithms are vulnerable against adversarial attacks, where the attacker can modify input data and lead to a misclassification result. An adversarial example is an input crafted to cause machine learning algorithms to misclassify the output. This process is performed during the test time after the algorithm has been trained (Papernot et al., 2016), and the process where the attacker crafts malicious input to fool the machine learning algorithms is called adversarial machine learning. This technique also involves designing and creating robust machine learning algorithms that can resist sophisticated attacks (Huang et al., 2011).

Relevant research related to adversarial machine learning describes specific characteristics of the attack model, the adversary, and the defenses. Barreno et al. (2006) presented three main properties for such an attack. Influence refers to the capability of the attacker and it could be causative (modify input training data) or exploratory (learn classifier decisions after sending instances to the classifier). The second property is security violation which covers integrity, availability, and privacy. The third property is specificity targeted (degrade classifier based on one instance target) and indiscriminate (the goal is to cause classifier failure based on a large number of classes). The threat model refers to the types of potential attacks considered by black-box attacks where the attacker has no information about the model or white-box attack where the attacker has access to all parameters of the model.

2.2 Adversarial Sample Generation

2.2.2 Jacobian based Saliency Map

One of the adversarial techniques evaluated in this work was Jacobian Based Saliency Map (JSMA) which was introduced by Papernot et al. (2016). This specific attack minimizes the L0 norm by iteratively calculating a saliency map and then perturbing the feature that will have the highest effect (Martins, 2019). 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* (Papernot et al., 2016):

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

Where F represents the second to last layer (Yuan et al., 2019). The perturbation is selected and the process continues until misclassification in the target class is achieved or the parameter for the maximum number of perturbed features is reached (Papernot et al., 2016). If it fails, then the algorithm selects the next feature and adds it to the perturbed sample (Rigaki & Elragal, 2017). The authors were successful by only modifying 4.02% of the input features per sample and achieved 97% adversarial success (Yuan et al., 2019). This process requires full knowledge of the targeted model's architecture and parameters (Papernot et al., 2016).

This attack can generate adversarial samples with a similar success rate as FGSM with the difference of using less feature modification with higher computing cost (Martins, 2019).

2.2.3 Fast Gradient Sign Method

Goodfellow et al. (2015) introduced a method called Fast Gradient Sign Method (FGSM) to generate adversarial examples. The perturbation is defined as:

$$\eta = \varepsilon * \mathrm{sign}(\nabla \mathrm{x}\, \mathrm{J}(\theta, \mathrm{x}, \mathrm{y})) \quad (2)$$

Where θ represents the parameters of a specific model, x is the input to the model, y represents the targets associated with x, J (θ, x, y) is the cost used to train the neural network (Goodfellow et al., 2015) and ε represents the magnitude of the attack, and the gradient can be obtained by backpropagation.

FGSM implements a loss function to decide the course of the input data in order to minimize this loss function (Wang, 2018). The authors successfully proved that this attack method is able to cause output misclassification with a variety of models (Goodfellow et al., 2015). The main goal of this attack was to be faster in the adversarial sample generation and it is not optimal in finding the minimal adversarial perturbations (Carlini & Wagner, 2017) which is why this attack method is the most efficient in terms of computing time.

2.2.4 Carlini Wagner

Carlini and Wagner (2017) proposed a powerful attack to evaluate the vulnerability of a secured model. This attack works by finding adversarial samples. The objective function is defined by:

$$\min_{\eta} \| \eta \|_p + c \cdot g(x+\eta) \quad s.t.\ x+\eta \in [0,1]^n \tag{3}$$

Where g(x') ≥ 0 if and only if f (x') = l' and l' is the label of the adversarial class in targeted adversarial examples (Yuan et al., 2019). In this way, the distance and penalty term can be better optimized (Wang, 2018). The authors provided an L2 attack to generate adversarial samples defined by (Carlini & Wagner, 2017):

$$\text{minimize } \|\frac{1}{2}(\tanh(w)+1) - x\|_2^2 + c \cdot f(\frac{1}{2}(\tanh(w)+1) \tag{4}$$

The main goal is to be able to lower distortion in the L2 metric. The authors also stated that this attack is robust against defensive distillation (Carlini & Wagner, 2017), making this attack the most powerful among the existing attacks against defenses. In this work, L2 was the one implemented by Cleverhans library (Papernot et al., 2017).

2.3 Dataset Overview

Datasets are fundamental in the process of developing new research. They play an important role in representing real-life network activity with labeled data, where each data point is assigned to a normal or attack class that will be used as evaluation criteria (Ring et al., 2017). Although there are many studies available in the intrusion detection domain, the lack of representative datasets which include a variety of attacks is one of the recurring issues, since most of the studies use the KDD'99 dataset or its derivation, the NSL-KDD (Rigaki & Elragal, 2017). The process to create new labeled datasets requires effort in addition to the complications that come with making a public dataset available since it would contain important information about the network, the users, and the current system environment (Javaid et al., 2016). In the last few years, there has been an effort in developing new datasets that contribute to new research as well as improving the quality of these datasets. The UNSW-NB15 dataset and the Bot-IoT dataset are two examples in this effort. Although the NSL-KDD dataset is the most commonly used dataset in the IDS domain, modern datasets such as the UNSW-NB15 and the Bot-IoT were used for this study. One of the reasons is because the NSL-KDD contains a large number of redundant records in the training set which definitely have an impact on results and multiple missing records affect the nature of the data (Moustafa & Slay, 2016). Another reason is that the NSL-KDD dataset is not a very comprehensive representation of an attack environment as the underlying network traffic of NSL-KDD dates back to 1998 (Ring et al., 2017). Both the UNSW-NB15 dataset and the Bot-IoT dataset represent realistic modern network traffic with diverse attack scenarios (Koroniotis et al., 2019).

2.3.1 UNSW-NB15 Dataset

This dataset was developed in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) (Moustafa & Slay, 2015). It represents new modern normal activities containing contemporary attacks. A partition of the full dataset is provided, divided into a training set and a test set according to the hierarchical sampling method, namely, *UNSW_NB15_training-set.csv* with 175,341 records and *UNSW_NB15_testing-set.csv* contains 82,332 records with a total of 257,673 records (Moustafa & Slay, 2016). The number of features is 43 with the class label. There are ten categories in total, one for normal class representing no attacks and nine attacks: shellcode, backdoor, exploits, worms, reconnaissance, generic, analysis, DoS, and fuzzers.

This dataset is more complex than the KDD'99 because it contains features where the attacks and normal classes have similar behaviors. Another reason is the correlation of the features for the UNSW-NB15 where training and test sets have the same distribution (Moustafa & Slay, 2016).

2.3.2 Bot-IoT Dataset

The Bot-IoT dataset was created by the Cyber Range Lab of the Centre of UNSW Canberra Cyber. The main characteristic of this dataset is that it represents a realistic network environment with more attacks

and network traffic in a realistic setting with its respective labels (Koroniotis et al., 2019). The dataset has normal IoT-related and other network traffic, along with various types of attack traffic commonly used by botnets. The attack categories in the dataset include Keylogging, Data exfiltration, DDoS, DoS, OS, and Service Scan. The dataset has 3 components: network platforms, simulated IoT services, extracting features and forensics analytics. The dataset has more than 72,000,000 records but a smaller version is available of around 3 million records. An extracted version of 5% of the total dataset was used in this study.

3 RELATED WORK

There has been extensive research on providing different techniques to generate adversarial samples. Szegedy et al. (2014) were the first to prove how vulnerable deep neural networks are against adversarial examples. Because of this research, the need to study adversarial attacks and defenses increased (Wang, 2018).

When it comes to extensive studies on defenses, Barreno et al. (2006) created a taxonomy of several attacks related to adversarial machine learning, as well as defensive methods and techniques against them. Also, they discussed how an adversary could successfully insert malicious data into the learning algorithm and provided detailed information about the attack model and its properties. Yuan et al. (2019) presented a taxonomy of adversarial attacks and defenses for deep neural networks, as well as challenges and possible defenses for these attacks.

Biggio et al. (2010) focused on improving classifiers under adversarial data manipulation, in addition to proposing a strategy for linear classifiers with Boolean features. Tan et al. (2020) focused on active adversarial attacks against machine learning algorithms, specifically on backdoor attacks, where the adversary manipulates training data and/or the training algorithm and parameters of the model to embed an adversarial sample.

Previous studies have created adversarial examples in the IDS domain successfully. Warzyński and Kołaczek (2018) studied the vulnerability of the NSL-KDD dataset against adversarial examples generated by FGSM. Wang (2018) researched the performance of attack algorithms against deep learning intrusion detection systems on the NSL-KDD dataset, as well as on the impact of feature selection in generating adversarial examples, demonstrating the vulnerability of deep learning algorithms against adversarial samples. He performed four attacks: FGSM, JSMA, Deepfool, and CW on a Multi-layer perceptron model. The most effective attack was targeted FGSM and the least efficient was CW. It also showed that it is not realistic to alter a very large set of features for an adversary, which explains why JSMA attacks are more popular than other attacks.

Rigaki and Elragal (2017) performed two types of attacks JSMA and FGSM on the NSL-KDD dataset. The study focused on evaluating the performance of several classifiers under attack such as RandomForest, Linear Support Vector Machine, Decision Tree, Multi-Layer Perceptron, and Voting Ensembles. The metrics used in this study were accuracy, AUC, and F1 score. The attacks implemented successfully decreased classifier performance by lowering accuracy on Linear SVM by 27% and Random Forest by 18%. The study demonstrated that adversarial methods can be implemented in the intrusion detection area. The authors showed that FGSM uses 100% of features to generate adversarial samples, while JSMA needs 6% of altered features to generate adversarial samples. The authors concluded that machine learning techniques need to be hand in hand with defensive methods against adversarial attacks.

Yang et al. (2018) showed how adversarial samples can have a negative impact on deep neural networks classifiers using the NSL-KDD dataset. They performed three attacks: CW, Zeroth-order Optimization (ZOO), and GAN. The process consisted of training a classifier to generate adversarial samples with CW and then attack another trained model. The same process was followed, but instead of CW, ZOO was implemented to get the gradient and then generate adversarial samples. Results showed a decrease of 70% in F1 score with the ZOO attack.

Martins et al. studied four attacks JSMA, DeepFool, CW, and FGSM on the NSL-KDDD and CICIDS2017 (Martins, 2019). They only performed the study on denial of service records and evaluated the attacks with Decision Tree, SVM, Naïve Bayes, Denoising Autoencoder (DAE), Neural Networks, and Random Forest. Results showed that overall performance is compromised by the attacks by decreasing 40% on CICDS2017 and 13% on the NSL-KDD. One key finding is that DAE was the most robust classifier.

Martins et al. (2020) presented a systematic review of adversarial machine learning applied to IDS and malware scenarios. In this work, the authors reviewed all existing research in the IDS field that

apply machine learning principles. One of the major findings is the lack of research using modern datasets. Of all the studies evaluated, six used the NSL-KDD, three implemented CTU-13, and one used CICIDS2017. According to the authors, the main reason is the lack of labelled intrusion datasets available. All studies observed performance degradation after adversarial attacks such as JSMA, WGAN, DeepFool, and FGSM. Results showed that the most affected classifiers are SVM, naïve Bayes, Decision Tree, while the most robust are Random Forest, and RBF SVMs.

4 EXPERIMENTAL EVALUATION

Our study is based on a multi-class classification problem. Bot-IoT has 5 classes while UNSW-NB15 has 10 classes. Based on this, four algorithms were selected to perform the experimental evaluation: Multi-Layer Perceptron (MLP), Decision Tree (DT), Random Forest (RD), and Support Vector Machine (SVM). The hyperparameters implemented for the experiment are presented in Table 1. In order to handle multi-class classification, OneVsRestClassifier was implemented to fit one classifier per class. The experimental evaluation, preprocessing and analysis were done with Python 3.6.5, Scikit-learn V.0.19.1 (Pedregosa et al., 2011), Tensorflow V.1.13.2 (Abadi et al., 2015), and Keras V.2.1.5 (Chollet, 2015). The attack algorithms were implemented with Cleverhans V.3.0.1 (Papernot et al., 2017), a library dedicated to assess machine learning vulnerability against adversarial samples.

4.1 Data Pre-processing

Data preparation is essential to train the machine learning algorithm. The first step was One-Hot encoding to convert all values to numerical data. The second step was standardization as not all features in the datasets are continuous values, but categorical or different data types that require pre-processing.

Table 1: Hyperparameters chosen for classifiers.

Classifier	Parameters
MLP	Dropout = 0.4, Layer 1 = 256, Layer 2 = 128, Activation = Relu, Loss = categorial crossentropy, Optimizer = Adam
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

4.1.1 One-Hot Encoding

One-Hot encoding was used to convert nominal values to numerical data. For example, the nominal values in the proto and service features need to be converted to numerical values. The UNSW-NB15 dataset has a total of 45 features and after One-Hot encoding, the number of features is 206. The Bot-IoT dataset has 39 features and after One-Hot encoding, the total number of features becomes 65.

4.1.2 Min-Max Normalization

After One-Hot encoding, Min-Max normalization was applied to all features on both datasets. The values were transformed to fit between 0 and 1. This step was important because datasets consist of numeric features whose values can be drawn from different distributions, have different scales, and, sometimes, contaminated by outliers (Wang, 2018). This affects results since features with very large values may cause imbalanced results by some classifiers. Besides, this step is needed by the attack methods so all features are within an interval (Martins, 2019).

4.2 Adversarial Sample Generation

The three selected adversarial attack algorithms include Jacobian Based Saliency Map Attack, Fast Gradient Sign Method, and Carlini Wagner attack. And they are available in the Cleverhans Library (Papernot et al., 2017). In addition, we chose targeted attacks since the goal of the experiment is to attack the normal class to be misclassified. The goal is to test the effectiveness of adversarial attacks on the Bot-IoT and UNSW-NB15 datasets and then evaluate the results based on the classifiers.

The study is divided into two main processes: model training for original data and adversarial sample generation. Model training for original data consists of pre-processing training and test data

(Figure 1) so it is scaled and prepared to train the machine learning algorithm. For this study, the model chosen for adversarial attack generation is a Multi-Layer Perceptron. The model was trained using training data to obtain the accuracy of normal samples. Then the evaluation of different machine learning algorithms such as Decision Tree, Support Vector Machine, and Random Forest were performed.

The same process was followed for the adversarial sample generation (Figure 2), where the pre-processed testing set was the input for the MLP to generate a new test set containing the poisoned samples by using three main attack methods JSMA, FGSM, and CW. After generating the poisoned adversarial test set, it was used for the evaluation of the classifier to make the predictions from the same machine algorithms and metrics mentioned above.

Targeted attacks for the normal class were performed on the datasets. The datasets are split into training and test sets. The study was performed in a white-box setting, where all the model information, target, and data are known by the attacker.

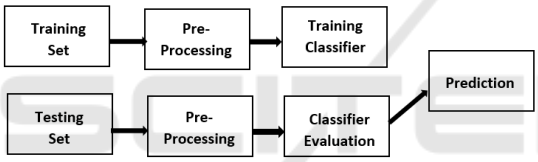

Figure 1: Training process for original data.

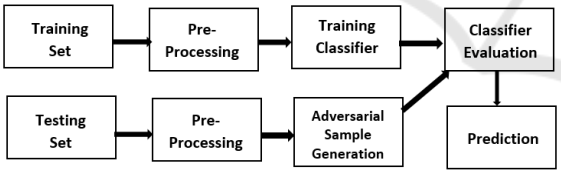

Figure 2: Adversarial sample generation process.

Multi-Layer Perceptron (MLP) was implemented as the source for the adversarial test set generation with Keras model (Chollet, 2015). Then, the JSMA attack algorithm was used to generate adversarial samples. The last step was evaluating the performance of the classifier with the original test set and the poisoned test set. The same process was followed for FGSM and Carlini Wagner attacks.

The default parameters used by the attacks are presented in Table 2. The values can be easily modified through Cleverhans (Papernot et al., 2017). For feature evaluation, we calculated the difference between the poisoned test set generated by the attacks and the original test set to obtain the list of features altered by the attacks.

Table 2: Parameters used by the attacks on both datasets.

Attacks	Parameters
JSMA	Theta = 1, Gamma = 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_const = 10

To obtain reasonable results for both datasets, the results were evaluated with averages provided by running the program 10 times for the UNSW-NB15 dataset and the Bot-IoT datasets.

5 EXPERIMENTATION RESULTS

As described in Section 4, the first process is to train and test all the baseline models on the original training set and test set from the UNSW-NB15 and Bot-IoT datasets. The second process is to generate an adversarial test set with the baseline model MLP with the attack algorithms including JSMA, FGSM, and CW to then target the machine learning classifiers including DT, RF, and SVM. After the adversarial attacks, we evaluated the results on the original data before the attacks and after perturbed samples are included. The metrics chosen for the evaluation are ROC AUC, accuracy, F1 score, and Recall. The final results are averaged from 10 runs in order to obtain more accurate final results to measure the impact of the adversarial samples on the classifiers.

5.1 UNSW-NB15 Dataset

Table 3: Accuracy results for the UNSW-NB15 dataset.

Classifier	Accuracy			
	Baseline	JSMA	FGSM	CW
MLP	0.72	0.38	0.39	0.21
SVM	0.60	0.25	0.16	0.23
DT	0.64	0.54	0.24	0.18
RF	0.64	0.63	0.34	0.32

Table 4: AUC results for the UNSW-NB15 dataset.

Classifier	AUC			
	Baseline	JSMA	FGSM	CW
SVM	0.89	0.36	0.57	0.73
DT	0.84	0.66	0.77	0.53
RF	0.92	0.92	0.83	0.78

Table 5: F1 score results for the UNSW-NB15 dataset.

Classifier	F1			
	Baseline	JSMA	FGSM	CW
SVM	0.70	0.32	0.23	0.35
DT	0.69	0.64	0.35	0.27
RF	0.73	0.72	0.33	0.38

Table 6: Recall results for the UNSW-NB15 dataset.

Classifier	Recall			
	Baseline	JSMA	FGSM	CW
SVM	0.66	0.25	0.26	0.28
DT	0.66	0.64	0.41	0.26
RF	0.64	0.63	0.23	0.26

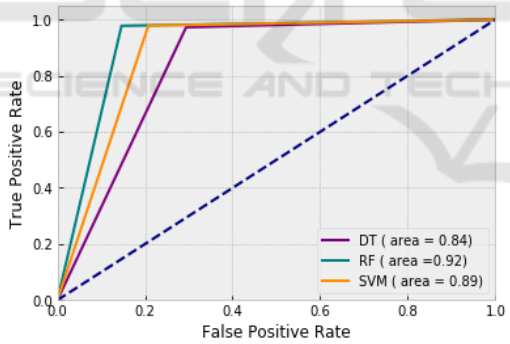

Figure 3: ROC Curve for baseline results on the UNSW-15 dataset.

5.1.1 Jacobian Saliency Map Attack

The results from the evaluation of the original data and after the attack on the UNSW-NB15 are presented in Tables 3-6. The baseline model MLP, used to generate adversarial samples on the test set is affected by the JSMA attack for this dataset as expected with a 34% drop in accuracy (from 0.72 to 0.38). Results showed the UNSW-NB15 presented a decrease in accuracy for SVM, DT, and RF that ranges from 1% to 35%, AUC from 0% to 53%, F1 score from 1% to 38%, and Recall from 1% to 41%. It demonstrated an overall decrease in performance of the JSMA on the UNSW-NB15.

The most severely affected algorithm after the evaluation with the perturbed samples was SVM in terms of accuracy, AUC, F1 score, and Recall (see Tables 3-6). SVM presented a decrease in accuracy of 0.35, AUC of 0.53, F1 score of 0.38, and Recall 0.41. This finding confirms the results obtained from Rigaki and Elragal (2017) where the weakest classifier was also linear SVM. Meanwhile, RF presented no change in AUC after the attack and only 0.01 drop in accuracy, F1 score, and Recall. DT was affected by the attack with a 0.1 drop in accuracy, 0.18 AUC, 0.05 F1 score, and 0.02 Recall.

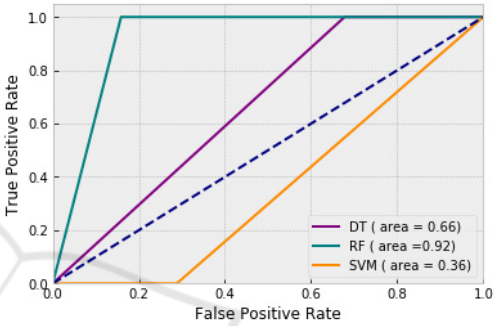

Figure 4: ROC Curve for normal class after JSMA attack on the UNSW-NB15 dataset.

The top 10 features used by the adversarial samples are *sbytes, smean, sinpkt, dur, spkts, sload, sttl, rate, ct_srv_src, ct_state_tt.* The feature categories include basic information features and connection/content related features (Moustafa & Slay, 2016). The number of unique features changed by the JSMA attack is 95 and the number of average features changed per data point is 22. The percentage of altered features is 11%. The adversarial sample generation process took 9 minutes and 34 seconds to finish. JSMA attacks are well known for its high computational cost which is why the attack is slower than FGSM (Yuan et al., 2019).

5.1.2 Fast Gradient Sign Method Attack

The metrics from the FGSM attack on the UNSW-NB15 are presented in Tables 3-6. The baseline model MLP used to generate adversarial samples is affected by the FGSM attack with 0.33 drop in accuracy (from 0.72 to 0.39), which successfully degraded the performance of the model. Also, the overall range decrease in accuracy is from 30% to 44%, AUC from 7% to 32%, F1 score from 34% to 47%, and Recall from 25% to 41%. It demonstrated

the vulnerability of the UNSW-NB15 dataset against FGSM.

The performance of the classifiers varies based on the metrics selected for the study. SVM decreased by 0.44 in accuracy, AUC by 0.32, F1 score by 0.47, and Recall by 0.4. Meanwhile, DT showed a drop in accuracy of 0.4, AUC by 0.07, F1 score by 0.34, and Recall by 0.25. RF showed a decrease in accuracy of 0.3, AUC by 0.09, F1 score by 0.4, and Recall by 0.41.

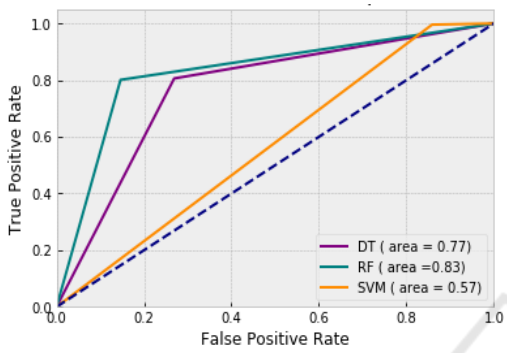

Figure 5: ROC Curve for normal class after FGSM attack on the UNSW-NB15 dataset.

ROC AUC curves for the models showed an overall decrease in performance against targeted misclassification related to the normal class (Figure 3 and Figure 5). It is clear SVM was severely affected by the attack with a decrease of 0.32. The classifiers RF and DT were similarly degraded by the attack with a decrease of 0.09 and 0.07 respectively.

The top 10 features used by the adversarial samples are *sbytes, spkts, sttl, dur, smean, sinpkt, sload, rate, ct_state_ttl, ct_srv_src.* Feature categories are basic information, connection, time-related, and content. Another finding is that the top features are more evenly distributed compared to JSMA attacks. The number of unique features changed is 196 and the number of average features changed per data point with FGSM is 162. The percentage of altered features is 78%.

The adversarial sample generation process took 5 seconds to complete. The FGSM attack was designed to run faster than other attack algorithms (Yuan et al., 2019) and this study has confirmed that FGSM attack is the fastest to generate adversarial samples for the UNSW-NB15 dataset.

5.1.3 Carlini Wagner Attack

The results for the CW attack on the UNSW-NB15 are presented in Tables 3-6. The baseline model MLP, used to generate adversarial samples for the normal class is affected by the CW attack for this dataset with about a 0.5 decrease in accuracy (from 0.72 to 0.21) as shown in Table 3. Results showed that CW attacks have a negative impact on accuracy with a decrease from 32% to 46%, AUC from 14% to 31%, F1 score from 35% to 42%, and Recall from 38% to 40%.

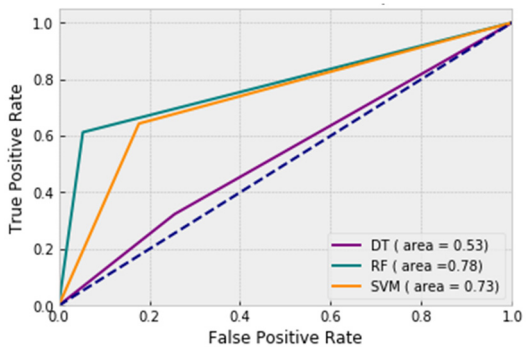

Figure 6: ROC Curve for normal class after CW attack on the UNSW-NB15 dataset.

The performance of RF is significantly better than SVM and DT in terms of accuracy. RF showed a decrease in accuracy of 0.32, AUC by 0.14, F1 score by 0.35, and Recall by 0.38. SVM had a decrease of 0.37 in accuracy, AUC by 0.16, F1 score by 0.35, and Recall by 0.38. Meanwhile, DT showed a decrease across all metrics with a drop by 0.46, AUC by 0.31, F1 score by 0.42, and Recall by 0.40.

In terms of AUC, RF and SVM performed similarly with a decrease of 14% and 16% respectively (Figure 3 and Figure 6). DT is shown as the most affected with a decrease of 31%.

The top 10 features used by the adversarial samples are *sbytes, smean, sinpkt, dur, sload, sttl, spkts, ct state ttl, rate, sjit.* The features chosen are related to basic, time, and connection features, similar to the results obtained by JSMA and FGSM attacks. Also, the number of unique features changed with the CW attack is 196, while the number of average features changed per data point is 133. The percentage of altered features is 65%. The adversarial generation process for this attack took 51 minutes and 24 seconds. This process is the slowest of all the three attacks for the UNSW-NB15.

5.2 Bot-IoT Dataset

Table 7: Accuracy results on the Bot-IoT dataset.

Classifier	Accuracy			
	Baseline	JSMA	FGSM	CW
MLP	0.90	0.39	0.37	0.35
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

Table 8: AUC results on Bot-IoT dataset.

Classifier	AUC			
	Baseline	JSMA	FGSM	CW
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 9: F1 score results on Bot-IoT dataset.

Classifier	F1			
	Baseline	JSMA	FGSM	CW
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 10: Recall results on Bot-IoT dataset.

Classifier	Recall			
	Baseline	JSMA	FGSM	CW
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

5.2.1 Jacobian Saliency Map Attack

The results from the evaluation of the original dataset and the attacks for the Bot-IoT are presented in Tables 7-10. The baseline model, MLP is affected by the JSMA attack for this dataset with a 51% drop in accuracy (from 0.94 to 0.48). The Bot-IoT showed an overall decrease in performance across all metrics with accuracy from 13% to 56% after the attack, AUC from 0% to 49%, F1 score from 3% to 38%, and Recall from 4% to 55%.

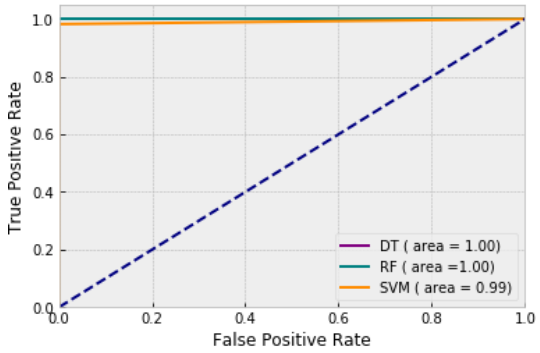

Figure 7: ROC Curve for baseline results on the Bot-IoT dataset.

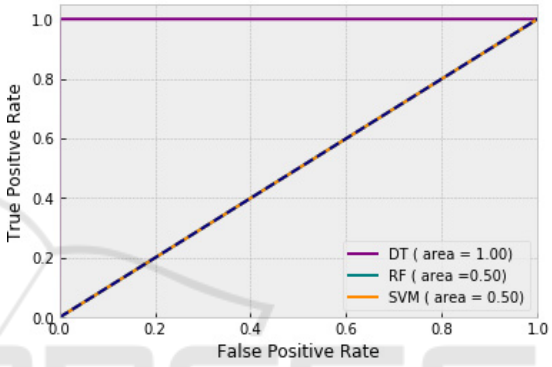

Figure 8: ROC Curve for normal class after JSMA attack on the Bot-IoT dataset.

DT presented a 0.51 drop in accuracy, no change in AUC, F1 score by 0.38, and Recall by 0.4. DT was the most affected with accuracy, F1, and Recall. RF showed similar results as SVM with a decrease in accuracy of 0.13, AUC of 0.49 0.03 F1 score, and 0.04 Recall. SVM was degraded by 0.46 in accuracy, AUC by 0.49, F1 score by 0.23, and Recall by 0.07.

In terms of AUC, SVM, and RF were equally affected by the attack as shown in Figure 7 and Figure 8 with 0.5 drop in accuracy. The AUC for DT was not affected after the JSMA attack.

The top 10 features used by the adversarial samples are *TnBPSrcIP, stime, TnBPDstIP, seq, Pkts_P_State_P_Protocol_P_DestIP, TnP_PSrcIP, TnP_PerProto, Pkts_P_State_P_Protocol_P_SrcIP, TnP_PDstIP*. These features belong to flow and basic categories. The number of unique features changed was 57 and the number of average features changed per data point is 28. The percentage of altered features is 43%.

The adversarial generation process for this attack took 16 minutes and 5 seconds.

5.2.2 Fast Gradient Sign Method Attack

The results for the FGSM attack on the Bot-IoT dataset are presented in Tables 7-10. The baseline model, MLP is affected by the FGSM attack for this dataset with a 53% drop in accuracy (from 0.90 to 0.37). All metrics were degraded for all algorithms implemented SVM, DT, and RF after the evaluation with the perturbed samples for the FGSM attack. The decrease in accuracy ranges from 51% to 53% for all classifiers, AUC from 1% to 2%, F1 score from 53% to 67%, and Recall from 58% to 60%.

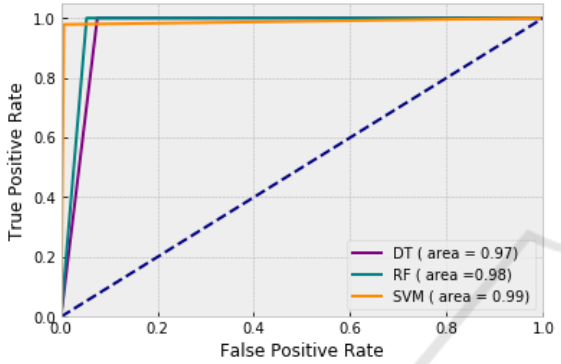

Figure 9: ROC Curve for normal class after FGSM attack on the Bot-IoT dataset.

Based on the results from all metrics, SVM had a decrease in accuracy by 0.54, AUC by 0.01, F1 score by 0.67, and Recall by 0.59. DT decreased by 0.51 in accuracy, AUC by 0.02, F1 score by 0.53, and Recall by 0.6. RF had a drop in accuracy by 0.52, AUC by 0.01, F1 score by 0.57, and Recall by 0.58. The results confirm that FGSM attacks had a negative impact on the Bot-IoT dataset across all metrics. AUC score was degraded for all algorithms SVM, DT and RF with a small decrease between 0.01 and 0.02 as shown in Figure 7 and 9.

The top 10 features used by the adversarial samples are *seq, TnP_PerProto, stime, bytes, Pkts_P_State_P_Protocol_P_DestIP, TnP_Per_Dport, TnBPSrcIP, TnBPDstIP, TnP_PDstIP, ltime*. These features belong to flow and basic categories. The number of unique features changed by the attack was 60 and the number of average features changed per data point is 34. The percentage of altered features is 52%.

The adversarial sample generation process for this attack finished in 33 seconds. FGSM was observed to generate adversarial samples faster than the other two attacks.

5.2.3 Carlini Wagner Attack

The results for the CW attack on the Bot-IoT dataset are presented in Tables 7-10. Accuracy for the baseline model MLP was affected by the CW attack for this dataset with an almost 0.5 decrease in accuracy (from 0.9 to 0.43). The results confirm that the CW attacks have an impact on accuracy for the Bot-IoT dataset with a decrease in accuracy from 34% to 46% for all classifiers, AUC from 2% to 4%, F1 score from 32% to 42%, and Recall from 40% to 42%. In terms of AUC, it is not severely affected by adversarial samples with an average drop in accuracy of 3% (Figure 7 and Figure 10), presenting a small difference of 0.02 to 0.04 (Table 8).

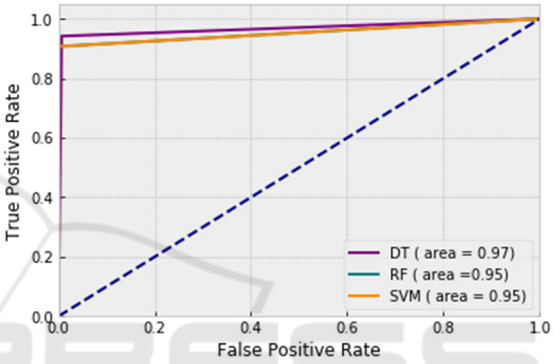

Figure 10: ROC Curve for normal class after CW attack on the Bot-IoT dataset.

Based on the results from all the metrics, all the classifiers were affected by the CW attack. SVM decreased in accuracy by 0.46, AUC by 0.04, F1 score by 0.42, and Recall by 0.41. DT had a decrease in accuracy by 0.34, AUC by 0.02, F1 score by 0.32, and Recall of 0.4. RF had a drop in accuracy of 0.39, AUC of 0.04, F1 score of 0.42, and Recall of 0.42.

The top 10 features used by the adversarial samples are *stime, bytes, TnP_Per_Dport, Pkts_P_State_P_Protocol_P_SrcIP, sbytes, AR_P_Proto_P_DstIP, AR_P_Proto_P_SrcIP, pkts, AR_P_Proto_P_Sport, dur*. These features belong to flow and basic categories. Most of them are related to protocol transactions in the network flow. The number of unique features changed was 59 and the number of average features changed per data point is 42. The percentage of altered features is 65%.

The adversarial generation process for this attack finished in 2 hours and 14 minutes as it is more computationally expensive compared to FGSM and JSMA.

6 DISCUSSIONS

The performance of JSMA, FGSM, and CW attacks varies based on the datasets used. On the UNSW-NB15 dataset, an average accuracy of all classifiers decreased by 33%, AUC by 20%, F1 score by 31%, and Recall 30%, while Bot-IoT showed an average decrease in accuracy by 47%, AUC by 12%, F1 score by 41%, and Recall 40%. Both datasets are vulnerable against JSMA, FGSM, and CW attacks. However, the UNSW-NB15 had a lower decrease across all the metrics compared to Bot-IoT. This is because Bot-IoT has fewer features, making it more vulnerable to adversarial attacks.

The robustness of the classifiers and attacks also varies between datasets. CW had the best performance in terms of accuracy and Recall on the UNSW-NB15 dataset with an overall decrease in accuracy by 42% and recall by 39%, while FGSM was the most efficient in decreasing the average performance for F1 score by 40%. In terms of AUC, JSMA decreased it by 24%. On the Bot-IoT dataset, results show FGSM is the most efficient attack with all the three metrics by decreasing the overall average by 53% in accuracy, F1 score by 59%, and Recall by 59%. The most efficient attack at degrading AUC was JSMA with an overall average decrease of 33%. JSMA and FGSM were shown as the most consistent attacks for both datasets with the least efficient performance. JSMA was the least efficient in terms of accuracy, F1, and Recall, while FGSM was the least successful in terms of AUC. This means the attacks' performances vary across datasets and affect the metrics differently which is an important factor for the attacker to consider when launching an attack. Time spent on adversarial sample generation is another factor to consider. The attack that takes the longest to generate perturbed samples is CW, while the most efficient attack is FGSM.

The classifier with the best performance on the UNSW-NB15 is RF with accuracy overall average decrease by 21%, AUC by 8%, and F1 score by 25%, while the best performance in terms of Recall was DT by 22%. The least robust classifier is SVM with an overall decrease across all the metrics with a 39% drop in accuracy, 34% decreased in AUC, 40% drop in F1 score and Recall. On the Bot-IoT dataset, the most robust classifier is RF in terms of accuracy, F1, and Recall by 35%, 34%, and 35% respectively. DT was the most robust in terms of AUC with a decrease of only 1%. On the contrary, the least robust classifier is RF and SVM in terms of AUC with an overall decrease of 18%. Meanwhile, SVM is the least robust in terms of accuracy and F1 with a decrease of 49% and 44% respectively, while DT had the worst performance in terms of Recall with a decrease of 52%.

The average number of features used by all attacks for the Bot-IoT is 59 out of 65, while the UNSW-NB15 dataset is 162 out of 206. JSMA required the least number of features to modify for both datasets with 11% and 43% for the UNSW-NB15 and Bot-IoT respectively. FGSM required 78% of features for the UNSW-NB15, and 52% for the Bot-IoT dataset. CW attack required 65% of all the features modified for both datasets. According to this finding, JSMA required the least number of features modified and it would need less effort for the attacker compared to FGSM and CW. Another observation is the feature perturbation distribution by the attack. On the UNSW-NB15, basic category features were the most perturbed by all attacks. For the Bot-IoT most of the features related to protocol and packet information were targeted by all attacks. Another finding is that CW also altered time-related features while JSMA targeted connection features in addition to basic features.

7 CONCLUSIONS

In this work, we use modern IDS datasets UNSW-NB15 and the Bot-IoT to study the impact of popular adversarial machine learning attacks JSMA, FGSM, and CW against machine learning classifiers. This study demonstrated that the above-mentioned attacks were able to effectively degrade the overall performance of the different classifiers SVM, DT, and RF used on the two IDS datasets. RF was shown as the most resilient classifier, while SVM was the least robust on both datasets. The attacks presented varied results based on the two datasets. Overall, JSMA was the least efficient on both datasets. CW was most efficient attack on the UNSW-NB15, while FGSM was the most efficient attack on the Bot-IoT dataset.

For the future work, we will incorporate a broader selection of contemporary IDS datasets, adversarial machine learning attacks and machine learning classifiers into the research. In addition, we will work on developing and evaluating various defensive techniques against adversarial machine learning attacks to improve the robustness of machine learning algorithms used in IDS systems.

REFERENCES

Abadi et al. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems*. Retrieved from https://www.tensorflow.org/about/bib

Barreno, M., Nelson, B., Sears, R., Joseph, A., & Tygar, J. (2006). Can machine learning be secure? *ASIACCS '06.*

Biggio, B., Fumera, G., & Roli, F., (2010). Multiple classifier systems for robust classifier design in adversarial environments. *International Journal of Machine Learning and Cybernetics*, 1, 27-41.

Carlini, N., & Wagner, D., (2017). Towards Evaluating the Robustness of Neural Networks. *2017 IEEE Symposium on Security and Privacy (SP),* 39-57.

Chollet, F. (2015). *keras*, Retrieved from https://github.com/fchollet/keras.

Goodfellow, I.J., Shlens, J., & Szegedy, C., (2015). Explaining and Harnessing Adversarial Examples. *CoRR, abs/1412.6572.*

Huang, L., Joseph, A., Nelson, B., Rubinstein, B.I., & Tygar, J., (2011). Adversarial machine learning. *In Proceedings of the 4th ACM workshop on Security and artificial intelligence, 43–58.*

Javaid, A., Niyaz, Q., Sun, W., & Alam, M., (2016). A Deep Learning Approach for Network Intrusion Detection System. *EAI Endorsed Trans. Security Safety*.

Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B., (2019). Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset. *Future Gener. Comput. Syst., 100*, 779-796.

Martins, N., (2019). Analyzing the footprint of classifiers in adversarial DoS contexts. *Proc. EPIA Conf. Artif. Intell, pp. 256–267.*

Martins, N., Cruz, J.M., Cruz, T., & Abreu, P.H., (2020). Adversarial Machine Learning Applied to Intrusion and Malware Scenarios: A Systematic Review. *IEEE Access, 8*, 35403-35419.

Moustafa, N., & Slay, J., (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *2015 Military Communications and Information Systems Conference (MilCIS)*, 1-6.

Moustafa, N., & Slay, J., (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective, 25*, 18 - 31.

Othman, S.M., Ba-Alwi, F., Alsohybe, N.T., & Al-Hashida, A.Y., (2018). Intrusion detection model using machine learning algorithm on Big Data environment. *Journal of Big Data, 5*, 1-12.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.Y., & Swami, A., (2016). The Limitations of Deep Learning in Adversarial Settings. *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, 372-387.

Papernot et al. (2017). Cleverhans v2.0.0: an adversarial machine learning library. *arXiv:1610.00768v4*

Pedregosa et al., (2011). Scikit-learn: Machine Learning in Python, *JMLR 12: 2825-2830.*

Rigaki, M., Elragal, A., (2017). Adversarial Deep Learning against Intrusion Detection Classifiers. *ST-152 Workshop on Intelligent Autonomous Agents for Cyber Defence and Resilience*.

Ring, M., Wunderlich, S., Grüdl, D., Dieter Landes, D., & Hotho, A., (2017). Flow-based benchmark data sets for intrusion detection. *European Conference on Cyber Warfare and Security (ECCWS), ACPI*, *361–369.*

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., & Fergus, R., (2014). Intriguing properties of neural networks. *CoRR, abs/1312.6199*.

Tan, T., Lester, J. & Shokri, R., (2020). Bypassing Backdoor Detection Algorithms in Deep Learning. *IEEE European Symposium on Security and Privacy (EuroS&P), pp. 175-183.*

Wang, Z., (2018). Deep learning-based intrusion detection with adversaries. *IEEE Access 6 38367-38384.*

Warzynski, A., & Kolaczek, G., (2018). Intrusion detection systems vulnerability on adversarial examples. *2018 Innovations in Intelligent Systems and Applications (INISTA)*, 1-4.

Yang, K., Liu, J., Zhang, C., & Fang, Y., (2018). Adversarial Examples Against the Deep Learning Based Network Intrusion Detection Systems. *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, 559-564.

Yuan, X., He, P., Zhu, Q., & Li, X., (2019). Adversarial Examples: Attacks and Defenses for Deep Learning. *IEEE Transactions on Neural Networks and Learning Systems, 30*, 2805-2824.