Computers, Materials & Continua
DOI: 10.32604/cmc.2022.029858
Article

Tech Science Press

Adversarial Training Against Adversarial Attacks for Machine Learning-Based Intrusion Detection Systems

Muhammad Shahzad Haroon* **and Husnain Mansoor Ali**

Department of Computer Science, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Karachi, 75600, Pakistan
*Corresponding Author: Muhammad Shahzad Haroon. Email: shahzad.haroon@szabist.edu.pk
Received: 13 March 2022; Accepted: 26 April 2022

Abstract: Intrusion detection system plays an important role in defending networks from security breaches. End-to-end machine learning-based intrusion detection systems are being used to achieve high detection accuracy. However, in case of adversarial attacks, that cause misclassification by introducing imperceptible perturbation on input samples, performance of machine learning-based intrusion detection systems is greatly affected. Though such problems have widely been discussed in image processing domain, very few studies have investigated network intrusion detection systems and proposed corresponding defence. In this paper, we attempt to fill this gap by using adversarial attacks on standard intrusion detection datasets and then using adversarial samples to train various machine learning algorithms (adversarial training) to test their defence performance. This is achieved by first creating adversarial sample based on Jacobian-based Saliency Map Attack (JSMA) and Fast Gradient Sign Attack (FGSM) using NSLKDD, UNSW-NB15 and CICIDS17 datasets. The study then trains and tests JSMA and FGSM based adversarial examples in seen (where model has been trained on adversarial samples) and unseen (where model is unaware of adversarial packets) attacks. The experiments includes multiple machine learning classifiers to evaluate their performance against adversarial attacks. The performance parameters include Accuracy, F1-Score and Area under the receiver operating characteristic curve (AUC) Score.

Keywords: Intrusion detection system; adversarial attacks; adversarial training; adversarial machine learning

1 Introduction

Machine learning models are currently being deployed in many domains [1]. Researchers are focusing on robustness of machine learning algorithms to maintain performance. One of the major threats to robustness of machine learning algorithms is adversarial attacks. These attacks are aimed to fool the machine learning model and misclassify the output. Models are often times trained with expectations in mind for ease of computation, such as feature independence and linear separability

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

of the data, but these types of convenience can at times open possibilities for adversarial attacks and make models vulnerable [2].

Adversarial attacks can be classified into two types: White-box attacks and black-box attacks. In white-box attacks, an adversary has the knowledge of the trained model, training data, network architecture hyper parameters etc. Whereas, in a black-box attack, an adversary has no access to training data and training model. Thus an adversary acts as a normal user and only knows the output of the model (label or confidence score).

Security concerns in enterprise networks remains a major worry as cyber threats increase day by day [3]. An intrusion detection system (IDS) is considered the main defence system against these cyber threats. Hackers are inventing new techniques frequently which can bypass the IDS. IDS are categorized into two major categories: Signature-based and anomaly-based. Signature-based IDS systems are developed by extracting information used in earlier attacks which are called a signature. Every time a new attack appears, the signature must be updated into the system [4]. Whereas anomaly-based IDS systems inspect traffic based on the behaviour of activities. Anomaly-based models are trained to classify normal and malicious traffic which can detect new attacks as well [5].

Researchers have used Machine Learning (ML) in anomaly-based IDS with the hope of improving intrusion detection. The limitation of ML models concerning the security of the model itself has been explored in the literature. Researchers have focused on the image processing domain and investigated it thoroughly [6–9]. Similarly, machine learning models have also been found to be vulnerable in the intrusion detection domain. There is a limited number of studies that have investigated the adversarial attack on machine learning-based intrusion detection systems like [2,5,10,11] while some papers have also studied their defence [12–14]. Some of the relevant papers related to current research are discussed in Section 3.

In this paper, the focus has been on adversarial defence. Multiple datasets are used for generation of adversarial attacks. The models trained using different ML datasets are then compared for performance. Models are also trained on adversarial attacks and their performance then analyzed. The paper is organized as follows: Section 2 discusses generation of adversarial attacks. Related literature is reviewed in Section 3. Experimental setup is discussed in Section 4 whereas results are discussed in Section 5. We conclude in Section 6.

2 Generation of Adversarial Attacks

Multi-layer Perceptron (MLP) [15] is used for adversarial attack generation. It is a feed-forward neural network and consists of fully connected three layers. The input layer receives the input to be processed. The output layer provides the predictions and classification of the received input. The hidden layer is the computation engine where all the inputs are processed. MLP is made of neurons called perceptron. The structure of a perceptron is given in Fig. 1.

In the MLP network, each perceptron receives $\boldsymbol{n}$ features as input ($x_1, x_2, \ldots x_n$) and each feature is associated with weights ($w_1, w_2, \ldots, w_n$). The input features are passed on to an input function $\boldsymbol{u}$, which computes the weighted sum of the input features as given in Eq. (1):

$$\mathrm{u}\,(\mathrm{x}) = \sum_{i=1}^{n} w_i x_i \tag{1}$$

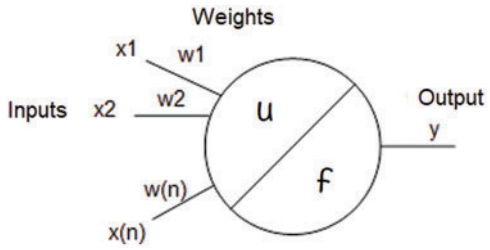

Figure 1: Structure of perceptron

The result of this computation is then passed onto an activation function $\boldsymbol{f}$, which will produce the output of the perceptron. For example, a step function can act as an activation function as given in Eq. (2):

$$y = f\left(u\left(x\right)\right) = \begin{Bmatrix} 1, & \text{if } \mathrm{u}\left(x\right) > \theta \\ 0, & otherwise \end{Bmatrix} \tag{2}$$

where $\boldsymbol{\theta}$ is the threshold parameter.

2.1 Jacobian-based Saliency Map Attack

Jacobian-based Saliency Map Attack (JSMA) was proposed in 2016 [16]. The aim was to misclassify by minimizing the modified features involved in an adversarial example generation. In this method, a saliency map is created for the input sample which has the saliency values for each feature. This saliency value suggests how much the classification process is manipulated. According to the saliency value each feature is selected in decreasing order. The process continues until the modified feature threshold is reached or misclassification occurs. This process creates adversarial examples close to the original sample [3].

For the white-box attack category, JSMA is more suitable for an adversary [2] but requires high computational power. In [10], Euclidean distance is used to measure the closeness of the original sample and adversarial sample which confirms 99% similarity. Only 6% of features are modified for the generation of adversarial samples in [4].

2.2 Fast Gradient Sign Attack

Fast Gradient Sign Attack (FGSM) was first proposed in 2014 [7]. The FGSM attack on neural networks is formulated by using gradients. The neural network minimizes the loss by adjusting weights through the feedback of back propagated gradients. To attack the neural network, the FGSM attack maximizes the loss using the same back propagated gradients.

The FGSM based adversarial attack is formulated as given in Eq. (3):

$$x' = x \ + \epsilon \ .sign\left(\nabla_x J\left(\theta, x, y\right)\right) \tag{3}$$

where x are the inputs to the model, ϵ is the magnitude of the perturbation and J(θ, x, y) is the gradient of the adversarial loss.

FGSM attack was initially evaluated on image related datasets like ImageNet [17], MNIST [18] and CIFAR [19] where they added a very small amount of carefully constructed noise to misclassify the object. It was later used in the intrusion detection domain like in [2,20]. FGSM is a white box category that modifies 100% features to generate adversarial samples [4]. The authors in [3] also summarized

FGSM as less effective but efficient with respect to computational time. Fig. 2 gives an example of how JSMA and FGSM changes attack traffic so that they appear as normal traffic to the classifiers.

0, udp,private, SF, 45, 44, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 505, 505, 0, 0, 0, 0, 0 , 1, 0 , 0, 255, 255, 1, 0, 1, 0, 0, 0, 0, 0	Attack
0, udp,private, SF, 45, 44, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 505, 505, 0, 0, 0, 0, 0 1, 0 , 0, 255, 255, 1, 0, 0, 1, 0, 0, 0, 0	Normal

Figure 2: An illustration how perturbation creates an adversarial attack traffic that is considered as normal traffic

3 Related Work

The following Section details adversarial machine learning training and attack related work done by others.

The authors in [4] used FGSM and JSMA for the generation of adversarial attacks in the white box category by using the NSLKDD [21] dataset. They conclude an important observation on the percentage of features modified by the attacks and preferred JSMA over FGSM. The features modified by FGSM is 100% on every sample, while only 6% on JSMA features. This makes JSMA a more practical attack. Due to domain-related limitations, the attacker has to relate which features he can modify. The author used MLP to generate JSMA based adversarial attacks. Testing was done by using the adversarial examples against Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF) and ensemble of these three classifiers called a voting classifier. The performance parameters included accuracy, F1-Score and AUC normal. It was found that the performance of all the baseline models decreased. The most affected classifier was SVM, and the most resilient was RF. The author also presented the top 20 contributed features involved in adversarial attacks. No solution or defence against attacks were provided. Also, there were no results based on FGSM and only JSMA results were provided on a single dataset.

The authors in [2] used JSMA, FGSM, Deepfool [22] and Carlini et al. (C&W) [23] attacks for the generation of adversarial examples using the NSLKDD dataset in a white box setting. The author used MLP as a neural network with ReLU [24] activation function. The MLP model was tested against adversarial samples. The main contribution of the author is to highlight the features that most participate in generating each attack. The performance parameters include accuracy, precision, recall, false alarm, F1-score. The accuracy for JSMA attack is 52.41%, untargeted FGSM is 40.78%, targeted FGSM is 50.66%, Deepfool is 41.03% and C&W is 64.8%. Author further concludes that C&W attacks seem to be less devastating than the other three attacks.

In [24], the authors proposed IDSGAN to generate adversarial attacks to defeat intrusion detection systems. The NSLKDD dataset is used in a black box scenario in which only malicious traffic is used to generate an adversarial attack. For the black box IDS training, a few algorithms are used like SVM, Naïve Bayes (NB), MLP, Logistic Regression (LR), DT, RF, k-nearest neighbors (K-NN). The author analyzes the algorithms on the detection rate and evasion increase rate which shows that IDSGAN successfully evaded the algorithms and DoS, U2R and R2L attacks are undetected by them.

The authors in [12] proposed DOS-WGAN, which uses Wasserstein generative adversarial network (WGAN) with a gradient penalty method to evade the classifier. The author uses standardized Euclidean distance which maps the adversarial samples to the original data distribution. Standardized Euclidean distance and information entropy is used to assess generative adversarial network (GAN)

training. The author uses KDDCUP 99 dataset. The author includes three types of experiments DoS-GAN with an accuracy of 69.7%, DoS-WGAN with clipping (WGAN-CLIP) accuracy of 57.1%, and DoS-WGAN with gradient penalty (WGAN-GP) accuracy of 47.6% with most stable training, respectively. The WGAN-CLIP model means that the weight of the generator is limited in [−0.01, 0.01] and the WGAN-GP model means DoS-WGAN uses gradient penalty.

In [11], the authors evaluated adversarial attacks in a black box scenario on the NSLKDD dataset. Three different types of black-box attacks were launched. The first attack in which the adversary trained a substitute model with white box limitations. The attacks generated on substitute models are created using C&W attacks. The second attack is based on zeroth-order optimization (ZOO). The third attack is generated using GAN. The following parameters are used to measure the performance of classifiers: Accuracy, precision, recall, false alarm and F1-score. The impact of the first attack approach that uses a substitute model is less as compared to the second and third approaches. The second approach performed better among three black-box attacks but required a large number of queries and computation power to calculate the gradients.

The authors in [5] used a neural network for the implementation of the intrusion detection system. The NSLKDD dataset has been used to train the model. FGSM was used for the generation of adversarial attacks. The author showed the results with various performance parameters. The overall results deteriorated after the adversarial attack.

In [20], the authors used four adversarial creation methods: Projected Gradient Descent (PGD) attack, Momentum Iterative Fast Gradient Sign Method, Limited-Memory Broyden-Fletcher-Goldfarb-Shanno method (L-BFGS) [6] attack, and Stochastic Approximation Simultaneous Perturbation (SPSA) [25]. For the experiment, they have used the NSLKDD dataset. For the testing of adversarial attacks, multiple algorithms have been selected like Deep Neural Network (DNN), SVM, RF and LR. The performance measurement parameter includes accuracy, precision rate, recall rate, F1-Score and success rate. All the performance parameters of all the targeted models were decreased after the attack.

The authors in [26] had chosen three methods to generate adversarial attacks: particle swarm optimization (PSO), a genetic algorithm (GA), and a GAN. The adversarial attacks have been tested on two datasets NSLKDD and UNSW-NB15 [27]. Multiple baseline classifiers have been used to test against an adversarial attack which includes SVM, DT, NB, K-NN, RF, MLP, Gradient Boosting (GB), LR, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA) and Bagging (BAG). All the trained classifiers demonstrated a decrease in evasion rate.

In [14], the authors used adversarial training methods to defend against adversarial attacks. They divided the CICIDS17 [28] dataset into four parts for training IDS, testing IDS, training adversarial detector and testing adversarial detector. The adversarial examples are generated using four white-box attacks method which includes FGSM, Basic Iterative method (BIM), C&W and PGD. Performance parameters include Precision, Recall F1-Score and Accuracy. RF & K-NN performed the same in all the parameters with the F1-score of 95%. The AdaBoost algorithm did not exceed the results of the RF with 87.66% accuracy while SVM was unsuccessful in picking up the adversarial attack with recall of 79%.

The authors in [13] demonstrated the adversarial attack and two defence methods. The author simulated realistic attacks by manually modifying three features; exchanged_bytes, duration, total_packets. The author evaluated RF, MLP and K-NN before and after the adversarial attack. The manually crafted features were added in the training set to retrain the models. The performance of all the models deteriorated but after retraining, it showed improvement. The author used feature removal

as the second method to defend against adversarial attacks. When compared to the non-adversarial setting, accuracy drops for each classifier such as RF from 99% to 97%, MLP and K-NN from 99% to 98% which affected the NIDS.

In [10], the authors experimented on NSLKDD and CICIDS17 datasets using only the DOS attack. For feature selection, Recursive Feature Elimination with Linear SVMs technique was used which provided the highest AUC on the original dataset. The selected features were 41 for NSLKDD and 77 for CICIDS17. Four adversarial attack methods were used for each distance metric FGSM, JSMA, Deepfool and C&W. The aim was to misclassify the attack record as a normal record. The author tested adversarial examples on DT, RF, NB, SVM, Neural Network (NN) and Denoising Autoencoder (DA). Evaluation on original datasets for baseline performance shows Decision Tree and RF are among the best while NB and DA underperformed. AUC decreased by 13% on the NSL-KDD dataset while on the CICIDS17 dataset it decreased by 40%. The model was then trained on three adversarial generation methods while one was left for testing purposes. The performance of the classifiers decreased by 4% on the NSLKDD and 18% on the CICIDS17 which was much better than the previous AUC score. In these conditions, RF was the most resilient which only suffered a 0.1% of AUC decrease on both datasets.

The authors in [29] used GAN to generate an adversarial attack. To validate the GAN based attack, black-box IDS have been trained on the baseline machine learning classifier such as DNN, RF, LR, NB, DT, K-NN, SVM, GB. The KDDCUP dataset is used which contains four types of attack classes: Denial-of-Service (DOS), Remote to User (R2L), PROBE and User to Root (U2R). The study targeted the classification of normal and probe class. All the classifiers were affected by adversarial examples generated by adding small perturbations using GAN. To demonstrate defence, the authors used the adversarial training methods in which adversarial examples were added to the training data to ensure the model learns about the possible perturbations. The author evaluated the performance of black box IDS on the accuracy, precision, recall and F1-Score. After GAN based adversarial training, LR performed better among all the classifiers with an accuracy of 86.64%.

In [30], the author used a deep neural network and static features from the DERBIN dataset [31]. The adversarial examples were manually crafted without impacting the malware functionality. The performance parameters were false-negative rates, misclassification rate and average distortion. The author found results with misclassification rates of up to 69% against models. The authors also demonstrated two defence methods: Adversarial training and Defense distillation. The adversarial training defence is non-adaptive and depends on training data. Defence Distillation did not perform well as it usually does in computer vision.

Tab. 1 summarizes the above literature review. As can be seen from the Tab. 1, none of the others have worked on all three i.e., NSLKDD, UNSW-NB15 and CICIDS17 datasets to provide comparison. Also, very few others have worked on defense against adversarial attacks. The contribution of this work can be given as follows:

- Detail comparison of adversarial attack on three benchmark datasets i.e., NSLKDD, UNSW-NB15 and CICIDS17
- For a defence against adversarial attacks, adversarial training is used by including adversarial dataset generated through FGSM and JSMA in training process.
- Various machine learning algorithms have been tested against adversarial attacks in seen (where model is aware of adversarial samples) and unseen (where model is unaware of adversarial samples) attacks.

Table 1: Literature comparison

Paper	Dataset	Classifier	Attack method	Defence	Attack type	Performance parameter	limitation
2017 [4]	NSLKDD	DT, SVM, RF, Voting	FGSM, JSMA	None	White box	Accuracy, F1-Score and AUC normal	Single dataset, results on JSMA only, No defence
2018 [2]	NSLKDD	MLP	JSMA, FGSM, Deepfool, C&W	None	White box	Confusion matrix, Accuracy, F1-Score, False alarm	Single dataset, No defence, One classifier
2018 [32]	NSLKDD	SVM, NB, MLP, LR, DT, RF, K-NN.	GAN	None	Black box	Detection Rate, Evasion increase rate	Single dataset, No defence
2019 [12]	KDDCup 99	CNN	DoS-WGAN	None	Black box	Accuracy	Single dataset, No defence
2019 [11]	NSLKDD	Naïve Bayes, RF, SVM, Proposed	C&W, ZOO, GAN	None	Black box	Accuracy, Precision, Recall, False alarm, F1-score	Single dataset, No defence
2018 [5]	NSLKDD	Neural Network	FGSM	None	White box	Confusion matrix, Accuracy, Precision	Single dataset, No defence, One classifier
2019 [20]	NSLKDD	DNN, SVM, RF, LR	PGD, MI-FSGM, L-BFGS, SPSA	None	White box	Accuracy, Precision Rate, Recall Rate, F1-Score, Success Rate	Single dataset, No defence
2020 [26]	NSLKDD, UNSW-NB15	SVM, DT, NB, K-NN, RF, MLP, GB, LR, LDA, QDA, BAG	PSO, GA, GAN	None	White box	Evasion Rate	No defence
2020 [14]	CICIDS17	ANN, RF, ADABoost, SVM	FGSM, BIM, C&W, PGD	Adversarial training	White box	Accuracy, Precision, Recall, F1-Score	Single dataset, Only Seen attacks, Adversarial labelled record in adversarial training
2019 [13]	Botnet [33]	RF, MLP and K-NN	Manually crafted	Adversarial training, Feature removal	White Box	Accuracy, Precision, Recall, F1-Score	Single dataset
2019 [10]	NSLKDD, CICIDS17	DT, RF, NB, SVM, NN, D A	FGSM, JSMA, Deepfool, C&W	Adversarial training	White Box	AUC	Only AUC was observed. Unknow attack in AT

(Continued)

Table 1: Continued

Paper	Dataset	Classifier	Attack method	Defence	Attack type	Performance parameter	limitation
2019 [29]	KDDCUP 99	DNN, RF, LR, NB, DT, K-NN, SVM, GB	GAN	Adversarial training	Black Box	Accuracy, Precision, Recall and F1-score	Single dataset
2021 [34]	CSE-CIC-IDS2018	Neural Network	MAT (FGSM, BIM, DeepFool, JSMA) and MGAN is GAN based	Adversarial training includes MAT and MGAN	Black Box	Accuracy, Precision, Recall and F1-score	Adversarial examples generated and tested on Neural network No other classifiers tested
2017 [30]	DERBIN [31]	Neural Network	Manually crafted	Adversarial training, Defence distillation	White box	False negative rates, misclassification rate, Average distortion	Used static features. Single dataset and neural network

4 Experimental Setup

In this study, NSLKDD, UNSW-NB15 and CICIDS17 datasets are utilized. For NSLKDD, there are 39 types of attacks and one normal class. All the attacks have been converted into one of the four classes ['dos', 'r2l', 'probe' and 'u2r']. For UNSW-NB15, there are 9 attack types and one normal class. For CICIDS17, there are 14 attack types and one normal class. All the datasets are evaluated as multi-classification problem. The categorical data were One-Hot Encoded as 1 for correct and 0 for all others. StandardScaler is used to resize the distribution of data so that the mean of the observed data is 0 and the standard deviation is 1. StandardScaler standardizes a feature by subtracting the mean and then scaling to unit variance. The unit variance uses the standard deviation as the scaling factor.

The Sklearn library is used for classification and the Cleverhans library [35] is for adversarial attacks generation. For the generation of adversarial examples, we have used MLP as shown in Fig. 3. With the help of MLP, we have created an adversarial test dataset using FGSM and JSMA for this study. Initially, the MLP model is trained on original dataset. The test set of the dataset is applied to MLP model with one of the adversarial attack methods in order to generate adversarial samples. The attack methods try to add small amount of perturbation in adversarial test dataset which can make model to take wrong decision.

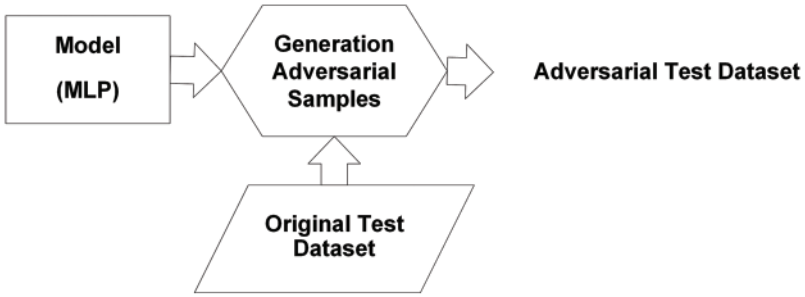

Figure 3: Adversarial test dataset generation

The adversarial examples were tested against multiple machine learning classifiers like Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbors, Logistic Regression and Naïve Bayes.

4.1 Evaluation Parameters

To evaluate the performance of machine learning classifiers, Accuracy, F1 Score and AUC Score are used

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5}$$

AUC is the area under the Receiver Operating Characteristic (ROC) curve which is drawn using the false positive rate (FPR) and true positive rate (TPR) metrics.

4.2 Types of Experiments

The experiments are divided into the following five types:

i. The baseline performance of each classifier
ii. Classifiers tested against JSMA AND FGSM based adversarial attacks (without adversarial training)
iii. Classifiers trained with adversarial samples and tested with the original dataset
iv. Performance of classifiers tested against JSMA AND FGSM after adversarial training with JSMA.
v. Performance of classifiers tested against JSMA AND FGSM after adversarial training with FGSM.

For the evaluation of experiments (i) and (ii), the model in Fig. 4 is implemented. Similarly, for the evaluation of the experiments of (iii) and (iv), the model implemented is shown in Fig. 5. The result of these models are observed in Tab. 2 for NSLKDD, in Tab. 3 for UNSW-NB15 and in Tab. 4 for CICIDS17 datasets.

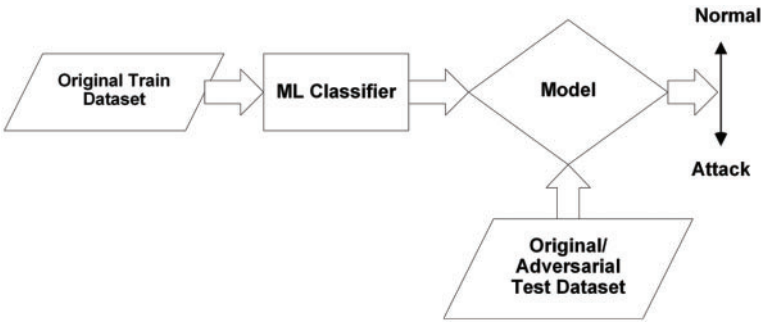

Figure 4: ML classifiers accuracy for original/adversarial test dataset

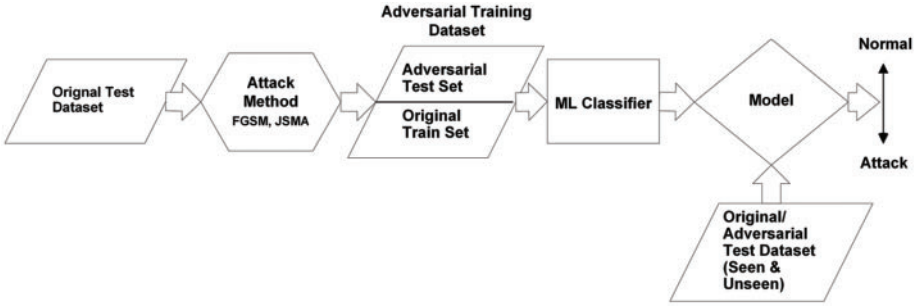

Figure 5: Performance of classifiers after adversarial training

Table 2: Evaluation of NSLKDD. ("Adv." represents adversarial)

NSLKDD		Original dataset (i)	Test on adv. dataset (JSMA) (ii-a)	Adv. training (JSMA) test on original dataset (iii-a)	(JSMA) Adv. training and adv. test dataset (iv-a)	Tested on unseen attack (FGSM) after adv. training (JSMA) (iv-b)	Test on adv. dataset (FGSM) (ii-b)	Adv. training (FGSM) test on original dataset (iii-b)	(FGSM) Adv. training and adv. test dataset (v-a)	Tested on unseen attack (JSMA) after adv. training (FGSM)(v-b)
Accuracy of NSLKDD										
1	DT	**0.811**	0.448	0.790	**1.000**	0.143	0.509	0.826	**0.998**	0.466
2	RF	0.797	0.448	**0.807**	0.999	0.342	0.543	0.804	0.997	**0.450**
3	SVM	0.766	**0.579**	0.697	0.833	0.155	0.196	0.836	0.912	0.536
4	K-NN	0.778	0.578	0.778	0.996	**0.787**	**0.778**	0.861	0.980	0.570
5	LR	0.805	0.315	0.787	0.974	0.145	0.261	**0.883**	0.946	**0.601**
6	NB	**0.466**	**0.466**	**0.466**	**0.466**	**0.466**	**0.466**	**0.466**	**0.466**	0.560
F1 Score of NSLKDD										
1	DT	0.569	0.128	0.567	**1.000**	0.112	0.294	0.653	**0.997**	0.168
2	RF	0.534	0.123	0.532	0.999	0.196	0.313	0.577	**0.997**	**0.126**
3	SVM	0.502	0.240	**0.361**	0.499	0.082	0.198	**0.544**	0.715	0.209
4	K-NN	0.546	**0.242**	0.545	0.958	**0.556**	**0.548**	0.732	0.931	0.242
5	LR	**0.613**	0.096	**0.592**	0.849	0.138	0.323	**0.745**	0.881	0.252
6	NB	**0.315**	**0.037**	0.437	**0.374**	**0.038**	**0.046**	0.614	**0.660**	**0.322**
AUC Score of NSLKDD										
1	DT	0.742	0.503	**0.734**	**1.000**	**1.000**	0.605	**0.783**	**0.783**	0.531
2	RF	0.857	**0.481**	**0.868**	0.999	0.584	0.770	0.893	**0.999**	0.609
3	SVM	0.903	**0.708**	0.839	0.993	0.461	0.580	0.990	0.994	0.538
4	K-NN	0.797	0.563	0.792	0.999	**0.797**	**0.797**	0.866	**0.999**	0.563
5	LR	**0.917**	0.490	0.862	0.995	0.555	0.659	0.992	0.994	**0.504**
6	NB	**0.730**	0.496	0.777	**0.896**	0.499	**0.502**	**0.964**	**0.962**	**0.818**

Table 3: Evaluation of UNSW-NB15. ("Adv." represents adversarial)

UNSW-NB15		Original dataset (i)	Test on adv. dataset (JSMA) (ii-a)	Adv. training (JSMA) test on original dataset (iii-a)	(JSMA) Adv. training and adv. test dataset (iv-a)	Tested on unseen attack (FGSM) after adv. training (JSMA) (iv-b)	Test on adv. dataset (FGSM) (ii-b)	Adv. training (FGSM) test on original dataset (iii-b)	(FGSM) Adv. training and adv. test dataset (v-a)	Tested on unseen attack (JSMA) after adv. training (FGSM)(v-b)
Accuracy of UNSW-NB15										
1	DT	0.730	**0.073**	0.730	**0.935**	0.388	**0.219**	0.739	**0.935**	0.449
2	RF	**0.744**	0.135	**0.744**	0.933	0.336	0.335	**0.741**	0.933	0.449
3	SVM	0.620	0.507	0.645	0.830	0.384	0.277	0.635	0.707	0.526
4	K-NN	0.663	0.083	0.662	0.883	**0.570**	**0.571**	0.665	0.828	0.068
5	LR	0.634	**0.508**	0.636	0.835	0.455	0.380	0.674	0.738	**0.536**
6	NB	**0.269**	0.449	**0.282**	**0.093**	**0.049**	0.449	**0.316**	**0.597**	**0.029**
F1 Score of UNSW-NB15										
1	DT	0.471	**0.013**	**0.476**	**0.765**	0.156	0.108	**0.479**	**0.765**	0.063
2	RF	**0.465**	0.023	0.475	0.753	0.138	0.134	0.451	0.751	0.060
3	SVM	0.290	0.115	0.297	0.390	0.134	0.120	0.298	0.326	**0.124**
4	K-NN	0.376	0.051	0.374	0.494	**0.273**	**0.275**	0.379	0.456	**0.034**
5	LR	0.300	**0.117**	0.299	0.399	0.198	0.171	0.319	0.341	0.119
6	NB	**0.149**	0.062	**0.156**	**0.080**	**0.009**	**0.062**	**0.178**	**0.267**	**0.034**
AUC Score of UNSW-NB15										
1	DT	0.817	0.500	0.825	**0.992**	0.552	0.528	0.809	**0.992**	0.501
2	RF	**0.910**	**0.508**	**0.915**	0.991	0.584	0.545	0.913	0.991	**0.537**
3	SVM	0.895	0.488	0.888	0.925	0.614	0.581	0.914	0.911	0.531
4	K-NN	0.799	0.476	0.801	0.966	0.686	**0.684**	**0.801**	0.962	0.458
5	LR	0.882	**0.467**	0.883	0.934	**0.706**	0.676	**0.920**	0.922	0.515
6	NB	**0.780**	0.500	**0.789**	**0.734**	**0.500**	**0.500**	0.817	**0.855**	**0.330**

Table 4: Evaluation of CICIDS17. ("Adv." represents adversarial)

CICIDS17		Original dataset (i)	Test on adv. dataset (JSMA) (ii-a)	Adv. training (JSMA) test on original dataset (iii-a)	(JSMA) Adv. training and adv. test dataset (iv-a)	Tested on unseen attack (FGSM) after adv. training (JSMA) (iv-b)	Test on adv. dataset (FGSM) (ii-b)	Adv. training (FGSM) test on original dataset (iii-b)	(FGSM) Adv. training and adv. test dataset (v-a)	Tested on unseen attack (JSMA) after adv. training (FGSM)(v-b)
Accuracy of CICIDS17										
1	DT	**0.998**	0.843	**0.998**	**0.858**	0.710	0.584	**0.998**	**0.999**	0.846
2	RF	**0.998**	**0.849**	**0.998**	**0.858**	0.829	0.817	**0.998**	**0.999**	**0.847**
3	SVM	0.803	0.803	0.804	0.803	0.751	0.705	0.803	0.803	0.803
4	K-NN	0.993	0.844	0.994	0.857	**0.842**	**0.843**	0.994	0.993	0.844
5	LR	0.967	0.831	0.955	0.841	**0.438**	**0.500**	0.960	0.975	0.836

(Continued)

Table 4: Continued

CICIDS17		Original dataset (i)	Test on adv. dataset (JSMA) (ii-a)	Adv. training (JSMA) test on original dataset (iii-a)	(JSMA) Adv. training and adv. test dataset (iv-a)	Tested on unseen attack (FGSM) after adv. training (JSMA) (iv-b)	Test on adv. dataset (FGSM) (ii-b)	Adv. training (FGSM) test on original dataset (iii-b)	(FGSM) Adv. training and adv. test dataset (v-a)	Tested on unseen attack (JSMA) after adv. training (FGSM)(v-b)
6	NB	**0.702**	**0.765**	**0.175**	**0.051**	0.802	0.803	**0.385**	**0.713**	**0.135**
F1 Score of CICIDS17										
1	DT	**0.893**	0.247	**0.910**	**0.437**	0.127	0.142	**0.859**	**0.999**	0.284
2	RF	0.834	**0.290**	0.886	0.425	0.103	0.086	0.845	0.973	**0.322**
3	SVM	**0.062**	**0.060**	**0.062**	**0.060**	0.062	**0.055**	**0.059**	**0.061**	**0.059**
4	K-NN	0.759	0.230	0.759	0.327	**0.356**	**0.356**	0.767	0.768	0.230
5	LR	0.356	0.172	0.388	0.193	0.071	0.063	0.420	0.544	0.179
6	NB	0.457	0.149	0.140	0.077	0.059	**0.059**	0.257	0.395	0.105
AUC Score of CICIDS17										
1	DT	**0.949**	0.564	0.949	**0.734**	0.534	0.530	**0.934**	**0.999**	0.575
2	RF	**0.980**	**0.642**	**0.995**	0.733	0.653	0.595	**0.990**	**0.999**	0.617
3	SVM	0.966	0.587	0.969	**0.598**	0.627	0.672	0.989	0.993	0.619
4	K-NN	0.971	0.568	0.971	0.644	**0.694**	0.695	0.975	**0.999**	**0.568**
5	LR	0.952	0.633	**0.839**	0.679	0.661	**0.821**	0.973	0.974	**0.651**
6	NB	0.971	**0.542**	0.942	0.669	**0.499**	**0.499**	0.942	**0.971**	0.645

5 Experiment Results

Tabs. 2–4 summarize the complete results for NSLKDD, UNSW-NB15, CICIDS17 respectively. The top and the lowest performer among classifiers for each category are in bold fonts. To explain the salient points of obtained results, a single row of NSLKDD (results of K-NN) is shown in Fig. 6.

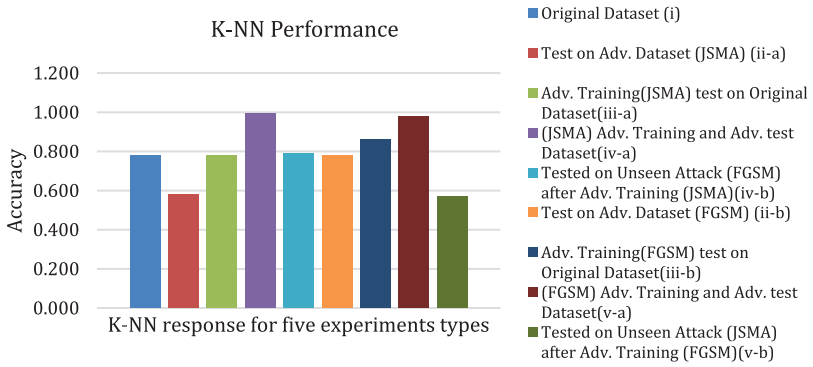

Figure 6: K-NN accuracy of NSLKDD dataset

The baseline performance obtained in (i) drops after the classifiers are tested against JSMA and FGSM based adversarial attacks in (ii-a) and (ii-b) (without adversarial training) on original datasets. The classifiers trained with adversarial samples and tested with the original dataset in (iii-a) and (iii-b) shows a similar trend as observed in (i). Performance of classifiers after adversarial training in (iv-a) and (v-a) shows improved results for the seen attack, whereas in (iv-b) and (v-b) drop of accuracy can be observed even after adversarial training for the unseen attack.

Tabs. 2–4 show the performance of baseline classifiers in column Original Dataset (i) for NSLKDD, UNSW-NB15, CICIDS17 respectively. The results obtained for baseline classifiers for each dataset are similar to those that have been obtained in other literature. These performance parameters are then used to compare with adversarial attack results.

Referring to experiment (ii) the impact of adversarial attack either with JSMA or FGSM without any adversarial training indicate the average drop of accuracy of around 25% to 30% on all the classifiers and datasets. On the other hand, K-NN shows better performance in accuracy, F1-Score and AUC among all the classifiers when tested against FGSM. Similarly, Random Forest performs better for CICIDS17 dataset when tested against JSMA.

The experiment types (iv-a) and (iv-b) is where models are trained on JSMA and tested on the JSMA and FGSM attacks respectively. The results in this type of experiment are better than type (ii) for all the classifiers as these classifiers are trained on the adversarial examples on which they have been tested. The experiment (iv-a) shows better results for the Decision Tree for all the datasets. Similarly, for the experiment (iv-b), K-NN has performed better against FGSM compared to other classifiers trained on JSMA based adversarial examples for all datasets. Whereas in the experiment (v-b), classifiers trained on FGSM and tested against JSMA based adversarial examples, Logistic Regression performs well in accuracy for NSLKDD and UNSW-NB15 datasets. For CICIDS17, Random Forest performs is better in accuracy among all classifiers. Considering accuracy for all the datasets, Naïve Bayes performs worst among all classifiers with the exception of a few results.

Analyzing the AUC Score, for experiment type (iv-a) and (v-a), observed results lies above 90% except for the CICIDS17 dataset in type (iv-a). Another interesting aspect for the experiment type (iv-b) for NSLKDD is, obtained results are not up to the mark for accuracy. A similar pattern is reflected by the F1 and AUC Score which results in a low score. Whereas the results of (v-a) are good for each classifier and are also supported by the F1 and AUC scores. These kinds of patterns validate our work according to the performance parameters included in our studies.

6 Conclusion

We have tested FGSM and JSMA based adversarial examples against multiple classifiers. The experiments have been conducted in five different scenarios. Initially, classifiers have been tested on clean data to compare the results with other experiments. The classifies tested against adversarial examples with or without adversarial training. The behaviour of classifiers on multiple datasets has been observed with performance parameters. Performance of NB is observed to be the worst overall whereas KNN performs better in NSLKDD and UNSW-NB15 datasets. For CICIDS17, Random Forest classifier gives better results.

The future work includes the adversarial training of the classifier with multiple adversarial datasets to increase the robustness of the classifier. Ensemble of classifiers can also be created to increase overall performance against adversarial attacks.

Funding Statement: The authors receive no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[2] Z. Wang, "Deep learning-based intrusion detection with adversaries," *IEEE Access*, vol. 6, no. 1, pp. 38367–38384, 2018.

[3] N. Martins, J. M. Cruz, T. Cruz and P. Henriques Abreu, "Adversarial machine learning applied to intrusion and malware scenarios: A systematic review," *IEEE Access*, vol. 8, no. 1, pp. 35403–35419, 2020.

[4] M. Rigaki and A. Elragal, "Adversarial deep learning against intrusion detection classifiers," *CEUR Workshop Proceedings*, vol. 2057, no. 1, pp. 35–48, 2017.

[5] A. Warzynski and G. Kolaczek, "Intrusion detection systems vulnerability on adversarial examples," in *IEEE (SMC) Int. Conf. on Innovations in Intelligent Systems and Applications (INISTA)*, Thessaloniki, Greece, pp. 1–4, 2018.

[6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan *et al.,* "Intriguing properties of neural networks," in *2nd Int. Conf. on Learning Representations (ICLR)-Conf. Track Proc.*, Canada, pp. 1–10, 2014.

[7] I. J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd Int. Conf. on Learning Representations (ICLR)-Conf. Track Proc.*, San Diego, CA, USA, pp. 1–11, 2015.

[8] J. Hang, K. Han, H. Chen and Y. Li, "Ensemble adversarial black-box attacks against deep learning systems," *Pattern Recognition*, vol. 101, pp. 107184, 2020.

[9] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh *et al.,* "Ensemble adversarial training: Attacks and defenses," in *6th Int. Conf. on Learning Representations (ICLR)-Conf. Track Proc.*, Vancouver, BC, Canada, pp. 1–20, 2018.

[10] N. Martins, J. M. Cruz, T. Cruz and P. H. Abreu, "Analyzing the footprint of classifiers in adversarial denial of service contexts," *EPIA Conference on Artificial Intelligence*, vol. 11805, no. 1, pp. 256–267, 2019.

[11] K. Yang, J. Liu, C. Zhang and Y. Fang, "Adversarial examples against the deep learning based network intrusion detection systems," in *Proc. - IEEE Military Communications Conf. (MILCOM*,), Los Angeles, CA, USA, pp. 559–564, 2019.

[12] Q. Yan, M. Wang, W. Huang, X. Luo and F. R. Yu, "Automatically synthesizing DoS attack traces using generative adversarial networks," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 12, pp. 3387–3396, 2019.

[13] G. Apruzzese, M. Colajanni, L. Ferretti and M. Marchetti, "Addressing adversarial attacks against security systems based on machine learning," in *11th Int. Conf. on Cyber Conflict (CyCon)*, Tallinn, Estonia, pp. 1–18, 2019.

[14] M. Pawlicki, M. Choraś and R. Kozik, "Defending network intrusion detection systems against adversarial evasion attacks," *Future Generation Computer Systems*, vol. 110, no. 1, pp. 148–154, 2020.

[15] S. Abirami and P. Chitra, "Energy-efficient edge based real-time healthcare support system," *Advances in Computers*, vol. 117, no. 1, pp. 339–368, 2020.

[16] N. Papernot, P. Mcdaniel, S. Jha, M. Fredrikson, Z. B. Celik *et al.,* "The limitations of deep learning in adversarial settings," in *Proc. - 2016 IEEE European Symp. on Security and Privacy, (EuroS&P)*, Saarbrücken Germany, pp. 372–387, 2016.

[17] L. Fei-Fei, J. Deng and K. Li, "ImageNet: Constructing a large-scale image database," *Journal of Vision*, vol. 9, no. 8, pp. 1037–1037, 2010.

[18] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[19] A. Krizhevsky, V. Nair and G. Hinton, "CIFAR-10 and CIFAR-100 datasets." https://www.cs.toronto.edu/~kriz/cifar.html, 2009.

[20] Y. Peng, J. Su, X. Shi and B. Zhao, "Evaluating deep learning based network intrusion detection system in adversarial environment," in *Proc. of IEEE 9th Int. Conf. on Electronics Information and Emergency Communication (ICEIEC)*, Beijing, China, pp. 61–66, 2019.

[21] Canadian Institute for Cybersecurity and University of New Brunswick, "NSL-KDD datasets Canadian institute for cybersecurity," https://www.unb.ca/cic/datasets/nsl.html, 2009.

[22] S. M. Moosavi-Dezfooli, A. Fawzi and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 2574–2582, 2016.
[23] N. Carlini and D. Wagner, "Defensive distillation is not robust to adversarial examples," arXiv Preprint arXiv:1607.04311, 2016.
[24] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," *arXiv Preprint arXiv:1803.08375*, 2018.
[25] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332–341, 1992.
[26] E. Alhajjar, P. Maxwell and N. D. Bastian, "Adversarial machine learning in network intrusion detection systems," *Expert Systems with Applications*, vol. 186, pp. 115782, 2021.
[27] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. of IEEE Military Communications and Information Systems Conf. (MilCIS)*, Canberra, ACT, Australia, 2015.
[28] I. Sharafaldin, A. H. Lashkai and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *4th Int. Conf. on Information Systems Security and Privacy (ICISSP)*, Portugal, 2018.
[29] M. Usama, M. Asim, S. Latif, J. Qadir and A. A. Fuqaha, "Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems," in *15th Int. Wireless Communications and Mobile Computing Conf. (IWCMC)*, Tangier, Morocco, pp. 78–83, 2019.
[30] K. Grosse, N. Papernot, P. Manoharan, M. Backes and P. McDaniel, "Adversarial examples for malware detection," in *22nd European Symp. on Research in Computer Security (ESORICS)*, Oslo, Norway, vol. 10493, pp. 62–79, 2017.
[31] D. Arp, M. Spreitzenbarth, M. Huebner, H. Gascon and K. Rieck, "Drebin: Efficient and explainable detection of android malware in your pocket," in *21th Annual Network and Distributed System Security Symp. (NDSS)*, San Diego, California, vol. 14, pp. 23–26, 2014.
[32] Z. Lin, Y. Shi and Z. Xue, "IDSGAN: Generative adversarial networks for attack generation against intrusion detection," arXiv Preprint arXiv:1809.02077, 2018.
[33] S. García, M. Grill, J. Stiborek and A. Zunino, "An empirical comparison of botnet detection methods," *Computers and Security*, vol. 45, no. 1, pp. 100–123, 2014.
[34] J. Wang, J. Pan, I. Alqerm and Y. Liu, "Def-IDS: An ensemble defense mechanism against adversarial attacks for deep learning-based network intrusion detection," in *Proc. - Int. Conf. on Computer Communications and Networks (ICCCN)*, Athens, Greece, 2021.
[35] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman *et al.,* "Technical report on the cleverhans v2.1.0 adversarial examples library," arXiv Preprint arXiv:1610.00768, 2016.