ENSEMBLE ADVERSARIAL TRAINING BASED DEFENSE AGAINST ADVERSARIAL ATTACKS FOR MACHINE LEARNING-BASED INTRUSION DETECTION SYSTEM

M.S. Haroon[*], *H.M. Ali*[*]

Abstract: In this paper, a defence mechanism is proposed against adversarial attacks. The defence is based on an ensemble classifier that is adversarially trained. This is accomplished by generating adversarial attacks from four different attack methods, i.e., Jacobian-based saliency map attack (JSMA), projected gradient descent (PGD), momentum iterative method (MIM), and fast gradient signed method (FGSM). The adversarial examples are used to identify the robust machine-learning algorithms which eventually participate in the ensemble. The adversarial attacks are divided into seen and unseen attacks. To validate our work, the experiments are conducted using NSLKDD, UNSW-NB15 and CICIDS17 datasets. Grid search for the ensemble is used to optimise results. The parameter used for performance evaluations is accuracy, F1 score and AUC score. It is shown that an adversarially trained ensemble classifier produces better results.

Key words: *adversarial attack, adversarial training, ensemble adversarial training, intrusion detection system, machine learning*

Received: February 20, 2023
Revised and accepted: June 15, 2023

DOI: 10.14311/NNW.2023.33.018

1. Introduction

An intrusion detection system (IDS) is an important tool to ensure the security of the network. Traditional intrusion detection systems mainly rely on expert knowledge to build rule sets to detect network attacks. However, the attack method of network attacks is changing rapidly, and traditional rule-based intrusion detection systems can not cope with this [1]. Therefore, in recent years, many researchers have begun to use machine learning (ML) algorithms to build intrusion detection systems [2].

Many machine learning-based intrusion detection systems have been proposed. However, it has been shown that ML algorithms are vulnerable to adversarial

[*] Muhammad Shahzad Haroon – Corresponding author; Husnain Mansoor Ali; Department of Computer Science, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST), Block 5 Clifton, Karachi, Sindh 75600, Pakistan, E-mail: shahzad.haroon@szabist.edu.pk, husnain.mansoor@szabist.edu.pk

attacks [3–5], which means that by adding a small perturbation to the input data, the machine learning algorithm can be fooled and the prediction results can be changed. Thus an attacker can change the prediction result of the IDS by using an adversarial attack, which implies that the IDS can be easily bypassed, greatly affecting the security of the network [6–10].

Generally, adversarial attacks can be divided into two types: white box and block box [11]. In the white box, the attacker has access to the parameters, the architecture of the ML algorithm and training

data. Therefore, the attacker can use the ML algorithm to train a substitute model and then use the substitute model to generate adversarial examples. The adversarial examples generated from the substitute model can also be used to attack the machine learning-based intrusion detection system. In the black box, the attacker does not have access to the parameters, architecture and training data. Therefore, an adversary acts as a normal user and only knows the output of the model.

In this paper, the ML models are tested against adversarial attacks on multiple datasets. To secure the IDS-based ML models ensemble-based adversarial training is proposed. The experiments are conducted in various scenarios and evaluated on performance parameters. The study conducts an experiment which includes the unseen attack methods. To the best of our knowledge, the evaluation of the unseen attack methods are very limited in literature as compared to the seen attacks. The paper is organized as follows: Section 2 explains the background knowledge to generate an adversarial attack. Section 3 discussed the related work of adversarial attack, adversarial training and ensemble-based adversarial training. Section 4 explains the experiments conducted duriabickang the study. In Section 5, results are discussed and in Section 6 we conclude our work.

2. Adversarial generation methods

For adversarial attack generation, different methods are used in the literature. This paper uses four different adversarial attack generation techniques: namely FGSM, JSMA, PGD and MIM which are described in the following section. For the generation of adversarial attacks, the procedure used is described in Fig. 1 [12]. The multi-layer perceptron model is developed for the generation of adversarial attacks. The test dataset is utilized from the original dataset. The adversarial test dataset is created separately for each attack method with the same procedure as mentioned in Fig. 1.

The adversarial dataset thus created using adversarial attack methods is tested in various experiments.

2.1 Multi-layer perceptron

Multi-layer perceptron (MLP) [13] is a deep learning technique which is used for adversarial attack generation. It is a feed-forward neural network with fully connected three layers. The first layer is an input layer that receives the input to be processed. The last layer is the output layer provides the predictions and classification of the received input. The hidden layer or the middle layer is the computation engine

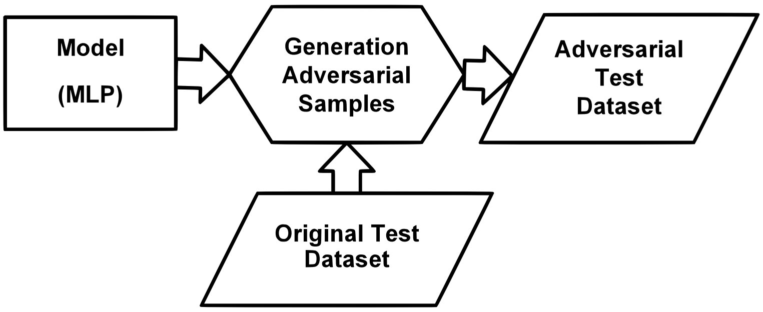

Fig. 1 *Adversarial generation method.*

where all the inputs are processed. MLP is made of neurons called perceptron. In Fig. 2, the structure of a perceptron is given.

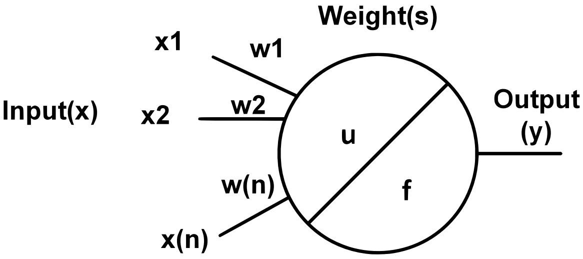

Fig. 2 *Structure of perceptron.*

In the MLP network, every perceptron receives n features $(x_1, x_2, \ldots, x_n)$ as input and each feature is associated with weights $(w_1, w_2, \ldots, w_n)$. The input features are passed on to an input function u, which processes the weighted sum of the input features as given in Eq. (3).

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

The outcome of this computation is then passed onto an activation function f, which will produce the output of the perceptron. For example, a step function can act as an activation function as given in Eq. (4).

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

where θ is the threshold parameter.

2.2 Fast gradient sign attack

Fast gradient sign attack (FGSM) was first proposed in 2014 [14]. The FGSM attack on neural networks is formulated with the help of gradients. The neural network reduces the loss by adjusting weights through the feedback of back-propagated gradients. To attack the neural network, the FGSM attack increases the loss using the same back-propagated gradients. The FGSM-based adversarial attack is formulated as given in Eq. (3):

$$x' = x + \epsilon \cdot \operatorname{sgn}(\nabla_x J(\theta, x, y)), \tag{3}$$

where x is the input to the model, ϵ is the magnitude of the perturbation and $J(\theta, x, y)$ is the gradient of the adversarial loss.

2.3 Jacobian-based saliency map attack

Jacobian-based saliency map attack (JSMA) was proposed in 2016 [15]. The aim was to misclassify by minimizing the modified features involved in a generation of adversarial examples. In this method, a saliency map is created for the input test sample which has the saliency values for each feature. This saliency value suggests how much the classification process is influenced. According to the saliency value each feature is selected in decreasing order. The process continues until the misclassification occurs due to the feature threshold. This process provides adversarial examples similar to the original sample [16].

2.4 Projected gradient descent

The projected gradient descent (PGD) [17] attack is widely regarded as one of the strongest attack methods. It is also competitive with C&W [18] and FGSM [14] attacks. This method adopts the multi-step variant of FGSM, i.e., projected gradient descent (PGD) on the negative loss function [17]

$$x^0_{t+1} = Clip_{x+\eta}(x^0_t + \alpha \cdot \operatorname{sgn}(\nabla_x \mathcal{L}(x^0, \theta, y))), \tag{4}$$

where α is the variant step size at step t, $\mathcal{L}$ is the cost function and the Clip function guarantees that the output falls in the valid input value $(0, \dots, 255)$.

PGD iteratively resumes from many points in the l_∞ balls around data points from the respective evaluation sets. As PGD does not explicitly minimise the l_p-norm of the perturbation, to evaluate vigorously at N distinct thresholds, PGD attack required to be restarted N times which linearly raises the cost of the attack.

2.5 Momentum iterative method

Boosting adversarial attacks with momentum is a technique that increases the effectiveness of these attacks by adding a momentum term to the input perturbations. The momentum term helps the perturbations to continue moving in the direction that causes the model to make a mistake, rather than getting stuck in a local minimum. This makes the attacks more likely to succeed, and can also make them more powerful by allowing them to evade detection by defensive mechanisms. The expression for boosting adversarial attacks with momentum is typically given as follows:

$$x' = x + \alpha \cdot \operatorname{sgn}(\nabla x J(x, y)) + \beta \cdot v, \tag{5}$$

where: x' is the perturbed input, x is the original input, α is the step size or learning rate, $J(x, y)$ is the cost function, $\nabla x J(x, y)$ is the gradient of the cost function with respect to x, β is the momentum term and v is the previous update direction. The $\operatorname{sgn}(\nabla x J(x, y))$ term is used to find the direction of the gradient and move the input in that direction. The $\beta \cdot v$ term is used to add momentum to the perturbations, making them continue moving in the same direction.

3. Related work

The following section details the related work done by others.

3.1 Adversarial attacks without defence

Tab. I summarises the previous work in which adversarial attacks are implemented without any defence method.

In [19], on the NSLKDD dataset [20], the authors evaluated adversarial attacks in a black box scenario. Three distinct black-box attack types were introduced. The adversary trained a substitute C&W [21] model in the first attack. The second attack utilises zero-order optimization (ZOO) while the third attack is generated with the GAN algorithm. Accuracy, precision, recall, false alarm, and F1 score are the parameters used to measure the performance of classifiers. The first attack employing a substitute model has less of an effect than the second and third strategies. The second method outperformed other black-box attacks, but it required a huge number of queries and computational capacity to calculate gradients.

The authors in [8] used a neural network for the development of the intrusion detection system. The model has been trained with the NSLKDD dataset. For the generation of adversarial attacks, FGSM is used. The study showed the results with various performance parameters. The overall results declined after the adversarial attack.

In [22], the authors employed four adversarial generation techniques: the limited-memory Broyden-Fletcher-Goldfarb-Shanno method (L-BFGS) [23] the PGD attack, the stochastic approximation simultaneous perturbation (SPSA) [24] and the momentum iterative fast gradient sign method MI-FGSM. Multiple algorithms, including deep neural network (DNN), support vector machine (SVM), random forest (RF) and LR were chosen for the experiment and trained using the NSLKDD dataset. Accuracy, precision rate, recall rate, F1 score, and success rate are measured following an adversarial attack. After the attack, the performance parameters of all targeted models decreased. The effective adversarial generation method is MI-FGSM in comparison with others.

The research in [25] selected three techniques to develop adversarial attacks: particle swarm optimization (PSO), a genetic algorithm (GA), and a generative adversarial network (GAN). For testing adversarial attacks, the NSLKDD and UNSW-NB15 [26] datasets are utilised. Multiple baseline classifiers, including K-NN (k-nearest neighbor), RF, MLP, SVM, DT, NB, gradient boosting (GB), LR, quadratic discriminant analysis (QDA), linear discriminant analysis (LDA), and bagging (BAG), have been tested against an adversarial attack. Across all trained classifiers, the evasion rate decreased. The author also indicates that the adversarial

Year	Dataset	ML Classifier	Attack Method	Attack Type	P.P	S.C
2019 [19]	NSLKDD	Naïve Bayes(NB), RF, SVM, Proposed	C&W, ZOO, GAN	Black box	Accuracy, Precision, Recall, False alarm, F1 score	Single dataset, No defence
2018 [8]	NSLKDD	Neural Network	FGSM	White box	Confusion matrix, Accuracy, Precision	Single dataset, No defence, One classifier
2019 [22]	NSLKDD	DNN, SVM, RF, LR	PGD, MI-FSGM, L-BFGS, SPSA	White box	Accuracy, Precision Rate, Recall Rate, F1 score, Success Rate	Single dataset, No defence
2020 [25]	NSLKDD, UNSW-NB15	SVM, DT, NB, *K*-NN, RF, MLP, GB, LR, LDA, QDA, BAG	PSO, GA, GAN	White box	Evasion Rate	No defence

Tab. I *Adversarial attacks without defense. ("P.P" represents performance parameter, "S.C" represents shortcoming.)*

sample created to mislead one ML classifier can also mislead others. The results on the UNSW-NB15 dataset using the PSO algorithm for the adversarial generation with the accuracy of 98% against GB, 85% against LDA and 97% against BAG.

3.2 Adversarial attacks with defence

Different authors have suggested various approaches for defence against adversarial attacks. A few of the authors also contributed to adversarial training as defense method against adversarial attacks as summarized in Tab. II.

In the study [10] a defence known as Def-IDS was suggested. The authors created attack cases using two different ways. Multi-source adversarial training (MAT), based on FGSM, BIM, DeepFool [27], and JSMA, is the first technique. The second technique, known as multi-class GAN (MGAN), uses the input as the original data and then uses GAN to produce mimic data. They test against each adversarial attack, such as FGSM, BIM, DeepFool, and JSMA, and then train the classifiers by mixing adversarial instances from both proposed techniques. The effectiveness of DEF-IDS against all attacks is greater than 90%. The work is limited in that only one dataset and neural networks were utilised to train and test the adversarial samples.

In [28], the authors used adversarial training methods to defend IDS against adversarial attacks. In this study, CICIDS17 [29] dataset is divided into four sets for training IDS, testing IDS, training adversarial detector and testing adversarial detector. The adversarial examples are produced using four attacks method which include FGSM, basic iterative method (BIM), C&W and PGD, all belong to the white box category. Performance parameters include precision, recall, F1 score and accuracy. RF & K-NN performed similarly in all the performance parameters with the F1 score of 95%. The AdaBoost algorithm did not surpass the results of the RF with 87.66% accuracy while SVM failed in picking up the adversarial attack with a recall of 79%.

In [9], the authors experimented on the DOS attack records taken from NSLKDD and CICIDS17 datasets. For feature selection, the recursive feature elimination with linear SVMs method was used which provided the highest AUC on the original dataset. The extracted features from the feature selection method provide 41 features for NSLKDD and 77 features for CICIDS17. Four adversarial attack methods were used FGSM, JSMA, Deepfool and C&W for each distance metric. The goal was to misclassify the attack record as a normal record. The authors have chosen multiple ML algorithms to test adversarial examples. Evaluation of original datasets for baseline performance shows DT and RF are among the finest while NB and denoising autoencoder (DA) underperformed. AUC decreased for both datasets by 13% on the NSL-KDD dataset and 40% on the CICIDS17 dataset. The model was then trained on three adversarial generation methods while one was selected for testing purposes. The performance of the classifiers declined by 4% on the NSLKDD and 18% on the CICIDS17. In these conditions, RF was the most robust which only experienced a 0.1% of AUC decrease on both datasets.

In the study [30] the author uses machine learning-based IDS in industrial control systems (ICS) and explored how vulnerable these systems might be to threats from the outside. The main goal of the study is to look into how adversarial learning can be used to attack supervised models and make adversarial samples using the JSMA attack. The goal is to figure out how adversarial samples affect the ability of two famous classifiers, random forest and J48, to do their job. Experiments done on a real dataset from a power system show that adversarial samples caused the models' performance to drop by 6 and 11 percentage points, respectively. But after going through adversarial training, the classifications were more reliable. The results show how important it is to think about hostile attacks and set up defences to improve the security of IDSs in ICS.

The study in [31] looks at how well common machine learning methods work as intrusion detection models to protect against attacks from malicious people. The author highlights the complexity of industrial internet of things (IIoT) systems and the possibility of adversarial attacks on machine learning-based intrusion detection systems (IDSs) in such environments. The paper describes a system called EIFDAA. The main parts of the EIFDAA framework are adversarial training and adversarial examination. Adversarial evaluation is used to find IDSs that do not work well in adversarial settings, while adversarial training is used to make weak IDSs work better. The framework uses five well-known adversarial attack algorithms, such as FGSM, BIM, PGD, DeepFool, and WGAN-GP, to turn attack samples into adversarial samples and mimic the adversarial environment. Experiments with the X-IIoTID dataset show that the lost adversarial detection rate to zero. The improved IDSs that were adversarially trained work well to stop adversarial attacks.

The author in [32] uses the CICIDS17 dataset. The author proposed adversarial training and ensemble learning with adversarial training as a defence method against adversarial attacks. For testing the classifiers, they have to include decision tree (DT), SVM, XGBoost, LR, and DNN. Six attack methods have been included: decision tree attack, JSMA, FGSM, PGD, C&W, and ZOO Attack. All classifiers are affected by adversarial attacks with an average performance loss of 0.45, indicating that adversarial attacks pose a danger to ML-based NIDS. LR is the most effective classifier against adversarial attacks for ML-based NIDSs, with an average F1 score of 0.52. The strongest transferability property is achieved by creating adversarial samples with DNN. Their findings indicate that 84% of adversarial attack methods may be transferred to DNN.

Our proposed defence method is ensemble-based adversarial training. The literature shows in Tab. II and Tab. III that adversarial training has been mostly used by researchers among other techniques. In our study, adversarial training based on an ensemble classifier is tested on the three datasets. We have used three datasets to validate our work as in the literature researchers have used a maximum of two. Our work also tested both the attack types white box as seen attack and black box as unseen attack. The use of ensemble classifiers as the defence method against adversarial attacks on machine learning-based IDS has also not been widely tested.

4. Ensemble adversarial training

Adversarial training is one of the techniques to defend against adversarial attacks for a ML-based intrusion detection system. In this study, we have proposed ensemble-based adversarial training to make our IDS perform better against ad-

Year	Dataset	ML Classifier	Attack Method	Defence	Attack Type	P.P	S.C
2021 [10]	CSE-CICIDS18	Neural Network	MAT (FGSM, BIM, Deep-Fool, JSMA) and MGAN is GAN based	Adv.T.	Grey/ Black Box	Accuracy, Precision, Recall and F1 score	Single dataset
2020 [28]	CICIDS17	ANN, RF, ADABoost, SVM	FGSM, BIM, C&W , PGD	Adv.T.	White box	Accuracy, Precision, Recall, F1 score	Single dataset, Only Seen attacks, Adversarial labelled record in Adv.T.
2019 [9]	NSLKDD, CICIDS17	DT, RF, NB, SVM, NN, DA	FGSM, JSMA, Deepfool, C&W	Adv.T.	White Box	AUC	Only AUC was observed.
2021 [30]	Power system	RF and j48	JSMA	Adv.T.	White Box	F1 score	Single dataset, only seen attack
2023 [31]	X-IIoTID [33]	SVM, DT, RF, K-NN, CNN, GRU, HyDL-IDS	FGSM, BIM, PGD, WGAN-GP	Adv.T.	Black Box	F1 score, EIR	Single dataset,

Tab. II *Adversarial training based defense for adversarial attack ("P.P" represents performance parameter, "S.C" represents shortcoming, "Adv.T." represents adversarial training).*

Year	2022 [32]
Dataset	CICIDS17
ML classifier	DT, SVM, XGBoost, LR, DNN
Attack method	DT Attack, JSMA, FGSM, PGD, C&W, ZOO Attack
Defence	Adv. T. and ensemble learning with Adv. T.
Attack type	White box
P.P	F1 score, AUC score
S.C	Single dataset

Tab. III *Ensemble adversarial training based defence ("P.P" represents performance parameter, "S.C" represents shortcoming, "Adv.T." represents adversarial training).*

versarial attacks. For the ensemble of classifiers, we have selected the following four techniques: DT, RF, K-NN and LR. SVM and NB have been removed from the ensemble as their performance is inconsistent and erratic results are sometimes generated against adversarial attacks [12].

This study uses the NSLKDD, UNSW-NB15 and CICIDS17 datasets. All three datasets are pre-processed which includes one hot encoding which is used to convert categorical data into ones and zeros, and StandardScaler for resizing the distribution of data. The attack classes in the datasets are treated as they are except for the NSLKDD where the 39 classes are converted into four ['dos', 'r2l', 'probe', 'u2r']. The datasets were further divided into train and test and evaluated as a multi-classification problem.

For the experiments, Python 3.7 version is used along with the Sklearn library for classification and the CleverHans library for the generation of adversarial attacks. Different types of experiments are conducted to evaluate each dataset.

4.1 Type of experiments

Tab. IV summarizes each type of experiment performed:

1. The baseline performance of each classifier.
2. Classifiers trained with the originals dataset and tested against adversarial attack FGSM-based in an unseen scenario.
3. Adversarial training with multiple attacks (JSMA, MIM and PGD) against adversarial attack (FGSM) in an unseen attack scenario.
4. Classifier trained with the original dataset for ensemble and tested against original datasets.
5. Classifiers trained with the original dataset for ensemble and tested against adversarial FGSM-based attack.
6. Classifiers trained with adversarial samples of multiple attacks (JSMA, MIM and PGD) for ensemble and tested against the original dataset.
7. Classifiers trained with adversarial samples of multiple attacks (JSMA, MIM and PGD) for ensemble and tested against adversarial attack (FGSM).

S.no.	Model trained on		Model test on		Ensemble
	Original	JSMA, PGD, MIM	Original	FGSM	
i.	Yes	–	Yes	–	–
ii.	Yes	–	–	Yes	–
iii.	–	Yes	–	Yes	–
iv.	Yes	–	Yes	–	Yes
v.	Yes	–	–	Yes	Yes
vi.	–	Yes	Yes	-	Yes
vii.	–	Yes	–	Yes	Yes

Tab. IV *Summarizes each type of experiment performed.*

We have selected the FGSM attack for testing while the models have not been trained on it (unseen attack scenario). (FGSM is used as an unseen attack as in our previous work, we tested JSMA and FGSM extensively in an unseen scenario [12]). In the first two experiments, Fig. 3 is used for the testing of the baseline performance of the classifiers on an original and adversarial dataset based on FGSM. The remaining experiments used Fig. 4. In the third experiment the ML classifiers are trained with the adversarial samples of JSMA, MIM and PGD. The adversarially

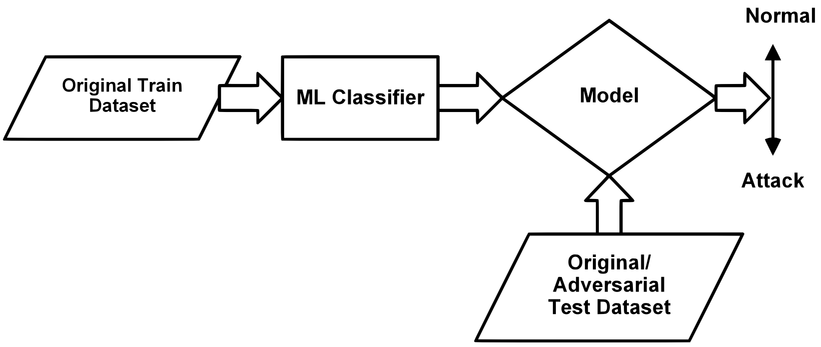

Fig. 3 *Machine learning classifiers accuracy for original/adversarial test dataset.*

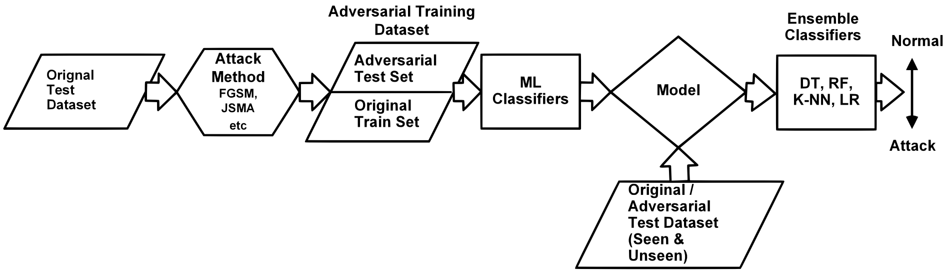

Fig. 4 *Ensemble adversarial training against original/adversarial test dataset.*

trained classifiers tested against FGSM for the unseen adversarial attack. In the fourth experiment, the ensemble of the classifiers is created and tested on the original dataset. In the fifth experiment, the ensemble created in the fourth experiment is tested against FGSM as an unseen attack.

In the sixth experiment, the ensemble of classifiers is trained on the multiple adversarial datasets created using JSMA, MIM and PGD. This is then tested against the original dataset while in the last experiment, the ensemble is tested against FGSM as an unseen adversarial attack. Three performance parameters have been used to compare the results of all the datasets, i.e., accuracy, F1 score and AUC score.

5. Experiment results

Tabs. V–VII summarize the complete results for NSLKDD, UNSW-NB15, and CICIDS17 respectively. Each table consists of the results of all seven experiments (as described in Tab. IV in columns i-vii). Each classifier result is shown in rows whereas the ensemble consists of DT, RF, K-NN and LR models. Accuracy, F1 score and AUC score are calculated for each of the experiments. An analysis of the results is presented below.

5.1 NSLKDD analysis

In the first experiment, the ML classifiers are tested for the baseline performance. All the classifiers perform above 76% except NB which has 46% of accuracy as shown in Tab. V(i). The second experiment is conducted for testing the adversarial samples based on the FGSM attack against ML classifiers trained on original datasets. The performance in terms of accuracy drops for all the classifiers except K-NNh which shows better performance against adversarial attacks which can be observed in Tab. V(ii). The SVM and LR are most affected and sensitive to adversarial attacks as shown by dropped accuracy of 57% and 54% respectively.

In the third type of experiment, classifiers have been trained by including the samples of multiple adversarial attack-based datasets generated through methods like JSMA, MIM and PGD. The adversarially trained classifiers tested against FGSM as an unseen attack. The results of this experiment can be observed in Tab. V(iii). All the classifiers performed well and were above 90% except SVM. When comparing experiments (ii) and (iii), as they both are unseen attacks, the DT improved by 41%, RF by 36%, SVM by 56%, K-NN by 20%, logistic regression (LR) by 65% and NB by 52%. This shows that when a model is trained on the adversarial attack, even if it encounters a new unseen attack, the performance is better when compared to no training at all. The performance of K-NN must be highlighted here that it is improved by 20% in Tab. V(iii) although it performed well as compared to the other classifiers in (i) and (ii). The performance of the classifiers against the unseen attack are considered as black box attack type.

Now considering the F1 score for the third experiment, the performance of all the classifiers has improved but prominent in this case are K-NN and LR with 92% and 85% respectively. Apart from these two classifiers, the remaining ones are still struggling against adversarial attacks. The AUC score in Tab. V(iii) also indicates the classifier health which is above 95% in the majority of cases due to the adversarial training of classifiers.

In the fourth experiment given in Tab. V(iv), the ensemble of classifiers is used to test the original dataset without any adversarial training. The result we have observed in Tab. V(iv) is the baseline performance for the ensemble classifier on the original dataset and is comparable with the results obtained in Tab. V(i).

In the fifth experiment, the ensemble classifiers are now tested against FGSM-based adversarial attacks in an unseen category in Tab. V(v). The experiment result shows that the ensemble classifier accuracy decreased as compared to experiment in Tab. V(iv) as is to be expected. But its performance is better when compared it to its counterpart in Tab. V(ii) where the individual classifier's performance decrease is greater except for K-NN.

In the sixth experiment, the ensemble classifiers are adversarially trained by including the adversarial samples of JSMA, MIM and PGD in Tab. V(vi) and tested against the original dataset. The comparison of this experiment with the results of Tab. V(iv) shows a major improvement with an increase of 13% in accuracy. The improvement in the result is due to the robust ensemble classifier which is adversarially trained. Comparing the result observed in Tab. V(vi) to all the previous result in Tab. V on original dataset, the ensemble based adversarially trained out beat all of them.

The same ensemble classifier is used in the seventh experiment and is now tested against FGSM as an unseen attack. An accuracy of 98.6% is obtained which outperforms both their counterparts in experiments Tab. V(iii) and Tab. V(v).

The F1 score is also reflecting the same as we have observed in the case of accuracy that as our defence becomes stronger the records are correctly identified. In experiments Tab. V(iv) and Tab. V(v), F1 score is low but the results for experiments Tab. V(vi) and Tab. V(vii) are up to the mark. The AUC score tells us the model's capability to identify the between the classes. The AUC score observed for NSLKDD is above 90% in almost every experiment.

5.2 UNSW-NB15 analysis

The results obtained for the UNSW-NB15 dataset are given in Tab. VI. The baseline performance for the classifiers is observed in the experiment Tab. VI(i) where DT and RF performed well with an accuracy of 73% and 74%. The second experiment shows similar trends as observed in the case of the NSLKDD dataset. All the classifiers have deteriorated performance in Tab. VI(ii).

The results in Tab. VI(iii) (classifiers trained with multiple adversarial attacks) are much better when compared with the results in experiment Tab. VI(ii). The prominent classifiers which showed improvement due to adversarial training with multiple attacks are DT, RF and K-NN which improved by 51%, 42% and 41% respectively. A similar trend is observed for the F1 score and the AUC score while the AUC score for the majority of the classifier is above 80%.

In experiment four when the ensemble is tested against the original dataset, it gives 73% accuracy. The same model accuracy is decreased by 20% when tested against FGSM in experiment five. When the ensemble is trained using multiple adversarial attacks and tested against the original dataset it provides 85% accuracy, an increase of 12%, as obtained in experiment Tab. VI(vi).

	NSLKDD	Classifier trained and tested on original dataset (i)	Classifier trained on original and tested on Adv. dataset (FGSM) (ii)	Tested on unseen attack (FGSM) after Adv.T. (multiple attacks) (iii)	En. without Adv.T., testing the original dataset (iv)	En. without Adv.T., testing unseen Adv. dataset (FGSM) (v)	En. with Adv.T. (multiple attacks), testing original test (vi)	En. with Adv.T. (multiple attacks), testing unseen Adv. dataset (FGSM) (vii)
Accuracy of NSLKDD								
1	DT	0.811	0.509	0.914	0.8	0.65	0.93	0.986
2	RF	0.797	0.543	0.9				
3	*K*-NN	0.778	0.778	0.97				
4	LR	0.805	0.261	0.912				
5	SVM	0.766	0.196	0.75				
6	NB	0.466	0.112	0.98				
F1 Score of NSLKDD								
1	DT	0.569	0.294	0.71	0.544	0.45	0.86	0.934
2	RF	0.534	0.313	0.64				
3	*K*-NN	0.546	0.548	0.92				
4	LR	0.613	0.323	0.85				
5	SVM	0.502	0.198	0.45				
6	NB	0.315	0.046	0.669				
AUC Score of NSLKDD								
1	DT	0.742	0.605	0.87	0.94	0.94	0.99	0.99
2	RF	0.857	0.77	0.95				
3	*K*-NN	0.797	0.798	0.99				
4	LR	0.917	0.659	0.984				
5	SVM	0.903	0.58	0.98				
6	NB	0.73	0.502	0.98				

Tab. V *Evaluation of NSLKDD dataset (“Adv.” represents adversarial, “Adv.T.” represents adversarial training, “En.” represents ensemble. SVM and NB are not included in ensemble.).*

The seventh experiment tests the ensemble against FGSM as an unseen attack which gives 78% accuracy. This validates the use of an ensemble as the performance of the ensemble is better than any of the individual classifiers as given in experiment Tab. VI(iii). The best of the individual classifiers is K-NN with 77% accuracy whereas the ensemble accuracy is better by 1%. This same trend is observed in all three datasets.

The results for the F1 score for the UNSW-NB15 dataset are improved to some extent. The result in experiment Tab. VI(vi) is improved by 8% and the result in experiment Tab. VI(vii) is improved by 14% when compared to their counterparts in experiments Tab. VI (iv) and Tab. VI(v) respectively. The AUC score is well above the mark for all the ensemble experiments.

In all the experiments in Tab. VI similar trend is observed as in Tab. V for NSLKDD dataset with different level of impact in different experiments.

5.3 CICIDS17 analysis

Tab. VII gives the results of the experiments conducted with the CICIDS17 dataset. In the first experiment, the baseline performance is obtained which is then compared with the second experiment results that are obtained by using the adversarial attack. As can be seen from second experiment results in Tab. VII(ii), all the classifiers have their accuracy degraded. The K-NN performs well among others with 84% accuracy while the worst performer against an adversarial attack is LR with 50% accuracy. It is worth to be noted here that K-NN performed well for multiple experiment conduct for NSLKDD and UNSW-NB15 datasets.

In the third experiment, all the classifier performances increase in comparison with Tab. VII(ii) due to adversarial training. Both the DT and K-NN perform well with an accuracy of 94% each. The results show a similar pattern as observed in other datasets that the nature of transferability in which the model trained on the different attacks can perform well even with the unseen attacks in adversarial cases.

In the fourth experiment Tab. VII(iv), the ensemble is trained and tested on the original dataset. The same ensemble in the fifth experiment is then tested against FGSM where the performance decreases by 17%. A similar drop in the F1 score can be seen in Tab. VII(v).

In the sixth and seven experiments Tab. VII(vi and vii), the ensemble is trained on multiple adversarial attacks. For the sixth experiment, the ensemble is tested against an original dataset and in the seventh experiment tested against FGSM. The performance of both experiments is up to the mark. The AUC score of the CICIDS17 dataset in comparison with the other two datasets is better.

5.4 Comparison of dataset

For all the ensemble-based experiments, the results obtained for accuracy in the CICIDS17 dataset are better than the other two datasets. Similarly, it is observed that the F1 score is much better and improved as can be seen when comparing the three Tab. VIII(iv with column vi) for the unseen adversarial-based attack. Comparing the overall performance with respect to datasets, it is observed that

NSLKDD		Classifier trained and tested on original dataset (i)	Classifier trained on original and tested on Adv. dataset (FGSM) (ii)	Tested on unseen attack (FGSM) after Adv.T. (multiple attacks) (iii)	En. without Adv.T., testing the original dataset (iv)	En. without Adv.T., testing unseen Adv. dataset (FGSM) (v)	En. with Adv.T. (multiple attacks), testing original test (vi)	En. with Adv.T. (multiple attacks), testing unseen Adv. dataset (FGSM) (vii)
Accuracy of UNSW-NB15								
1	DT	0.73	0.219	0.72	0.73	0.52	0.85	0.78
2	RF	0.744	0.335	0.75				
3	*K*-NN	0.663	0.571	0.77				
4	LR	0.634	0.38	0.68				
5	SVM	0.62	0.277	0.68				
6	NB	0.269	0.449	0.52				
F1 Score of UNSW-NB15								
1	DT	0.471	0.108	0.32	0.45	0.211	0.53	0.35
2	RF	0.465	0.134	0.301				
3	*K*-NN	0.376	0.275	0.34				
4	LR	0.3	0.171	0.28				
5	SVM	0.29	0.12	0.25				
6	NB	0.149	0.062	0.26				
AUC Score of UNSW-NB15								
1	DT	0.817	0.528	0.678	0.94	0.77	0.96	0.93
2	RF	0.91	0.545	0.818				
3	*K*-NN	0.799	0.684	0.818				
4	LR	0.882	0.676	0.862				
5	SVM	0.895	0.581	0.875				
6	NB	0.78	0.5	0.86				

Tab. VI *Evaluation of UNSW-NB15 dataset ("Adv." represents adversarial, "Adv.T." represents adversarial training, "En." represents ensemble. SVM and NB are not included in ensemble.).*

NSLKDD		Classifier trained and tested on original dataset (i)	Classifier trained on original and tested on Adv. dataset (FGSM) (ii)	Tested on unseen attack (FGSM) after Adv.T. (multiple attacks) (iii)	En. without Adv.T., testing the original dataset (iv)	En. without Adv.T., testing unseen Adv. dataset (FGSM) (v)	En. with Adv.T. (multiple attacks), testing original test (vi)	En. with Adv.T. (multiple attacks), testing unseen Adv. dataset (FGSM) (vii)
Accuracy of CICIDS17								
1	DT	0.998	0.584	0.94	0.998	0.827	0.998	0.999
2	RF	0.998	0.817	0.923				
3	*K*-NN	0.993	0.843	0.901				
4	LR	0.967	0.5	0.944				
5	SVM	0.803	0.705	0.803				
6	NB	0.702	0.803	0.582				
F1 Score of CICIDS17								
1	DT	0.893	0.142	0.562	0.836	0.187	0.84	0.993
2	RF	0.834	0.086	0.524				
3	*K*-NN	0.759	0.356	0.521				
4	LR	0.356	0.063	0.456				
5	SVM	0.062	0.055	0.081				
6	NB	0.457	0.059	0.29				
AUC Score of CICIDS17								
1	DT	0.949	0.53	0.756	0.996	0.897	0.993	0.999
2	RF	0.98	0.595	0.88				
3	*K*-NN	0.971	0.695	0.817				
4	LR	0.952	0.821	0.946				
5	SVM	0.966	0.672	0.969				
6	NB	0.971	0.499	0.952				

Tab. VII *Evaluation of CICIDS17 dataset ("Adv." represents adversarial, "Adv.T." represents adversarial training, "En." represents ensemble. SVM and NB are not included in ensemble.).*

the UNSW-NB15 dataset has the worst performance. To improve the result of the UNSW-NB15 dataset, optimization of the ensemble classifier is undertaken. Grid-search method for the optimization is used and results are presented in Tab. VIII. The before and after optimization results reflect the improvement achieved. In the fourth experiment, the accuracy and F1 score are improved by 1% as a result of grid search optimization. In the fifth experiment, the accuracy and F1 score increased by 3%. For the seventh experiment where the ensemble is trained on multiple adversarial attacks and tested against FGSM, there is an improvement of 3%.

UNSW-NB15 experiment	Performance parameters	Before optimization	After optimization
En. without Adv.T., testing the original dataset (iv)	Accuracy F1 Score AUC Score	0.73 0.45 0.94	0.74 0.46 0.94
En. without Adv.T., testing unseen Adv. datasets (FGSM) (v)	Accuracy F1 Score AUC Score	0.52 0.21 0.77	0.55 0.23 0.77
En. with Adv.T. (multiple attacks), testing original test (vi)	Accuracy F1 Score AUC Score	0.85 0.53 0.96	0.85 0.53 0.96
En. with Adv.T. (multiple , attacks) testing unseen Adv. dataset (FGSM) (vii)	Accuracy F1 Score AUC Score	0.78 0.35 0.93	0.81 0.35 0.93

Tab. VIII *Optimization of UNSW-NB15 dataset on ensemble classifiers ("Adv." represents adversarial, "Adv.T." represents adversarial training, "En." represents ensemble).*

6. Conclusion

In this research, the ML models for intrusion detection systems are tested against adversarial attacks in a variety of scenarios. ML classifiers are also trained on adversarial datasets generated by JSMA, MIM, and PGD. The adversarial attack is built with FGSM and tested against ML classifiers in an unknown environment in which the model is uninformed of the attack. Transferability is a concept that describes how adversarial examples developed on one model can be utilised against other models. In this paper, we suggest building an ensemble classifier based on the best-performing ML algorithm against adversarial attacks. The ensemble classifier is then trained on a variety of adversarial datasets and evaluated against adversarial attacks. By integrating adversarial datasets in our training, we can defend the ensemble classifier with transferability. When compared to an untrained ensemble, the outcome of an adversarial-trained ensemble is superior.

References

[1] CHEN J., WU D., ZHAO Y., SHARMA N., BLUMENSTEIN M., YU S. Fooling intrusion detection systems using adversarially autoencoder, *Digital Communications and Networks*, no. September 2019, pp. 1–8, 2021, doi: 10.1016/j.dcan.2020.11.001.

[2] JORDAN M.I., MITCHELL T.M. Machine learning: Trends, perspectives, and prospects, *Science*, 349(6245), pp. 255–260, 2015, [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aaa8415.

[3] HANG J., HAN K., CHEN H., LI Y. Ensemble adversarial black-box attacks against deep learning systems, *Pattern Recognition*, 2020, 101, doi: 10.1016/j.patcog.2019.107184.

[4] GROSSE K., PAPERNOT N., MANOHARAN P., BACKES M., MCDANIEL P. Adversarial examples for malware detection, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10493 LNCS, 2017, pp. 62–79, doi: 10.1007/978-3-319-66399-9_4.

[5] METZEN J.H., GENEWEIN T., FISCHER V., BISCHOFF B. On detecting adversarial perturbations, *5th International Conference on Learning Representations, ICLR 2017 – Conference Track Proceedings*, pp. 1–12, 2017.

[6] WANG Z. Deep Learning-Based Intrusion Detection with Adversaries, *IEEE Access*, 2018, 6, pp. 38367–38384, doi: 10.1109/ACCESS.2018.2854599.

[7] RIGAKI M., ELRAGAL A. Adversarial deep learning against intrusion detection classifiers, *CEUR Workshop Proceedings*, 2017, 2057(0), pp. 35–48.

[8] WARZYNSKI A., KOLACZEK G. Intrusion detection systems vulnerability on adversarial examples, *2018 IEEE (SMC) International Conference on Innovations in Intelligent Systems and Applications, INISTA, 2018*, doi: 10.1109/INISTA.2018.8466271.

[9] MARTINS N., CRUZ J.M., CRUZ T., ABREU P.H. *Analyzing the Footprint of Classifiers in Adversarial Denial of Service Contexts*, 11805 LNAI. 2019, doi: 10.1007/978-3-030-30244-3_22.

[10] WANG J., PAN J., ALQERM I., LIU Y. Def-IDS: An Ensemble Defense Mechanism against Adversarial Attacks for Deep Learning-based Network Intrusion Detection, *Proceedings – International Conference on Computer Communications and Networks, ICCCN*, 2021, July, 2021, doi: 10.1109/ICCCN52240.2021.9522215.

[11] CHEN P.Y., ZHANG H., SHARMA Y., YI J., HSIEH C.J. ZOO: Zeroth order optimization based black-box atacks to deep neural networks without training substitute models, *AISec 2017 – Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, co-located with CCS 2017*, 2017, pp. 15–26, doi: 10.1145/3128572.3140448.

[12] HAROON M.S., ALI H.M. Adversarial Training Against Adversarial Attacks for Machine Learning-Based Intrusion Detection Systems, *Computers, Materials and Continua*, 2022, 73(2), pp. 3513–3527, doi: 10.32604/cmc.2022.029858.

[13] ABIRAMI S., CHITRA P. Energy-efficient edge based real-time healthcare support system, *Advances in Computers*, 2020, 117(1), pp. 339–368, doi: 10.1016/bs.adcom.2019.09.007.

[14] GOODFELLOW I.J., SHLENS J., SZEGEDY C. Explaining and harnessing adversarial examples, *3rd International Conference on Learning Representations, ICLR 2015 – Conference Track Proceedings*, 2015, pp. 1–11.

[15] PAPERNOT N., MCDANIEL P., JHA S., FREDRIKSON M., CELIK Z.B., SWAMI A. The limitations of deep learning in adversarial settings, *Proceedings IEEE European Symposium on Security and Privacy, EURO S and P*, 2016, pp. 372–387, doi: 10.1109/EuroSP.2016.36.

[16] MARTINS N., CRUZ J.M., CRUZ T., HENRIQUES ABREU P. Adversarial Machine Learning Applied to Intrusion and Malware Scenarios: A Systematic Review, *IEEE Access*, 2020, 8, pp. 35403–35419, doi: 10.1109/ACCESS.2020.2974752.

[17] KURAKIN A., GOODFELLOW I.J., BENGIO S. Adversarial machine learning at scale, *5th International Conference on Learning Representations, ICLR 2017 – Conference Track Proceedings*, 2017, pp. 1–17.

[18] CARLINI N., WAGNER D. Towards Evaluating the Robustness of Neural Networks, *Proceedings IEEE Symposium on Security and Privacy*, 2017, pp. 39–57, doi: 10.1109/SP.2017.49.

[19] YANG K., LIU J., ZHANG C., FANG Y. Adversarial Examples Against the Deep Learning Based Network Intrusion Detection Systems, *Proceedings - IEEE Military Communications Conference MILCOM*, 2019, pp. 559–564, doi: 10.1109/MILCOM.2018.8599759.

[20] Canadian Institute for Cybersecurity & University of New Brunswick, NSL-KDD Datasets Research Canadian Institute for Cybersecurity |UNB, 2009, https://www.unb.ca/cic/datasets/nsl.html (accessed Mar. 03, 2022).

[21] CARLINI N., WAGNER D. Defensive Distillation is Not Robust to Adversarial Examples, 2016, 0, pp. 1–3, [Online]. Available: http://arxiv.org/abs/1607.04311.

[22] PENG Y., SU J., SHI X., ZHAO B. Evaluating deep learning based network intrusion detection system in adversarial environment, *ICEIEC 2019 – Proceedings of 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication*, 2019, pp. 61–66, doi: 10.1109/ICEIEC.2019.8784514.

[23] SZEGEDY C., ZAREMBA W., SUTSKEVER I., BRUNA J., ERHAN D., GOODFELLOW I., FERGUS R. Intriguing properties of neural networks, *2nd International Conference on Learning Representations, ICLR 2014 – Conference Track Proceedings*, 2014, pp. 1–10.

[24] SPALL J.C. Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation, *IEEE Transactions on Automatic Control*, 1992, 37(3), pp. 332–341, doi: 10.1109/9.119632.

[25] ALHAJJAR E., MAXWELL P., BASTIAN N. Adversarial machine learning in Network Intrusion Detection Systems, *Expert Systems with Applications*, 2021, 186, pp. 1–25, doi: 10.1016/j.eswa.2021.115782.

[26] MOUSTAFA N., SLAY J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), *Proceedings Military Communications and Information Systems Conference, MilCIS*, 2015, doi: 10.1109/MilCIS.2015.7348942.

[27] MOOSAVI-DEZFOOLI S.M., FAWZI A., FROSSARD P. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2574–2582, doi: 10.1109/CVPR.2016.282.

[28] PAWLICKI M., CHORAŚ M., KOZIK R. Defending network intrusion detection systems against adversarial evasion attacks, *Future Generation Computer Systems*, 2020, 110, pp. 148–154, doi: 10.1016/j.future.2020.04.013.

[29] SHARAFALDIN I., LASHKARI A.H., GHORBANI A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization, *ICISSP 2018 – Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018, Cic, pp. 108–116, doi: 10.5220/0006639801080116.

[30] ANTHI E., WILLIAMS L., RHODE M., BURNAP P., WEDGBURY A. Adversarial attacks on machine learning cybersecurity defences in Industrial Control Systems, *Journal of Information Security and Applications*, 2021, 58, p. 102717, doi: 10.1016/j.jisa.2020.102717.

[31] LI S., WANG J., WANG Y., ZHOU G., ZHAO Y. EIFDAA: Evaluation of an IDS with function-discarding adversarial attacks in the IIoT, *Heliyon*, 2023, 9(2), doi: 10.1016/j.heliyon.2023.e13520.

[32] LIN Y.D., PRATAMA J.H., SUDYANA D., LAI Y.C., HWANG R.H., LIN P.C., LIN H.Y., LEE W.B., CHIANG C.K. ELAT: Ensemble Learning with Adversarial Training in defending against evaded intrusions, *Journal of Information Security and Applications*, 2022, 71, pp. 1–14, doi: 10.1016/j.jisa.2022.103348.

[33] AL-HAWAWREH M., SITNIKOVA E., ABOUTORAB N. X-IIoTID: A Connectivity-Agnostic and Device-Agnostic Intrusion Data Set for Industrial Internet of Things, *IEEE Internet of Things Journal*, 2022, 9(5), pp. 3962–3977, doi: 10.1109/JIOT.2021.3102056.