computers

MDPI

Article

An Adversarial Approach for Intrusion Detection Systems Using Jacobian Saliency Map Attacks (JSMA) Algorithm

Ayyaz Ul Haq Qureshi [1], Hadi Larijani [1,*], Mehdi Yousefi [1], Ahsan Adeel [2] and Nhamoinesu Mtetwa [3]

[1] School of Computing, Engineering and Built Environment, Glasgow Caledonian University, Glasgow G4 0BA, UK; Ayyaz.qureshi@gcu.ac.uk (A.U.H.Q.); Mehdi.Yousefi@gcu.ac.uk (M.Y.)
[2] deepCI.org, Edinburgh, United Kingdom and School of Mathematics and Computer Science, University of Wolverhampton, Wolverhampton WV1 1LY, UK; ahsan.adeel@deepci.org
[3] Barclays Bank Plc., Glasgow G2 7JT, UK; nhamo.mtetwa@barclays.com
* Correspondence: H.Larijani@gcu.ac.uk

Received: 1 June 2020; Accepted: 15 July 2020; Published: 20 July 2020

Abstract: In today's digital world, the information systems are revolutionizing the way we connect. As the people are trying to adopt and integrate intelligent systems into daily lives, the risks around cyberattacks on user-specific information have significantly grown. To ensure safe communication, the Intrusion Detection Systems (IDS) were developed often by using machine learning (ML) algorithms that have the unique ability to detect malware against network security violations. Recently, it was reported that the IDS are prone to carefully crafted perturbations known as adversaries. With the aim to understand the impact of such attacks, in this paper, we have proposed a novel random neural network-based adversarial intrusion detection system (RNN-ADV). The NSL-KDD dataset is utilized for training. For adversarial attack crafting, the Jacobian Saliency Map Attack (JSMA) algorithm is used, which identifies the feature which can cause maximum change to the benign samples with minimum added perturbation. To check the effectiveness of the proposed adversarial scheme, the results are compared with a deep neural network which indicates that RNN-ADV performs better in terms of accuracy, precision, recall, F1 score and training epochs.

Keywords: intrusion detection; adversarial attacks; JSMA; NSL-KDD; network security

1. Introduction

Internet services are responsible for the creation of a digital world of connectivity with the help of sensors and actuators which are widely used for the automation of smart grids, homes, health monitoring equipment, and several other systems [1]. Even though such systems are often considered intelligent and secure, but network proliferation makes them vulnerable to many external threats. In the past few decades, the applications of cognitive computation has been continuously evolving due to advances in self-organized and automated networks connected to the internet [2]. Thanks to the immense research in the area of machine learning (ML), this proposition is turning into a reality. We use ML application in daily lives from smartphones to virtual personal assistants, but despite its success, they are susceptible to privacy and data breach. Network security is now considered as one of the primary issues faced by modern computer networks. A significant extension of intelligent services to the users has raised strong concerns among the research communities to design and develop efficient intrusion detection systems (IDS) [3]. Not only does it enforce the safety of the communication system, but it could also be useful as an alternative to traditional firewalls. IDS monitors traffic in a dynamic manner and raises the alarm when it determines any intrusion in the network. They are categorized as

signature-based or anomaly-based [4]. Signature-based IDS requires a constant update to the known signature data so that IDS could work efficiently. Anomaly-based IDS learns from the existing data and uses patterns to quantify intrusion in the network. The latter approach is more realistic and used widely to design intrusion detection applications.

Machine learning (ML) has been extensively used to develop such anomaly-based intelligent IDS where a designated system learns from the historical network data. It works on the principle that a model is trained with known data points and tested with benign samples. Recent research has found that ML techniques can be fooled by adding careful perturbation to the data known as adversaries [5–8]. Adversarial samples can be defined as a set of inputs that are carefully generated by an adversary after adding a light perturbation to minimize the true classification rate of an ML technique and increase false-positives [2].

Adversarial sample perturbations have the following goals:

1. It can reduce the trust of a classifier by utilizing class ambiguity.
2. It can quantify output which is different from the original class.
3. It may force classifier to generate results which resemble a targeted output class.
4. It may use a targeted output to add noise and misclassify it further as another targeted class.

Different approaches have been proposed in the past to check the effects of adversarial samples on ML platforms. In this research, we aim to propose a Random Neural Networks based Adversarial Intrusion Detection System (RNN-ADV). To craft the adversarial samples, the Jacobian Saliency Map Attack (JSMA) algorithm is adopted. Features are identified and then alter with carefully crafted perturbations. The adversarial samples produced are then used to estimate the performance of RNN-ADV. The NSL-KDD dataset and adversarial data are used to train/test the system. The performance is then compared with Multi-Layer Perceptron based deep neural networks and results are explained in further sections.

The primary contributions of this paper are:

- For adversarial attack detection, a random neural network based intrusion detection system (RNN-ADV) is presented.
- Adversarial samples are crafted by computing forward derivative using Jacobian Saliency Map Attacks (JSMA) algorithm.
- Performance of RNN-ADV is compared with Multi-Layer Perceptron based deep neural networks in terms of accuracy, precision, recall, F1 score and total number of epochs.

The rest of the paper sections are organized as follows: review of the literature reported in this research related to adversarial attacks, intrusion detection and random neural networks (RNN) is discussed in Section 2. The methodology for Adversarial attack crafting using Jacobian Saliency map Attacks (JSMA) algorithm is explained in Section 3. The experimental results and analysis are presented in Section 4 while the conclusion and future research direction are outlined in Section 5.

2. Background

In [9], authors have developed an intrusion detection system by comparing the performance of traditional machine learning techniques with deep learning. Deep neural networks (DNN) are utilized, and excessive features from NSL-KDD are reduced using principal component analysis (PCA). However, the training and testing accuracy of the classifier is relatively low for multiclass detection. Kunal et al., in [10] used State Preserving Extreme Learning Machine(SPELM) algorithm for the application of network security and concluded that it quantifies intrusions on the network with a better accuracy rate than traditional deep belief network (DBN) Algorithm. NSL-KDD dataset was used for training while Chaithanya et al., in [11] proposed Moth-Flame Optimization Algorithm-based Random Forest (MFOA-RF) to achieve this task.

In [12], the authors have developed an intrusion detection system for cybersecurity. A harder approach is utilized using the random forest (RF) machine learning algorithm due to its effectiveness. Unlike previous techniques where samples are classified as benign or malicious, the RF-based IDS uses the ability of flexible training by applying probability labels on the data-points for the classification of adversarial samples. The results show that the proposed scheme has enhanced attack detection many folds for different class labels

Neural networks are capable of performing a recognition and classification task due to their cognitive ability. However, Szedgy et al., in [13], considering the following minimization problem, found out that neural networks are prone to adversarial attacks as they map inputs to the outputs which could be intermittent based on the data used.

$$\arg\min_{\xi} f(a+\xi) = l \text{ s.t. } (a+\xi) \in D \tag{1}$$

This means to adversarial sample $a^* = a + \xi$ is crafted in the input domain D, when benign sample is infected with perturbation/noise ξ which then reduce the classification trust and then makes neural networks to mis-classify. They used Broyden–FletcherGoldfarb–Shanno (LBFGS) optimization algorithm to craft such adversarial samples and concluded that even though adversary affects the performance but it may not have the same effect for any other ML technique.

Several techniques are developed to craft adversarial samples which can be used to dodge the detection capabilities of a system. Goodfellow et al., [14] used the MNIST dataset to discuss the non-linear behaviour of neural networks. Adversarial samples were generated from the Fast Gradient Sign (FGSM) algorithm. Unlike previously proposed Broyden–FletcherGoldfarb–Shanno (LBFGS) optimization algorithm, the FGSM calculates the gradient of the loss of the given activation function for the corresponding input to increase the efficiency. Therefore, to generate adversarial samples I_{adv}, the existing data are altered by adding perturbation which is the combination of input gradient and controlled parameter.

$$I_{adv} = I + \varepsilon * sign(\nabla_I C(I, j)) \tag{2}$$

where, C denominates the cost function, ε is the controlled variation hyper parameter and ∇_I represents the gradient with respect to correct label j and normal sample I.

In [15], Moosavi et al. proposed another technique to craft adversarial samples called DeepFool. Minimum perturbations are generated that could affect the classification labels. To demonstrate the efficiency of the proposed scheme, several ML classifiers are trained with different datasets. Adversarial samples are generated by minimizing Euclidean Distance between the crafted and benign samples. To execute the attack, a linear boundary between different class labels is first identified and perturbation is added in an iterative fashion until the adversary is located. To guarantee better performance, fine-tuning parameters are used. Final perturbation vector α_i^* for affine, binary classifier, is computed as:

$$\arg\min_{\alpha_i} \| \alpha_i \|_2 \text{ s.t. } f(p_i) + \nabla f(p_i)^T \alpha_i = 0, \tag{3}$$

where, current point is p_i the perturbation α_i is calculated in iterative process. Generated perturbations had the similar effect as FGSM instances but it is smaller in size. It can be further upgraded to use for multi-class classifiers.

The sensitivity of the model is highly dependent upon the inputs which are used to train the system. To create misclassification, an adversary can perturb only the selected features or add noise to whole input dimensions. The latter is achieved using the FGSM algorithm used in [14], which follows a gradient-based approach. Papernot et al. focus on the former by addressing the anomalies in the training phase of neural networks. In [6] where they proposed another adversarial sample crafting technique called Jacobian-based Saliency Map Attack (JSMA) algorithm. Only the selected inputs are mapped to output to generate adversarial samples. By computing saliency maps, each feature is assigned some value which are the clear indicators of the minimum amount of change that a

feature is required to affect the classifier and flag false alarms. Since we want to study the impact of misclassification, therefore unlike previous techniques which perturb large dimensions of input samples are more likely to get detected. We are adopting JSMA technique due to unique ability to cause maximum change with minimum added noise. It is also reported to be the optimal choice to cause source-target misclassification [5,16,17].

Although a lot of contributions have been made to adversarial attack detection in the areas of image recognition. There is an open research gap to estimate the performance of intrusion detection systems (IDS) using the adversaries since very few of them, such as [5,7], have completed this task. In this research, we are using the Jacobian-based Saliency Map Attack (JSMA) algorithm to generate adversarial samples. NSL-KDD dataset is used, which is a benchmark for IDS studies. Random Neural Networks based Adversarial (RNN-ADV) Intrusion Detection System is proposed, and the effects of adversaries on existing performance are calculated.

Adversarial Random Neural Network Model

A novel class of artificial neural networks was proposed by Gelenbe named Random Neural Network (RNN) [18]. RNNs have been used extensively for pattern recognition [19]. However, a little research has been reported to analyse the effectiveness of RNNs for intrusion detection systems using NSL-KDD dataset, such as our previous work at [3]. RNN has recently got colossal attention due to its less complexity and better generalisation capabilities. We have tested it for traditional as well as swarm optimisation techniques; it would be interesting to understand the impacts of adversaries.

Figure 1 depicts an RNN model where n_i and n_j are neighbouring neurons which can send and receive information to the rest of the network with the arrival of positive and negative signals either from the neighbours or outside environment. Since RNN is also inspired by biological settings, at a given time t each neuron represents a state $K_i(t)$; a non-negative integer. Upon reception of inhibition (negative) signal, the state remains idle, and no update takes place. Whereas, excitatory (positive) signal at n_i, alters the states to $K_i(t) > 0$, and therefore fires/transmits an impulse signal towards n_j with rate r_i. As a result of which weight are biases are calculated across the network. This fired spike may trigger the response with the following probabilities:

- It can reach neuron n_j with probability of $p^+(i,j)$ as an excitation signal.
- It can reach neuron n_j with probability of $p^-(i,j)$ as an inhibitory signal.
- It can depart the neural network with probability of $c(i)$.

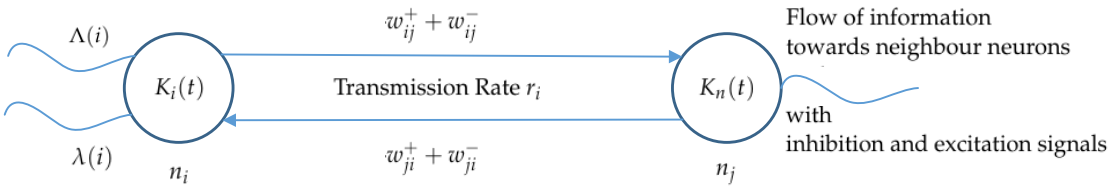

Figure 1. Random neural network model for adversarial analytics.

Mathematically,

$$c(i) + \sum_{j=1}^{N} [p^+(i,j) + p^-(i,j)] = 1, \forall i, \tag{4}$$

$$w^+(i,j) = r_i p^+(i,j) \geq 0, \tag{5}$$

similarly

$$w^-(i,j) = r_i p^-(i,j) \geq 0. \tag{6}$$

Combining Equations (4), (5) and (6)

$$r(i) = (1 - c(i))^{-}1 \sum_{j=1}^{N} [w^{+}(i,j) + w^{-}(i,j)] \tag{7}$$

where,

- $r(i)$ is the firing transmission rate by which information flows towards other neighbouring neurons.
- w^{+} and w^{-} denominates the weight updates for neuron i and j.

The output activation function for N neurons connected and transferring information upon reception of spiking signals at Poison rate $\Lambda(i)$ for positive and $\lambda(i)$ for negative signals, can be written as:

$$q(i) = \frac{\lambda^{+}(i)}{r(i) + \lambda^{-}(i)}, \tag{8}$$

where

$$\lambda^{+}(i) = \sum_{j=1}^{n} q(j) r(j) p^{+}(j,i) + \Lambda(i), \tag{9}$$

and

$$\lambda^{-}(i) = \sum_{j=1}^{n} q(j) r(j) p^{-}(j,i) + \lambda(i). \tag{10}$$

Interested reader can further understand the network operation in [18] and our previously published work at [3,20,21].

3. Methodology

Jacobian Saliency Map Attacks (JSMA) algorithm is utilized to generate adversarial attacks from benign samples. There are a few necessary steps required to prepare data before they are fed to the classifier. RNN-ADV consists of one input, two hidden and one output layers that work in a feed-forward way. The number of hidden layers can be increased according to the optimization problems, to learn complex representations. Since there is no feature selection involved, the data are first normalized. RNN-ADV is then trained with traditional gradient descent algorithm, and performance is estimated based on different metrics.

3.1. Jacobian Saliency Map Attacks

Adversarial samples are intended to enhance the false alarms of the system due to their novel ability to increase misclassification. In neural networks, gradients are used to adjust the weighs during training which helps to generate the desired output [7]. However, when it comes to adversarial sample crafting, most of the techniques used previously utilize gradients to amend original inputs which would then be detected by the particular system.

In [6], Papernot et al. proposed a new algorithm towards adversarial attack crafting using the Jacobian Matrix known as Jacobian Saliency Map Attacks (JSMA). The opposite approach is adopted where a direct mapping is established from the input vector to the desired output, which then generates adversaries. For an RNN model, consider activation function $F : A \mapsto B$ where A original input features extracted from the dataset used and B represents the corresponding output.

In order to generate the adversary A^* from original data, consider the following mathematical optimization problem:

$$\arg\max_{\sigma_a} \parallel \sigma_a \parallel \text{ s.t. } F(A+\sigma_a) = B^*, \tag{11}$$

where,

- σ_a is the perturbation vector;
- $\parallel . \parallel$ is the relevant norm for RNN input comparison;
- B^* is the required adversarial output data points/features;
- $A + \sigma_a = A^*$ is the adversarial sample.

It is to be noted that for the original dataset, $F(A)$ is considered to be $F(A) = B$ and we want to generate adversaries A^* in such a way that it should be almost similar to original sample A but RNN misclassify it as $F(A^*) = B^* \neq B$. Since σ_a is the perturbation which would create new adversarial sample for output class B^*, forward derivate approach is adopted. It is a Jacobian Matrix of given function which is learned during the training phase of neural network [6]. For a one-dimensional vector space, it can be defined as:

$$\nabla F(A) = [\frac{\delta F(A)}{\delta d_1}, \frac{\delta F(A)}{\delta d_2}] \tag{12}$$

where, d_1 and d_2 are non-negative integers and taking forward derivate not only reduced the adversarial data search space but also demonstrates the amount of change happened in original features to generate adversaries. The whole process is depicted step by step in Figure 2. Considering Equation (12), the generation of adversarial samples $F(A^*)$, modified by τ with maximum distortion ζ is explained in Algorithm 1 using the following pseudo-code.

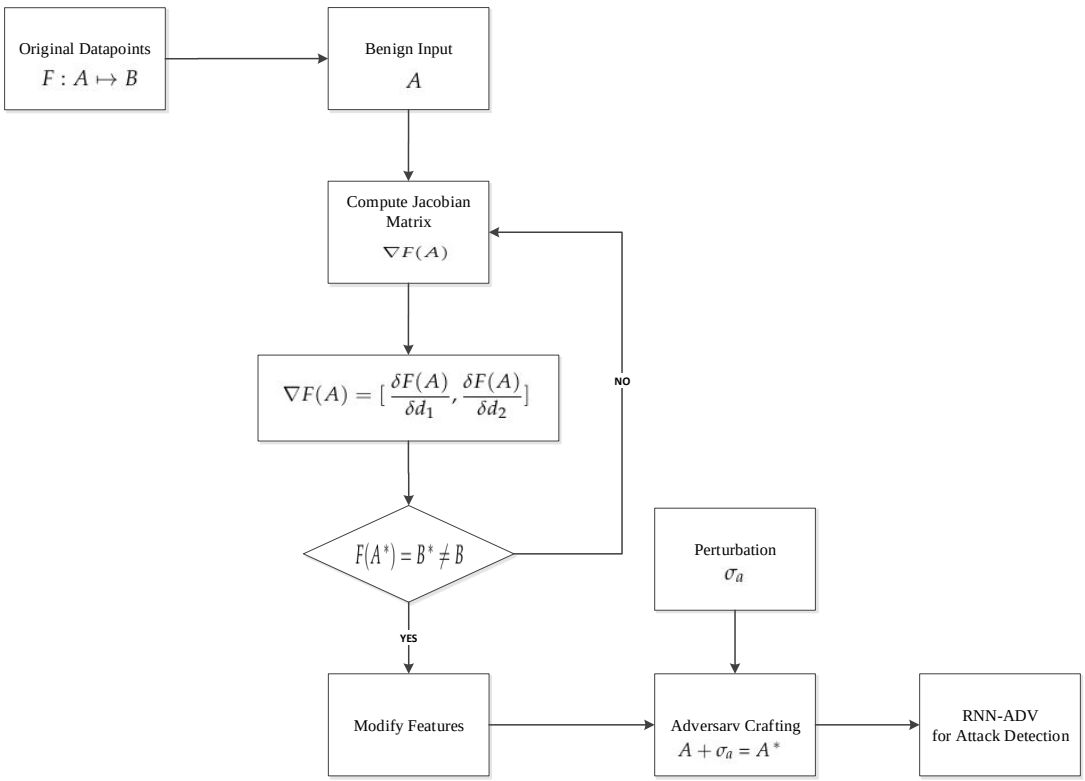

Figure 2. Adversarial attack crafting using Jacobian saliency map attacks (JSMA) algorithm.

Algorithm 1 Adversarial sample generation.

Input: $A, B^*, F, \zeta, \tau, \alpha$

Output: A^*

Initialization:

$A^* \leftarrow A$

for $F(A^*) \neq B^*$ and $\| \sigma_a \| < \zeta$ **do**

Initiate Saliency Map

Compute $\nabla F(A^*)$

Alter the the inputs by τ

$\| \sigma_a \| \leftarrow (A^*) - (A)$

end for

return A^*

3.2. Distance Metrics

For a given Random Neural Network (RNN) model, the original sample s is used to generate adversarial samples $s^{'}$. Precisely, for an adversary $s^{'}$ which belong to the output class α_s, having threshold of ϵ and distance metric M, the output classification function F^s is written as:

$$F(s) = \alpha_s \wedge F(s') \neq \alpha_s \wedge M(s, s') \leq \epsilon, \tag{13}$$

where,
$M(s, s')$ ensures the resemblance between original sample and adversarial sample
$F(s') \neq \alpha_s$ guarantees that the adversary is incorrectly classified by the model
$F(s) = \alpha_s$ confirms the correct categorization of the normal samples

In order to verify the similarity between adversary and the original sample, distance metric (M) is utilized by L_p norms vectorization. It is represented as:

$$\| s - s' \|_p = (\sum_{j=1}^{m} | s_j - s'_j |^p)^{\frac{1}{p}} \tag{14}$$

Following distance metrics are used to quantify adversarial perturbation.

- L_0 metric is used to identify the features that have been perturbed between the original sample s and adversarial sample s'
- L_2 is also reffered as Euclidean norm since it measures the euclidean distance between original sample s and adversarial sample s'.
- L_∞ is used to measure the maximal change in the features during adversarial attack crafting. Mathematically it is written as:

$$\| s - s' \|_\infty = max(| s_j - s'_j | ..., | s_n - s'_n |) \tag{15}$$

The Dataset and Pre-Processing

Several intrusion detection datasets are reported to train and test the systems [22] such as UNSW-NB15 which contains 49-dimensional features obtained by configuring three virtual servers. Although it includes several new types of attacks, it is not widely used due to class imbalance. CICIDS2017 is another emulated dataset which contains modern-day records such as DoS, DDoS spread

along 79 features. International Knowledge Discovery and Data Mining Tools gathered traffic records and formulated a benchmark dataset known as KDD'99. For a long time, it was considered to be the benchmark to design predictive models for detecting the intrusions in a network, but it had several limitations, such as:

- Vast feature space which consists of many redundant and obsolete records.
- Ambiguous network attack class definitions.
- No cross validation employed to consider the possibility of dropped packets during data collection phase.

Travalle et al. [23], performed statistical evaluations to highlight the limitations stated above, and concluded that due to bias in previous datasets, they are no longer feasible to train attack detection mechanisms. To reduce the complexity of the system, they addressed the inadequacies and proposed a new benchmark for IDS known as NSL-KDD [24]. We have used this dataset in our research. From the total of 42 features, first 41 feature is used as inputs which contain the traces from collected network traffic such as *Dstbytes*, *Protocol*, *ServerrRate*, etc. The 42nd feature is the output label which contains information about the targeted class. The attack in the output label varies from Probe, Root-to-local (R2L), Denial-of-Service and User-to-root (U2R). The dataset is widely used due to its packet distribution among normal and abnormal packets in training and testing sets, where it contains 46.5% attack patterns.

The dataset also lists three features that are nominal in nature, and we know the fact that the ML algorithm can only process the information is binary or numeric in nature. For this purpose, we have utilized "One-hot Encoding" where these features are converted to the designated numeric values. Many of the researchers [3,20] utilized feature selection by using the gain ratio (GR), correlation-based feature extraction (CFS) and information gain (IG), which helps reduce the complexity of classifier. For this study, we use the complete feature space of the dataset and performance is estimated.

To truncate the training time of proposed RNN-ADV, data are normalized and then used as corresponding input. Min-Max normalization is exploited, and data are processed accordingly. Mathematically,

$$n_i = \frac{g_i - \min(g)}{\max(g) - \min(g)}, \tag{16}$$

where:

- $g = (g_1, \ldots, g_n)$ is required output
- $n_(i)$ is input'

4. Experimental Results and Analysis

In this section, we have explained the experimental results and presented an analysis of the effects of adversarial attacks on random neural networks. Several performance matrices such as Precision, Recall, F1 Score, False Detection Rate and Accuracy are used to elaborate on the results where are denoted as τ, ψ, υ, and ω the True Positives (TP), True Negatives (TN), False Positives (FP) and False Negative (FN) respectively.

4.1. Accuracy

Accuracy measures the correctness by which proposed RNN-ADV detects the intrusions in the network.

Mathematically,

$$Accuracy\,(RNN - ADV) = \frac{\tau + \psi}{\upsilon + \psi + \omega + \tau} \tag{17}$$

4.2. Precision

Precision measures the relevance of proposed RNN-ADV by considering true positives and false-positive instances in a given data.

$$Precision = \frac{\tau}{\upsilon + \tau} \tag{18}$$

4.3. Recall

Recall measures the ratio of actual positives by considering true positive and false negatives instances in a give data.

$$Recall = \frac{\tau}{\omega + \tau} \tag{19}$$

4.4. F1-Score

Since precision and recall, both are proportionate. In order to use the optimal value, we have used the F1 score which is the harmonic mean of both values.

$$F1\ Score = 2 \times (\frac{Precison \times Recall}{Precision + Recall}) \tag{20}$$

In this research, we use random neural networks as a classifier which would quantify the intrusions from normal packets after training it with required data. Many of the machine learning models have been reported to be vulnerable to the adversarial threats where an attacker can trick the trained model into misclassifying hence increasing false alarms. We are focusing on the bit to understand the RNN performance under adversarial threats. The network consists of two hidden layers and trained at the learning rate of 0.001. Cleverhans python library [25] is also utilised.

Initially, we would train RNN-ADV for the required iterations so that weights are updated in the network accordingly. Then we would test the performance of the proposed system using original benign samples as well as adversarial samples. Adversarial crafting is executed using Jacobian Saliency Map Attacks Algorithm (JSMA) through which perturbations are added and returns the features which affect the overall performance with enduring the minimum change. Following scenarios are designed to understand the adversarial attack effects on RNN-ADV:

4.5. Scenario I

To establish a baseline performance for RNN-ADV, it is trained with original NSL-KDD $Train^+$ and tested against $Test^+$. This is essential as it could serve as a clear indicator of how much performance is deteriorated when an adversary is introduced to the existing system.

4.6. Scenario II

To check the overall behaviour of the classifier, the system is trained and tested with JSMA generated adversarial features using the settings listed in Table 1. This would enable us to understand the network over-fitting when an adversary is introduced to a system in black-box settings.

Table 1. Adversarial attack crafting settings.

Algorithm	Distance Metric	Perturbation Metric
Jacobian Saliency Map Attacks Algorithm (JSMA)	L_0	0.5

4.7. Scenario III

To conclude how much performance is effected by RNN-ADV when tested with unknown data-points, in this scenario, the system is trained with benchmark NSL-KDD $Train^+$ but tested agaisnt adversarial sample as generated by the JSMA attacks algorithm. It also serves as an attack scenario.

As mentioned before, to check the baseline performance of the system, the proposed random neural network-based adversarial intrusion detection system (RNN-ADV) has been trained with the benchmark NSL-KDD dataset. For that purpose, it has been trained with $Train^+$ with the help of a traditional gradient descent algorithm. At this stage, no adversary has been introduced to the system. For the multi-class attack category of the data used, the experimental results are reported in Table 2. According to the outcome, RNN-ADV has successfully classified normal patterns form anomalies with an accuracy of 82.14%.

Table 2. Scenario-I: Performance of RNN-ADV based on $Train^+$ and $Test^+$.

Attack Label	Accuracy	Precision	Recall	F1-Score
Normal	82.14	99.42	91.82	95.46
Denial-of-Service (DoS)	92.62	99.12	94.24	96.61
Probe	91.51	97.21	75.25	85.55
User-to-Root (U2R)	44.82	92.74	48.77	63.92
Root-to-Local (R2L)	61.35	95.24	39.24	55.58

It is worth mentioning that we have used the performance matrices of precision and recall since we are taking a harmonic mean of the values and presenting them in the form of an F1-score. If we look at the attack detection, RNN-ADV has successfully classified DoS attacks by 92.62%, a probe by 91.51%, U2R by 44.85% and R2L by 61.35% with a minimum precision level of 95.24%. The false detection rate is also very low. F1-score for normal and attack classes is high which indicates the high prediction in RNN-ADV, for the case of no adversary.

After analyzing the performance of RNN-ADV with the benchmark dataset, in scenario-II, we would check its effectiveness in adversarial settings, we use the Jacobian Saliency map Attack Algorithm (JSMA). Adversarial samples are crafted by the procedure mentioned in Algorithm 1 [5] and Figure 2. Adversaries usually affect the performance of a classifier either by assuming that it has all the knowledge required to carry on an attack on the system, e.g., number of layers, weights, etc., it only limited knowledge about system-generated output. Latter settings are known as black-box where carrying on the attack is difficult and requires more sophistication than earlier mentioned white-box settings. In this research, we assume that the adversary has information about our system and the JSMA algorithm would generate adversarial samples that we then use for attack analysis.

In the attack generation phase, the JSMA algorithm would find the feature which can influence the performance of a classifier. From full feature space total of 9 features were altered for Normal class and four attack classes, namely DoS, Probe, R2L, and U2R. The same number of features are reported in [5], where authors have also used L_0 norm with the help of the attack algorithm. The results reported in Table 3, suggest that for an adversarial only scenario, the performance of RNN-ADV has been deteriorated where accuracy for normal attacks is decreased by 20%. On the other hand, the false alarms have increased, which is 42.31% for normal packets and 30.47% for denial-of-service attacks. This consolidates the claims made in previous research findings that neural networks are vulnerable to the adversarial setting. F1-score, which represents the confidence in classification, is also reduced to 74.58%, which is almost 40% less than benchmark values discussed in Table 2.

In this research, till now, we have analysed the performance of RNN-ADV with benchmark dataset and with adversarial samples as reported in Tables 2 and 3. For scenario-III, consider an attack on proposed RNN-ADV, trained with benchmark $Train^+$ data which can quantify the normal patterns from attack traffic in general settings. However, it would be interesting to notice the performance

of RNN-ADV when it has no prior information about the test set and tested with adversarial data. Results are presented in Table 4, where performance is estimated for such an attack, and it can be concluded that the performance of RNN-ADV is further deteriorating where the accuracy for the detection of denial-of-service attacks has reached 67.97%. Precision and recall values are also declining with the increase in misclassification, which results in a higher false detection rate and low F1 score. It is interesting to note the accuracy for normal patterns in increase than adversarial only scenario.

Table 3. Scenario-II: Performance of RNN-ADV based on training with adversarial data and tested with adversarial data.

Attack Label	Accuracy	Precision	Recall	F1-Score
Normal	63.41	53.87	42.61	47.58
Denial-of-Service (DoS)	71.38	47.23	31.22	37.59
Probe	79.49	32.41	30.54	31.44
User-to-Root (U2R)	39.25	3.94	5.67	4.64
Root-to-Local (R2L)	52.91	27.81	28.48	28.10

Table 4. Scenario-III: Performance of RNN-ADV based on benchmark $Train^{+}$ and tested with adversarial data.

Attack Label	Accuracy	Precision	Recall	F1-Score
Normal	60.58	40.70	29.21	34.01
Denial-of-Service (DoS)	67.97	33.18	18.99	24.15
Probe	71.41	21.01	16.43	18.43
User-to-Root (U2R)	32.11	1.28	2.47	1.68
Root-to-Local (R2L)	48.17	11.62	17.84	17.84

To verify the results with other ML platforms, the proposed RNN-ADV is compared with MLP based deep neural network [5] for JSMA attacks. F1-score metric is utilized for such estimating the effectiveness of both schemes in Figure 3. The performance of proposed RNN-ADV is almost 11% or more than the MLP IDS where it has an F1 score of 47.58% of a normal class, and for attack class, it ranges from 37.59% to 28.10% for DoS, Probe, U2R and R2L attacks. The epochs to carry out our simulation were 80, 92, and 98, respectively. This means even if the false detection of a system is high, it is still quantifying adversaries with from normal packets with better accuracy and high confidentiality.

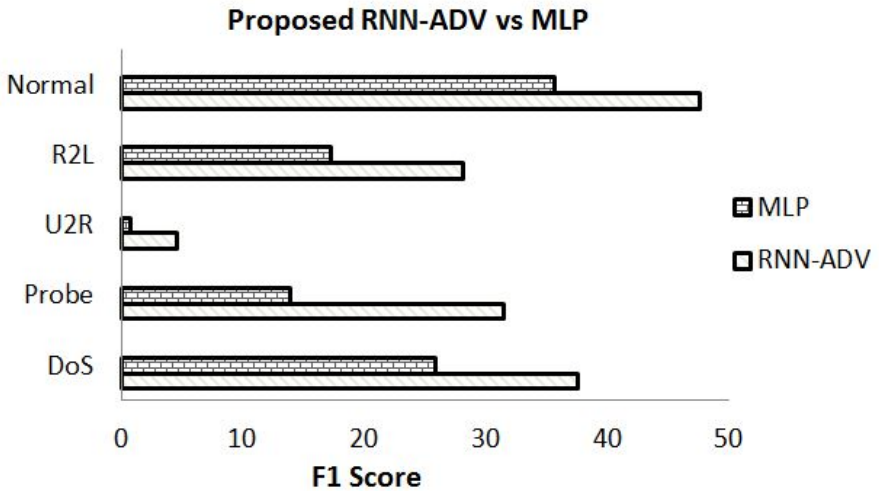

Figure 3. Performance comparison between adversarial RNN-ADV and MLP.

As discussed previously, the adversary reduces the trust of classier hence causing ambiguity as a result of which, the false detection rate increases and accuracy of prediction decreases. Figure 4,

represents the output prediction confidence of proposed RNN-ADV for all three scenarios used in this research. Since we are dealing with the class unbalance where a system with a high F1 score is preferred over the other due to its ability to denominate the RNN-ADV's precision and robustness. For attack categories, in all scenarios, if we consider the denial-of-service as the most vicious attack, then it can be seen that its detection rate is decreasing with the decrease in F1 score and a similar thing is happening for other attack patterns. U2R and R2L have the lowest detection values due to low pattern occurrence in the feature space of the dataset. It can be concluded that Scenario-III where RNN-ADV is trained with benign samples but tested with adversarial data, performed the worst and it is proof that adversaries affect the performance of the neural network in different ways depending upon the training/testing criteria.

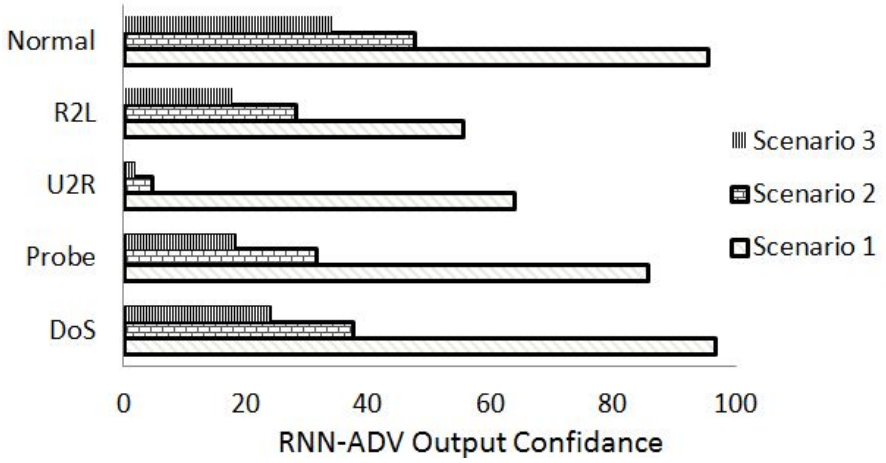

Figure 4. RNN-ADV output confidence based on harmonic mean of precision and recall values.

5. Conclusions

In this paper, we proposed RNN-ADV: A random neural network-based adversarial intrusion detection system. This novel scheme uses the Jacobian Saliency map Attacks (JSMA) algorithm for the generation of adversarial attacks. Different scenarios were adopted where RNN-ADV is trained and tested with benchmark NSL-KDD data and Adversarial only data. An attack scenario is also conceived for more real-time execution of IDS operation where RNN-ADV is trained with benchmark data and tested with adversarial data. Results are further compared with MLP based deep neural networks, and they suggest that like other ML algorithms, the proposed scheme is also prone to adversarial attacks.

It is worth mentioning that for the benchmark scenario, F1-score, which represents the confidence in classification, was 95.6% for normal and 96.61% for denial-of-service attacks while the precision value remained around 99.12%. However, when Adversarial only data were used, it falls to 47.58% for normal and minimum of 28.10% for other attack classes. This happened due to the class imbalance in the benchmark data and can be improved by using better feature extraction techniques. JSMA algorithm is more feasible to craft adversarial samples since it changes only a few features and adds perturbations which makes it a perfect choice for use in limited resources. The results generated indicate that proposed RNN-ADV performs better in terms of accuracy, recall, precision value and F1 score.

In the future, we will extend our work in the following directions:

- To execute transferability analysis and understanding the effects of adversarial attacks in a deep network where more number of hidden layers are used and trained with different learning rates.
- To craft adversarial samples using other techniques such as Fast Gradient Sign Method (FGSM), DeepFool and CW attack algorithms.
- To understand the reliability of RNN-ADV with different datasets such as CICIDS2017 and UNSW-NB15 etc.

- To optimize and fine-tune the network by choosing different training algorithms such as Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO).

Author Contributions: Conceptualization, H.L.; Data curation, N.M.; Formal analysis, M.Y.; Investigation, A.U.H.Q.; Project administration, H.L.; Supervision, H.L.; Writing—original draft, A.U.H.Q.; Writing—review & editing, H.L., M.Y., A.A. and N.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ferdowsi, A.; Saad, W. Generative Adversarial Networks for Distributed Intrusion Detection in the Internet of Things. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019.
2. Usama, M.; Qadir, J.; Al-Fuqaha, A.; Hamdi, M. The Adversarial Machine Learning Conundrum: Can The Insecurity of ML Become The Achilles' Heel of Cognitive Networks? *IEEE Netw.* **2019**, *34*, 196–203. [CrossRef]
3. Qureshi, A.U.H.; Larijani, H.; Mtetwa, N.; Javed, A.; Ahmad, J. RNN-ABC: A New Swarm Optimization Based Technique for Anomaly Detection. *Computers* **2019**, *8*, 59. [CrossRef]
4. Qureshi, A.; Larijani, H.; Javed, A.; Mtetwa, N.; Ahmad, J. Intrusion Detection Using Swarm Intelligence. In Proceedings of the 2019 UK/ China Emerging Technologies (UCET), Glasgow, UK, 21–22 August 2019; pp. 1–5. [CrossRef]
5. Wang, Z. Deep Learning-Based Intrusion Detection With Adversaries. *IEEE Access* **2018**, *6*, 38367–38384. [CrossRef]
6. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The Limitations of Deep Learning in Adversarial Settings. In Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS P), Saarbrücken, Germany, 21–24 March 2016; pp. 372–387. [CrossRef]
7. Martins, N.; Cruz, J.M.; Cruz, T.; Abreu, P.H. Analyzing the Footprint of Classifiers in Adversarial Denial of Service Contexts. In *Progress in Artificial Intelligence*; Moura Oliveira, P., Novais, P., Reis, L.P., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 256–267.
8. Brendel, W.; Rauber, J.; Kurakin, A.; Papernot, N.; Veliqi, B.; Mohanty, S.P.; Laurent, F.; Salathé, M.; Bethge, M.; Yu, Y.; et al. Adversarial Vision Challenge. In *The NeurIPS '18 Competition*; Escalera, S., Herbrich, R., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 129–153.
9. Rawat, S.; Srinivasan, A.; R, V. Intrusion detection systems using classical machine learning techniques versus integrated unsupervised feature learning and deep neural network. *arXiv* **2019**, arXiv:1910.01114.
10. singh, K.; Mathai, K.J. Performance Comparison of Intrusion Detection System Between Deep Belief Network (DBN)Algorithm and State Preserving Extreme Learning Machine (SPELM) Algorithm. In Proceedings of the 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 20–22 February 2019; pp. 1–7. [CrossRef]
11. Chaithanya, P.S.; Gauthama Raman, M.R.; Nivethitha, S.; Seshan, K.S.; Sriram, V.S. An Efficient Intrusion Detection Approach Using Enhanced Random Forest and Moth-Flame Optimization Technique. In *Computational Intelligence in Pattern Recognition*; Das, A.K., Nayak, J., Naik, B., Pati, S.K., Pelusi, D., Eds.; Springer: Singapore, 2020; pp. 877–884.
12. Apruzzese, G.; Andreolini, M.; Colajanni, M.; Marchetti, M. Hardening Random Forest Cyber Detectors Against Adversarial Attacks. *arXiv* **2019**, arXiv:1912.03790.
13. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
14. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2014**, arXiv:1412.6572.
15. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.

16. Chakraborty, A.; Alam, M.; Dey, V.; Chattopadhyay, A.; Mukhopadhyay, D. Adversarial Attacks and Defences: A Survey. *arXiv* **2018**, arXiv:1810.00069.
17. Qiu, S.; Liu, Q.; Zhou, S.; Wu, C. Review of Artificial Intelligence Adversarial Attack and Defense Technologies. *Appl. Sci.* **2019**, *9*, 909. [CrossRef]
18. Gelenbe, E. Random Neural Networks with Negative and Positive Signals and Product Form Solution. *Neural Comput.* **1989**, *1*, 502–510. [CrossRef]
19. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
20. Qureshi, A.; Larijani, H.; Ahmad, J.; Mtetwa, N. A Novel Random Neural Network Based Approach for Intrusion Detection Systems. In Proceedings of the 2018 10th Computer Science and Electronic Engineering (CEEC), Colchester, UK, 19–21 September 2018; pp. 50–55. [CrossRef]
21. Qureshi, A.U.H.; Larijani, H.; Ahmad, J.; Mtetwa, N. A Heuristic Intrusion Detection System for Internet-of-Things (IoT). In *2019 Springer Science and Information (SAI) Computing Conference*; Springer: Cham, Switzerland, 2019. [CrossRef]
22. Datasets Available For Intrusion Detection. Available online: https://www.unb.ca/cic/datasets/index.html (accessed on 17 July 2020).
23. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A Detailed Analysis of the KDD CUP 99 Data Set. Technical report. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009.
24. NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity |. Available online: http://www.unb.ca/cic/datasets/nsl.html (accessed on 3 May 2018).
25. Papernot, N.; Faghri, F.; Carlini, N.; Goodfellow, I.; Feinman, R.; Kurakin, A.; Xie, C.; Sharma, Y.; Brown, T.; Roy, A.; et al. Technical Report on the CleverHans v2.1.0 Adversarial Examples Library. *arXiv* **2018**, arXiv:1610.00768.