Defending Network Intrusion Detection Systems against Adversarial Evasion Attacks

Marek Pawlicki[a,b,], Michał Choraś[a,b], Rafał Kozik[a,b]

[a] *ITTI Sp. z o.o.*
[b] *UTP University of Science and Technology*

Abstract

Intrusion Detection and the ability to detect attacks is a crucial aspect to ensure cybersecurity. However, what if an IDS (Intrusion Detection System) itself is attacked; in other words what defends the defender? In this work, the focus is on countering attacks on machine learning-based cyberattack detectors. In principle, we propose the adversarial machine learning detection solution. Indeed, contemporary machine learning algorithms have not been designed bearing in mind the adversarial nature of the environments they are deployed in. Thus, Machine Learning solutions are currently the target of a range of attacks. This paper evaluates the possibility of deteriorating the performance of a well-optimised intrusion detection algorithm at test time by crafting adversarial attacks with the four of the recently proposed methods and then offers a way to detect those attacks. The relevant background is provided for both artificial neural networks and four ways of crafting adversarial attacks. The new detection method is explained in detail, and the results of five different classifiers are compared. To the best of our knowledge, detecting adversarial attacks on artificial neural networks has not yet been widely researched in the context of intrusion detection systems.

1. Introduction

In this paper, two immensely important and emerging topics are discussed. The first regards the use of Machine Learning (ML) techniques for Intrusion Detection Systems (IDS) in the Cybersecurity domain; a practice which is widespread and, one could say,

URL: marek.pawlicki@utp.edu.pl (Marek Pawlicki)

unavoidable with the volumes of traffic present in contemporary network usage. The topic has been gaining traction for some time now and massive amounts of brilliant research work have been conducted to further the speed, accuracy, precision and other valuable measures for ML-based IDS.

The other topic has become significantly more relevant over the last couple of years. With ML-based solutions and Artificial Intelligence (AI) technologies finding their way into virtually every facet of present-day's life, a number of new challenges regarding these methods came to surface. One of the most pressing issues comes in the form of Adversarial Attacks. In recent years, the data-driven algorithms, the mechanisms that comprise the inner workings of many intelligent systems, have themselves been the subject of a barrage of attacks.

The case of adversarial attacks, as brought to light by [1], is rapidly developing into a serious peril for modern AI applications, principally with the current proliferation of data-driven technologies in extraordinarily vital applications, like autonomous driving, biometrics or cybersecurity [2]. As noted in [3], adversarial attacks can be dangerous if used, for example, to change the classification of stop road-signs into yield [3] or speed limit [4]. By the same token, a slight perturbation can allow malware to avoid detection [5]. A well executed adversarial attack against a Network IDS can circumvent the detection, which is a direct challenge to the existence of the machine-learning based intrusion detection systems. Answering this predicament is the motivation of this work.

The major contribution of this paper comes in the proposition and the examination of a detection system capable of intercepting adversarial attacks on IDS.

The proposed solution is a novel approach to handling adversarial attacks against artificial neural networks. In this work:

- four different adversarial attack methods are implemented
- a method of detecting those attacks is introduced
- the method does not influence the detection results of the IDS
- the solution is experimentally tested on a recent IDS benchmark dataset

- five different pattern recognition algorithms were tested in the detection pipeline, and the results are provided

The paper is structured as follows: in Section 2 related works in the relevant domains are disclosed. In Section 3 the proposal for a novel method of evasion attack detection is outlined, in Section 4 the experimental setup and results are reported and in Section 5 the conclusions are given.

2. Related Work

In this section, an overview of recent and relevant papers in the domain is given. Firstly, information on Artificial Neural Networks (ANN) and their usage in intrusion detection systems (IDS) and the task of detecting cyberattacks is provided. Then related works in adversarial attacks and adversarial attacks detection are presented. In particular, the following types of attacks are tackled: Fast Gradient Sign Method, Basic Iterative Method, Carlini and Wagner Attack, and Projected Gradient Descent.

2.1. Artificial Neural Networks

The Artificial Neural Network (ANN) is a well-established, versatile modelling algorithm that over the years found immense success in a wide range of applications, being capable, in its many variants, to handle regression, classification, clustering and time series analysis. The premise of an ANN is that it resembles, to a point, the cognitive abilities of a biological neural network [6]. The staggering success of ANN architectures stems from the way they fit to training data, having strong approximation capacities, a critical property for ML algorithms deployable in real-world applications.

ANN fits to data by adjusting the weights of the neural nodes on successive layers ensuing batches of data. This procedure allows for exceptional recognition of the relations among the inputs and grants the ability to generalise well enough to perform successfully on unforeseen data [7]. Fundamentally the process is like fitting a line, or a plane, or a hyper-plane through a set [8].

The algorithm allows to build a virtually infinite number of architectures, with different number of layers, different numbers of neurons on each layer, the neurons having a range of options in terms of their activation function, batch size, number of epochs; the optimiser can be chosen as well, all of that topped with a custom loss function. The optimisation of an ANN and all its hyperparameters for the use in Intrusion Detection has been evaluated in-depth in [9].

An ANN of just one layer is usually referred to as a perceptron. With all the features from the feature vector fed to the input layer, they go directly to the computational layer. The input layer consists of d nodes that speak for d features $X = [x_1...x_d]$ and edges of weight $W = [w_1...w_d]$. The output neuron computes $W \cdot X = \sum_{i=1}^{d}(w_i x_i)$. As a way to deal with possible distribution imbalance, bias b can be utilised.

The prediction of $\hat{y}$ comes then as the result of the following:

$$\hat{y} = sign\{W \cdot X + b\} = sign\{\sum_{i=1}^{d} w_i x_i + b\}$$

It is apparent in this case, with $sign$ as the activation function $\Phi(v)$ that the result will be binary. One of the most commonly used activation functions in contemporary ANNs is the Rectified Linear Unit (ReLU). This is also the activation used in this work for both the IDS ANN and the Detector.

The loss function can be defined as the minimisation of the error, understood as the difference between the test value and the predicted value, so $E(X) = y - \hat{y}$, therefore

$$\sum_{(X,y)\in D} (y - sign\{W \cdot X\})$$

2.2. *Artificial Neural Networks in Intrusion Detection*

A wide number of different ANN setups for the use in IDS has been proposed over the last few years. The authors of [10] evaluate a range of different ANN architectures to check the relationship among precision, recall, accuracy, and the complexity of the network. In general, the authors conclude that more complex networks are able to, to a point, improve the detection results. This, however, comes at the cost of a higher computational requirement of the setup.

[11] tests a simple, but GridSearch optimised ANN over a part of the CIC-IDS dataset. The authors claim the testing accuracy of 99.97%.

A range of algorithms is evaluated in [12] to model anomaly detection in HTTP requests.

In [13] the authors present an evaluation of shallow and deep NN architectures for IDS. Testing a range of setups over the KDD99 dataset, the best performing one - a Deep NN of 3 layers, achieves results better than other architectures and a range of classical machine learning algorithms alike. The reported accuracy is 0.930, with the precision of 0.997, recall of 0.915 and the f1-score of 0.955.

In [14] a Convolutional Neural Network is suggested for feature extraction and classification. The architecture, which consists of three convolutional layers intertwined with three max-pooling layers, achieves 99.23% accuracy in experiments conducted on the KDD99 dataset [15].

Similarly, a CNN architecture utilising 3 convolutional layers - one 1x1, one 3x3 and one 5x5 with batch normalisations after each layer (except the fully connected layer) is evaluated in [16]. To counter the vanishing/exploding gradient problem a constraint to normal distribution of $N(0, 1)$ is introduced on each layer. The convolutional layers are set up to follow the 'Inception' model. ReLU is used as the activation function. The proposed architecture achieves 94.11% accuracy, trained and tested over the KDD99 dataset [16]. [17] describes combining two neural network architectures for network intrusion detection, a Deep Confidence Neural Network for feature extraction and an ANN for classification. The experiments are performed over the KDD'99 [15] dataset. The Deep Confidence NN is a variant of the Restricted Boltzman Machine (RBM) [18], a neural network used to figure out the probability distribution over its set of inputs. The authors prove better results as compared to Principal Component Analysis (PCA) [19] by almost 10 percentage points in four tests.

Some researches study the possibility of using a Long Short-Term Memory (LSTM) network for intrusion detection. An LSTM is was conceived to recognise long-term dependencies in the data. In [20] the network has its hyperparameters optimized, with the ADAM optimizer selected. The authors find the LSTM architecture in conjunction with the ADAM optimizer valid for IDS.

Similarly, the authors of [21] test the LSTM setup for validity as IDS. It is trained on the KDD'99 benchmark as well. The LSTM is optimized for preferable hyperparameters and when stacked against traditional Machine Learning algorithms, the results are promising.

There are studies that examine one of the most novel ANN builds, the Gated Recurrent Unit (GRU) [22] for the use in IDS [23] and [24], with KDD'99 and NSL-KDD datasets utilised respectively. The GRU, concisely put, is an extension of the recurrent neural network architecture, with the ability to recognise temporal patterns in the data, but not as prone to overfitting as the Long Short-Term Memory variant. The achieved accuracy exceeds 98% for [24] and 99% for all the GRU setups in [23].

In [25] a systematic review of 43 articles on the use of ANN in IDS can be found. According to the authors, the research in the field is on the rise and is predicted to be a trending topic in the upcoming years.

2.3. Adversarial Machine Learning and Attack Generation

Over the last few years, the research into the curious properties of ML has exploded. The fact that a skilfully crafted feature vector can fool even the classifiers that exceed human performance on a benchmark dataset has riveted the attention of the AI scientific community. With the awareness of the issue rising, a range of soft spots has been found [26]. Adversarial examples are the samples that for all intents and purposes look almost identical to correctly classifiable data; however, with a small, intentional, worst-case perturbation that can cause a range of ML algorithms, most notably artificial neural networks, to fail [27].

2.3.1. Fast Gradient Sign Method

The authors of [28] found a rapid approach to dependably produce adversarial examples that lead an array of ML methods to misclassify. The method was initially demonstrated on ImageNet, MNIST [29] and CIFAR-10 [30] datasets. It relies on finding a small adversarial noise vector that when summed up corresponds with the sign of the elements of the gradient of the cost function for the evaluated sample. The Fast Gradient Sign Method can be defined as the linearization of the cost function around the current value of Θ, obtaining an optimal max-norm constrained perturbation of

$$\eta = \epsilon sign(\nabla_x J(\Theta, x, y))$$

Where Θ stands for the parameters of the model, x for the inputs to the model, y for the targets of the corresponding x and $J(\Theta, x, y)$ is the cost utilised to train the ANN [28]. The method is referred to as Fast Gradient Sign Method (FGSM), Fast Gradient Method (FGM) or simply Fast Method in literature.

2.3.2. Basic Iterative Method

The authors of [31] offer an extension to the FGM method, applying it multiple times using a small step size, clipping the values after every transitional pace, also using $\alpha = 1$, which translates to changing the value of each element (i.e. pixel) by 1. In their work the authors have chosen the number of iterations heuristically to be enough to reach the border of the ϵ max-norm ball. The formula used is as follows:

$$X_0^{adv} = X_1,\, X_{N+1}^{adv} = Clip_{X,\epsilon}\big\{X_N^{adv} + \alpha sign(\nabla_x J(X_N^{adv}, y_{true}))\big\}$$

2.3.3. Carlini and Wagner Attack

[32] offer the solution to the adversarial example creation by formulating the optimisation problem in a way that can be dealt with by current algorithms. The optimisation problem is formally defined as

$$\begin{aligned} minimise \quad & D(x, x+\delta) \\ such\, that \quad & C(x+\delta) = t \\ & x + \delta \in [0,1]^n \end{aligned}$$

where x does not change, so one aims to find δ that minimises $D(x, x+\delta)$. In other words - finding δ that will change the classification. D is a distance metric; in their paper the authors evaluate three of them - L_0, L_2 and L_∞; however, for the use in this paper L_2 was selected.

To make the formula solvable, the authors redefine an objective function f so that $C(x+\delta) = t$ when $f(x+\delta) \leq 0$ and offer a range of options for the f formula. In this work the attack defined in [32] as the L_2 attack was used.

2.3.4. Projected Gradient Descent

In [33] Projected Gradient Descent is put forward as the strongest attack and the universal 'first order adversary', as it is the definitive method for constrained large-scale optimization. Essentially, the abovementioned FGM is a one-step method for generating adversarial examples; in theory a more dangerous option would be the multi-step procedure, which the authors call 'essentially projected gradient descent', formulated as follows:

$$X^{t+1} = \Pi_{x+S}(x^t + \alpha sgn(\nabla_x L(\Theta, x, y)))$$

2.3.5. Summary of Adversarial Attack Generation

Following the summary found in [33], the attack model can be definitively formulated as a two-level optimisation problem, expressed by the following:

$$\min_{\theta} \rho(\Theta),\ where\, \rho(\Theta) = E_{(x,y)\sim D}[\max_{\delta \in S} L(\Theta, x + \delta, y)]$$

where S is the set of allowed perturbations, D is the distribution, L is the loss and $E_{(x,y)\sim D}$ is the perturbed input before the sample is fed to the loss. This formulation allows to consider the $inner\ maximisation$ and the $outer\ minimisation$ problems. In essence, the four abovementioned attacks are four different approaches to solving this formula.

2.4. Adversarial Capabilities

The hazard level of an adversary is driven by the intelligence it can gather on the workings of the targeted algorithm; this, in turn, influences the variety of attacks they can employ. In literature, this level of acquaintance is called Adversarial Capabilities, categorised as black box and white box [34, 26]. Simply put, black box adversaries do not know anything about the algorithm they attack, in contrast to white box adversaries, who possess full knowledge of the algorithm and thus are the strongest possible adversaries [34].

The adversarial model in this paper assumes full knowledge of the algorithm.

In a real-life scenario model extraction, which is a different kind of adversarial attack on machine learning, can be used to steal a working model and use it to create evasion adversarial attacks [35].

2.5. Countering Adversarial Attacks

A number of possible defences against the effects of adversarial examples have been put forward. One of the defences to be proposed is adversarial retraining, either by trying to correctly classify the adversarial example itself [27, 28, 36, 37] or by creating a separate class for adversarial examples.

This method has its merits, but is not effective on unforeseen attacks, and causes a deterioration of the model in many applications.

A different way of countering the problem comes in the form of Defensive Distillation. Proposed in [38], it was initially a procedure used to form a Deep Neural Network (DNN) with knowledge gained from another DNN, but with reduced computational complexity, transferring the knowledge to smaller architectures [39]. The method infers supplementary observations about training samples coming in the form of class probabilities, the information which is then used as training input for the original DNN. The method generates smoother classification models, thus reducing lowering their responsiveness to adversarial examples [38]. Carlini and Wagner attack proves effectiveness against Defensive Distillation in [32].

There are some researchers who propose training a second classifier to detect Adversarial Examples [40]. The authors claim robustness to FGM and Jacobian Saliency Map Attacks [41]. The approach, however, learns to distinguish adversarial examples from the non-adversarial ones using the same distribution and thus can be evaded by formulating the attack to find adversarial examples that fool both classifiers at the same time, as demonstrated by [42].

An approach most similar to the one suggested in this paper can be found in [43], where the authors inspect the activations of certain layers in the ResNet [44] Convolutional Neural Network. The detector, as the authors put it, performs surprisingly well on CIFAIR10 and ImageNET datasets.

In contrast, the work contained in this paper takes into consideration all the neural activations in all the layers, thanks to the fact that the architecture used in intrusion detection, with satisfactory results, uses fewer layers and fewer neurons on those layers. To the best of our knowledge, this is the first time using neural activations will be applied to detecting adversarial examples in intrusion detection.

The general consensus among the researches with regard to the defensive measures is that no fully safe system has been put forward and no truly field-proven solutions exist [45]. The methods developed to this point apply to certain kinds of attacks, but do not provide defence against all possible adversarial attacks. Some of those solutions lead to the deterioration in ML performance [26].

3. Proposed Method for Evasion Attack Detection in Neural Networks

In this section, the overall approach for Evasion Attack Detection in Neural Networks will be presented. Firstly, the utilised dataset is disclosed, which is followed by the applied dataset preprocessing and the IDS training pipeline. Then the attacks on the IDS are performed and tested. The neural activations of those attacks, as well as the activations for clear samples are gathered; finally the attack detector is trained and tested.

3.1. Intrusion Detection based on an Artificial Neural Network

In this work, the dataset was first cut into four parts in a stratified fashion to ensure full coverage of all kinds of attacks included in the dataset in all its sub-parts. This procedure results in the following setup:

- Dataset A - used to train the IDS classifier
- Dataset B - used to test the IDS classifier and to craft the adversarial attacks, and test them on the original IDS ANN, then to acquire the activations of neural nodes in the IDS network of benign, attack and adversarial samples to train the Adversarial Detector
- Dataset C and D - used to craft test adversarial samples and acquire the activations for the neural nodes of benign, attack and adversarial samples

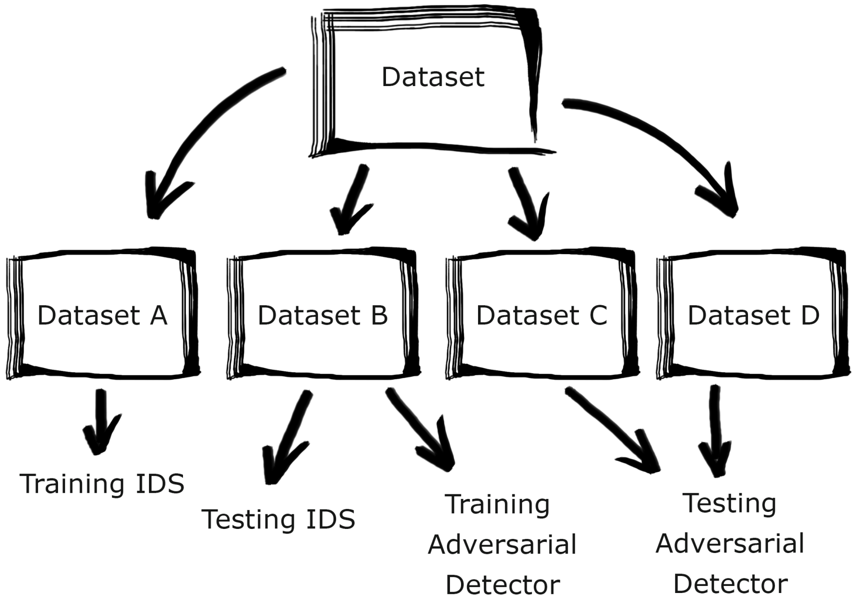

Figure 1: The utilisation and partitioning of the dataset for training and testing of IDS ANN and the Adversarial Detector

All of the sub-parts were then turned into a binary classification task, leaving all the benign samples as 'BENIGN', but changing all the names of possible attacks to simply 'ATTACK'. The utilisation and partitioning of the dataset is depicted in Fig. 1.

The pipeline of the IDS training/testing process is showcased in Fig.4. As seen in the figure, the binarized dataset is fed to the architecture described above, and the training procedure results in building a model capable of binary classification.

3.2. Adversarial Attacks

After testing the trained IDS (the optimisation procedure of an ANN-based IDS can be found in [9]) four different adversarial attacks were crafted based on the ATTACK class of Dataset B. The algorithms used for the creation of evasion attacks were:

- Carlini and Wagner attack (CW) [32]

Table 1: ’IDS ANN’ trained on Dataset A and tested on Dataset B

	precision	recall	f1-score	support
ATTACK	0.96	0.97	0.97	139675
BENIGN	0.99	0.99	0.99	405383
micro avg	0.98	0.98	0.98	545058
macro avg	0.97	0.98	0.98	545058
weighted avg	0.98	0.98	0.98	545058
samples avg	0.98	0.98	0.98	545058

- Fast Gradient Sign Method (FGM) [28]
- Basic Iterative Method (BIM) [31]
- Projected Gradient Descent (PGD) [33]

1397 samples of the ’ATTACK’ class were randomly extracted from Dataset B and turned into adversarial samples with the use of those four algorithms. The IDS ANN classified those as 1353 ATTACKS and 44 BENIGNS. Using adversarial attacks we were able to force the IDS ANN to classify 1296 ’ATTACK’ as ’BENIGN’ samples for BIM and PGD, 1324 for FGM and 59 for CW.

The abovementioned procedures introduce adversarial noise to the samples. This noise resulted in negative values in some of the features. Those negative values were supplanted by zeros with some loss of effectiveness of those attacks -

55 more attacks were classified as benign samples with the original BIM and PGD methods, 295 more for CW and interestingly, 21 fewer for FGM.

The samples from Dataset B not used in crafting Adversarial Attacks were annotated as ’nonadversarial’, the Adversarial Attacks were labelled ’adversarial’. With 5588 adversarial attack samples, a matching number of nonadversarial records was randomly picked from the unused samples of Dataset B to form the base for a balanced ’Adversarial Training Dataset’ for the adversarial attack detector. The procedure is depicted in Fig. 3. Dataset D was subjected, except for the balancing, to the exact same

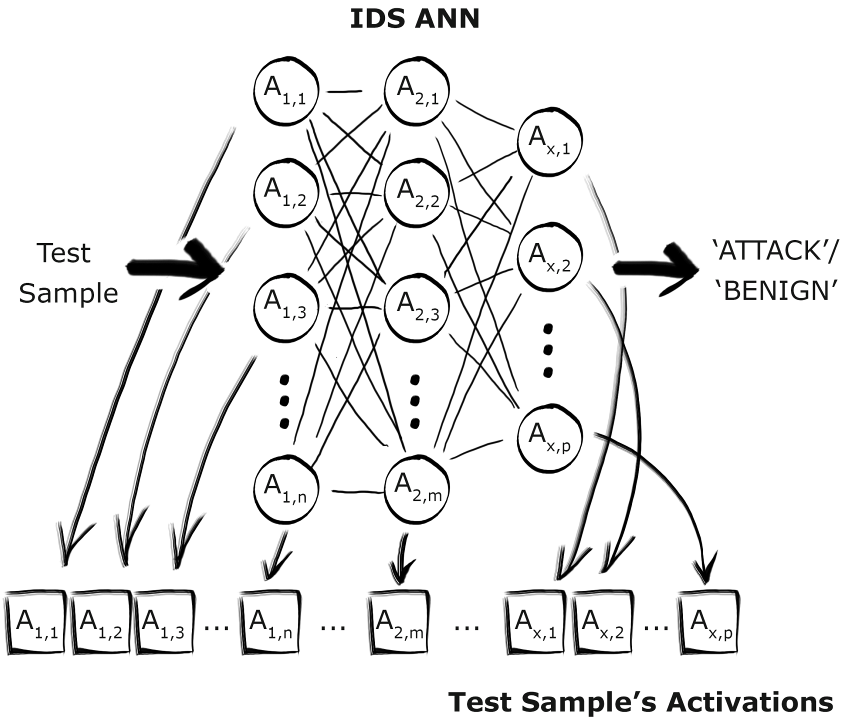

Figure 2: The acquisition of IDS ANN activations for a given test sample

crafting/annotation procedure to form the base for the testing dataset for the detector.

Algorithm 1: Acquisition of the activation dataset

```
Result: Neural Activations Dataset
AttackedDataset = dataset + AdversarialAttacks;
for row in AttackedDataset do
    activationsVector = get_activations(row) ;
    if sample = adversarial then
        activationsVector.Label = 'adversarial';
    else
        activationsVector.Label = 'non-adversarial';
    end
    NeuralActivationsDataset += activationsVector;
end
```

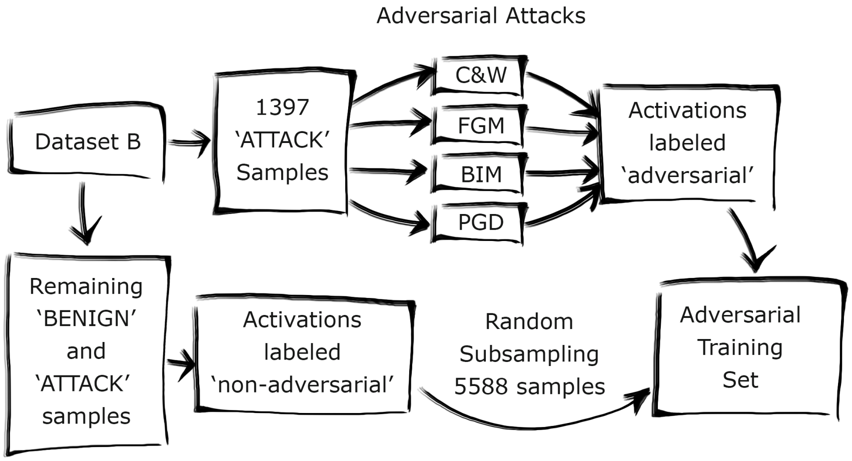
Adversarial Attacks
Dataset B
1397 'ATTACK' Samples
C&W
FGM
BIM
PGD
Activations labeled 'adversarial'
Remaining 'BENIGN' and 'ATTACK' samples
Activations labeled 'non-adversarial'
Random Subsampling 5588 samples
Adversarial Training Set

Figure 3: Forming the Adversarial Training Dataset from Dataset B

3.3. Detection Method

The training and testing activation datasets were fed to the IDS ANN and the activations for all 102 neurons (including the softmax layer), as shown in Fig.2, were recorded and annotated as adversarial or nonadversarial respectively.

4. Experimental Setup and Results

4.1. IDS ANN Setup

The IDS setup was as follows: An Artificial Neural Network of 3 hidden layers was compiled, with 40 neurons on the first hidden layer, 40 on the second and 20 on the third layer. The Rectified Linear Unit activation function was utilised, and the optimiser selected was ADAM. With batch size of 100 and 10 epochs, the network achieved the accuracy of 0.9827 when trained with Dataset A and tested on Dataset B. The precision, recall and f1-score are showcased in Tab. 1

4.2. Detector Setup

The recorded activations were used to train the detector artificial neural network. The architecture of the detector is as follows: 3 hidden layers with the ReLU activation

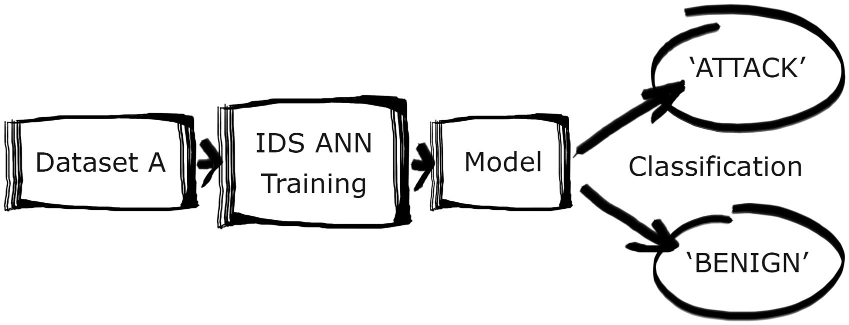

Figure 4: The IDS ANN pipeline

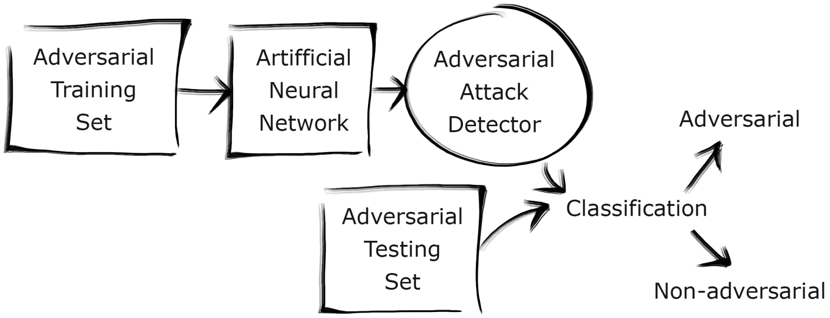

Figure 5: The Adversarial Detector Training/Testing Pipeline

function, 51, 51 and 25 neurons respectively and the ADAM optimiser. The training / testing pipeline is presented in Fig. 5. Using batch size of 100 and just 10 epochs, the detector achieved the accuracy of 0.8506 on the testing set. The detailed results are assembled in Tab. 2.

The detector achieves high accuracy and the recall for the adversarial class signifies that it can recognise the attacks with great promise. The precision however is the evidence of a high number of false-positives.

Since in the process of creating the ANN-based adversarial attack detector a dataset of neural activations of the IDS architecture was created, the authors proceeded to test

Table 2: Results of Adversarial Attack Detector over the test set activations using various ML classifiers

			ANN			RandomForest	
	precision	recall	f1-score	precision	recall	f1-score	support
adversarial	0.06	0.91	0.11	0.11	0.99	0.20	5588
non-adversarial	1.00	0.85	0.92	1.00	0.91	0.95	543661
macro avg	0.53	0.88	0.51	0.56	0.95	0.58	549249
weighted avg	0.99	0.85	0.91	0.99	0.91	0.95	549249
			ADABoost			SVM	
adversarial	0.07	0.90	0.13	0.11	0.79	0.19	5588
non-adversarial	1.00	0.88	0.93	1.00	0.93	0.97	543661
macro avg	0.53	0.89	0.53	0.55	0.86	0.58	549249
weighted avg	0.99	0.88	0.93	0.99	0.93	0.96	549249

Table 3: Results of Nearest Neighbour-based Adversarial Attack Detector over the test set activations

	precision	recall	f1-score	support
adversarial	0.11	0.99	0.20	5588
nonadv	1.00	0.91	0.95	543661
macro avg	0.56	0.95	0.58	549249
weighted avg	0.99	0.91	0.95	549249

the approach using other well-established classifiers. In Tab. 2 the results of detection with Random Forest (RF) are presented. As immediately apparent, with this kind of data, RF achieves results superior to ANN, with higher recall and better precision. Notably, the accuracy of this approach exceeds 91% (91.24).

Following the success of the RF-based classifier, another ensemble method was tested. The ADABoost algorithm did not surpass the results of the Random Forest, getting up to only 87.66% accuracy. This, however, is still a better result than the ANN approach. The details can be found in Tab 2.

The tests were then followed by building a model relying on the Support Vector Machine algorithm. As can be noticed from investigating the Tab. 2, the SVM was not as successful in picking up on the adversarial attacks based on the activations as the other algorithms, with the recall of only 0.79.

Finally, the same activation dataset was utilised to train a nearest neighbour classifier. The procedure achieved results on par with the Random Forest method. The details can be found in Tab. 3

4.3. k-fold cross-validated paired t-test

To back the analysis of the experimental results by statistical analysis, the k-fold cross-validated paired t-test has been performed. The comparison of the best-performing and worst-performing classifiers revealed that the t-value is 1.3017 and the p-value is 0.104717. The result is not significant at $p < 0.05$; therefore, the presented procedure is suitable for any of the presented classifiers.

5. Conclusions and future work

The major contribution of this work comes in the form of using neuron activations at test time to detect adversarial attacks performed with the use of four known evasion attack algorithms, namely: Fast Gradient Sign, Basic Iterative Method, Carlini and Wagner attack, and Projected Gradient Descent in the context of cybersecurity. The authors collected the test time neural activations of an ANN trained on a part of the CICIDS2017 dataset [46] and the neural activations of the adversarial examples crafted for this ANN. Using these activations, the authors trained and tested five different ML classifiers to detect adversarial examples, achieving the recall of 0.99 for adversarial attacks with two of the classifiers - Random Forest and the Nearest Neighbour classifier.

The reported results are promising, and suggest the possibility of building an adversarial attack detector which does not affect the classification results of the protected model, which can drive further research in the direction of defending the networks and intrusion detection systems based on machine learning algorithms. However, further lowering of the false positive rate is of utmost importance for the future development of this method.

While the topic of evasion attacks on artificial neural networks has been gaining in popularity recently, most of the research is performed on image recognition datasets. For this reason, it is hard to compare the results of this innovative work with other defence or detection systems.

Acknowledgement

This work is funded under the SPARTA project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 830892.

References

[1] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, F. Roli, Is feature selection secure against training data poisoning?, CoRR abs/1804.07933. arXiv:1804.07933.
URL http://arxiv.org/abs/1804.07933

[2] P. Wei Koh, J. Steinhardt, P. Liang, Stronger data poisoning attacks break data sanitization defenses (11 2018).

[3] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, A. Swami, Practical black-box attacks against machine learning, in: Proceedings of the 2017 ACM on Asia conference on computer and communications security, 2017, pp. 506–519.

[4] T. Gu, K. Liu, B. Dolan-Gavitt, S. Garg, Badnets: Evaluating backdooring attacks on deep neural networks, IEEE Access 7 (2019) 47230–47244. doi:10.1109/ACCESS.2019.2909068.

[5] B. Kolosnjaji, A. Demontis, B. Biggio, D. Maiorca, G. Giacinto, C. Eckert, F. Roli, Adversarial malware binaries: Evading deep learning for malware detection in executables, in: 2018 26th European Signal Processing Conference (EUSIPCO), IEEE, 2018, pp. 533–537.

[6] O. Maimon, L. Rokach, Data Mining and Knowledge Discovery Handbook, 2nd ed, 2010.

[7] I. N. da Silva · Danilo Hernane Spatti Rogerio Andrade Flauzino Luisa Helena Bartocci Liboni Silas Franco dos Reis Alves, Artificial Neural Networks A Practical Course, 2017. doi:10.1007/978-3-319-43162-8.

[8] S. B. A. E. F. C. M. E. Pasero, Advances in Neural Networks, 2016. doi:10.1007/978-3-319-33747-0.

[9] M. Pawlicki, R. Kozik, M. Choraś, Artificial neural network hyperparameter optimisation for network intrusion detection, in: Intelligent Computing Theories and Application - 15th International Conference, ICIC 2019, Nanchang, China, August 3-6, 2019, Proceedings, Part I, 2019, pp. 749–760. doi:10.1007/978-3-030-26763-6_72.
URL https://doi.org/10.1007/978-3-030-26763-6_72

[10] R. Tavoli, et al., Providing a method to reduce the false alarm rate in network intrusion detection systems using the multilayer perceptron technique and backpropagation algorithm, in: 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), IEEE, pp. 001–006.

[11] V. Kanimozhi, T. P. Jacob, Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset cse-cic-ids2018 using cloud computing, in: 2019 International Conference on Communication and Signal Processing (ICCSP), IEEE, 2019, pp. 0033–0036.

[12] M. Choraś, R. Kozik, Machine learning techniques applied to detect cyber attacks on web applications, Logic Journal of IGPL 23 (2015) 45–56. doi:10.1093/jigpal/jzu038.

[13] K. R. Vigneswaran, R. Vinayakumar, K. Soman, P. Poornachandran, Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security, in: 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, 2018, pp. 1–6.

[14] R. U. Khan, X. Zhang, M. Alazab, R. Kumar, An improved convolutional neural network model for intrusion detection in networks, in: 2019 Cybersecurity and Cyberforensics Conference (CCC), 2019, pp. 74–77. doi:10.1109/CCC.2019.000-6.

[15] K. Cup, Available on: http://kdd. ics. uci. edu/databases/kddcup99/kddcup99.html (2007).

[16] L. Yong, Z. Bo, An intrusion detection model based on multi-scale cnn, in: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), IEEE, 2019, pp. 214–218.

[17] W. Peng, X. Kong, G. Peng, X. Li, Z. Wang, Network intrusion detection based on deep learning, in: 2019 International Conference on Communications, Information System and Computer Engineering (CISCE), IEEE, 2019, pp. 431–435.

[18] P. Smolensky, Information processing in dynamical systems: Foundations of harmony theory, Tech. rep., Colorado Univ at Boulder Dept of Computer Science (1986).

[19] K. Pearson, Liii. on lines and planes of closest fit to systems of points in space, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2 (11) (1901) 559–572.

[20] S. Althubiti, W. Nick, J. Mason, X. Yuan, A. Esterline, Applying long short-term memory recurrent neural network for intrusion detection, in: SoutheastCon 2018, 2018, pp. 1–5. doi:10.1109/SECON.2018.8478898.

[21] J. Kim, J. Kim, H. L. T. Thu, H. Kim, Long short term memory recurrent neural network classifier for intrusion detection, in: 2016 International Conference on Platform Technology and Service (PlatCon), 2016, pp. 1–5. doi:10.1109/PlatCon.2016.7456805.

[22] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078.

[23] C. Xu, J. Shen, X. Du, F. Zhang, An intrusion detection system using a deep neural network with gated recurrent units, IEEE Access 6 (2018) 48697–48707.

[24] M. Pawlicki, A. Marchewka, M. Choraś, R. Kozik, Gated recurrent units for intrusion detection, in: Image Processing and Communications - Techniques, Algorithms and Applications, IP&C'2019, Bydgoszcz, Poland, 11-13 September 2019, Proceedings, 2019, pp. 142–148. doi:10.1007/978-3-030-31254-1_18.
URL https://doi.org/10.1007/978-3-030-31254-1_18

[25] M. U. ÖNEY, S. PEKER, The use of artificial neural networks in network intrusion detection: A systematic review, in: 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), IEEE, 2018, pp. 1–6.

[26] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, D. Mukhopadhyay, Adversarial attacks and defences: A survey, CoRR abs/1810.00069. arXiv:1810.00069.
URL http://arxiv.org/abs/1810.00069

[27] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199.

[28] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples (2014). arXiv:1412.6572.

[29] Y. LeCun, The mnist database of handwritten digits, http://yann. lecun. com/exdb/mnist/.

[30] A. Krizhevsky, V. Nair, G. Hinton, The cifar-10 dataset, online: http://www. cs. toronto. edu/kriz/cifar. html 55.

[31] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial examples in the physical world, arXiv preprint arXiv:1607.02533.

[32] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 IEEE Symposium on Security and Privacy (SP), IEEE, 2017, pp. 39–57.

[33] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks (2017). `arXiv:1706.06083`.

[34] N. Papernot, P. McDaniel, A. Sinha, M. P. Wellman, Sok: Security and privacy in machine learning, in: 2018 IEEE European Symposium on Security and Privacy (EuroS P), 2018, pp. 399–414. `doi:10.1109/EuroSP.2018.00035`.

[35] M. Choraś, M. Pawlicki, R. Kozik, The feasibility of deep learning use for adversarial model extraction in the cybersecurity domain, in: International Conference on Intelligent Data Engineering and Automated Learning, Springer, 2019, pp. 353–360.

[36] B. Li, Y. Vorobeychik, X. Chen, A general retraining framework for scalable adversarial classification (2016). `arXiv:1604.02606`.

[37] K. Grosse, P. Manoharan, N. Papernot, M. Backes, P. McDaniel, On the (statistical) detection of adversarial examples (2017). `arXiv:1702.06280`.

[38] N. Papernot, P. McDaniel, X. Wu, S. Jha, A. Swami, Distillation as a defense to adversarial perturbations against deep neural networks, 2016 IEEE Symposium on Security and Privacy (SP)`doi:10.1109/sp.2016.41`.
URL `http://dx.doi.org/10.1109/SP.2016.41`

[39] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network (2015). `arXiv:1503.02531`.

[40] Z. Gong, W. Wang, W.-S. Ku, Adversarial and clean data are not twins (2017). `arXiv:1704.04960`.

[41] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, A. Swami, The limitations of deep learning in adversarial settings, 2016 IEEE European Symposium on Security and Privacy (EuroSP)doi:10.1109/eurosp.2016.36.
URL http://dx.doi.org/10.1109/EuroSP.2016.36

[42] N. Carlini, D. Wagner, Adversarial examples are not easily detected, Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security - AISec '17doi:10.1145/3128572.3140444.
URL http://dx.doi.org/10.1145/3128572.3140444

[43] J. H. Metzen, T. Genewein, V. Fischer, B. Bischoff, On detecting adversarial perturbations (2017). arXiv:1702.04267.

[44] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[45] X. Liao, L. Ding, Y. Wang, Secure machine learning, a brief overview, in: 2011 Fifth International Conference on Secure Software Integration and Reliability Improvement - Companion, 2011, pp. 26–29. doi:10.1109/SSIRI-C.2011.15.

[46] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization., in: ICISSP, 2018, pp. 108–116.