Adversarial Training in Deep Learning-Based Intrusion Detection System

Kuang Yenting, kuang.ye@northeasterm.edu
Gao Siyuan, gao.siyuan1@northeastern.edu
Sun Yiwei, sun.yiwe@northeastern.edu

Abstract

Intrusion Detection Systems (IDS) play a critical role in protecting networks against malicious activities, but their performance can degrade under adversarial attacks. For instance, someone deliberately attempts to manipulate inputs and evade detection. This study investigates the application of adversarial training to enhance the robustness of deep learning-based IDS against such attacks. We evaluate a Convolutional Neural Network (CNN) model on the UNSW-NB15 dataset, focusing on its resilience to three well-known adversarial attack methods: Projected Gradient Descent (PGD), Basic Iterative Method (BIM), and Fast Gradient Sign Method (FGSM). For feature selection, we employ a random forest to improve model performance. Our results demonstrate that adversarially trained IDS significantly enhance detection accuracy and robustness. Specifically, under PGD and BIM attacks, the model maintains an accuracy of 66%, while under FGSM attacks, it achieves an accuracy of 83%. These findings highlight the potential of adversarial training to strengthen IDS against sophisticated threats, contributing to more secure real-time network monitoring solutions.

Key words*: Adversarial Training, Intrusion Detection Systems, Convolutional Neural Network, Projected Gradient Descent, Basic Iterative Method, Fast Gradient Sign Method, Random Forest.*

1. Introduction

As the rapid development of deep learning in recent years and its application in more and more fields, deep neural networks are increasingly used in the field of network security applications like network intrusion detection systems (IDS). Since IDS itself is adversarial in nature, the importance of security, stability and accuracy in its deep neural network increased and became a key goal in the DNN design progress.

Many works have studied the adversarial attack in different neural networks that mainly focus on picture classification but there is not much research on adversarial training neural network models for IDS. Using deep neural networks such as CNN in IDS can effectively distinguish between benign and malicious input. But adversarial samples may expose the blind spots in inputs or deep neural network models which will lead to incorrect distinguishing results. So, in this paper we will use adversarial training to create a more robust and with high stability CNN model to solve the IDS problem which will have more practical value in real activities

In this work, we will use UNSW-NB15[1][2][3][4][5] as our modeling dataset. We first preprocess the dataset and perform dimensionality reduction. To do this, we use random forest to select the top 8 import features. Random forest can filter important features and retain the original feature semantics. Also, it is highly robust. Then we construct the basic CNN deep neural network using dataset UNSW-NB15 after dimensionality reduction. CNN is a more commonly used DNN model. It is suitable for processing high-dimensional data and has high computational efficiency. We only use CNN algorithm as our basic neural network model, but the same research method as this paper can also be used for IDS problems based on other DNN models.

We use min-max[6][7] formulation in this paper to be the adversarial training method. The model's robustness to adversarial attacks in this algorithm is improved by minimizing the model's loss on adversarial samples (min) and generating adversarial samples with maximum attack effects (max). The first step is to generate adversarial samples. In this work, we use Fast Gradient Sign Method (FGSM), BIM and Projected Gradient Descent (PGD) to form the adversarial attack and adversarial samples. FGSM is a kind of simple one step attack while PGD and BIM are more powerful multi-step. These attack methods are used to form adversarial samples and to solve the inner maximization problem in the min-max formulation. We selected three different attack methods to ensure the integrity of adversarial training and the optimization of the final results.

Finally, we use the adversarial samples formed above and the original samples to train the basic CNN model together. The loss needed to be minimized in the process. We test the performance and robustness of the model on adversarial samples and normal samples and compare the result to see if after adversarial training, the robustness, stability and accuracy of the model have been improved accordingly.

2. Background

Recent research [8] has shown that deep neural networks are susceptible to adversarial paradigms, where the input closely resembles natural material but leads to incorrect classification. This vulnerability demonstrates an inherent weakness in deep learning models where minor perturbations can destroy accuracy. To solve this problem, robust optimization has become a key defense method, providing a unified framework for understanding adversarial robustness. It provides security guarantees, especially against first-order adversaries that exploit gradients. By training the network with adversarial samples, robust optimization significantly increases resistance to a wide range of attacks, marking a step toward fully resilient deep learning models.

Khamis et al. (2020) [9] studied the robustness of deep learning-based Intrusion Detection Systems (IDS) using a min-max optimization strategy. This involves generating adversarial samples to maximize loss (max step) and retraining the IDS to minimize it (min step). Their experiments with the UNSW-NB15 dataset showed that adversarial training improves IDS resilience. Applying Principal Component Analysis (PCA) for dimensionality reduction further enhanced robustness by retaining critical features while reducing complexity.

Building on this, Khamis and Matrawy (2020) [10] provided a comprehensive evaluation of adversarial training's effectiveness across different neural network architectures, including Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs). Utilizing datasets such as UNSW-NB15 and NSL-KDD, they examined the models' resilience to attacks generated by various methods such as FGSM, BIM, PGD, CW, and DeepFool. The study also incorporated feature selection techniques like Recursive Feature Elimination (RFE) and PCA, which helped in identifying and retaining the most relevant features for training. Their findings emphasize that while adversarial training enhances IDS robustness, the degree of improvement varies across different neural network architectures, underscoring the importance of tailored defenses for specific models.

In our study, we applied similar adversarial training techniques to a CNN-based IDS model, demonstrating improved resilience to various attacks. Our approach also emphasizes the importance of feature selection reduction, enabling accurate threat detection even under adversarial conditions. This underscores the value of combining adversarial training with strategic feature engineering for robust IDS systems.

3. Approach

3.1 Feature selection

Because the UNSW-NB15 dataset has 49 features, we need to perform downscaling. We select the features and choose the features with high importance for model prediction in CNN modeling. In order to do this, we use random forests.

Random forest is an extension of bagging. RF builds on the foundation of Bagging ensembles using decision trees as base learners and further introduces randomness in the process of training decision trees. The core principles for random forest feature selection is as follows. Random Forest is composed of multiple decision trees, where each tree is trained on a different subset of the data. During the training process, each decision tree selects the features for splitting based on their effectiveness in improving the split. Random Forest then calculates the "importance scores" of these features based on their contribution to tree splits and combines the results from all trees to provide an overall importance evaluation for each feature.

Using Gini Impurity in random forest to calculate the feature importance. Gini Impurity is calculated as follows:

$$G = 1 - \sum_{i=1}^{C} p_i^2$$

Where C is the total number of classes and p_i is the proportion of samples belonging to class i in the current node.

Assume a node is split into two child nodes, L and R. The Gini index before the split is denoted as G_{before}=G_{root} , representing the Gini index of the current node before division.

After the split, the weighted Gini index of the two child nodes is calculated as:

$$G_{after} = \frac{n_L}{n} G_L + \frac{n_R}{n} G_R$$

where G_L and G_R are the Gini indices of the left and right child nodes, n_L and n_R are the numbers of samples in the left and right child nodes, and n is the total number of samples in the current node.

The reduction in the Gini index, also referred to as the split gain, is expressed as:

$$\Delta G = G_{before} - G_{after}$$

The importance of a feature is determined by the sum of all split gains across all nodes where the feature is used for splitting.

3.2 Adversarial attack methods

3.2.1 Fast Gradient Sign Method (FGSM):

A single-step adversarial attack, designed to maximize prediction error while keeping the perturbation minimal. FGSM is widely used as a baseline for evaluating model robustness. This method generates adversarial samples quickly and effectively, since it applies one perturbation to the input by adding a small, ϵ-scaled adjustment in the direction of the loss gradient. The equation is shown as follow:

$$x^{adv} = x + \epsilon \cdot sign\left(\nabla_x L(\Theta, x, y)\right)$$

Where x is the original input, y is the original input label, ϵ controls the perturbation magnitude to ensure the perturbations are small, L is the loss function, and Θ is the model parameters.

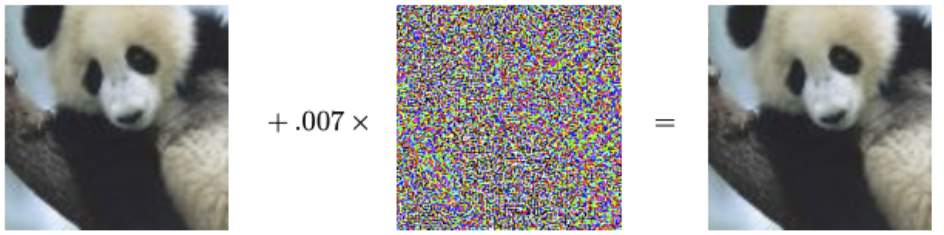

Figure 1. Fast Gradient Sign Method (FGSM) example

3.2.2 Basic Iterative Method (BIM)

An iterative version of FGSM by applying multiple small steps iteratively. This iterative approach makes BIM more potent than single-step attacks in generating adversarial samples. Each step adjusts the input based on the sign of the loss gradient and clips the total perturbation to stay within a threshold. The equation is shown as follow:

$$x^{k+1} = Clip_{x,\epsilon}\left(x^k + \alpha \cdot sign\left(\nabla_x L(\Theta, x^k, y)\right)\right)$$

Where x^k is the perturbed input at iteration k, y is the original input label, α controls the perturbation magnitude, $Clip_{x,\epsilon}$ ensures the perturbations remain within ϵ of the original input x^0, L is the loss function, and Θ is the model parameters.

3.2.3 Projected Gradient Descent (PGD)

An iterative adversarial attack that perturbs inputs to maximize the model's prediction error. Similar to BIM, PGD adjusts the input using the gradient of the loss function and projects it back into a bounded region (an ϵ-ball) around the original input x^0 at each step. This iterative approach makes PGD highly effective at uncovering model vulnerabilities. The iterative equation of the PGD is shown as below [11]:

$$x^{t+1} = \prod_{x+S} \left(x^t + \alpha \cdot sign\left(\nabla_x L(\Theta, x, y) \right) \right)$$

where L is a loss function, x is the input to the model whose parameters represented by Θ, y is the original input label, $\prod_{x+S}$ is a projection operator with perturbation set $x + S$, and α is a gradient step size.

3.3 Deep Neural Network-CNN

In this work, we use CNN as our basic deep neural network model. The basic schematic of CNN is shown below.

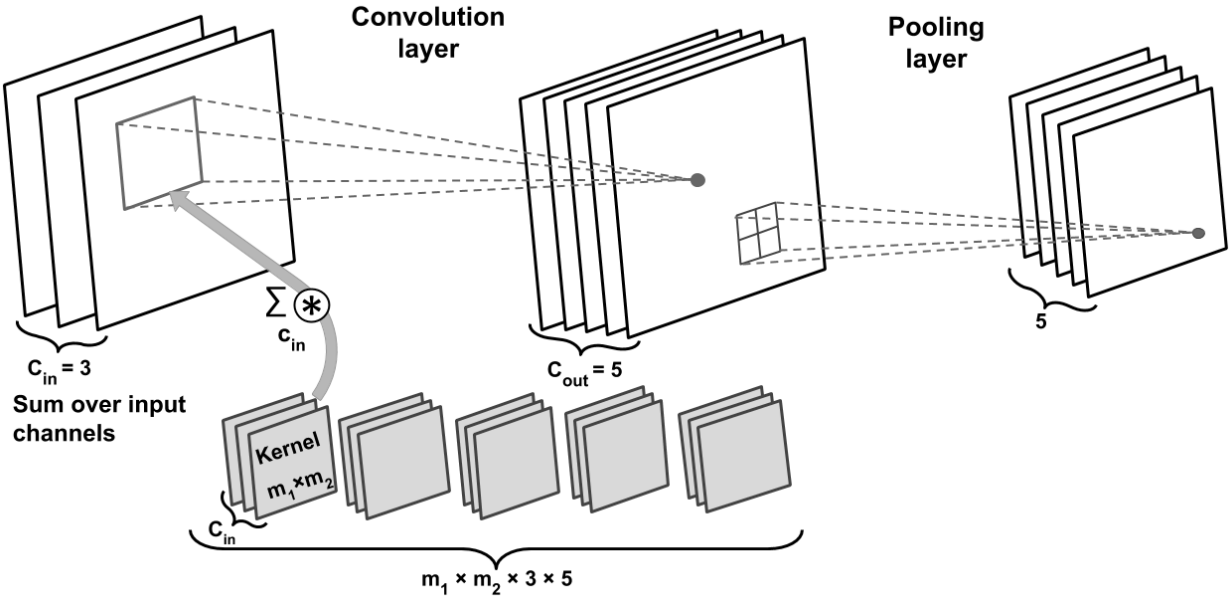

Figure 2. CNN Basic Schematic Diagram

Convolutional Neural Networks (CNNs) is a commonly used classification deep neural network model which can process structured data. CNN is composed of convolutional and pooling layers, which work together to extract meaningful features while reducing computational complexity.

The convolutional layers apply learnable filters (kernels) across input feature maps to extract spatial patterns. Each kernel performs a sliding operation, computing weighted sums

followed by non-linear activations (e.g., ReLU). Stacking these layers enables the network to learn hierarchical features, from low-level edges to high-level textures and shapes.

Pooling layers perform spatial down-sampling (e.g., max or average pooling) to reduce feature map dimensions while retaining critical information. This not only decreases computational costs but also improves the network's robustness to small input variations.

By alternating convolutional and pooling layers, CNNs progressively learn more abstract representations of the input. This hierarchical structure is highly effective for pattern recognition tasks such as classification and segmentation.

Finally, fully connected layers or global pooling map the learned features to the output space. For classification tasks, a softmax activation is typically used to produce class probabilities.

This layered architecture allows CNNs to automatically learn and extract meaningful features, making them a powerful tool for tasks involving high-dimensional structured data. The proposed approach employs CNNs to extract robust features for specific task, as described in subsequent sections.

4. Results and Discussion

4.1 Experiment Structure

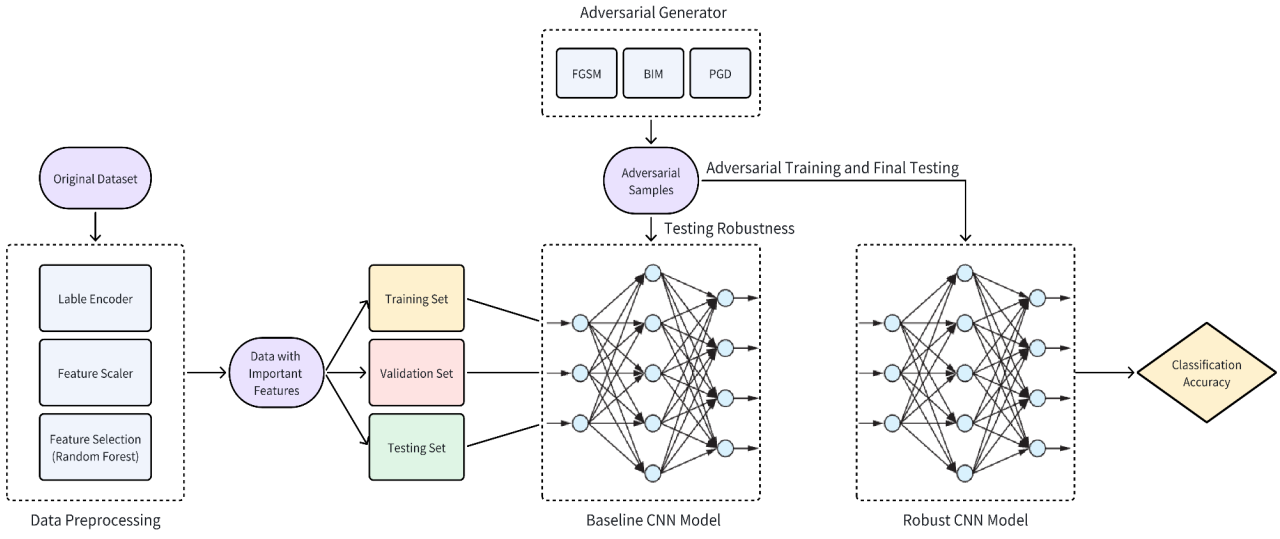

Figure 2. Experiment Structure

The structure of this paper's implementation model can be divided into five main parts:

1. Data Preprocessing and Dataset Splitting

- Label Encoding: Convert the string-type categorical features ('proto', 'service', 'state') into numeric values using a label encoder.
- Feature Scaling: Normalize all numeric feature columns to a range between 0 and 1 using a feature scaler.
- Feature Importance Evaluation: Use the Random Forest algorithm to evaluate the importance of all feature columns with respect to the target variable.
- Dataset Splitting: Retain the most important features and split the dataset into training, validation, and testing subsets.

2. Baseline CNN Model Training
   - Train a baseline CNN model using the training, validation, and testing sets.
3. Robustness Testing of the Baseline CNN Model Using Adversarial Samples
   - Adversarial Sample Generation: Generate adversarial samples using three attack methods: FGSM, BIM, and PGD.
   - Baseline Robustness Testing: Evaluate the baseline CNN model's accuracy and robustness by testing it on the generated adversarial samples.
4. Adversarial Training of the CNN Model
   - Incorporate adversarial samples into the training dataset and retrain the CNN model to enhance its robustness.
5. Final Robustness Testing of the Adversarially Trained CNN Model
   - Evaluate the robustness of the adversarially trained CNN model using adversarial samples and compare the results with the baseline performance to analyze the improvements.

This structured approach ensures a comprehensive evaluation of the CNN model's baseline performance, its vulnerability to adversarial attacks, and the effectiveness of adversarial training in improving robustness.

4.2 Data Preprocessing

In our paper we use UNSW-NB15[1][2][3][4][5] as our benchmark datasets. This dataset has already been cleaned and split into a training set and testing set. Therefore, in this paper we will not perform data cleaning operations such as outlier processing etc.. The UNSW-NB15 datasets contain 49 latitude data indicators including destination port number, destination bits per second, record start time, class labels etc.. Because in this article we only perform binary classification and do not classify different attack types, we delete the “attack_cat” and “id” indicator when preprocessing the data.

We then use random forest to reduce the dimensionality of the dataset. After feature selection, we retain the top 8 important indicators of the original 49 data indicators as the data for our subsequent use.

4.3 Baseline CNN Model

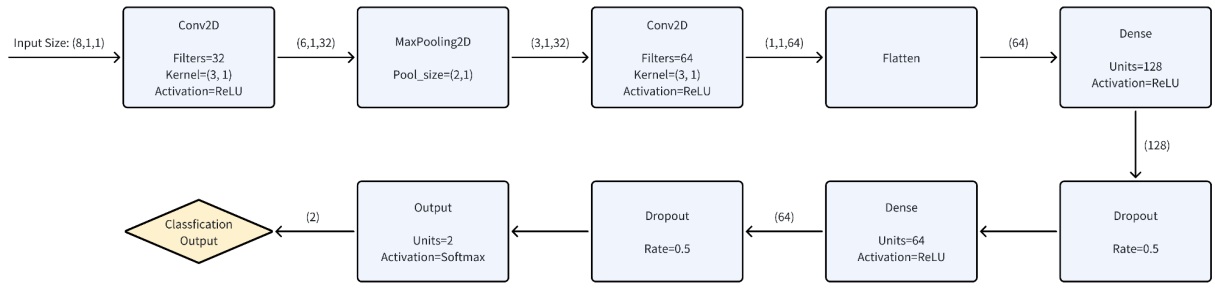

Figure 3. Baseline CNN Model Structure

Figure 3 above shows the baseline CNN model structure we constructed in this work. This CNN model processes input data of shape (8, 1, 1) through multiple layers. The first Conv2D layer applies 32 filters of size (3, 1) with ReLU activation, reducing the height of the input to (6, 1, 32). A MaxPooling2D layer with a pooling size of (2, 1) further reduces the height to (3, 1, 32). A second Conv2D layer with 64 filters and kernel size (3, 1) and ReLU activation processes the data, resulting in a shape of (1, 1, 64). The data is then flattened into a vector of size 64.

The flattened vector is passed through a series of fully connected (Dense) layers. The first dense layer has 128 units with ReLU activation, followed by a Dropout layer with a rate of 0.5 to prevent overfitting. A second dense layer has 64 units with ReLU activation, followed by another Dropout layer with the same rate. Finally, the output layer, which is a Dense layer with 2 units and a Softmax activation function, provides the classification result. This design efficiently captures spatial features and outputs a two-class probability distribution.

4.4 Results of Adversarial Training

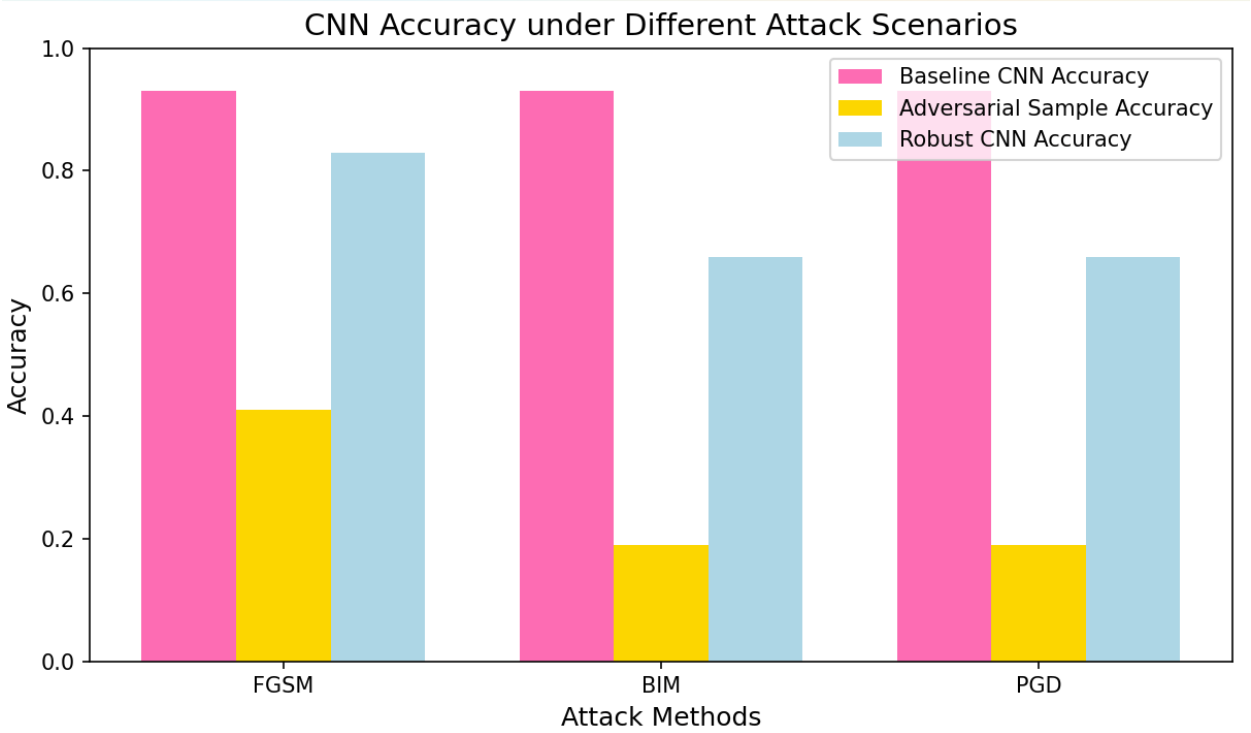

Figure 4. Results

- Baseline CNN Accuracy

In the baseline CNN model, we used dropout technology to prevent overfitting and finally achieved a good accuracy of 93%.

- Robustness testing using adversarial samples

From Figure 4, it can be observed that the generated adversarial samples have a significant impact on the 2D CNN model. The classification accuracy of adversarial samples generated using the FGSM method drops to 41% on the baseline CNN model, while the accuracy for adversarial samples generated using the BIM and PGD methods is only 19%.

- Robustness testing after adversarial training

After completing adversarial training on the baseline CNN model, the impact of adversarial samples generated by the three attack methods on the CNN model is reduced. The accuracy of the adversarially trained CNN model on adversarial samples from the three attack methods improves to 83%, 66%, and 66%, respectively.

5. Conclusion

Through this paper, we verify that a variety of network attack modes do have great interference to the basic neural network model of attack detection, and the accuracy of the model

to identify network attacks can be greatly improved after the adversarial training. The accuracy of the adversarial trained CNN model on adversarial samples from the three attack methods improves to 83%, 66%, and 66%, respectively. This shows that after adversarial training, the model is more stable and accurate than the initial CNN model and can resist certain disturbances and better classify IDS problems. However, there are still some limitations in this paper.

Limitations

- Due to computational limitations, we only tested the detection performance of the CNN model against network attacks. Other deep neural networks, such as ANN and RNN, might achieve better results in detection performance compared to CNN.
- We only used FGSM, BIM, and PGD methods to generate adversarial samples. In fact, there are many other adversarial sample generation methods, such as CW and Deepfool. A more comprehensive approach to adversarial sample generation might provide a more thorough robustness testing.
- Our discussion of attack methods was limited to binary classification (attack or normal). However, network attacks can take many forms, such as DoS, Backdoor, Worms, and Fuzzers. In more detailed classification tasks, it might be possible to develop more precise adversarial training methods for different types of attacks.

Future work

First, given the computational constraints of this work, we focused solely on testing the robustness of CNN models. Future work could expand this investigation to other deep neural networks, such as ANN and RNN, which might exhibit better detection performance against network attacks. Second, we restricted adversarial sample generation to three methods (FGSM, BIM, PGD). Future research should incorporate a broader range of adversarial sample generation techniques, such as CW and Deepfool, to achieve more comprehensive robustness testing and better understand model vulnerabilities. Finally, our evaluation focused on binary classification of attack versus normal traffic. Real-world scenarios often involve diverse types of network attacks, such as DoS, Backdoor, Worms, and Fuzzers. Future studies could explore fine-grained classification tasks and develop adversarial training strategies tailored to specific attack types, enhancing the precision and effectiveness of intrusion detection systems.

Reference

[1] Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, 2015.

[2] Moustafa, Nour, and Jill Slay. "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset." Information Security Journal: A Global Perspective (2016): 1-14.

[3] Moustafa, Nour, et al. "Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks." IEEE Transactions on Big Data (2017).

[4] Moustafa, Nour, et al. "Big data analytics for intrusion detection system: statistical decision-making using finite dirichlet mixture models." Data Analytics and Decision Support for Cybersecurity. Springer, Cham, 2017. 127-156.

[5] Sarhan, Mohanad, Siamak Layeghy, Nour Moustafa, and Marius Portmann. NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems. In Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020, Proceedings (p. 117). Springer Nature.

[6] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083

[7] A. Al-Dujaili *et al.*, “Adversarial deep learning for robust detection of binary encoded malware,” in *2018 IEEE SPW*, IEEE, 2018.

[8] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083

[9] R. A. Khamis, M. O. Shafiq and A. Matrawy, "Investigating Resistance of Deep Learning-based IDS against Adversaries using min-max Optimization," ICC 2020 - 2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 2020, pp. 1-7, doi: 10.1109/ICC40277.2020.9149117.

[10] R. A. Khamis and A. Matrawy, "Evaluation of Adversarial Training on Different Types of Neural Networks in Deep Learning-based IDSs," 2020 International Symposium on Networks,

Computers and Communications (ISNCC), Montreal, QC, Canada, 2020, pp. 1-6, doi: 10.1109/ISNCC49221.2020.9297344.

[11] M. S. Ayas, S. Ayas and S. M. Djouadi, "Projected Gradient Descent Adversarial Attack and Its Defense on a Fault Diagnosis System," 2022 45th International Conference on Telecommunications and Signal Processing (TSP), Prague, Czech Republic, 2022, pp. 36-39, doi: 10.1109/TSP55681.2022.9851334.

[1]

[1] Siyuan Gao, Yenting Kuang, Yiwei Sun all participated in the implementation, writing and review of Abstract, Introduction, Background, Approach, Results and Discussion, Conclusion of the paper.