Adversarial Machine Learning

Homework Assignment 4

The assignment is due by the end of the day on Tuesday, April 25.

Objectives:

- Become familiar with ML classification models used in cybersecurity applications and implement adversarial attacks against such models.

Part 1: Attacks on ML systems for Network Intrusion Detection (50 marks)

Dataset: We will use the NSL-KDD dataset for network intrusion detection. The dataset contains 41 features extracted from network data packet records (PCAP files), which include duration of the connection, protocol type, data bytes send from source to destination, number of failed logins, and similar. The dataset has 125,973 records in a training set, and 22,544 records in a test set. The label for each record indicates whether it is normal traffic or an attack. You can download the dataset and a starter code for loading the dataset from the Shared folder on OneDrive. The Data Loader provides short data analysis and preprocessing using the Pandas library. The lecture notes on Network Intrusion Detection provide basic information about the NSL-KDD dataset, as well as more detailed information can be found at this link.

Step 1: Train the following conventional ML models to classify the network connections into normal and attack classes: Decision Trees, Random Forest, k-Nearest Neighbors, naïve Bayes, and Support Vector Machines. It is not expected to finetune these models, but feel free to perform some finetuning if you wish to. An example of classification with Logistic Regression is provided in the Data Loader. Report the classification accuracies on the test dataset by the different models.

Estimated run time: between 5 and 10 minutes.

Step 2: Train a deep-learning model for classification of the network connections into normal and attack classes. You can consider a network with several fully-connected layers and dropout layers, or you can consider other types of layers if you wish. Report the classification accuracy on the test dataset, and display the learning curves. For full marks, it is expected to obtain test accuracy greater than 77%.

Estimated run time: between 2 and 20 minutes.

Step 3: Implement a non-targeted white-box FGSM evasion attack against the deep learning model from Step 2. For creating the adversarial samples, use a random set of 1,000 records from the test dataset. Apply the following perturbation magnitudes: ϵ= [0.01, 0.025, 0.05, 0.1, 0.2, 0.3].

Estimated run time: between 1 and 2 minutes.

Step 4: Implement a non-targeted white-box JSMA (Jacobian Saliency Mapp Attack) against the deep learning model from Task 2. In the ART library, JSMA is implemented using SaliencyMapMethod.

For creating the adversarial samples, use the same random set of 1,000 records from Step 3. Apply three JSMA attacks with different values of the parameters for the function in the ART library. Discuss the results and compare the performance to the FGSM attack from Step 3.

Estimated run time: between 5 and 15 minutes.

Step 5: Implement a non-targeted white-box FGSM against the Logistic Regression model from Step 1. The following notebook in the ART toolbox provides an example of implementing attacks against Logistic Regression. For creating the adversarial samples, use the same random set of 1,000 records from Step 3. Apply the following perturbation magnitudes: ϵ= [0.01, 0.025, 0.05, 0.1, 0.2, 0.3]. Discuss the results and compare the performance to the attacks against the deep learning model from Steps 3 and 4.

Estimated run time: between 1 and 2 minutes.

Part 2: Attacks on ML systems for Malware Detection (50 marks)

Dataset: For this task, we will use a dataset of malware and goodware samples. The dataset consists of 216,351 records. Each record includes features extracted from Portable Executable (PE) files for Microsoft Windows OS. The raw features from the PE files were converted into 53 numerical features in the dataset. The file for the dataset "malware_dataset.csv" can be downloaded from the Shared folder on OneDrive.

Step 1: Load the dataset using the Pandas library. If you need to refresh your knowledge regarding Pandas, you can review these two notebooks: Data Manipulation with Pandas and Data Exploration and Preprocessing, but also you can find tons of resources about Pandas on the web.

Print basic information about the columns and rows in the dataset to ensure that the data is loaded correctly.

Perform basic data exploration, e.g., to check whether there are missing values in the dataset. If there are, remove the records with the missing values.

Estimated run time: between 1 and 2 minutes.

Step 2: Use the last column "legitimate" as labels for training a classifier. The assigned values in the column are 0 for benign files, and 1 for malware files. Use the remaining 53 columns as features for training a classification model.

Split the data into a training dataset (80%) and test dataset (20%).

Scale the data into the range [0, 1], e.g., by using the Scalar functions in scikit-learn.

Estimated run time: between 1 and 2 minutes.

Step 3: Train the following ensemble models for classification of the records: Random Forest, XGBoost, LightGBM, and CatBoost. You can import Random Forest from scikit-learn, and you need to install the other models (e.g., by using pip install). For each model, report the classification accuracy on the test dataset, and display the confusion matrix. For full marks, it is expected to obtain test accuracy greater than 98%. You should be able to achieve it without any finetuning.

Estimated run time: between 5 and 15 minutes.

Step 4: Train a deep-learning model for classification of the records into benign and malicious PE files. You can use several fully-connected layers and dropout layers, or you can consider other types of layers if you wish. Report the classification accuracy on the test dataset, plot the learning curves, and display the confusion matrix. For full marks, it is expected to obtain test accuracy greater than 97%.

Discuss the results, and compare the performance of the deep learning model and the ensemble models from Step 3.

Were you able to exceed the accuracy of the conventional and ensemble ML models by the deep learning models in Parts 1 and 2 of this assignment? If not, elaborate why and whether you could have done anything else to improve the performance of the deep learning models.

Estimated run time: between 5 and 20 minutes.

Step 5: Implement a white-box FGSM evasion attack against the deep learning model from Step 4. For creating the adversarial samples, use a random set of 1,000 records from the test dataset. Apply the following perturbation magnitudes: ϵ= [0.01, 0.025, 0.05, 0.1, 0.2, 0.3].

Report the classification accuracy and the average perturbation for the different levels of perturbation.

Estimated run time: between 1 and 2 minutes.

Step 6: Implement a white-box Carlini & Wagner ℓ_2 and ℓ_∞ attacks against the deep learning model from Step 5. For creating the attacks, use a random set of 100 records from the test dataset.

Report the classification accuracy and the average perturbation.

Estimated run time: between 5 and 10 minutes.

Step 7: Implement the Zoo evasion attack against the Random Forest model. The following notebook in the ART library provides an example of how to implement the attack with a Random Forest model. For creating the attack, use a random set of 20 records from the test dataset.

Report the classification accuracy. Elaborate on the difference between the results of the FGSM and C&W attacks.

Estimated run time: between 15 and 30 minutes.

Submission documents

All notebooks and a brief report with tables and results (either in MS Word/PDF or inserted inline in the Jupyter notebooks).