NETWORK SECURITY

Project

KOCKAN Melisa / CANIN Raphaël / BIBILA Jodry / BACHA Eliès

E4 AIC

Introduction

The purpose of this lab is to use machine learning approaches in order to detect intrusion on the use case of IoT.

To do so, the dataset we are going to use is a wireshark capture which depicts the traffic of a real network receiving attacks.

The dataset at stake is available using this link : click here. In this dataset, each line corresponds to an element from the traffic received.

After identifying the different types of attacks in our dataset, we will analyze how they can impact an IoT system in order to select the relevant features enabling us to predict if a line in the traffic corresponds to an attack.

Identification of the attacks

After exploring our dataset on our ipynb file, we identified 4 different types of attacks :

- Mirai
- DoS
- Scan
- Spoofing

According to the definition of Oracle, the Internet of Things (IoT) refers to « the network of physical objects—“things”—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet ».

To understand how these 4 attacks can affect an IoT system, let’s define each of them beforehand.

- Mirai
  Mirai is a malware looking for IoT systems using default usernames and passwords in order to infect them

- DoS
  Fact of sending indefinitely requests to a service in order to make it unavailable

- Scan
  Analyzing ports to establish a list of open ports to attack

- Spoofing
  Attackers disguise into well known and safe company in order to retrieve informations

Mirai does not only make IoT devices unavailable ; it leads to the creation of BotNet, which can lead to launching a DoS attack. By making an IoT device unavailable, DoS will indirectly make unavailable any other IoT devices depending on them. Scan permits to pave a path to spoofing

Using a machine learning model

Using python, We train a machine learning model upon these data and evaluate its performances. We used several algorithms to compare them and choose the one with the highest performance

- KNN (93.7% accuracy)
- Logistic regression (79.1%)
- Linear discriminant analysis (76.8%)
- Support vector machine (79.3%)
- Trees (96.6%)
- Random forest (94.2%)
- A neural network for classification (84.0%)
- Gradient Boosting (97.2%)
- **Stacking classifier (97.6%)**

Please find our *Project_Network_security* notebook to see the result.

❖ Explanation of the different steps

First, after importing the dataset, we need to take a very important step for the results :

- **PREPROCESSING**

Preprocessing is an important step in Machine Learning because it strongly impacts the results of our model.

- In datasets, we rarely have a clean dataset. The first step in preprocessing is often to deal with null values. Indeed, these zero/nan values do not allow us to train our model correctly.

- In this dataset we have different types of data. However, to train a model we need only numeric features. So, we have to search for the different categorical features that we have and convert them into numeric features. The categoricals datas are identified with the type Object. We therefore isolate this type of data. The LabelEncoder method allows each categorical data to be coded into numerical labels. Then, we convert each integer into one hot encoded (i.e. one binary number) hence the method name one-hot encoded. In this dataset, the only categorical datas is the column CLASS.

- To learn more about the dataset, we will also use another tool: the correlation matrix. This tool allows you to find the possible dependencies between the variables of the dataset.

➢ **TRAINING AND TESTING DIFFERENT MODELS**

First, we divide the dataset into two parts: one for the training and the other for the test.

We implement a function that calculates the average precision over n different training sessions for a given model as parameter.

Then, we calculate the accuracy for different algorithms on normal data and normalized data.

Relevance of each features

Sometimes, the columns of a dataset aren't all relevant to take into account for our analysis. Before studying each column individually, we printed the matrix correlation of our data set.

On this matrix correlation, we had no surprise noticing that some columns such as 'Forward_Packets_Length_Max', 'Forward_Packets_Length_Min' and 'Forward_Packets_Length_Mean' were linked together.

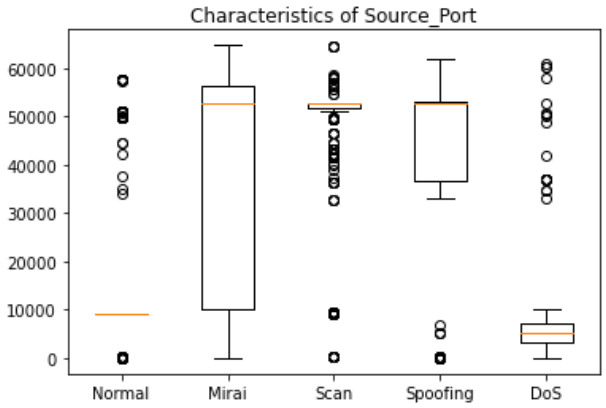
To take a further look at each column, we draw a box plot for each of them owing to a function we implemented during our Machine Learning project.

In the 32 columns of our dataset , we highlighted in green those who seemed the most meaningful to distinguish our attacks :

'Source_Port', 'Destination_Port', 'Protocol', 'Flow_Duration',
'Total_Forward_Packets', 'Total_Backward_Packets',
'Total_Length_Forward_Packets', 'Total_Length_Backward_Packets',
'Forward_Packets_Length_Max', 'Forward_Packets_Length_Min',
'Forward_Packets_Length_Mean', 'Backward_Packets_Length_Max',
'Backward_Packets_Length_Min', 'Backward_Packets_Length_Mean',
'Packets_Flow_IAT_Mean', 'Packets_Flow_IAT_Standard_Deviation',
'Packets_Flow_IAT_Max', 'Packets_Flow_IAT_Min', 'Forward_IAT_Total',
'Forward_IAT_Mean', 'Forward_IAT_Min', 'Backward_IAT_Total',
'Backward_IAT_Mean', 'Backward_IAT_Min', 'Forward_Header_Length',
'Backward_Header_Length', 'Forward_Packets_per_Second',
'Backward_Packets_per_Second', 'Packets_Length_Min',
'Packets_Length_Max', 'Packets_Length_Mean',
'Packets_Length_Standard_Deviation',

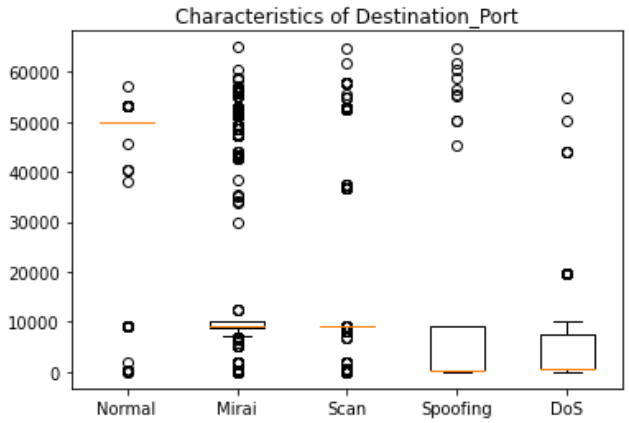
Let’s take a further look into these most impactful columns individually.

‘Source Port’



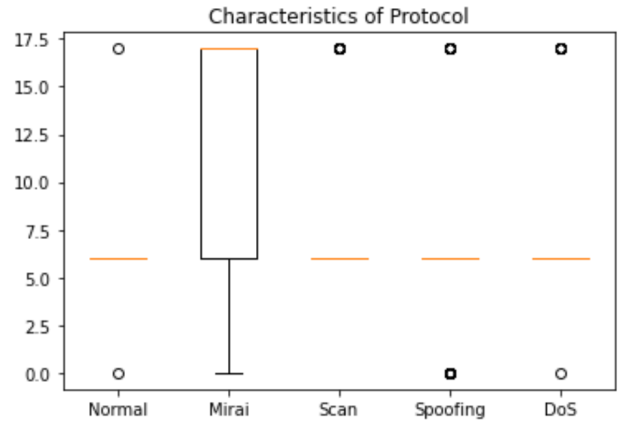
This column is one of the more useful to detect each attack. As we can see, each type of attack emits from a pretty different segment of ports.

‘Destination Port’



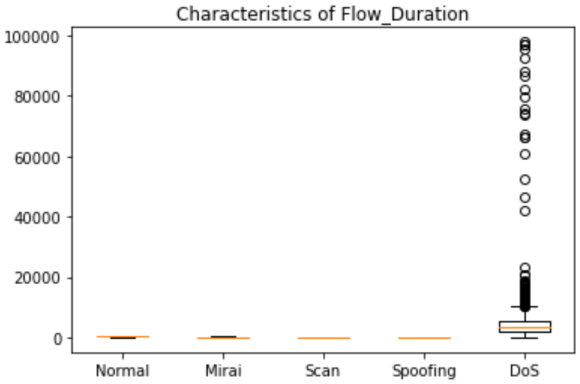
Like the previous column, this one has a strong influence on the accuracy of predictions.

'Protocol'



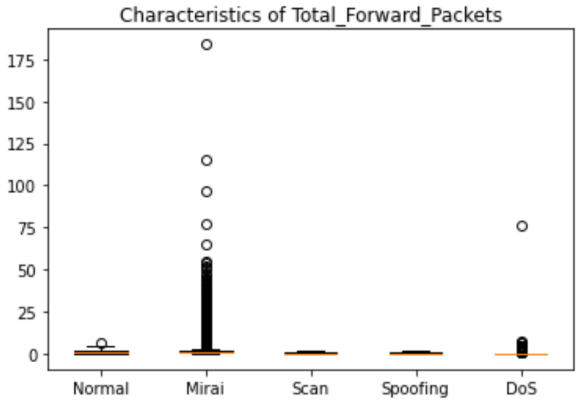
This parameter is relevant to detect a Mirai attack. Mirai is a malware, compared to the three other attacks ; hence, it seems coherent to say that the protocol used by this attack is suspicious.

'Flow_Duration'



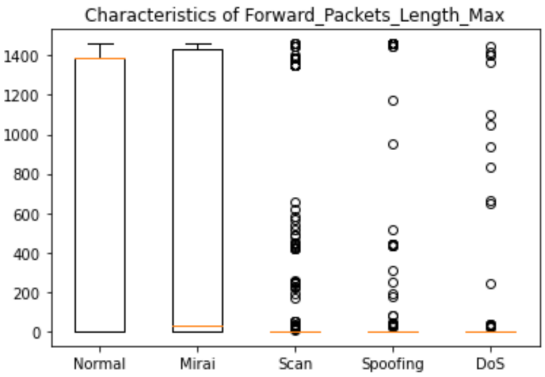
This parameter is characteristic of DoS. During DoS, requests are sent succinctly and indefinitely ; hence, the flow duration is drastically affected.

'Total_Forward_Packets'



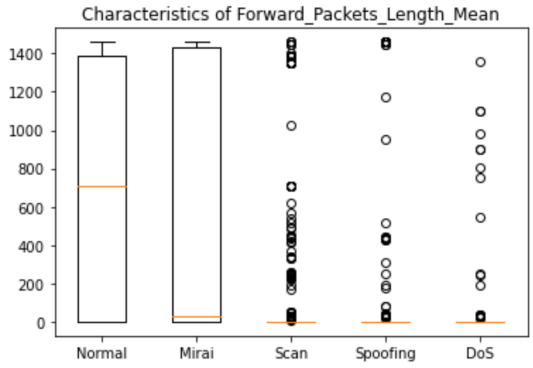
This parameter is relevant to detect Mirai attacks.

'Forward_Packets_Length_Max'



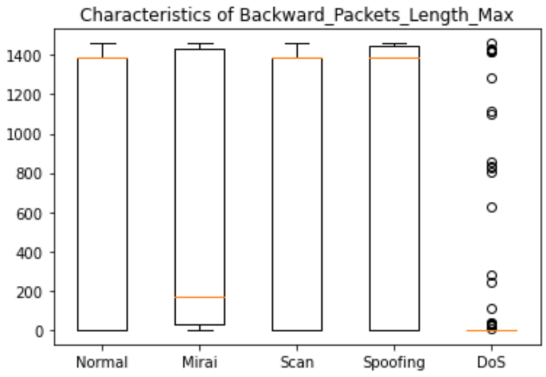
This parameter enables us to detect the fourth of our attacks.

'Forward_Packets_Length_Mean'



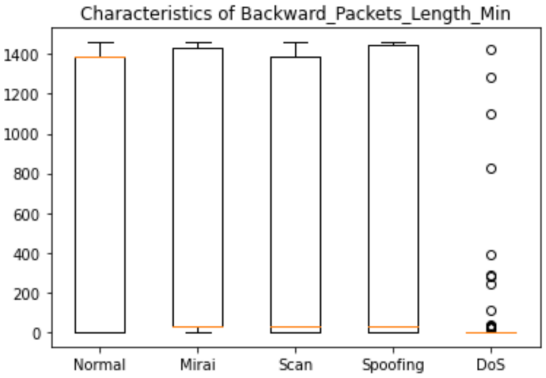
This parameter enables us to detect the fourth of our attacks.

'Backward_Packets_Length_Max'



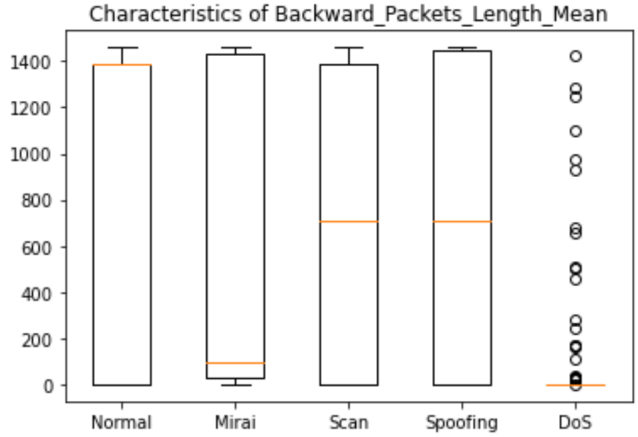
This parameter helps to detect Mirai and DoS.

'Backward_Packets_Length_Min',



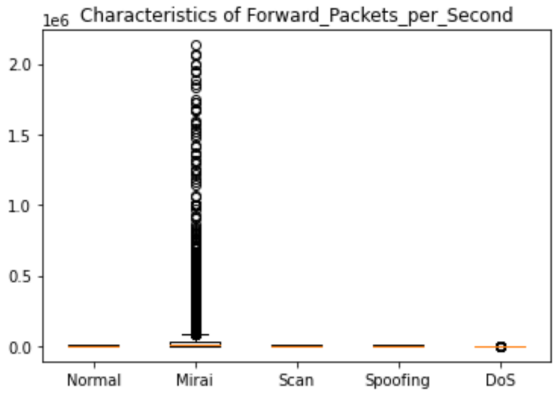
This parameter enables us to detect the fourth of our attacks.

'Backward_Packets_Length_Mean'



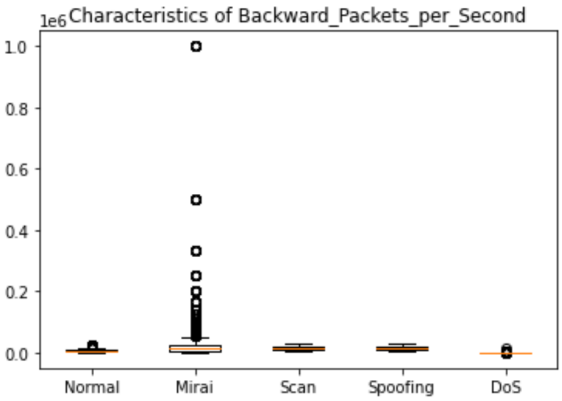
This parameter enables us to detect the fourth of our attacks, but especially for Mirai and DoS.

'Forward_Packets_per_Second'



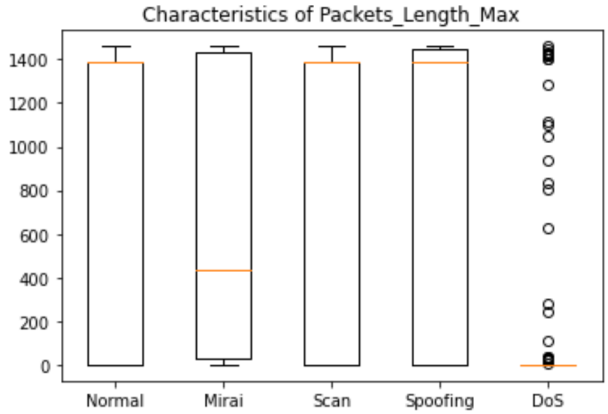
This parameter is meaningful for Mirai attacks.

'Backward_Packets_per_Second'



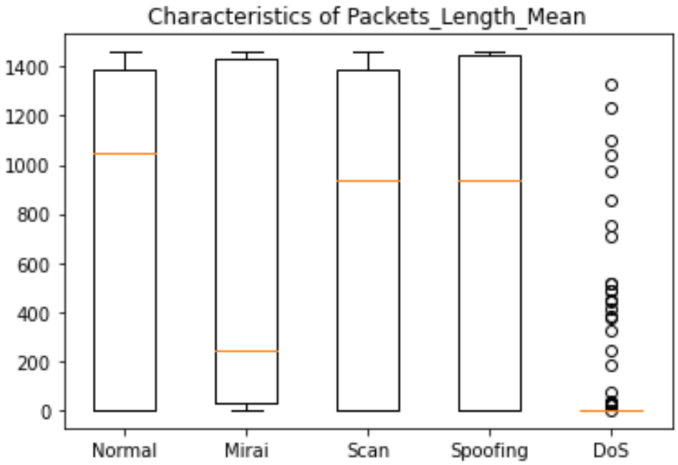
This parameter is meaningful for Mirai attacks.

'Packets_Length_Max'



This parameter is meaningful for Mirai and DoS attacks.

'Packets_Length_Mean'



This parameter is meaningful for Mirai and DoS attacks.

We reduced our dataset to these columns, and retrained our models.

Finally we obtained the following performance:

- KNN (93.7% accuracy)
- Logistic regression (77.97%)
- Linear discriminant analysis (75.0%)
- Support vector machine (79.06%)
- Trees (97.17%)
- Random forest (95.8%)
- A neural network for classification (83.36%)
- Gradient Boosting (97.27%)
- **Stacking classifier (97.73%)**

The accuracy is a little better than before reducing our dataset. We can also point out the fact that reducing the dataset enabled us to reduce the calculation time.