People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Abdelhamid MEHRI - Constantine 2 -

Faculty of New Technologies of Information and Communication
Department of Software Technologies and Inforamtion Systems

Master Software Engineering

Statistics and data analysis project

Theme

Network Intrusion Detection System (NIDS) using Machine Learning

Realiezd by :

- BENDJAMAA Heitham
- BENABDESSADOK Safa

Directed by :

- Dr MARIR Naila

2022/2023

1 Introduction

Always there is a motivation that motivates hackers to impinge the security system of others. That is what made security a critical aspect of companies, governments, and even civilized people. According to the power of artificial intelligence and machine learning, hackers use it to do their malicious behavior. So, improving and enhancing intrusion detection systems (IDS) is a need. Therefore, we exploit the strength of artificial intelligence and implement a network intrusion detector model based on statistical analysis and machine learning algorithms.

2 Data Description

In this project, we use CICIDS2017 dataset provided by Canadian Institute of Cybersecurity released in 2017. Which was collected during 5 days from 9 a.m., Monday, July 3rd, 2017 to 5 p.m., Friday, July 7, 2017. The dataset is represented in CSV files.

The CICIDS2017 dataset consists TrafficLabelling CSV files, which include full packet payloads in pcap format, the corresponding profiles, and the labeled flows.

Also, CSV files named MachineLearningCVE, which include full packet payloads in pcap format and the labeled flows. We choose MachineLearningCVE files because TrafficLabelling contains features ”Flow ID, Source IP, Source Port, Destination IP, Protocol, Timestamp”, these features are not good to be considered in a ML model because:
- **Flow ID** contains repeated information which is **Destination IP**, **Source IP**, **Destination Port**, **Source Port**, and **Protocol**.
- **Source IP**, **Source Port**, and **Destination IP** are changing continuously.
- **Protocol** can be deduced from **Destination Port**.

This dataset contains benign cases and several attacks (Brute Force, Heartbleed, Web attacks, Infiltration, DoS, and DDoS) represented in a way very close to the true real-world data (PCAPs).

1. **Monday-WorkingHours.pcap_ISCX.csv** contains 529918 benign cases, which are Normal human activities.

2. **Tuesday-WorkingHours.pcap_ISCX.csv** contains 445909 data flow examples divided into 432074 benign, 7938 FTP-Patator, and 5897 SSH-Patator.

3. **Thursday-WorkingHours-MorningWebAttacks.pcap_ISCX.csv** contains 170366 data flow examples divided into 168186 benign, 1507 Web Attack (Brute Force), 652 Web Attack (XSS), and 21 Web Attack (SQL Injection).

4. **Thursday-WorkingHours-Afternoon Infilteration.pcap_ISCX.csv**, the data file contains 288566 benign cases and 36 Infiltration where the total number of cases is 288602.

5. **Wednesday-workingHours.pcap_ISCX.csv**, record Data representation contains 440031 BENIGN, 231073 DoS Hulk, 10293 DoS GoldenEye, 5796 DoS slowloris, 5499 DoS Slowhttptest, and 11 Heartbleed, where the total number is 692703 data flow cases.

6. **Friday-WorkingHours-Morning.pcap_ISCX.csv** record Data representation contains 189067 BENIGN cases and 1966 Bot.

7. the **Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv** file is with 286467 instances representing 158930 PortScan attack, and 127537 benign.

8. and the **Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv** file is with 225745 instances representing 128027 DDoS attack, and 97718 benign.

Thus, all recorded traffic data for each day contain both benign and anomaly cases.

To use this data in our statistical and data analysis project, we need data merged into a single file. So, we concatenated these dataset files and save them in a single file (Machine-LearningCVE.csv) containing all recorded data. Then, we read out the following information:

- The total number of instances is **2830743** and **78** features.
- Fifteen **(15)** distinct class Labels, **one** is normal and **14** are attack types.
- The following table describes the number of instances for each attacks type.

BENIGN	2273097					
DoS Hulk	**PortScan**	**DDoS**	**DoS GoldenEye**	**FTP-Patator**	**SSH-Patator**	**DoS slowloris**
231073	158930	128027	10293	7938	5897	5796
DoS Slowhttptest	**Bot**	**Web Attack (Brute Force)**	**Web Attack (XSS)**	**Infiltration**	**Web Attack (SQL Injection)**	**Heartbleed**
5499	1966	1507	652	36	21	11

We have **557646** anomaly cases, while **2273097** normal cases.

3 Statistical analysis

Statistical analysis is the collection and interpretation of data, in order to understand it and root out useful information.

3.1 Analyzing and cleaning data

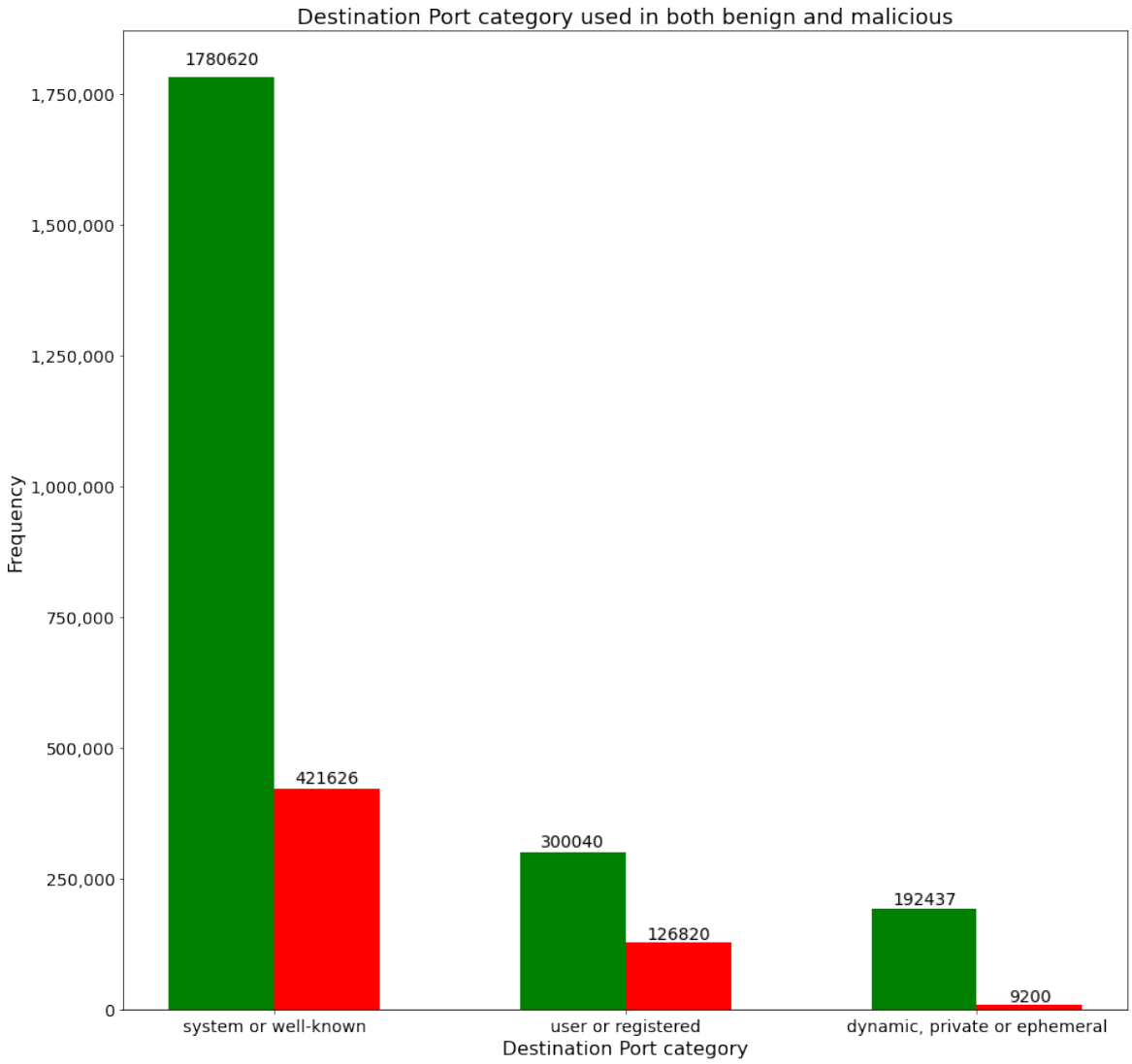
We observed that there is a duplicated column named ”FwdHeaderLength”, and there are **1358** missing values in **Flow Bytes/s** column. Also, there are **308381** duplicated instances.

Applying the *describe*() python method of our data show that there are **8** columns with value does not vary. It has 0 for all instances. Moreover, We remark that **Flow Bytes/s** and **Flow Packets/s** columns have instances with the **infinity** value.

Otherwise, In networking Ports are grouped to 3 categories as follows :
- Ports with numbers in range of **0 to 1023** are named system or well-known ports,
- Ports with numbers in range of **1024 to 491513** are called user or registered ports,
- and, ports with numbers in range of **49152 to 65535** are called dynamic, private, or ephemeral ports.

We checked if there is an unnormalized destination port, which means out of port ranges to discover fake data, and try to visualize the most used ports for normal and malicious activities.

From the following histogram, we observed that system or well-known ports are the most used port for normal human activities and attackers’ activities.



Furthermore, the correlation is a statistical measure to notice the dependence between features is causal or not, and has a negative or positive correlation. Audition of this measure shows that:

- The correlation of these features **Bwd PSH Flags**, **Bwd URG Flags**, **Fwd Avg Bytes/bulk**, **Fwd Avg Packets/Bulk**, **Fwd Avg Bulk Rate**, **Bwd Avg Bytes/Bulk**, **Bwd Avg Packets/Bulk**, and **Bwd Avg Bulk Rate** with other features is undetermined.
- There are features that have a higher correlation (more than 0.95) which means are more linearly dependent. Figure 1 is a visualization of these features.

3.2 Preparing data

Accordingly, as motioned that there are two columns named ”FwdHeaderLength” and have the same values, these two features say the same thing. So, the machine learning model won’t learn any functional thing by keeping both of them in features in training. Further, features with invariant values ”0” for all instances have no effect on the result, moreover they are the same features that can’t determine their correlation. So, dropping these columns is better.

As well, missing values affect the accuracy of machine learning model and disadvantage

the results. In our case, we remove the rows with missing values in **Flow Bytes/s** column which represent only 0.047% of the whole data. Also, removing the duplicated instances because it will overfit the machine learning model. Besides, we dropped 1211 instances that have infinity values. where keeping them acts negatively on our machine learning algorithm.

Since using all of the highly correlated features is a redundancy, we choose one of them as a feature for training our machine learning model. So, we keep *"Bwd IAT Max"* and remove *"Bwd IAT Std"* because they are highly correlated to each other, we keep *"Fwd PSH Flags"* and remove *"SYN Flag Count"* because they are highly correlated to each other, we keep *"Avg Fwd Segment Size"* and remove both *"Fwd Packet Length Max"* and *"Fwd Packet Length Std"* because they are highly correlated to each other, we keep *"Subflow Bwd Bytes"*, and remove all of *"Subflow Bwd Packets"*, *"Subflow Fwd Packets"*, *"Total Backward Packets"*, *"Total Fwd Packets"*, *"Total Length of Bwd Packets"*, and *"act_data_pkt_fwd"* because they are highly correlated to each other, we keep *"ECE Flag Count"* and remove *"RST Flag Count"* because they are highly correlated to each other, we keep *"CWE Flag Count"* and remove *"Fwd URG Flags"* because they are highly correlated to each other, we remove *"Idle Max"* because it is highly correlated with *"Idle Mean"*.

The new data have 56 features that have a numerical data type, that's means no need for data transformation, and one categorical target feature "Lable". And we have 2520798 instances.

The new distribution of data is represented in the following Table:

BENIGN	2095057					
DoS Hulk	**PortScan**	**DDoS**	**DoS GoldenEye**	**FTP-Patator**	**SSH-Patator**	**DoS slowloris**
172846	90694	128014	10286	5931	3219	5385
DoS Slowhttptest	**Bot**	**Web Attack (Brute Force)**	**Web Attack (XSS)**	**Infiltration**	**Web Attack (SQL Injection)**	**Heartbleed**
5228	1948	1470	652	36	21	11

As part of the preparation of data for machine learning, there is the normalization of data which is a technique used to be applied to change the values of numeric columns to use a common scale without distorting differences in the ranges of values or losing information. In our case, we used **z-score Scaling**.

Generally in machine learning, data is split into two or three sets of data. For our own part, we took 20% of the original data as a testing dataset and left 80% for the training dataset. The attack types were distributed as follows:

Category	Number of instances in training dataset	Number of instances in testing dataset
BENIGN	1676045	419012
DoS Hulk	138276	34570
DDoS	102411	25603
PortScan	72555	18139
DoS GoldenEye	8228	2058
FTP-Patator	4744	1187
DoS slowloris	4308	1077
DoS Slowhttptest	4182	1046
SSH-Patator	2575	644
Bot	1558	390
Web Attack (Brute Force)	1176	294
Web Attack (XSS)	521	131
Infiltration	28	8
Web Attack (SQL Injection)	16	5
Heartbleed	8	3
Total	2016631	504167

4 Model

In our own training dataset, we have more than 50 samples and we have to predict a category, also our dataset is labeled and containe over and above 100K samples. Thus, the Linear SVC is extremely a good choice to train our model.

Linear SVC alos know as Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM is more effective in high dimensional spaces, and we have 56 feature so 56 dimension. also, SVM is relatively memory efficient.

If the chosen model don't give a good result, we will work with K-nearest-neighbor (KNN) algorithm or Naive Bayes (NB).

5 Results obtained

After training our SVM model we got an accuracy of 0.97, and, this is an excellent outcome. Where, from its consequence confusion matrix, we can deduce that the trained algorithm distinguishes 412212 benign cases, which is equivalent to 98.38% of benign test cases. This is an excellent result, and, 6929 attack cases, which means that the model could distinguish 91.86% of all attack categories.

These results are excellent for a network intrusion detection system (NIDS) to detect malicious behavior. However, the cause of not detecting some attack types is that we have a low number of samples in the dataset.

As a result, our trained model can be deployed as a Network Intrusion Detection System.

6 Conclusion

As a result of what we went through from the previous stages, and what we learned from this module, we want to clarify what our targets were and what the results we had in store.

As a starting point, after we got our data, we learned that good analysis and description of data is the most important stage, because, it allows an understanding and clarification of the relation between the data information (we used several Python libraries to visualize the data and it analyzes) and from it the identification of the main objective of the work.

The next stage teaches us how to make a decision in filtering data and choosing the best links that help in raising production efficiencies.

And before concluding, we learned about the various algorithms that may help us reach our goal, and what is the difference between them, especially in the field of classification (example of algos: SVM used in the project, KNN our alternative proposal algorithm, BN).

In conclusion, as a result of the good progress of the previous stages, we got a high accuracy of **0.97**. Also in the future, we intend to expose the project to a deep learning system.

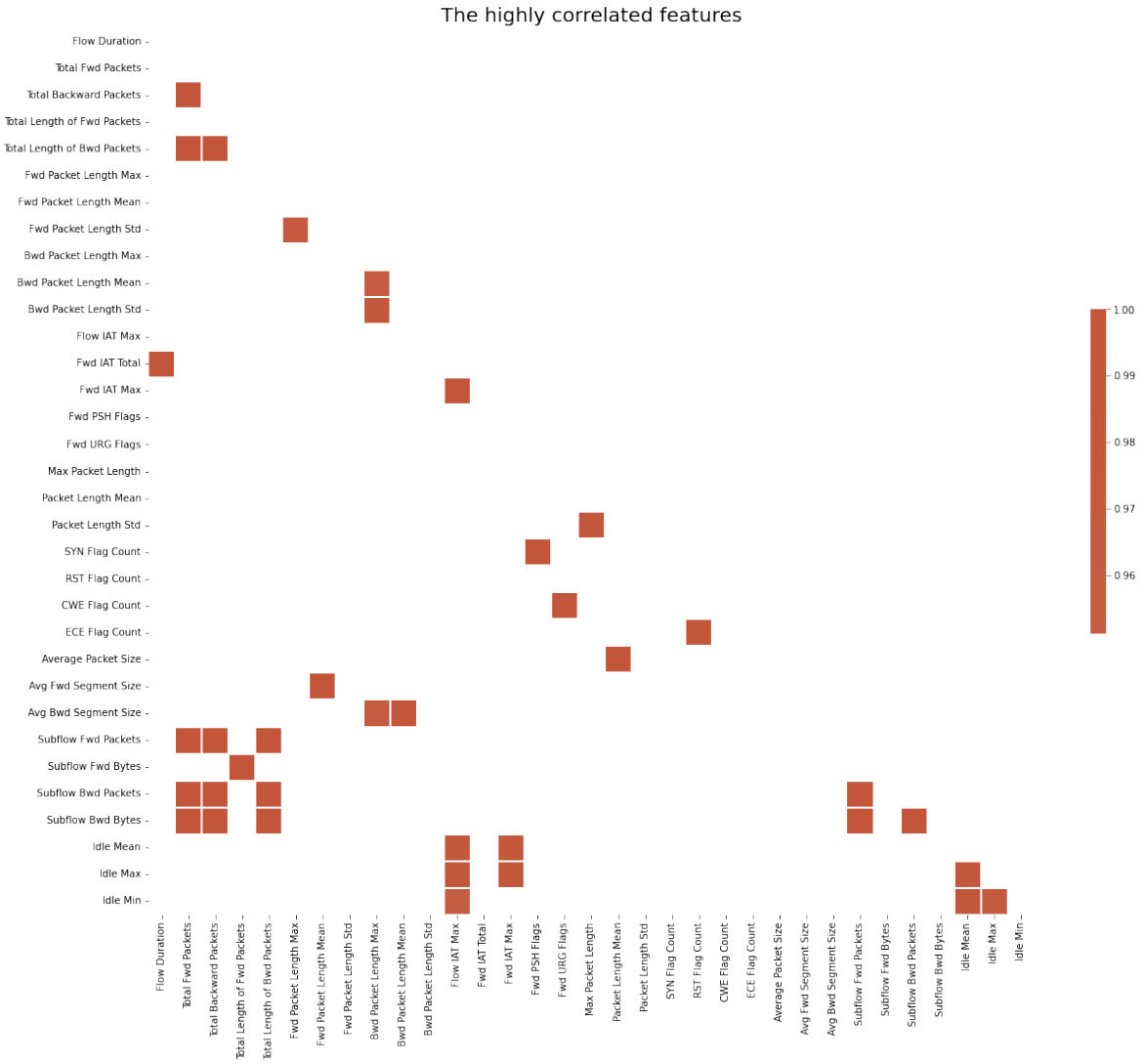


Figure 1: The highly correlated features