T.C.

ALTINBAS UNIVERSITY

Electrical and Computer Engineering

COMPARATIVE STUDY OF INTRUSION DETECTION SYSTEM USING MACHINE LEARNING

Mahmood Imad Abdulkareem

Master Thesis

Supervisor:
Prof. Dr. Osman Nuri UÇAN

Istanbul, 2019

COMPARATIVE STUDY OF INTRUSION DETECTION SYSTEM USING MACHINE LEARNING

by

Mahmood Imad Abdulkareem

Department

Electrical and Computer Engineering

Submitted to the Graduate School of Science and Engineering

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2019

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Prof. Dr. Osman Nuri UCAN

Supervisor

Examining Committee Members (the first name belongs to the chairperson of the jury and the second name belongs to supervisor)

Prof. Dr. Osman Nuri UCAN	School of Engineering and Natural Sciences, Altinbas University	_____
Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	School of Engineering and Natural Sciences, Altinbas University	_____
Asst. Prof. Dr. Adil Deniz DURU	Faculty of Sport Sciences Marmara University	_____

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Asst. Prof. Dr. Cagatay AYDIN

Head of Department

Approval Date of Graduate School of Science and Engineering: ____/____/____

_____

Prof. Dr. Oguz BAYAT

Director

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Mahmood Imad Abdulkareem

DEDICATION

To my mother and father the fountain of patience and optimism and hope who were in existence after Allah and His Messenger , the supporters and my strength and sanctuary after God , those who raised me on themselves and who taught me the science of life, to those who showed me what is more beautiful than the life of my brothers and my friends.

ABSTRACT

COMPARATIVE STUDY OF INTRUSION DETECTION SYSTEM USING MACHINE LEARNING

Abdulkareem, Mahmood Imad Abdulkareem,

M.S, Electrical and Computer Engineering, Altınbaş University,

Supervisor: Prof. Dr. Osman Nuri UCAN

Date: 11/2019

Pages: 59

Users and hundreds of foundation communicate over the internet so in order to secure these communications we need to detect any abnormally Network traffic may refer to an unauthorized intrusion in the network ,That's anomalies detection led to prevent network security ,For us as researchers that’s a sensitive and dynamic research area specially these days detecting abnormal network traffic based on automated learning classification techniques, (IDSs) is the most powerful security tools against the unauthorized login and network attacks , because the real need for a suitable size of test and validation datasets, abnormally intrusion detection approaches are tribulation from consistent and accurate performance evolutions, most of the other datasets suffer from the lack of traffic and sizes, and it may not contain all attack types ,to analyze the standing of traffic performance and take a look at the aptitude of traffic analysis, detection of malicious attacks that come back through high load traffic, in this thesis, we presents machine learning and neural network models to get the best accuracy for multi-classification most researchers are trying to emerging science that is implemented in many different technologies and in many ways with multi binary classification .so so we have to look into how machine learning could be used in network security to do the multi-classification or a better job than solutions available today. For this idea to be further developed there has to be some proofs-of-

concept or studies, which indicates that this idea is possible to develop. the aim to detect abnormal traffic detection using machine learning and neural network models. The algorithms used (Logistic Regression, Random Forest, QDA, SGD, and ANN) and the implementation are done by using a data set to train and test the accuracy. The data set used in this thesis is Canadian Institute for Cybersecurity datasets, named CICIDS2017 which has updated in 2018, so in we are trying to compare between different multi and binary classification technique for random set of data, most automatic learning techniques provide an accuracy higher than 90% , However the Random Forest model achieved the best performance accuracy among other models.

Keywords*:* Machine Learning, Security, Dataset.

TABLE OF CONTENTS

Pages

ABSTRACT ..... **vi**

LIST OF FIGURES ..... **xiii**

LIST OF ABBREVIATIONS ..... **xv**

1. INTRODUCTION **1**

1.1 MOTIVATION 1

1.2 PROBLEM DEFINITIONS 3

1.3 CONTRIBUTION OF THESIS 3

1.4 THESIS ORGANIZATION 3

2. LITERATURE REVIEW **4**

3. METHODOLOGY **6**

3.1 ANOMALY DETECTION 6

3.2 DATASET 6

3.3 MACHINE LEARNING(ML) 8

3.4 MACHINE LEARNING METHODS 8

3.4.1 Supervised Machine Learning 8

3.4.2 Unsupervised Machine Learning 8

3.5 FEATURE SELECTION TYPES 9

3.5.1 Filter BASE 9

3.5.2 Wrapper Based 9

3.5.3 Embedded 9

3.6 ALGORITHMS 9

3.6.1 Random Forest 10

3.6.2 Logistic Regression 11

3.6.3 QDA (Quadratic Discriminant Analusis) 12

3.6.4 SGD (Stochastic Gradient Descent) 12

3.7 ANN (ARTIFICIAL NEURAL NETWORK) 12
3.8 EVALUATION METRICS 13
3.8.1 Accuracy 14
3.8.2 Recall (the sensitivity) 14
3.8.3 Precision 14
3.8.4 F-measure: 14
3.9 ATTACKS 15
3.9.1 DDOS and DOS Hulk 17
3.9.2 TCP-SYN Flood 18
3.9.3 HTTP-GET Flood 19
3.9.4 Portscan 20
3.9.5 DOS Golden Eye 20
3.9.6 FTP-Patator 21
3.9.7 SSH- Patator 21
3.9.8 DOS Slowris 21
3.9.9 DoS SlowHTTPTEST 22
3.9.10 Botnet 22
3.9.11 Web Attack 23
3.9.12 Infetration 23
3.9.13 Heartbleed 24
3.10 PLATFORM 25
4. IMPEMENTATION 28
4.1 DATA SELECTING AND CLEANING 29
4.2 PREPROCESSING DATA 29
4.3 CREATION OF TRAINING AND TEST DATA 30
4.4 FEATURE SELECTION 31
4.5 IMPLEMENTATION OF MACHINE LEARNING ALGORITHMS 33

5. RESULTS AND DISCUSSION 35

5.1 FEEDING DATASET TO THE MODEL. 35

5.1.1 Model 1 35

5.1.2 Model 2 38

5.2 EVALUATION 44

5.3 CONCLUSION AND FUTURE WORK 46

5.3.1 Conclusion 46

5.3.2 Future Work 46

REFERENCES 48

APPENDIX A 55

LIST OF TABLES

Pages

Table 3.1: CICIDS 2017 (Intrusion Detection Evaluation Dataset) files description..... 7

Table 3.2: BENIGN and attack types in the select 10% of data 16

Table 3.3: Bening, DDos, port scan, and Dos attack types in the dataset..... 17

Table 3.4: Bening, DDos, port scan, and Dos attack types after append all Dos attack types together 17

Table 4.1: Features Importance Weights For the Best 20 Features using an EMBEDDED method 32

Table 4.2: Best 10 Features By Using Wrapper-Based Method..... 33

Table 5.1: Accuracy of RF and ANN for 10% of the dataset that contain all attacks 35

Table 5.2: 10% of 8 obtained segmented files of the dataset with best 10 selected features, using Random Forest for All Attacks..... 36

Table 5.3: 10% of 8 obtained segmented files of the dataset with best 10 selected features confusion matrix , using Random Forest for All Attacks..... 36

Table 5.4: 10% of 8 obtained segmented files of the dataset with best 10 selected features, using ANN for All Attacks 37

Table 5.5: 10% of 8 obtained segmented files of the dataset confusion matrix using ANN for All Attacks 37

Table 5.6: The Score..... 38

Table 5.7: Precision 38

Table 5.8: Recall..... 39

Table 5.9: F1-score 39

Table 5.10: Compare this study results with Previous[55] 44

Table 5.11: Compare this Random Forest DoS Attack 45

Table 5.12: Compare this ANN of our study results with Previous[7] 45

LIST OF FIGURES

Pages

Figure 1.1: The Increasing Number of Internet Users (1995-2017) [1]. 1

Figure 1.2: Explain The Process Of Analyzing On The Real Traffic Data[4]. 2

Figure 3.1: Random Forest Classifier. 10

Figure 3.2: How gradient descent minimizes the cost function[19] 11

Figure 3.3: Neural network with two-Dense- Hidden Layer 13

Figure 3.4: Confusion matrix 15

Figure 3.5: Attack and Benign Percent 16

Figure 3.6: DDoS Attack 18

Figure 3.7: TCP-SYN FLOOD Attack with normal 3-way handshake 19

Figure 3.8: Single and multiple HTTP GET Flood Attack 20

Figure 3.9: Anaconda Editor 25

Figure 3.10: Scikit-Learn Library 26

Figure 3.11: Scikit-Learn Library 26

Figure 4.1: The Implementation Process 28

Figure 4.1: Correlation of the best 20 features using an EMBEDDED method 31

Figure 4.2: The correlation of the best 10 features using Wrapper-based method 32

Figure 5.1: The Implementation of the confusion matrix for All Dos Attack 40

Figure 5.2: The Implementation of the confusion matrix for All DDos Attack 41

Figure 5.3: The Implementation of the confusion matrix for PortScan Attack 42

Figure 5.4: The Implementation of the confusion matrix for 43

LIST OF ABBREVIATIONS

ML	:	Machine Learning
ANN	:	Artificial Neural Network
RF	:	Random Forest
LR	:	Logistic Regression
QDA	:	Quadratic Discriminant Analysis
SGD	:	Stochastic Gradient Descent
IDS		Intrusion Detection System

1. INTRODUCTION

1.1 MOTIVATION

In the past two decades, computer networks are rapidly getting bigger with more various traffic activates and many companies are using many information systems like distributed online cloud storage systems and authentication login systems [1].

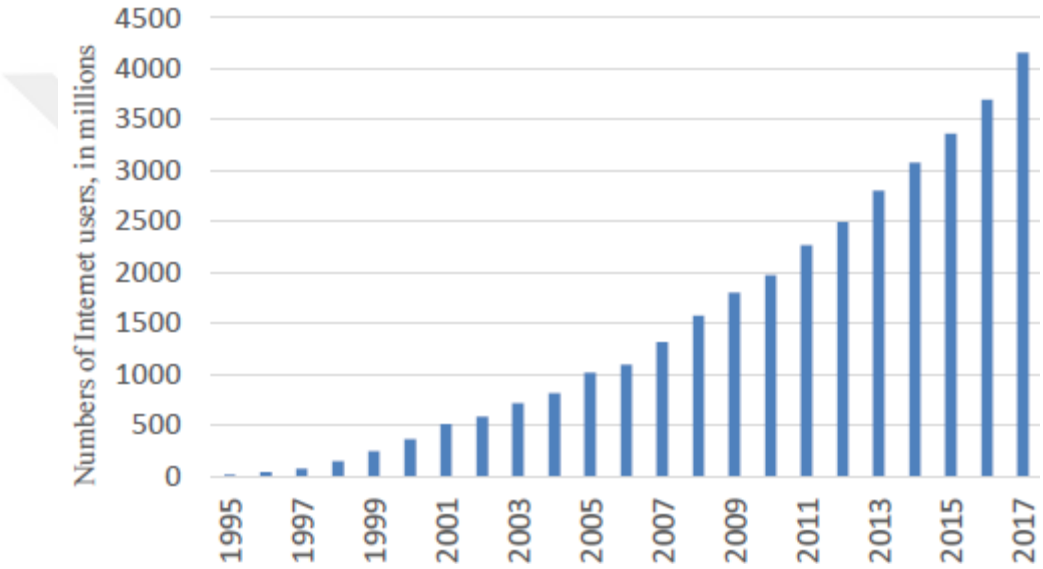

Figure 1.1: The Increasing Number of Internet Users (1995-2017) [1].

Coincides with this huge increasing a large numbers of attacks and it becomes impossible to detect intrusion of abnormal traffic on these networks so it's unuseful to use classic security tools like user authentication access and firewalls, therefore, to increase security for computer networks there is a need for a network intrusion detection system to detect attacks against computer networks and systems automatically so building models with supervised and unsupervised techniques [2] are the best ways , IDS technology provide effectively and success as new technologies detect attacks against computer networks [3].

We will build our models that detect abnormal automatically, the aim of detection anomaly traffic by algorithms (the models) are their ability to detect unseen attacks because their ability to detect new or unusual attacks by learning from examples.

The two main categories of detection techniques, supervised and unsupervised, In supervised learning given a set of normal data to train on and given a new set of test data, building a model to determine whether the test data is ‘normal’ or anomalous for the binary classification as an example, unsupervised learning attempts to detect anomalous behavior without using any knowledge about the training data, Unsupervised anomaly detection approaches are based on statistical approaches, clustering, outlier detection schemes, state machines, etc.

Figure 1.2 explains the process of analyzing the real traffic data by using the system, Input the data collected using a data capturing unit (CICFlowMeter)[4]. By capturing packet header information , build one-way sessions (flows), The data are stored in files and the data is fed into the module but before that will be filtered from unwanted data and missing, For example source and destination physical address MAC. The first step in the model is extracting unwanted features that are used in the data mining analysis after that the selected like source and destination ports, protocol, flags, number of bytes, These features are especially useful in separating sources of high volume connections per unit time from the rest of the traffic such as fast scanning activities.

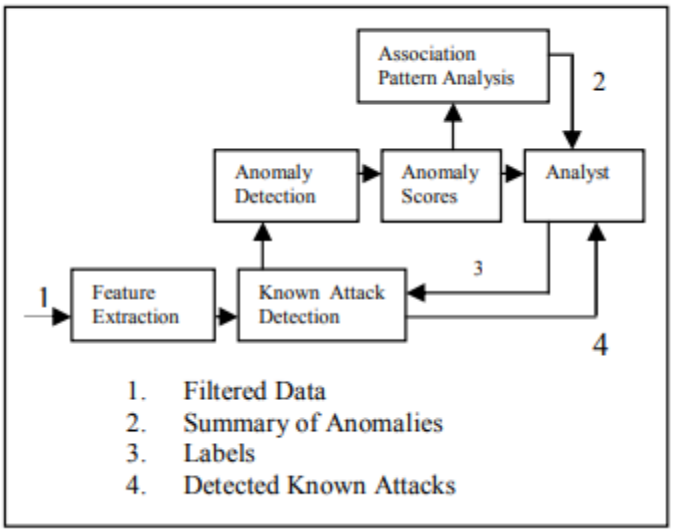

Figure 1.2: Explain The Process Of Analyzing On The Real Traffic Data[4].

1.2 PROBLEM DEFINITIONS

There are many problems that the researchers in this subject especially the privacy of the traffic, Size of real traffic and the various types of attacks , in most of researchers thesis we noticed that they are using multi binary classification but we are going to use multi binary classification with multi-classification models to detect network threats by using ML algorithms and ANN model.

1.3 CONTRIBUTION OF THESIS

The goal and objectives are to Examinant ML(RF, QDA, LOR, SGD) and ANN that can be used to detect network attacks in a fast and effective way by comparing the results with previous studies, In the detection of network anomalies and making classification and compare the results with their.

By improving that we can get a close result by using multi-classification by Selecting the appropriate software platform and professional ways to deal with data set selecting and preprocessing before applying dataset to the model.

1.4 THESIS ORGANIZATION

CHAPTER ONE: This chapter (introduction) presents an overview of the topics

CHAPTER TWO: This chapter presents an overview of background related work description.

CHAPTER THREE: In this chapter, we discuss the information on the problem.

CHAPTER FOUR: The procedure of the experiments that the author has run. Some details about machine learning, algorithms and the equations that can used in this study.

CHAPTER FIVE: Discusses the results and future work. Confusion matrices and accuracy metrics including the analyzed data.

2. LITERATURE REVIEW

In this chapter, we present a literature review for the previous related works of IDS as the following:

- In 2017 a study [5], used machine learning methods (Random Forest, ID3, Naïve Bays, Multilayer perception, Ad boost, K- Nearest Neighbors, and Quadratic Discriminate Analysis were used to detect attack types, they used the same data set, CICIDS2017 was used as the dataset. The performance in this study are as follows: RF 0.97%, MLP 0.76%, Ad boost 0.77%, K- Nearest Neighbors 0.96%, ID3 0.98%, QDA 0.92%.

- In 2019 a study [6], used machine learning methods, C4.5, Random Forest, and Forest by Penalizing Attributes to make an Efficient Network Intrusion Detection System Based on Feature Selection and Ensemble Classifier CICIDS2017 data set were used too and the get the following performance RF 0.949, C4.5 0.937, ForestPA 0.948.
- In 2018 researchers from Blekinge Institute of Technology, Sweden[7] used a deep neural network with 10 hidden layers with binary classification and the best accuracy rate for DDoS was 95 %,

- In 2019 researchers [8]used First 80 different features, used machine learning methods, to build a Network Intrusion Detection System they get the following results DT 98.87%, Random Forest 98.95%, KNN 99.16%, QDA 99.18 %.

- A study targeting DDoS attacks was conducted by Suresh and Anitha [56], in 2011. In this study, CAIDA was chosen as the dataset while many machine learning algorithms such as Fuzzy C-Means, Naive Bayesian, SVM, K-Nearest Neighbours (KNN), Decision Tree and K-means Clustering were used. The success rates of this study are as follows: Fuzzy C Means: 98.7%, Naive Bayesian: 97.2%, SVM: 96.4%, KNN: 96.6%, Decision Tree: 95.6%, and K-Means Clustering: 96.7%.
- In the study[57] conducted in 2012, Naive Bayes Classifier, combined with feature reduction method, achieved high performance in determining four types of attack on NSL-KDD dataset. (DoS: 98.7%, Probe: 98.8%, R2L: 96.1%, U2R: 64%).

- In 2013, another study[58], a new architecture was achieved by combining K-Means Clustering and Naïve Bayes Classifier methods to deal with common false alarm, (the benign data classified as attack, false positive) problems in machine learning methods. In this study, the new method was tested with the ISCX 2012 Intrusion Detection Evaluation Dataset. At the end of this process, high performance (99.8%) and low false alarm rate (0.13%) were obtained.
- In a study [59]conducted in 2017, seven commonly used machine learning methods (Naive-Bayes (NB), Random Forest (RF), K- Nearest Neighbours (KNN), Multilayer perceptron, Adaboost, ID3, and Quadratic Discriminant Analysis (QDA) were used to detect 15 different types of attack. During this process, CICIDS2017 was used as the dataset. The performance ratios obtained in this study are as follows: Naive-Bayes 0.84%, KNN 0.96%, RF 0.97%, MLP 0.76%, Adaboost 0.77%, ID3 0.98%, QDA 0.92%.

3. METHODOLOGY

3.1 ANOMALY DETECTION

There are many point for anomaly we need to recognize it like a sample that does not have a normal behavior, it's like an unusual data sample doesn't like the dataset with her the properties like in case of using credit card but in Christmas and valentine day it's a collective anomaly that the increase in spending is normal [9].

3.2 DATASET

To make our classification a big amount of network traffic to train and test is needed. As we know it's impossible to use real network traffic because of privacy issues so for that datasets have been produced by recording only the headers, many data sets are available to use but we have to decide to choose CICIDS 2017 (Intrusion Detection Evaluation Dataset) [10].

Which was created by the Canadian Institute for Cybersecurity at the University of New Brunswick, the data stream was recorded on a network created by computers using up-to-date operating systems such as Windows, Mac, Ubuntu Linux, and Kali Linux.

This dataset are better than others because:

• It's from the real-world data and It includes the FTP, HTTP, SSH protocols.

• It was collected from various updated operation systems computers (Mac, Windows, and Linux) among attackers and victims systems and data sets were labeled, was applied and 85 features and data set files are large (47.9 GB) in multi separate files unlike the KDD99 and NSL-KDD datasets which are separated files for train and test.

• IT contains benign and the most up-to-date common attacks that could be seen in table 3.1.

Table 3.1: CICIDS 2017 (Intrusion Detection Evaluation Dataset) files description

We used 10% of the date set in the implementation which is about 250000 simple randomly.

Flow Recording Day	Duration	CSV File Size	Attack Name	Flow Count
Monday	Working hours	172.782 MB	Benign (Normal human activities)	529918
Tuesday	Working hours	131.914 MB	Brute Force, FTP-Patator SSH-Patator	445909
Wednesday	Working hours	219.890 MB	DoS / DDoS, DoS slowloris DoS Slowhttptest DoS Hulk, DoS Golden Eye	692703
Thursday	Morning	50.8 MB	Web Attack – Brute Force Web Attack XSS Web Attack – SQL Injection	170366
	Afternoon	81.155 MB	Infiltration–Dropbox download Meta exploit Win Vista	288602
Friday	Morning	56.9MB	Botnet ARES	192033
	Afternoon	75.13 MB	DDoS	225745
	Afternoon	75.10MB	PortScan	286467

3.3 MACHINE LEARNING(ML)

It's a science and art that enables the programmed computers to learn from the given data, it is a branch of artificial intelligence (AI), it works to design and development of algorithms for the input data and make predictions thought to be features of the data by building models for data [11], It can be used Network intrusion detection(nid), Email spam Blocking, Web search, ads, Text analyzing , Facebook uses [12].

3.4 MACHINE LEARNING METHODS

There are two main categories of machine learning:

3.4.1 Supervised Machine Learning

It categorizing data from the prior data (labeled data) with the target. It used frequently in data science problems so the computer predicts the next task and performs it by analyzing the data provided to it previously by building a classifier to classify or labeled incoming patterns. The data accessed to the model are formed in training and testing sets[13].

3.4.2 Unsupervised Machine Learning

It's the opposite of supervised learning by categorizing data from the prior data by generating rules to map input data to output data it does not need labeled data, it learns from the input data by well-distributed the whole picture and gathers output by processing the whole input [14].

3.5 FEATURE SELECTION TYPES

3.5.1 Filter BASE

This method uses a ranking or sorting algorithm to filter out those features that have less usefulness so in this type filtering only the subset of the relevant features then The model is built [15].

3.5.2 Wrapper-based

This method considers the selection of a set of features as a search problem. So we need a machine-learning algorithm to use it for testing performance of a model then we add or remove the features[16].

3.5.3 Embedded

This method uses algorithms that have built-in feature selection methods. For instance, Lasso[17].

3.6 ALGORITHMS

It's a procedure to solve a problem based on a sequence of specified actions it classify and decide what to do with the information gathered or at hand. So it's the way of thinking, many algorithms are available with different purposes.

In this thesis, we are going to use supervised machine learning techniques by applying an algorithm and compare the results of the algorithms to see which is the best results and compare them with other studies.

3.6.1 Random Forest

It is a machine learning algorithm used for binary and multi-classification, This algorithm used to the performance of various problems, it was proposed by Breiman [18]. It was based on decision trees for classification or regression, the model made up of many decision trees not just one simply averaging the prediction of trees figure 3.1.

There is no need for cross-validation or a test set to get an unbiased estimate of the test error,So it creates by assembling a huge number of decision tree structures and generates N (number) decision trees from the training dataset, In the same time it randomly resample's the training set for each one of the trees which are obtained, the last step is voting by selecting new estimates from estimates made by N trees, The best value the rating is determined as the final selected value [18].

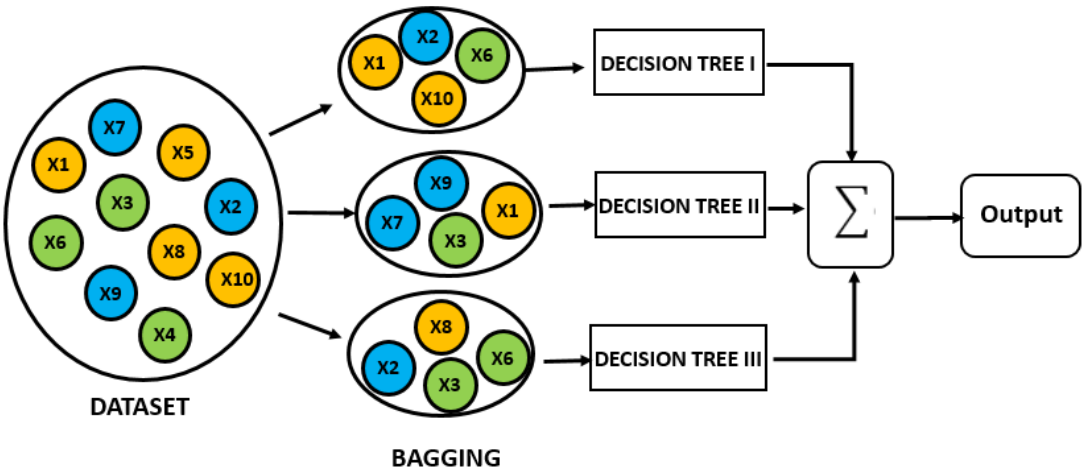

Figure 3.1: Random Forest Classifier.

Usually, we use it for classification and for feature selection because it's the best way to avoid over fitting problems because it does not count the most frequently decision tree, especially with big data sets.

3.6.2 Logistic Regression

It is a machine learning algorithm used for binary and multi-classification, it is a predictive analysis algorithm and based on the concept of probability by transforming the output using the logistic sigmoid function to return a probability value for binary classification The hypothesis limit of the cost function between 1 and 0 and minimize the cost value by using Gradient descent[19]figure 3.2 and equations 3.1 and 3.2 explain how does it work.

$$\theta J \coloneqq \theta J - \alpha \frac{\partial}{\partial \theta J} J(\theta) \tag{3.1}$$

Minimizing cost function by running gradient descent over each parameter

$$Repeat\ \{$$

$$\theta J \coloneqq \theta J - \alpha \sum_{i=1}^{m} (h\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \tag{3.2}$$

$$\}$$

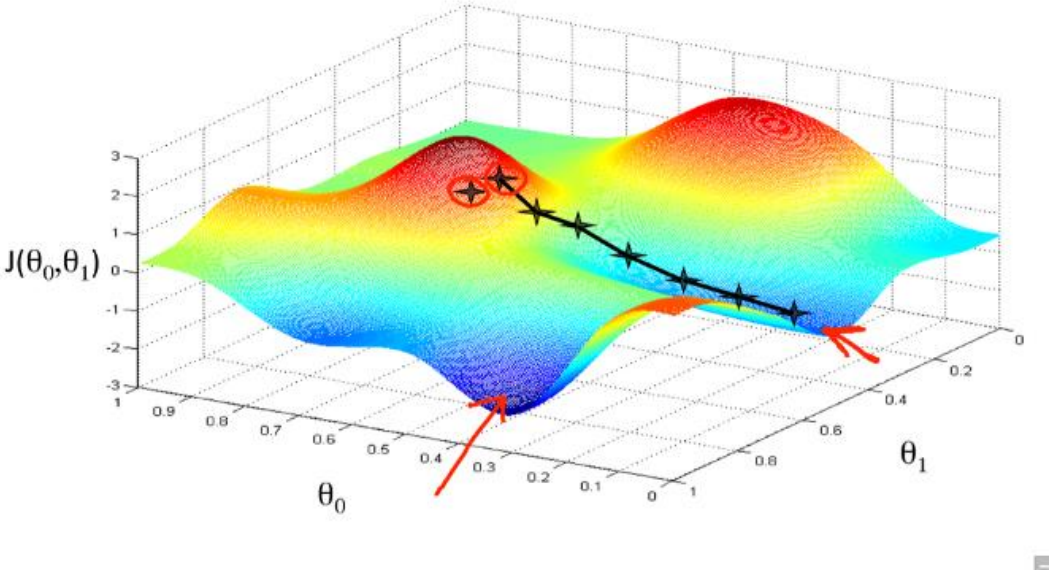

Figure 3.2: How gradient descent minimizes the cost function[19]

3.6.3 QDA (Quadratic Discriminant Analysys)

It is a statistical classification method, it Assigns a group within multiple sum of groups by processing each feature in a single patch so we get the minimum error rate and the measurements of each class are distributed[20].

3.6.4 SGD (Stochastic Gradient Decent)

It is a simple yet very efficient approach to discriminative learning of linear classifiers under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. Even though SGD has been around in the machine learning community for a long time, it has received a considerable amount of attention just recently in the context of large-scale learning[21].

3.7 ANN (ARTIFICIAL NEURAL NETWORKS)

It is a specific subfield of machine learning, method trying to mimic the way the human brain works such as making decisions and finding new deriving information, ANN made up of interconnected hierarchical artificial cells and layers in each layer there are neurons or nodes which are elementary units in an artificial [18]as shown in figure 3.3.

We are going to create artificial neural networks that are consists of :

- Input layer: it is the first layer of the model consist of many neurons depends on the model input (90 dens neurons) which is fully connected, it is not responsible of processing of any information it's just responsible for receiving data set and transmitted to the next layer(hidden layer).
- Hidden layers: the data that has been sent from the input layer is processed then transmitted to the output layer consist of (90 dens neurons) which is fully connected. the number of hidden layers or the number of artificial neurons in this layer different from model to other models, the number of the hidden layers represent the depth of neural network we have two hidden layers the first on (45 dens neurons) and (20 dens neurons) to map the input to the output.

- Output layer: in this layer, the number of neurons depends on the target and each neuron is connected to all the neurons in the hidden layer, the processed data in the hidden layer are served to this layer.

Although it is difficult to build the network structure of ANN but its good at
with missing data complicated datasets and for Initializations we use to define the way to set the initial random weights in Keras layers in all layers and the activation function is responsible for transforming the summed weighted input from the node into the activation of the node or output for that input.

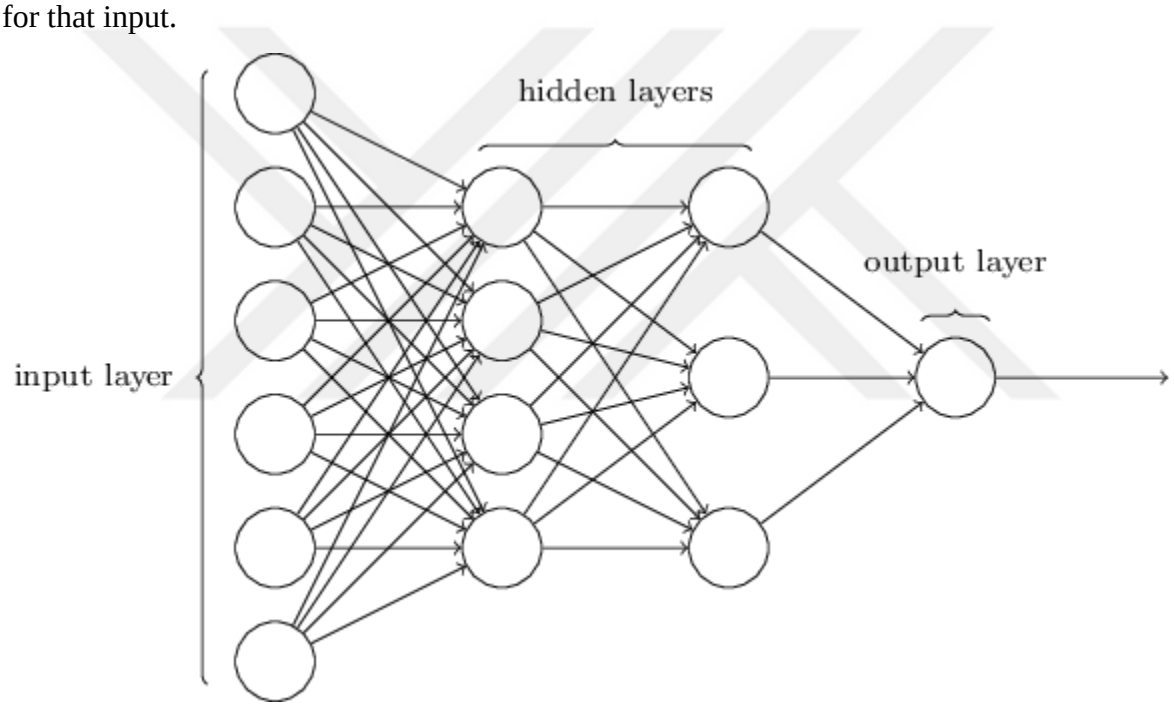

Figure 3.3: Neural network with two-Dense- Hidden Layer

3.8 EVALUATION METRICS

Measuring the performance of Machine Learning of our study are evaluated according to accuracy, precision, f-measure, and recall[23].

3.8.1 Accuracy

Model ratio of successfully categorized train and test data can be described in equation (3).

$$Accuracy = \frac{TN+TP}{FP+TN+TP+FN} \quad (3.3)$$

3.8.2 Recall (the sensitivity)

The ratio of classified non-normal data to all data.

$$Recall = \frac{TP}{TP+FN} \quad (3.4)$$

3.8.3 Precision

Ratio of how much success is the classified data to all data.

$$Precision = \frac{TP}{FP+TP} \quad (3.5)$$

3.8.4 F-measure:

The measure of sensitivity and precision, we will calculate it using the following equation :

$$F\ Measure = \frac{2}{\frac{1}{Recall}+\frac{1}{Precision}} \quad (3.6)$$

Following are the list of various alarm for calculating these four items:

• **TP**: True Positive: The original input data is classified as an attack.

• **FP**: False Positive The original input is normal data is detected by IDS data and classified as an attack.

• **FN**: False Negative The original input data is an attack and classified as benign.

• **TN**: True Negative The original input data is normal which is detected by IDS classified as benign.

which are used to define the metrics below. Note that TP (True Positives) and TN (True Negatives) are the correctly classified instances and the sum TP + FN + FP + TN is the size of the testing dataset figure 3.4 show a simple confusion matrix.

- False-Positive Rate (FPR), True-Positive Rate (TPR), False-Negative Rate (FNR), and True-Negative Rate (TNR) – represent the rates for TP, FN, FP, and TN considering the respective actual classes, equations (2.9) shows the evaluation metrics that is,

$$FPR = \frac{FP}{FP+TN}, TNR = \frac{TN}{FP+TN}, FNR = \frac{FN}{FN+TP} \text{ and } TPR = \frac{TP}{FN+TP} \quad (3.7)$$

This distribution is presented by visualizing Confusion matrix in Figure 11.

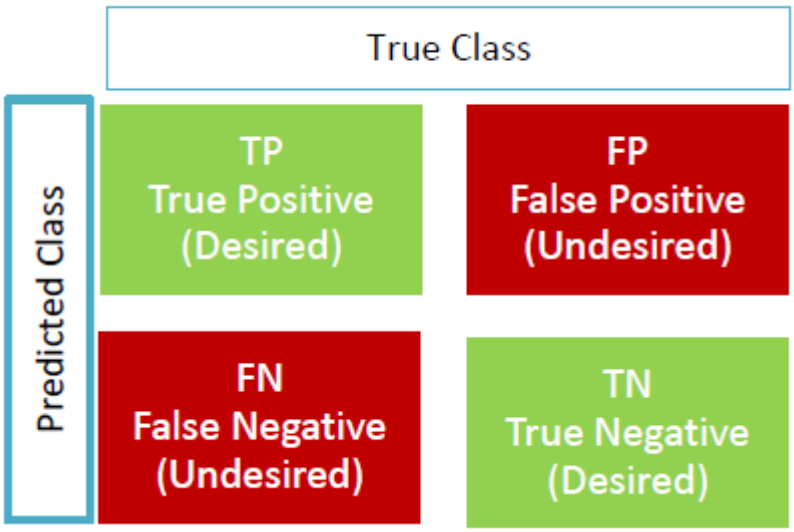

Figure 3.4: Confusion matrix

3.9 ATTACKS

The data set contains many types of attacks are and they are tagged with 15type, The Benign tags represent normal network and the others represent attack types. The benign record, Formed using FTP, Mail services, and HTTPS protocols represent a non-harmful attack are shown in table (3.2)and percents in figure 3.6.

Table 3.2: BENIGN and attack types in the select 10% of data

types	numbers
BENIGN	193767
PortScan	16663
FTP-Patator	558
SSH-Patator	402
Web Attack Brute Force	286
DoS slowloris	245
Web Attack SQL Injection	5
Heartbleed	2
DDoS	17101
DoS Hulk	9876
DoS Golden Eye	428
Bot	290
DoS Slowhttptest	251
Web Attack XSS	122
Infiltration	4

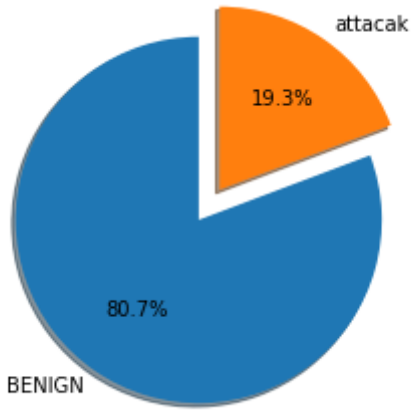

Figure 3.5: Attack and Benign Percent

When the numbers of these attacks are examined, it is immediately apparent that the numbers of some attacks are very high. For example, The DoS HULK attack is almost half of all attacks and the PortScan attack is one-third of all attacks, The reason for this imbalance in the distribution is the nature of these attacks. Both DoS and PortScan attacks cause too much data and packet flows during the attack. Therefore, during these attacks, it is completely natural to observe more intense traffic from normal usage and other types of attacks ,that's why in this multi binary classification we focus on Dos (DoS Hulk, DoS Golden Eye, DoS Slowhttptest), DDoS and port scan like in

tables(3.3)(3.4) because they are related and for ANN and random forest we feed the model with all attack types.

Table 3.3: Bening, DDos, port scan, and Dos attack types in the dataset

Attack	types
BENIGN	45434
DDoS	17101
PortScan	16663
DoS Hulk	9876
DoS Golden Eye	428
DoS Slowhttptest	251
DoS slowloris	245

Table 3.4: Bening, DDos, port scan, and Dos attack types after append all Dos attack types together

Attack	types
BENIGN	44848
DDoS	17101
PortScan	15203
DoS	8626

The dataset contain the following attacks :

3.9.1 DDoS and DoS HULK

the effectiveness of DoS and DDoS attacks on the computer (server) load first-hand, so it disable the server so it has the ability to conceal the user client and use a different template for each attack request. for DoS attacks one computer send that request to the server but in DDoS attacks, many

computers send many requests through many connections to the same server so it disable the server faster than DoS attack figure 3.7.

when the attack starts it creates TCP-SYN (Transmission Control Protocol - Synchronization) floods and multiple HTTP-GET flood requests. [23][24][25].

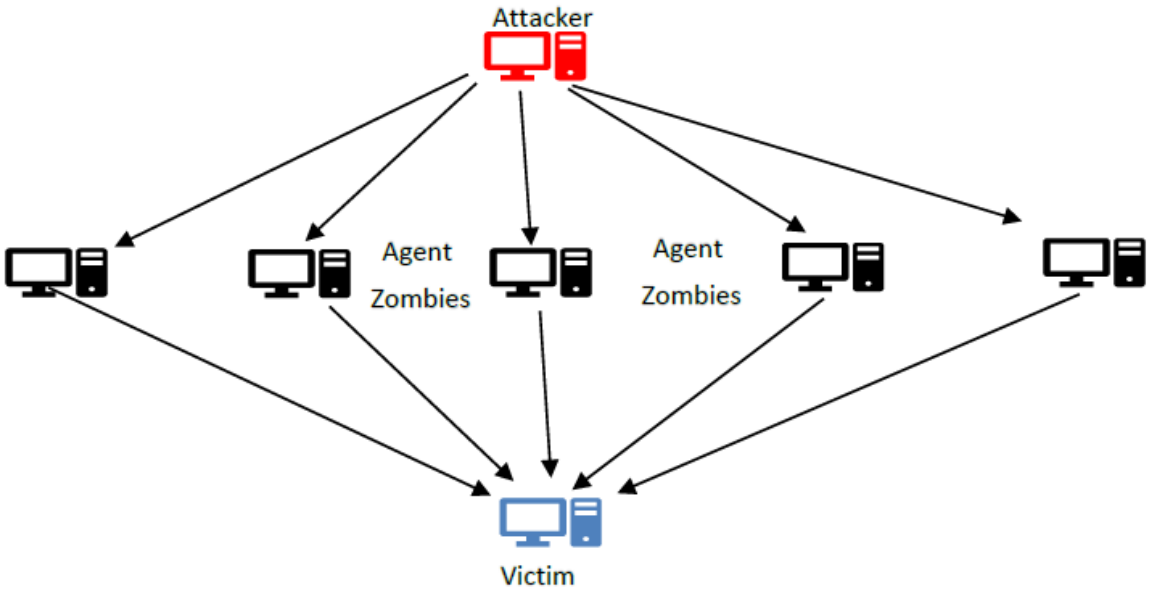

Figure 3.6: DDoS Attack

3.9.2 TCP-SYN Flood

This type of attack is a 3-way handshake method to consume server resources by sending SYN packet to the (server) from the client (attacker) requesting to connect the victim then the server(victim) responds with the SYN-ACK packet in response. But the client doesn't send the ACK packet and the connection waits as incomplete. The server reserves a resource by keeping these requests in a log queue figure 3.8. After a while, the number of requests will increases so the server becomes inaccessible and collapse [26].

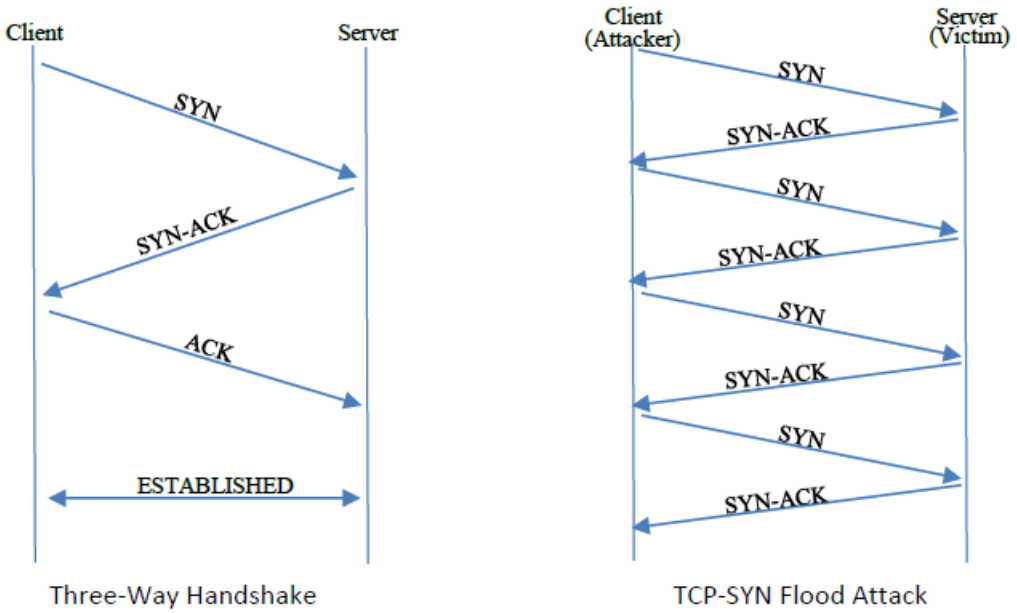

Figure 3.7: TCP-SYN FLOOD Attack with normal 3-way handshake

3.9.3 HTTP-GET Flood

HTTP GET method is used by the client from a web browser to request a file from a web server. The attackers aim the bottleneck resources of the server as network bandwidth, CPU processing, database bandwidth by sending one or multiple HTTP GET requests on each TCP connection using the HTTP to increase the number of HTTP connections that the web server can keep open so the server unresponsive to legitimate HTTP GET requests 3.9 feature. Moreover, these requests do not require high network traffic[27].

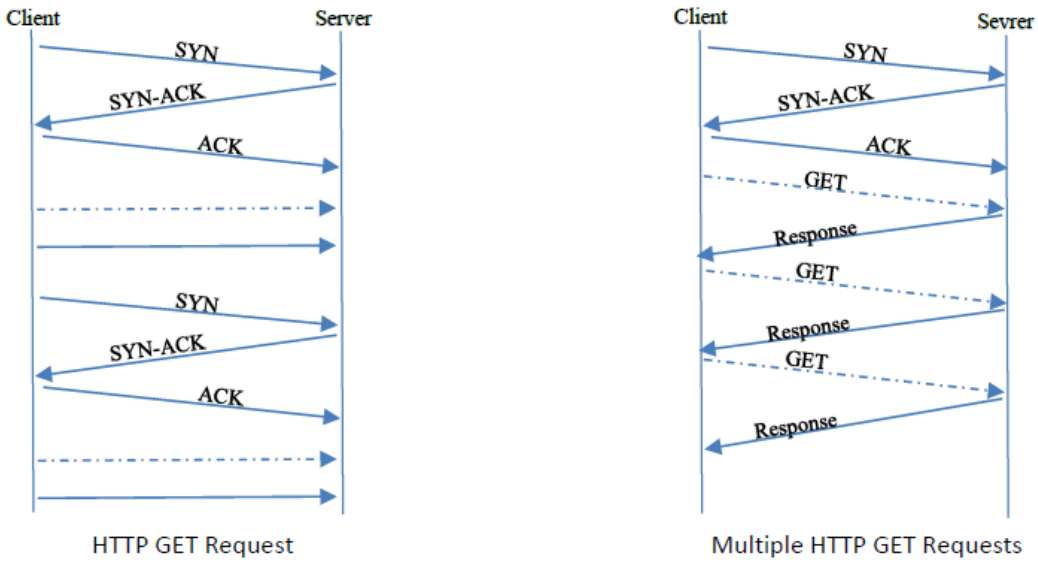

Figure 3.8: Single and multiple HTTP GET Flood Attack

3.9.4 PortScan

It is a popular attack type used to gather information about how attackers can get into the system, in the computers (servers) there' are 65535 Ports used to determine what type of service on the server , a server could run a SQL server, FTP server at the same time using ports 80 and 21, by using a listening service (nmap) to see which ports are open on the server [28]. In order to determine which services are running on the server to find exploitation on it.

3.9.5 DoS Golden Eye

It is a Python-based DoS attack. It consumes the victim's system resources, it is a multithreading attack, which can influential launch an Http Flood attack using multithreaded CPU and memory hardware efficiently[27].

It does not encrypt packets during an attack and does not create a fake source IP (Internet Protocol) address (IP spoofing). It can be run on all Linux, Mac and Windows operating systems[29][30][31].

By using the Keep-Alive method, it maximizes the file size it transmits over a single TCP connection. Also, with the No-Cashed message feature, it disables HTTP- Cache-Control. By using these two features, system resources are consumed very quickly [32].

3.9.6 FTP-Patator

In this Python-based attack, attacker try to find your username and password and login legally by guessing your username and password. By using a Software that automatically generates users (admin usually) and passwords over a short period of time is a characteristic feature for brute force attacks and as we know users tend to prefer meaningful and easy to remember passwords so they are generally weaker than they are thought to be against brute-force attacks[33].

3.9.7 SSH-Patator

SSH-Patator is a tool written in the Python program, using SSH to log on to a remote system[34], and just like (FTP-Patator) the attackers try to generates users (admin usually) and passwords over a short period of time and to gain remote access to a system and gain full control over it by
Scanning the server to learn about the server to find the host running SSH by performing a scan on a specific port number for IP blocks on a network or subnets. And then log in to the system
As an admin.

3.9.8 DoS Slowloris

this attack is written in Perl programming language to consume the system resources of the server and legitimate clients from receiving the service.
By creating a TCP-SYN flood to reduce the number of users that the target server could handle and serve with of incomplete TCP packets with SYN flags are observed.[26][35].

3.9.9 DoS SlowHTTPTest

This attack abuses the window size feature of TCP, consuming the resources of the server so that the rightful users cannot benefit from the service.

The window size is a feature used to prevent overflow and stacking of data in TCP. In this way, the recipient limits the maximum size of the file be received and the server makes submissions based on this limitation. By this method, avoids the overflow and collapsing of the transmitted data due to the slowness of the connection during transmission[36].

During the SlowHTTPTest attack, the attacker sends a normal request to the server, but when receiving the response from the server, it sets the window size to a value too close to 0, which slows down the receiving process as much as possible. In such a case, the server will have to allocate resources to process and store the remaining large part after sending a very small part of the file. In the event of an increase in these requests, the server cannot meet the demands and becomes out of service. In addition to the attack, the attacker continues sending SYN and ACK packets to prevent the connection from breaking due to timeout[37].

This attack is quite easy to make because it does not require high bandwidth or hardware. Moreover, this attack is difficult to detect in the internet stream because it looks like innocent HTTP requests which need a long time. However, connections that have window sizes that are much smaller than normal flow will give you a hint about this attack.

3.9.10 Botnet

Botnet attack is made using the Ares [38], an attack tool written in the Python program. This attack takes place on Friday-morning-records within the CICIDS2017 dataset[10].

The botnet attack is a network created by malware-infected computers. These computers that makeup botnet are called bots or zombies. Because of the malicious software, these computers perform various harmful activities without the knowledge of their owners. These computers can be remotely controlled by bot-masters and used to make SPAM (unsolicited emails) or DDoS attacks[31][32].

The detection of the bots' network behavior is only possible in the active state (during the attack). When the botnet is passive, there is no network activity and cannot be detected. In the active state, the number of bytes per packet and the number of packets with the PSH flag make it possible to detect this attack.

3.9.11 Web Attack

Web Attack takes place on Thursday-afternoon -records in the CICIDS2017 dataset [10]. In this group of attacks, three different web-based attacks were launched against a vulnerable PHP (Hypertext Pre-processor - a script programming language)/ MySQL (an open-source database administration framework) web application identified as the victim. These attacks are:

Brute Force: This topic has been elaborated under the heading FTP-Patator[39].

XSS: XSS (Cross-Site Scripting): a popular web-based attack type, is implemented by injecting code into a web page. When the victim wants to view this web page, this malicious code fragment can lead to undesirable results such as data theft, session seizing, special code execution, and fraud [40].

SQL Injection: SQL (Structured Query Language – a database management system) Injection attack is one of the most popular and most dangerous attacks on the web. A typical dynamic web page keeps the user's various information in the database for later use and occasionally makes various queries in the database to reach this data [41].

These attacks are usually done by exploiting a security vulnerability in the database layer of a web application. The attacker can access the database using the exploits of the connection between the web application and the database. Then he could be able to steal, delete, or modify the data by sending malicious commands to the database server.

3.9.12 Infiltration

Infiltration attack takes place on Thursday-afternoon -records in the CICIDS2017 dataset. However, the infiltration concept used in this dataset is not a general attack, but rather a specialized attack concept for a specific scenario. In this situation, there is an information-gathering attack on a network that has become vulnerable due to a virus file entering the system.

This attack scenario is as follows: the virus enters the system through a file downloaded from the Dropbox by the victim using Windows and a file copied from a USB flash drive by the victim using the Macintosh. At the next stage of the attack, the attacker exploits the vulnerability created by this virus to perform port scanning attacks on the network [42].

3.9.13 Heartbleed

This attack is made using the Heartleech[43], a Heartbleed exploit tool written in C programming language. Heartbleed is a crucial vulnerability in OpenSSL (an open-source implementation of SSL and TLS protocols) and exploits from heartbeat, a library of TSL protocols.

The normal operation of heartbeat is as follows: The client transmits to the server a request consisting of random payload. the server replies to this request with a packet containing the same payload.

During the attack, a specially created heartbeat request is sent to the server. This demand is actually empty but points out that it carries very high amounts of data. At this stage, due to a vulnerability in OpenSSL, the server sends a piece of memory in the specified length in response to this message. These parts contain various confidential information such as personal information, usernames, and passwords that should not be sent. The size of these packages can be up to 16 KB and the attacker can repeat this operation without any limit.

During this attack, some anomalies on the network flow are observed. The length of the incoming messages is less than the heartbeat minimum message size of 20 bytes. Outgoing 16 message lengths are in the kilobyte level, and for a heartbeat response, this size is too big. The difference in size between outgoing and incoming messages is another way to understand the attack. In normal heartbeat messages, the lengths of the outgoing and incoming messages are the same. In case of attack, incoming message lengths are too small, outgoing message lengths are too large.

3.10 PLATFORM

The Software: Python Anaconda Jupiter editor[44] , A free package manager and open source object-oriented programming language based on python with a simple syntax editor and dynamic structure. In Anaconda editor, it's simple and easy for coding and writing notes. and there are many libraries support "machine learning" and neural networks such as MATLAB and C/C++.

Figure 3.9: Anaconda Editor

Sklearn (Scikit-learn) [45] it is a machine learning library that can be used with the Python programming language and it's used in machine learning algorithms.

Figure 3.10: Scikit-Learn Library

Dense, Activation [46] powerful tools for building a neural network**.**

Pandas[47] it's a powerful data analysis library running on Python. When working with a large dataset, Pandas allows you to easily perform many operations such as filtering, bulk column/row deletion, addition, because of all these advantages, the Pandas library has been used.

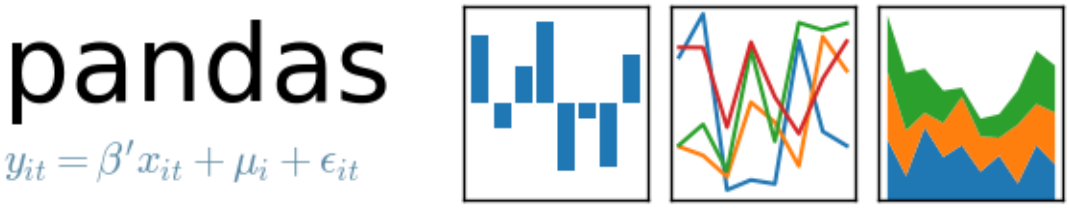

Figure 3.11: Scikit-Learn Library

NumPy[49], a library for performing mathematical and logical (numerical) operations easily.

Label Encoder[50] Its an embedded library of the Sklearn Learn in Python which converts categorical and text to numerical values.

Train_test_split[51] it's used to split train and test data.

Early Stopping[52] Its a function that used in at certain stages of the training process by monitoring the loss value at each epoch and after the test loss has not increase after a number of epochs the training will interrupt and stop.

ModelCheckpoint[53] Save the model after every epoch.

Random Seed [54] Random number generator in Python.

Hardware Minimum Requirements:

CPU	:	Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz 2.60 GHz
Random Access Memory (RAM)	:	4 GB
Operating System	:	Windows 7 Pro 64-bit
Graphics Processing Unit	:	INTEL ® HD GRAPHIC 4000

4. IMPLEMENTATION

In this chapter, many preprocessing operations are performed to detect anomaly by ML AND ANN techniques, selecting and cleansing processes are the first thing to do .to get a clean data without missing or mistake data then dividing dataset into training and testing parts,after that feeding data set to our model and start the implementation of machine learning and neural network algorithms. Figure 4.1 illustrates the implementation process in detail.

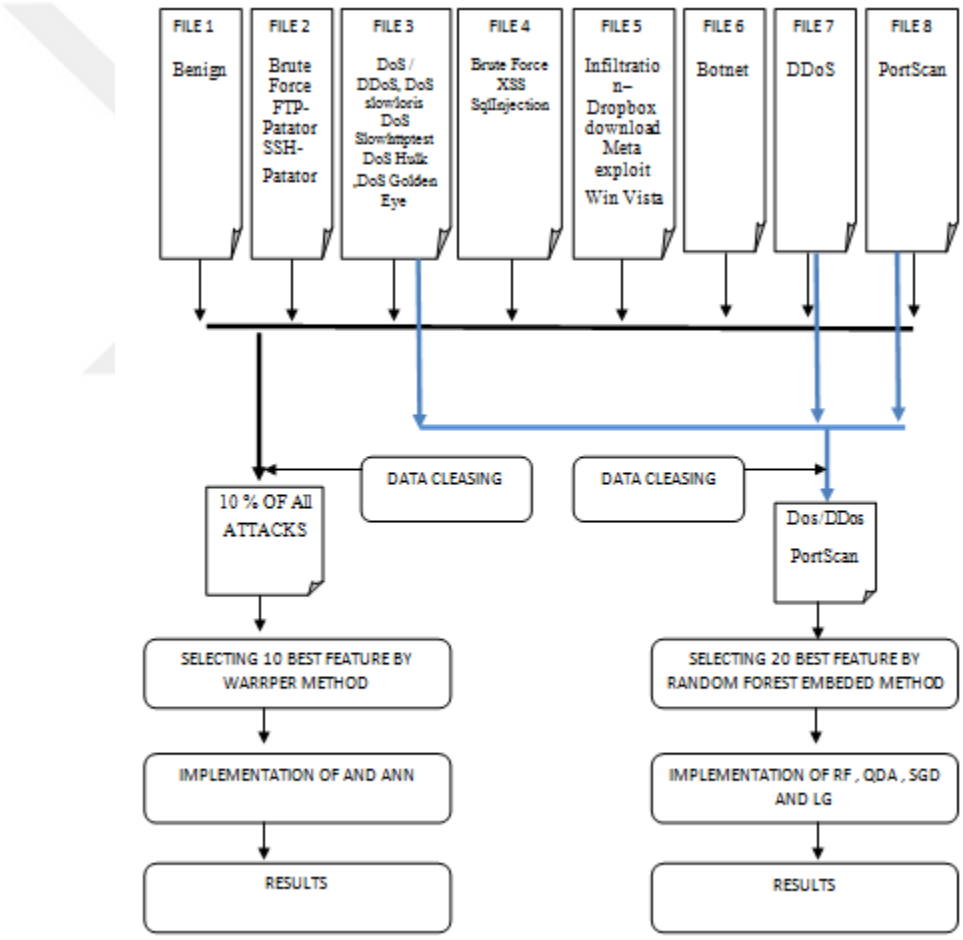

Figure 4.1: The Implementation Process

4.1 DATA SELECTING AND CLEANING

CICIDS2017 [10] dataset contain 8 recorded files for different days , Each file contain one or more attack type so to make our multi-classification we have to obtain all these files in one big file then to make selection for random simples and then make cleansing for unnecessary data and fill missing data to make the model more efficient, defining functions to impute missing data . The first function to find null columns then we use the second function to explore the selected column then we fill the null value with the third function for imputation and another one to remove outliners values because ML tends to work better with well-distributed data, data contain special characters and NAN values so we convert it and fill 0 instead of it, The dataset file consists of 79 columns that define the flow properties for better performance we convert the properties including the categorical and string values into numerical data before feeding the model by using LabelEncoder [58] from Sklearn classes, so the "Label" tags are a categorical feature so we convert it to numerical values.

Finally, some minor structural changes are made to the dataset, including:

The dataset file contains 3119345 stream records in 8 files. The first step in the pre-processing process will be to select 10% (Row, Column) (240000, 79) 30000 random simple from each file of this dataset and shuffling the data by using random. seed [54] from Sklearn , classes removing three attack types (Heartbleed, Infiltration and Web Attack SQL Injection) because of lack of simples in the new created file And even in the original file , for multi binary classification we used data of 3 files that contain (all Dos , DDos and Portscan) attacks with final data shape for Dataset (Row, Column) : (85780, 79).

4.2 PREPROCESSING DATA

Deleting corrupted data and converting special characters, Filling the NAN value, converting non-numerical values to numerical values by using label encoder[50], Then by using StandardScaler from sklearn [54] library applied for the train and test to the data so the results are visualized and a clear difference noted.

Appending Dos, DDos and port Scan files in one file and prepare this file to apply it to our model then scaling our model by(standardization and normalization) are very important preprocessing point for many machine learning algorithms to enhance the accuracy of the model. because ML tends to work better with well-distributed features, rescaling it so they have the properties of a standard normal distribution with a mean of zero and a standard deviation of one.

4.3 CREATION OF TRAINING AND TEST DATA

The training and test data are needed to train and evaluate the performance of the model, The algorithm acquires a skill on the training data and applies it to the test data. The result of the test data is the performance of the model [40], The dataset used contains a single unbundled dataset., It should be divided into training and test, By using a function we define it perversely to split the target with is the label column from the data set then we use the same present for training and testing by using sklearn command, train_test_split[59] but for neural network we define a function to split the target with is the label column from the data set then we use the same percent for training and testing , this command divides the data into 25% test, 75% training [40]. By using The train_test_split to makes the selection random with a cross-validation to ensure that we get the best score but for neural network we define a function to split the target with is the label column from the data set then we use the same percent for training and testing.

4.4 FEATURE SELECTION

It is the process of selecting useful features in the dataset and to remove the bad effects of having unnecessary features in order to get the best score in the model, Applying Random Forest to select features based on feature importance (240000 rows of data), the feature list obtained is shown in Table 4.1 and figure 4.2.

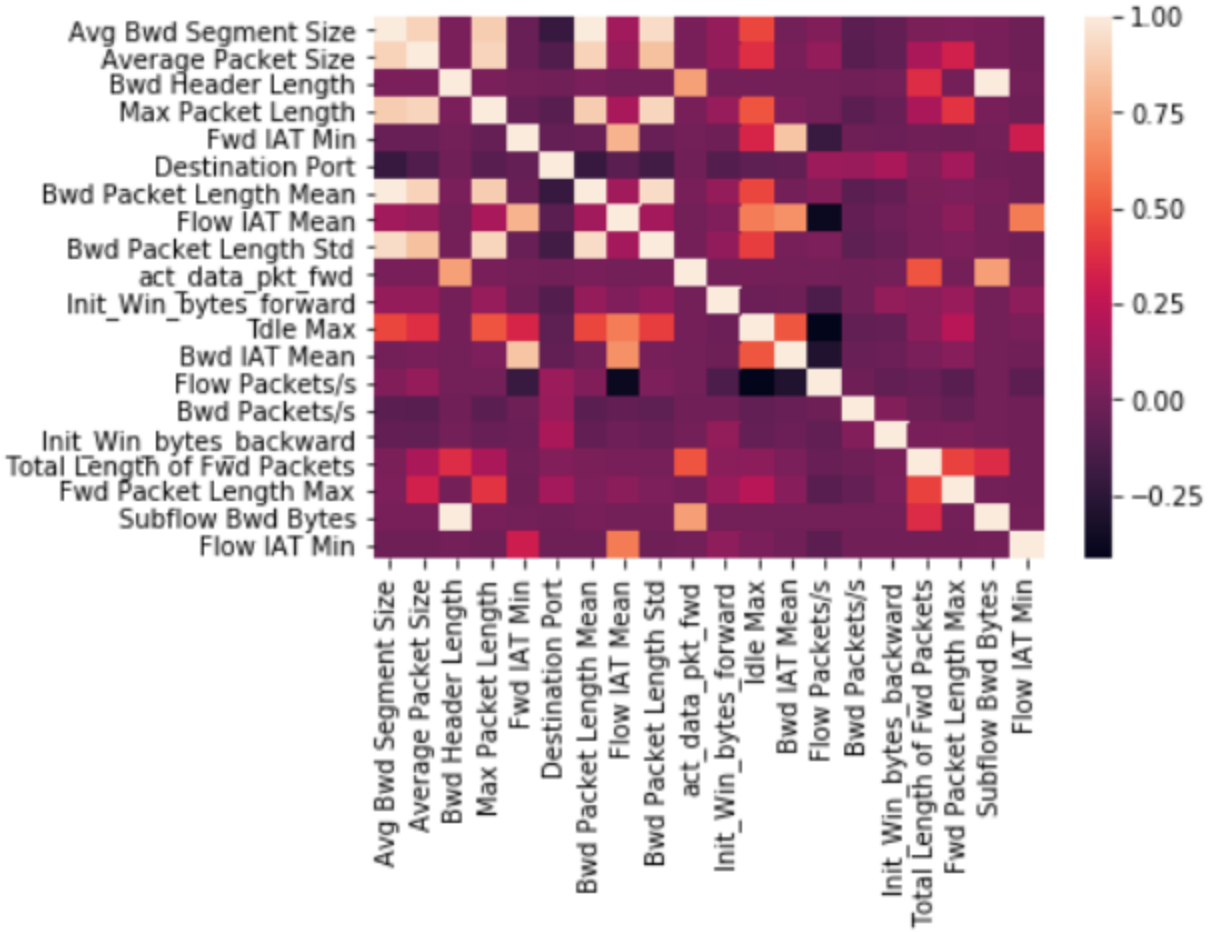

Figure 4.1: Correlation of the best 20 features using an EMBEDDED method

Table 4.1: Features Importance weights for the best 20 features using an EMBEDDED method

Feature name	Importance weights	Feature name	Importance weights
Avg Bwd Segment Size	0.1797	Init_Win_bytes_forward	0.0218
Average Packet Size	0.1167	Idle Max	0.0208
Bwd Header Length	0.0861	Bwd IAT Mean	0.0197
Max Packet Length	0.0801	Flow Packets/s	0.0176
Fwd IAT Min	0.0797	Bwd Packets/s	0.0145
Destination Port	0.0709	Init_Win_bytes_backward	0.0104
Bwd Packet Length Mean	0.0442	Total Length of Fwd Packets	0.0103
Flow IAT Mean	0.0322	Fwd Packet Length Max	0.0100
Bwd Packet Length Std	0.0267	Subflow Bwd Bytes	0.0093
act_data_pkt_fwd	0.0229	Flow IAT Min	0.0089

Using warp method to select the best features needs an expensive computer resource and time we used it to select best 10 features then we use those feature in ANN model and RF algorithm to improve the accuracy figure 4.3 and table 4.2 explain shows the weights.

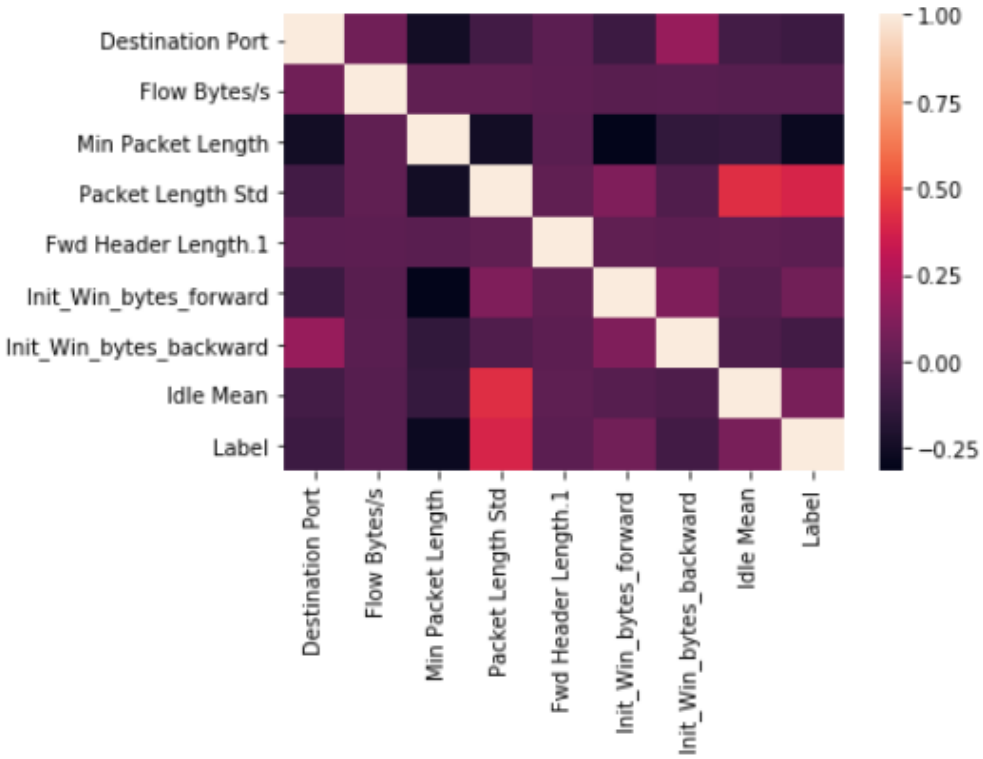

Figure 4.2: the correlation of the best 10 features using Wrapper-based method

Table 4.2: Best 10 features by using Wrapper-based method

Feature name	Score
Destination Port	0.385
Flow Bytes/s	0.576
Min Packet Length	0.708
Packet Length Std	0.769
Fwd Header Length.1	0.789
Fwd Avg Bytes/Bulk	0.797
Init_Win_bytes_forward	0.80
Init_Win_bytes_backward	0.801
Idle Mean	0.804

4.5 IMPLEMENTATION OF MACHINE LEARNING ALGORITHMS

The 10% of data set features after preprocessing and cleaning we feed the model with data, we used machine learning algorithms Random forest and we build an ANN.

The first method, we apply Feature Selection for the best 10 features with the highest importance-weight as we saw in table 4.2 by using RF Wrapper-based method aiming to observe the effectiveness and performance of different machine learning methods on different attack types, which evaluated each feature. The feature with the best score is selected out of all the features and combination with all the other features. The combination of two features that proceeds the best algorithm performance is selected. The process continues until the ten features are selected, all features in the file are used contains(attacks types and benign tags). ModelCheckpoint[53] and EarlyStopping[52] are used to monitor and check the process of the model in order not fall in local minimum and give the process a push just like a rolling all to avoid to fall in local minimum and get the global minimum , we set patience to five to make sure of that ,we compare multi classification RF and ANN in figure 4.3 with previous studies [8] they forget an important point especially the disruption of the dataset because as we know machine learning algorithms tend to work better with well-distributed data.

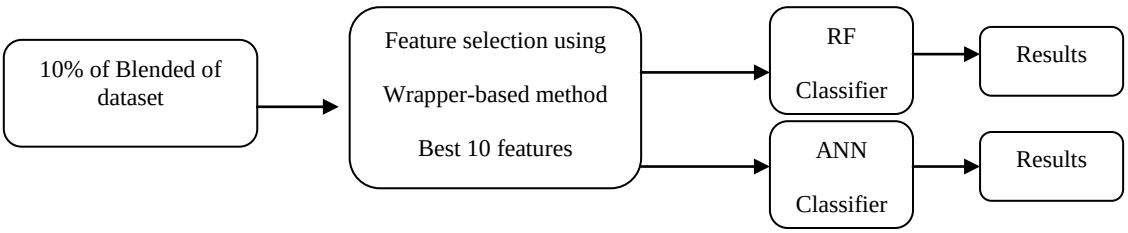

Figure 4.3: Implementation of Segmented and Selected Data For RF and ANN

In the second method of binary classification we select three attacks file (Dos, DDoS, PortScan) with normal traffic simples which has been filtered and preprocessing processes applied as shown in data SELECTING AND Cleaning, we use embedded method with RF classifier to select best 20 features of data set the selected attacks and normal data ,then feeding the model and apply (RF ,QDA,SGD AND LOGISTIC REGRESSION) algorithms as shown in figure 4.4, after splitting it to multi-data frames (Dos table , DDoS table and Port scan table) and train and test, finally we append them in one table which contain the three attacks with normal .

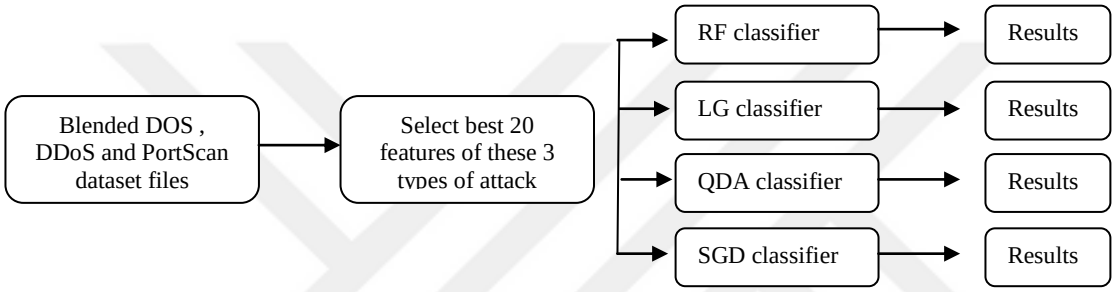

Figure 4.4: Implementation of Segmented and Selected Data For

The RF ,QDA,SGD AND LOGISTIC REGRESSION

Many researchers[7], [55] tend to work with weka which is one of data mining tool specially with binary classification , preprocessing and cleansing operations on data set could not performed on data set that's mean the data will feed to the model directly which decrease the performance of their models and many problems with categorical and missing values .

5. RESULTS AND DISCUSSION

In this section ,The results of the study done in the implementation section are presented ,In this context the assessment carried out so the evaluation criteria are presented via the data of the F-measure, Precision and recall.

graphs are created to illustrate the consistency of the results and the change between them.

5.1 FEEDING DATASET TO THE MODEL.

5.1.1 Model1

The first step we apply selected 10 features with the best weights obtained According to all datasets. with no significant change in the algorithms of RF and ANN ,based on F-measure and the confusion matrix for each algorithm in our model and the result we get shown in table 5.1, Classification report in table 5.2 and confusion matrix in table 5.3 for random forest and the Classification report for Neural network in table 5.4 and confusion matrix in table 5.5 for random forest.

Table 5.1: Accuracy of RF and ANN for 10% of the dataset that contain all attacks

Machine Learning Algorithms For All Data Set	**With 10 selected features**
RF	0.996
ANN	0.98

The algorithm of the RF score 0.996. and ANN 0.98. However, implementation that uses an Wrapper-based method for feature selection can be seen in Table 4.2. In this method, we used 10 best features with the highest score.

Table 5.2: 10% of 8 obtained segmented files of the dataset with best 10 selected features, using Random Forest for All Attacks

Accuracy metrics for Random Forest	normal	Attacks										
	BENIGN	DoS Hulk	PortScan	DDoS	DoS slowloris	SSH-Patator	DoS Slowhttptest	FTP-Patator	Bot	Web Attack Brute Force	DoS GoldenEye	Web Attack XSS
F-measure	0.99	0.99	1.00	0.99	0.99	0.97	0.8	0.06	0.97	0.99	0.36	0.98
Precision	0.99	1.00	1.00	0.99	0.99	0.98	0.92	0.5	0.97	1.00	0.95	1.00
Recall	0.99	0.99	1.00	.99	.99	.97	0.70	0.03	.98	0.99	0.22	.96

Table 5.3: 10% of 8 obtained segmented files of the dataset with best 10 selected features confusion matrix , using Random Forest for All Attacks

confusion matrix	BENIGN	DoS Hulk	PortScan	DDoS	DoS slowloris	SSH-Patator	DoS Slowhttptest	FTP-Patator	Bot	Web Attack	DoS GoldenEye	Web Attack
BENIGN	48435	0	0	30	10	0	4	1	1	0	1	0
DoS Hulk	1	4313	0	0	0	0	0	0	0	0	0	0
PortScan	0	0	140	0	0	0	0	0	0	0	0	1
DDoS	16	0	0	4082	3	0	0	0	0	0	0	0
DoSslowloris	1	0	0	0	2440	0	0	0	2	0	0	0
SSH-Patator	2	0	0	0	0	66	0	0	0	0	0	0
DoSSlowhttptest	21	0	0	0	0	0	50	0	0	0	0	0
FTP-Patator	29	0	0	0	0	0	0	1	0	0	0	0
Bot	2	0	0	0	0	0	0	0	99	0	0	0

WebAttack Brute	1	0	0	0	0	0	0	0	0	112	0	18
DoS GoldenEye	65	0	0	0	0	0	0	0	0	0	19	0
Web Attack XSS	1	0	0	0	0	1	0	0	0	0	0	49

Table 5.4: 10% of 8 obtained segmented files of the dataset with best 10 selected features, using ANN for All Attacks

Accuracy metrics for ANN	normal	Attack types										
	BENIGN	DoS Hulk	PortScan	DDoS	DoS slowloris	SSH-Patator	DoS Slowhttptest	FTP-Patator	Bot	Web Attack Brute Force	DoS GoldenEye	Web Attack XSS
F-measure	0.99	0.96	0.79	0.97	0.88	0.65	0.60	0.00	0.66	0.68	0.00	0.72
Precision	0.99	0.98	0.99	0.94	0.81	0.66	0.59	0.00	0.83	0.93	0.00	0.64
Recall	0.98	0.93	0.66	0.99	0.97	0.64	0.61	0.00	0.55	0.54	0.00	0.81

Table 5.5: 10% of 8 obtained segmented files of the dataset confusion matrix using ANN for All Attacks

	BENIGN	DoS Hulk	PortScan	DDoS	DoS slowloris	SSH-Patator	DoS Slowhttpte	FTP-Patator	Bot	Web Attack Brute	DoS GoldenEye	Web Attack
BENIGN	57701	8	0	260	251	5	39	0	12	4	0	14
DoS Hulk	11	4749	0	0	293	0	0	0	0	0	0	0
PortScan	1	0	120	0	60	0	0	0	0	0	0	0
DDoS	26	0	0	4913	6	5	0	0	0	0	0	1
DoSslowloris	32	46	0	0	2847	2	0	0	1	0	0	0
SSH-Patator	12	0	1	0	2	52	0	0	0	0	0	14
DoSSlowhttptest	37	0	0	0	0	0	58	0	0	0	0	0
FTP-Patator	20	0	0	0	2	0	0	0	0	0	0	0

Bot	38	0	0	0	5	13	0	0	69	0	0	0
WebAttack Brute	44	0	0	1	3	1	0	0	0	58	0	0
DoS GoldenEye	94	0	0	0	0	0	0	0	1	0	0	0
Web Attack XSS	8	0	0	0	4	0	0	0	0	0	0	53

5.1.2 Model 2

The next tables will explain the results of 20 best-selected features by the embedded method with (RF, QDA, LG, AND SGD) including confusion matrix, F-measure, Precision and Recall for Dos, DDoS and Port Scan could be seen in tables below.

Table 5.6: The Score, precision, recall, and f1-score for DoS Attack and benign

Attack type	precision	recall	f1-score	Accuracy	Algorithm
DoS	0.99	0.99	0.99	0.997	RF
DoS	0.99	0.98	0.99	0.985	QDA
DoS	0.99	0.99	0.99	0.989	LR
DoS	0.99	0.98	0.99	0.986	SGD

Table 5.7: The Score, precision, recall, and f1-score for DDoS Attack and benign

Attack type	precision	recall	f1-score	Accuracy	Algorithm
DDoS	0.99	1.00	0.99	0.999	RF

DDoS	0.87	1.00	0.93	0.896	QDA
DDoS	0.99	0.99	0.99	0.994	LR
DDoS	0.99	0.99	0.99	0.994	SGD

Table 5.8: The Score, precision, recall, and f1-score for PortScan Attack and benign

Attack type	precision	recall	f1-score	Accuracy	Algorithm
PortScan	0.99	0.99	0.99	0.999	RF
PortScan	0.99	1.00	0.99	0.997	QDA
PortScan	0.99	0.99	0.99	0.997	LR
PortScan	0.99	0.99	0.99	0.996	SGD

Table 5.9: The Score, precision, recall, and f1-score for DoS, DDoS and PortScan

Attack type	precision	recall	f1-score	Accuracy	Algorithm
Abnormal	0.99	0.99	0.99	0.999	RF
Abnormal	0.91	0.91	0.94	0.948	QDA
Abnormal	0.97	0.98	0.98	0.98	LR
Abnormal	0.94	0.97	0.96	0.962	SGD

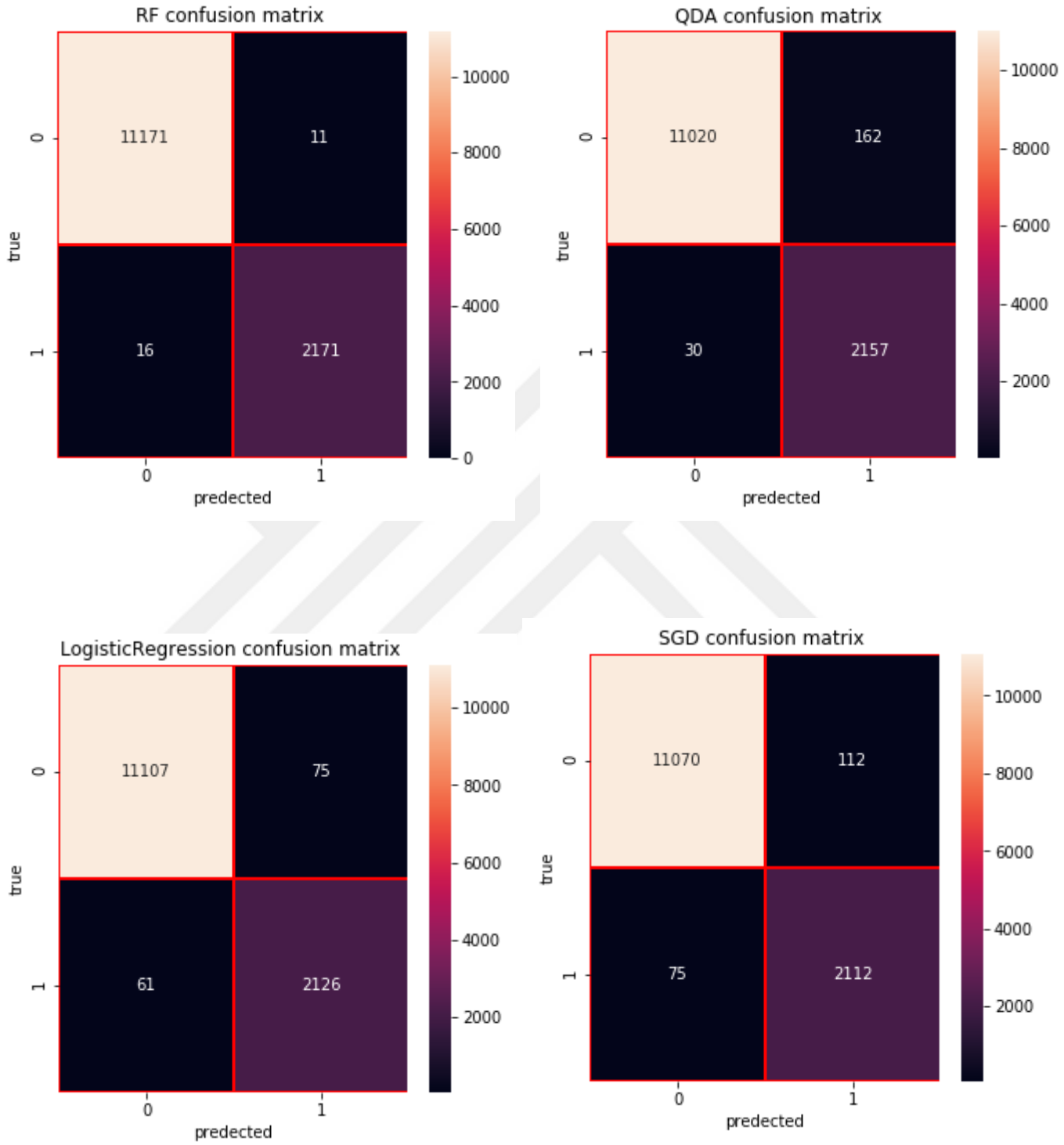

Figure 5.1: The implementation of the confusion matrix for All Dos Attack

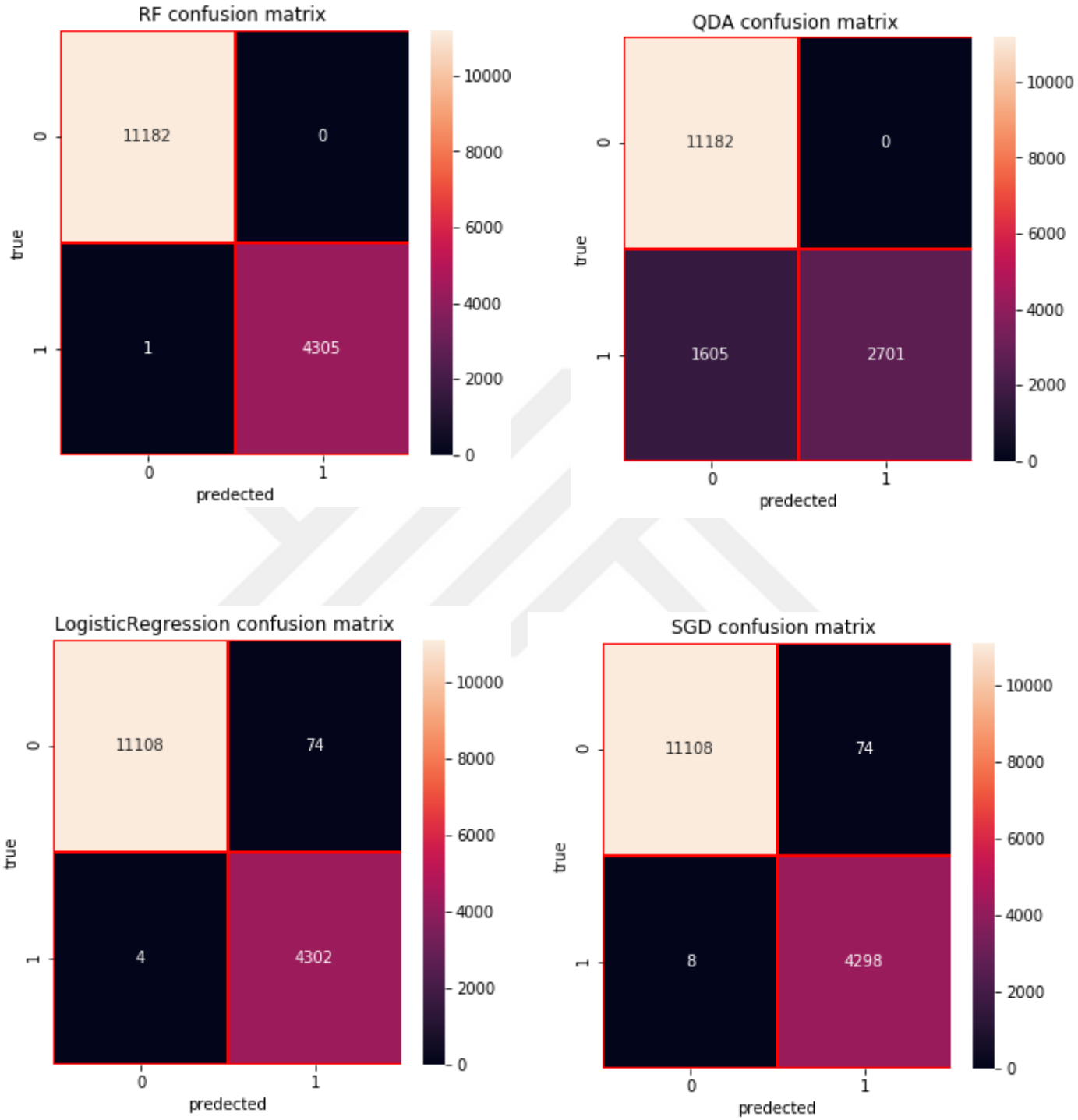

Figure 5.2: The implementation of the confusion matrix for All DDos Attack

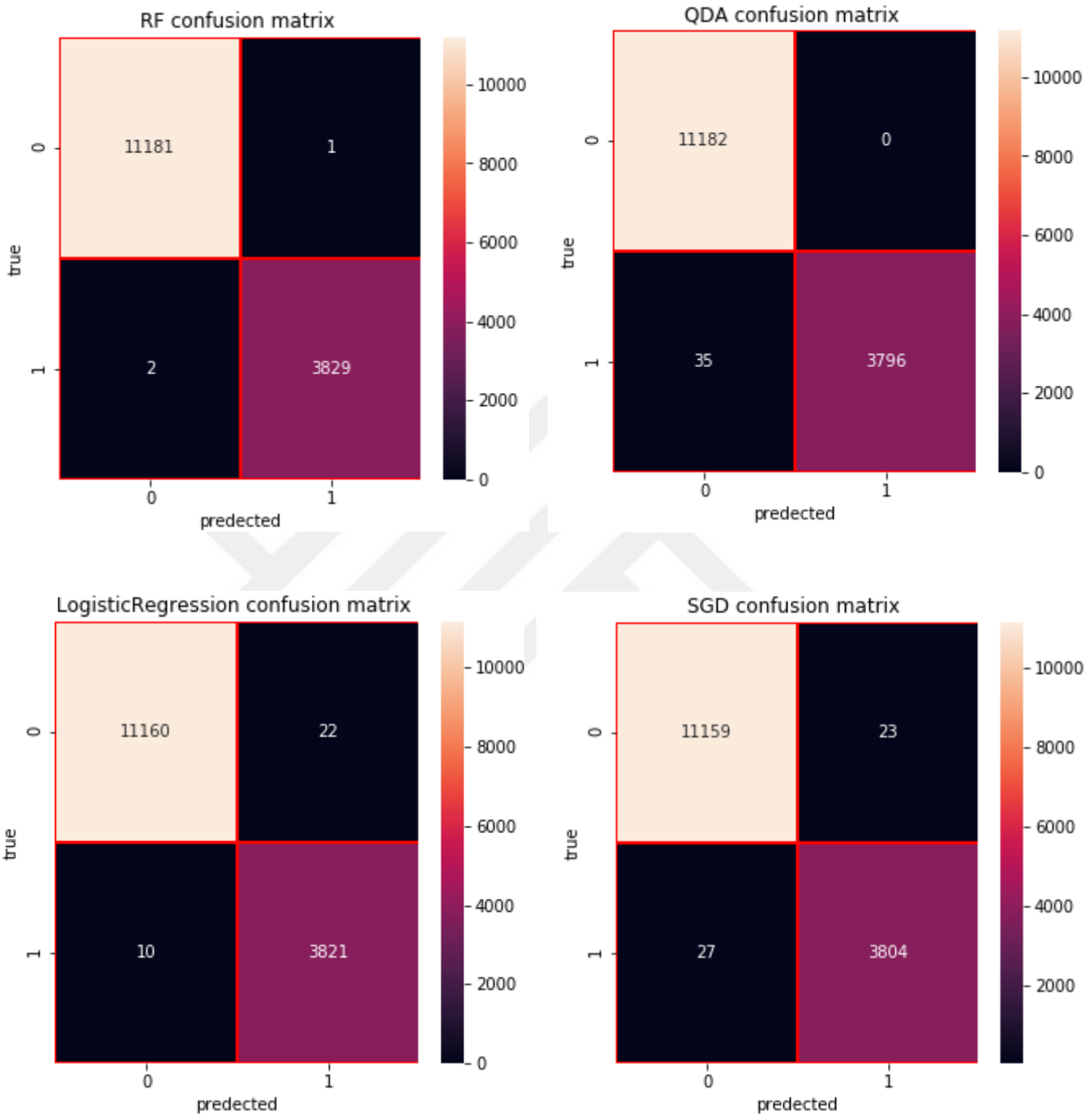

Figure 5.3: The implementation of the confusion matrix for PortScan Attack

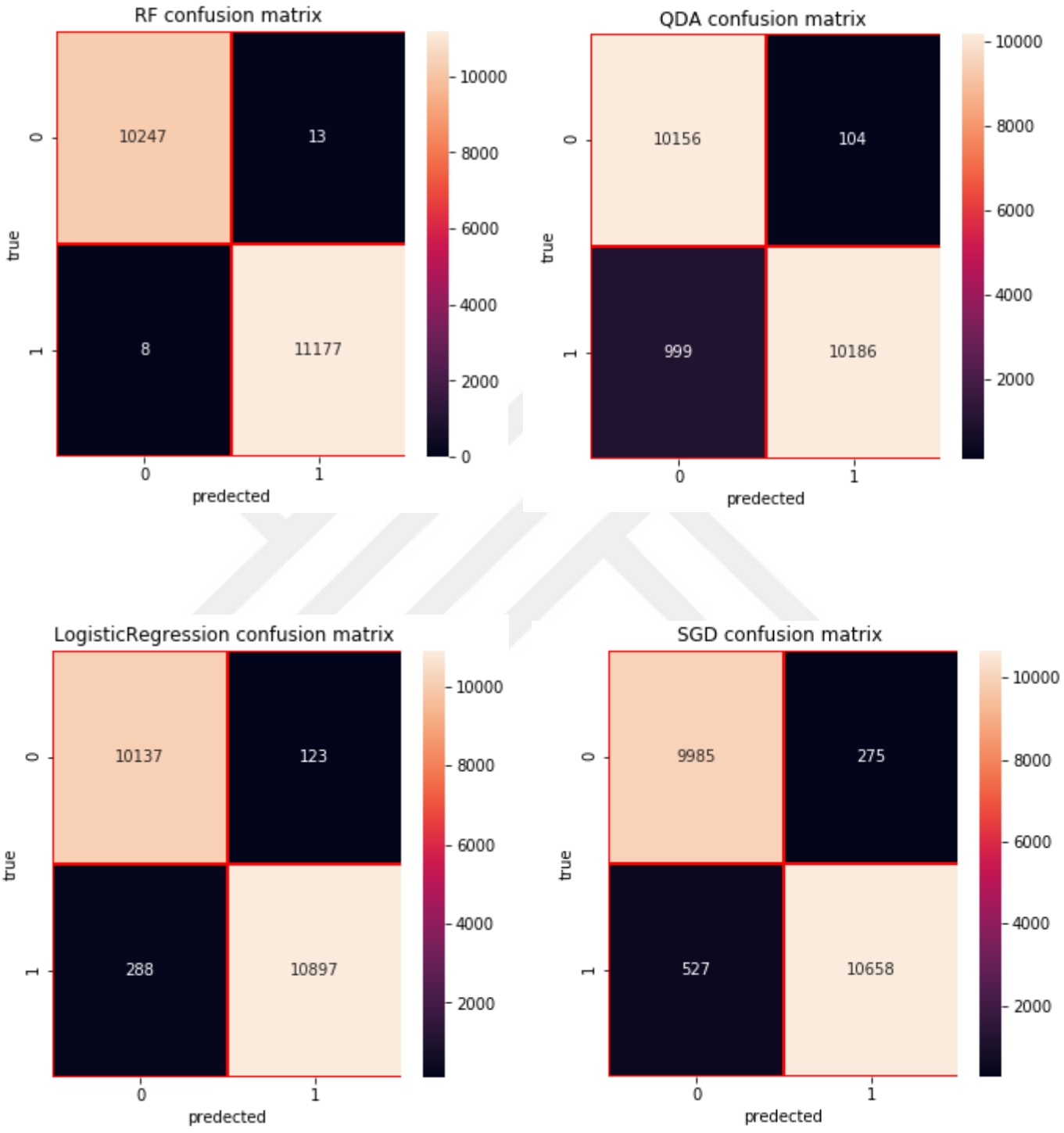

Figure 5.4: The implementation of the confusion matrix for

Dos, DDos and PortScan Attacks the process times of all algorithms are reduced, The reason for this reduction in execution time is that 20 and 10 attributes are used instead of all features and we get good result as we can see in the previous tables and only 10 attributes are used This reduction in feature count has reduced the running time of machine learning algorithms RF and ANN but the time for feature selection was too long and that's a negative point of wrapper based method that's cost a lot of computer resources and time and the positive point is high accuracy, un like Wrapper-based method the embedded method are faster with a lower accuracy .

5.2 EVALUATION

In this study we compare our results to other researchers and we find various different result but in most of the models we get better result by using machine learning and neural network because this study focus on data preprocessing and slandered scalar which increase the score because machine learning tend to work better with well-distributed data and for the outliners deleting three types of attack in multi-classification model because lack of simples and as we saw the lowest score was for attack type with lower number of simples and for the binary model we focus on Dos, DDos and Port Scan attacks .

Feature selection in more than one method un like other researchers they focus on the model design without taking care of data preprocessing because in machine learning models they used weka to apply algorithms and feature selection, so in model 2 we use the embedded method with random forest to select best 20 feature.

Table 5.10: Compare this study results with Previous[55]

Machine Learning Algorithms	Our study				Previous [55]			
	F-Measure	Precision	Recall	Score	F-Measure	Precision	Recall	Score
QDA	**0.94**	**0.91**	**0.91**	**0.94**	0.86	0.87	0.88	0.88
Random Forest	**0.99**	**0.99**	**0.99**	**0.99**	0.94	0.94	0.94	0.94
SGD	0.96	0.94	0.97	0.96	-	-	-	-
LG	0.98	0.97	0.98	0.98	-	-	-	-

In model 1 we use neural network and random forest to classify dataset because both of them work better than other algorithms in over fitting, iterations and missing

Feature selection in this model warped base method it's one of the best methods with high accuracy although its so expensive we achieved the best results in the model as we can see in table 5.11.

Data with good preprocessing for the dataset leads to get a better result in most of the attacks and as we can see from the results and as we can notice attacks with a low number of attacks have the lowest score but the others with the high number of simples there score are better.

Table 5.11: Compare this Random Forest DoS Attack of our study results with Previous[8]

Machine Learning Algorithms	Our study				Previous [8]			
	F-Measure	Precision	Recall	Score	F-Measure	Precision	Recall	Score
Random Forest	**0.99**	**0.99**	**0.99**	**0.99**	0.96	0.97	0.96	0.98

Table 5.12: Compare this ANN of our study results with Previous[7]

Attack type	Our study			Previous[7]		
	Precision	Recall	Score	Precision	Recall	Score
DDoS	0.94	**0.99**	**0.98**	0.98	0.96	0.95
PortScan	**0.99**	0.66	**0.98**	0.96	0.93	0.92

5.3 CONCLUSION AND FUTURE WORK

5.3.1 Conclusion

In this study classifying the CICIDS2017 dataset because it's up to date in order to detect network threats using machine learning and ANN, the dataset contains 79 features defining network flow. the importance of weight calculation was made with the Random Forest [18] algorithm to decide which of these features will be used in machine learning methods. Two approaches have been used when making these calculations. In the first place, The best 10 importance weights are calculated separately for selected attack types by a warped based method. In the second method, The selected attacks are collected under a single group and select 10 of heights weights for this group are calculated. that is, the common properties that are important for DOS, DDoS and Port Scan attacks. Finally, machine learning algorithms, which are widely used and have different qualities, have been applied to this data. These algorithms and the achieved performance ratios according to F-measure are as follows (F-measure takes a value between 0 and1, Random Forest: 0.99, QDA: 0.94, SGD: 0.96, LG: 0.98 and ANN: 0.98.

It's clear after getting the results most of the attacks are related and that makes multi-classification for all attack types are important and building model with best-selected features to increase the speed of the processing time so classification attacks don't need for a more complex method to detect attacks

5.3.2 Future Work

Many algorithms can be used for classification. In this thesis, the most common classification algorithms were selected for comparison with other studies. Many people choose to go with a full set of algorithms to see more results, but the positive thing about using only these are that they can be improved and this is our aim.

However, a data set consisting of 8 CSV files containing 79 features obtained from the network traffic headers were used for training and test. but as we can see this method is not practically viable in real systems .so by adding a module that catches and record the traffic network data and makes it workable with the machine learning algorithm in our models.

And to make a solution for the lack of some attacks by focusing on them and build a multi-classification model for all attack type with the best-selected weights to decrease the process time.

REFERENCES

[1] K. Kostas, "Anomaly Detection in Networks Using Machine Learning," Research Proposal, 23 Mar 2018, 2018.

[2] Raghunath, B. R., & Mahadeo, S. N. (2008). Network Intrusion Detection System (NIDS). 2008 First International Conference on Emerging Trends in Engineering and Technology.doi: 10.1109/icetet.2008.252.

[3] R. A. Kemmerer and G. Vigna, “Intrusion Detection: A Brief History and Overview,” pp. 27–30, 2002.

[4] Arash Habibi Lashkari, Gerard Draper-Gil, Mohammad Saiful Islam Mamun and Ali A. Ghorbani, "Characterization of Tor Traffic Using Time-Based Features", In the proceedings of the 3rd International Conference on Information System Security and Privacy, SCITEPRESS, Porto, Portugal, 2017.

[5] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, "Towards a reliable intrusion detection benchmark dataset," Software Networking, vol. 1, no. 1, pp. 177-200, 2017.

[6] Yu-Yang Zhou and Guang Cheng " An Efficient Network Intrusion Detection System Based on Feature Selection and Ensemble Classifier ", School of Cyber Science and Engineering, Southeast University 2019.

[7] Saddam Hussein and Anirudh Janagam, " Analysis of Network Intrusion Detection System with Machine Learning Algorithms (Deep Reinforcement Learning Algorithm) "Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden October 2018

[8] VILHELM GUSTAVSSON "Machine Learning for a Network-based Intrusion Detection System "EXAMENSARBETE INOM ELEKTRONIK OCH DATORTEKNIK, GRUNDNIVÅ, 15 HP STOCKHOLM, SVERIGE 2019.

[9] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," Journal of Network and Computer Applications, vol. 60, pp. 19-31, 2016.

[10] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization”, 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018.

[11] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., … Wang, C. (2018). Machine Learning and Deep Learning Methods for Cybersecurity. IEEE Access, 6, 35365–35381.doi:10.1109/access.2018.2836950.

[12] DDoS. Ello social network hit by suspected bloody DDoS attack. [Online]. Available: http://www.ddosattacks.net/ello-socialnetwork-hit-by-suspected-bloody-dDoS-attack (2014).

[13] Choudhary, R., & Gianey, H. K. (2017). Comprehensive Review On Supervised Machine Learning Algorithms. 2017 International Conference on Machine Learning and Data Science (MLDS).doi:10.1109/mlds.2017.11

[14] Tamilselvi, P., & Kumar, K. A. (2017). Unsupervised machine learning for clustering the infected leaves based on the leaf-colors. 2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM).doi: 10.1109/iconstem.2017.8261265.

[15] Pramokchon, P., & Piamsa-nga, P. (2016). Effective threshold estimation for filter-based feature selection. 2016 International Computer Science and Engineering Conference (ICSEC). DOI:10.1109/icsec.2016.7859919.

[16] El Aboudi, N., & Benhlima, L. (2016). Review on wrapper feature selection approaches. 2016 International Conference on Engineering & MIS (ICE). DOI:10.1109/icemis .2016.7745366.

[17] Liu, H., Zhou, M., & Liu, Q. G. (2019). An embedded feature selection method for imbalanced data classification. IEEE/CAA Journal of Automatica Sinica, 1–13. DOI:10.1109/jas .2019.1911447.

[18] Bernard, S., Heutte, L., & Adam, S. (2009). On the selection of decision trees in Random Forests. 2009 International Joint Conference on Neural Networks.doi: 10.1109/ijcnn.2009.5178693.

[19] Zhou, Y., & Yan, J. (2016). A Logistic Regression-Based Approach for Software Test Management. 2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). DOI: 10.1109/cyberc.2016.59.

[20] Pal, S., Woods, R. P., Panjiyar, S., Sowell, E., Narr, K. L., & Joshi, S. H. (2017). A Riemannian Framework for Linear and Quadratic Discriminant Analysis on the Tangent Space of Shapes. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).doi:10.1109/cvprw.2017.102.

[21] McSherry, T. (1976). A General Steepest Descent Algorithm. IEEE Transactions on Aerospace and Electronic Systems, AES-12(1), 12–22. DOI:10.1109/taes.1976.308210

[22] Mishra, M., & Srivastava, M. (2014). A view of Artificial Neural Network. 2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014).doi:10.1109/icaetr.2014.7012785 .

[23] Sasikala, B. S., Biju, V. G., & Prashanth, C. M. (2017). Kappa and accuracy evaluations of machine learning classifiers. 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT).doi:10.1109/rteict.2017.8256551 .

[23] Chao-yang, Z. (2011). DOS Attack Analysis and Study of New Measures to Prevent. 2011 International Conference on Intelligence Science and Information Engineering. DOI:10.1109/isie.2011.66.

[24] M. Chhabra, B. Gupta, and A. Almomani, "A novel solution to handle DDOS attack in MANET," Journal of Information Security, vol. 4, no. 03, p. 165, 2013.

[25] Douligeris, C., & Mitrokotsa, A. (n.d.). DDoS attacks and defense mechanisms: a classification. Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology (IEEE Cat. No.03EX795). DOI: 10.1109/isspit.2003.1341092.

[26] Geetha, K., & Sreenath, N. (2014). SYN flooding attack — Identification and analysis. International Conference on Information Communication and Embedded Systems (ICICES2014). DOI:10.1109/icices.2014.7033828.

[27] Yatagai, T., Isohara, T., & Sasase, I. (2007). Detection of HTTP-GET flood Attack Based on Analysis of Page Access Behavior. 2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing. DOI: 10.1109/pacrim.2007.4313218.

[28] Gadge, J., & Patil, A. A. (2008). Port scan detection. 2008 16th IEEE International Conference on Networks. DOI: 10.1109/icon.2008.4772622.

[29] Papadimitriou, S., & Moussiades, L. (2018). Mac OS versus FreeBSD: A Comparative Evaluation. Computer, 51(2), 44–53. DOI:10.1109/mc.2018.1451648

[30] Bokhari, S. N. (1995). The Linux operating system. Computer, 28(8), 74–79. DOI: 10.1109/2.402081.

[31] Peng Chao, Yang Chao, Niu Zhen-zhou, & Liu Xiang-peng. (2009). Research on Windows operating system education. 2009 IEEE International Symposium on IT in Medicine & Education. DOI:10.1109/itime.2009.5236327

[32] Poongothai, M., & Sathyakala, M. (2012). Simulation and analysis of DDoS attacks. 2012 International Conference on Emerging Trends in Science, Engineering, and Technology (INCOSET). DOI:10.1109/incoset.2012.6513885 .

[33] R, V., Alazab, M., KP, S., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep Learning Approach for Intelligent Intrusion Detection System. IEEE Access, 1–1. DOI:10.1109/access .2019.2895334.

[34] Hofstede, R., Jonker, M., Sperotto, A., & Pras, A. (2017). Flow-Based Web Application Brute-Force Attack and Compromise Detection. Journal of Network and Systems Management, 25(4), 735–758. DOI:10.1007/s10922-017-9421-4

[35] Yevsieieva, O., & Helalat, S. M. (2017). Analysis of the impact of the slow HTTP DOS and DDOS attacks on the cloud environment. 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T). DOI:10.1109/infocommst.2017.8246453.

[36] K. Hong, Y. Kim, H. Choi, and J. Park. SDN-Assisted Slow HTTP DDoS Attack Defense Method. IEEE Communications Letters, PP(99):1–1, 2017.

[37] "slowhttptest," GitHub. [Online]. Available: https://github.com/shekyan/slowhttptest/wiki. [Accessed: 12 Aug 2018].

[38] Zhang, L., Yu, S., Wu, D., & Watters, P. (2011). A Survey on Latest Botnet Attack and Defense. 2011IEEE 10th International Conference on Trust, Security, and Privacy in Computing and Communications. DOI:10.1109/trustcom.2011.11 .

[39] Gautam, T., & Jain, A. (2015). Analysis of brute force attack using TG — Dataset. 2015 SAI Intelligent Systems Conference (IntelliSys). DOI:10.1109/intellisys .2015.7361263.

[40] Yusof, I., & Pathan, A.-S. K. (2014). Preventing persistent Cross-Site Scripting (XSS) attack by applying pattern filtering approach. The 5th International Conference on Information and Communication Technology for The Muslim World (ICT4M). DOI:10.1109/ict4m.2014.7020628

[41] Mukherjee, S., Sen, P., Bora, S., & Pradhan, C. (2015). SQL Injection: A sample review. 2015 6th International Conference on Computing, Communication, and Networking Technologies (ICCCNT). DOI: 10.1109/icccnt.2015.7395166.

[42] Reaves, B., & Morris, T. (2009). Discovery, infiltration, and denial of service in a process control system wireless network. 2009 eCrime Researchers Summit. DOI:10.1109/ecrime.2009.5342612

[43] Kyatam, S., Alhayajneh, A., & Hayajneh, T. (2017). Heartbleed attacks implementation and vulnerability. 2017 IEEE Long Island Systems, Applications, and Technology Conference (LISAT). DOI: 10.1109/lisat.2017.8001980.

[44] Lovejoy, M. R., & Wickert, M. A. (2015). Using the IPython notebook as the computing platform for signals and systems courses. 2015 IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE). DOI:10.1109/dsp-spe.2015.7369568

[45] "scikit-learn," scikit-learn 0.19.1 documentation. [Online]. Available: http://scikit-learn.org/stable/. [Accessed: 18-Aug-2018].

[46] “keras.layers.Dense,” TensorFlow Core r2.0

documentation. [Online]. Available: https://www.tensorflow.org. [Accessed: 04-Oct-2019].

[47] “Python Data Analysis Library,” pandas: powerful Python data analysis toolkit - pandas 0.22.0 documentation. [Online]. Available: https://pandas.pydata.org/.[Accessed: 04-Oct-2019].

[48] “Matplotlib: Python plotting - Matplotlib 2.2.2 documentation,” Matplotlib. [Online]. Available: https://matplotlib.org/. [Accessed: 04-Oct-2019].

[49] “NumPy,” NumPy developers. [Online]. Available: http://www.numpy.org/. [Accessed: 4-Oct-2019].

[50] “sklearn.preprocessing.LabelEncoder,” 1.4. Support Vector Machines - scikit-learn 0.19.1 documentation. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html. [Accessed: 19-Aug-2018].

[51] “train_test_split,” scikit-learn 0.19.1 documentation. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html. [Accessed: 04-Oct-2019].

[52] “Early Stopping,” by Jason Brownlee on December 10, 2018, in Deep Learning Performance [Online]. Available: https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/. [Accessed: 04-Oct-2019].

[53] “ModelCheckpoint,” by Jason Brownlee on June 15, 2016, in Deep Learning Available: https://machinelearningmastery.com/check-point-deep-learning-models-keras/[Accessed: 04-Oct-2019].

[54] “Random.seed,” by How to Get Reproducible Results with Kerasby Jason Brownlee on June 14, 2017, in Deep Learning Available: https://machinelearningmastery.com/reproducible-results-neural-networks-keras/ [Accessed: 04-Oct-2019].

[55] K. Kostas, “Anomaly Detection in Networks Using Machine Learning,” Thesis for Master of Science in Computer Networks and Security, Advisor: Martin Reedno. August 2018.

[56]M. Suresh and R. Anitha, "Evaluating machine learning algorithms for detecting DDoS attacks," in International Conference on Network Security and Applications, 2011, pp. 441-452: Springer.

[57]S. Mukherjee and N. Sharma, "Intrusion detection using naive Bayes classifier with feature reduction," Procedia Technology, vol. 4, pp. 119-128, 2012.

[58]W. Yassin, N. I. Udzir, Z. Muda, and M. N. Sulaiman, "Anomaly-based intrusion detection through k-means clustering and naives bayes classification," in Proc. 4th Int. Conf. Comput. Informatics, ICOCI, 2013, no. 49, pp. 298-303.

[59] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, "Towards a reliable intrusion detection benchmark dataset," Software Networking, vol. 1, no. 1, pp. 177-200, 2017.

APPENDIX A

FEATURE'S EXPLANATIONS

The list from http://www.netflowmeter.ca/netflowmeter.html

Feature Name	Description
Feduration	Duration of the flow in Microsecond
total_fpackets	Total packets in the forward direction
total_bpackets	Total packets in the backward direction
total_fpktl	Total size of the packet in the forward direction
total_bpktl	Total size of packet in the backward direction
min_fpktl	The minimum size of the packet in forwarding direction
min_bpktl	The minimum size of the packet in the backward direction
max_fpktl	Maximum size of the packet in the forward direction
max_bpktl	Maximum size of the packet in the backward direction
mean_fpktl	Mean size of the packet in the forward direction
mean_bpktl	Mean size of the packet in the backward direction
std_fpktl	Standard deviation size of packet in forwarding direction
std_bpktl	Standard deviation size of the packet in the backward direction
total_fiat	Total time between two packets sent in the forward direction

total_biat	Total time between two packets sent in the backward direction
min_fiat	Minimum time between two packets sent in the forward direction
min_biat	Minimum time between two packets sent in the backward direction
max_fiat	Maximum time between two packets sent in the forward direction
max_biat	Maximum time between two packets sent in the backward direction
mean_fiat	Meantime between two packets sent in the forward direction
mean_biat	Meantime between two packets sent in the backward direction
std_fiat	Standard deviation time between two packets sent in the forward direction
std_biat	Standard deviation time between two packets sent in the backward direction
fpsh_cnt	Number of times the PSH flag was set in packets traveling in the forward direction (0 for UDP)
bpsh_cnt	Number of times the PSH flag was set in packets traveling in the backward direction (0 for UDP)
furg_cnt	Number of times the URG flag was set in packets traveling in the forward direction (0 for UDP)
burg_cnt	Number of times the URG flag was set in packets traveling in the backward direction (0 for UDP)

total_fhlen	Total bytes used for headers in the forward direction
total_bhlen	Total bytes used for headers in the backward direction
fPktsPerSecond	Number of forwarding packets per second
bPktsPerSecond	Number of backward packets per second
flowPktsPerSecond	Number of flow packets per second
flowBytesPerSecond	Number of flow bytes per second
min_flowpktl	Minimum length of a flow
max_flowpktl	The maximum length of a flow
mean_flowpktl	Mean length of a flow
std_flowpktl	Standard deviation length of a flow
min_flowiat	Minimum inter-arrival time of packet
max_flowiat	Maximum inter-arrival time of packet
mean_flowiat	Mean inter-arrival time of packet
std_flowiat	Standard deviation inter-arrival time of packet
flow_fin	Number of packets with FIN
flow_syn	Number of packets with SYN
flow_rst	Number of packets with RST
flow_psh	Number of packets with PUSH
flow_ack	Number of packets with ACK
flow_urg	Number of packets with URG
flow_cwr	Number of packets with CWE

flow_ece	Number of packets with ECE
downUpRatio	Download and upload ratio
avgPacketSize	Average size of packet
fAvgSegmentSize	Average size observed in the forward direction
fAvgBytesPerBulk	The average number of bytes bulk rate in the forward direction
fAvgPacketsPerBulk	The average number of packets bulk rate in the forward direction
fAvgBulkRate	The average number of bulk rate in the forward direction
bAvgSegmentSize	Average size observed in the backward direction
bAvgBytesPerBulk	The average number of bytes bulk rate in the backward direction
bAvgPacketsPerBulk	The average number of packets bulk rate in the backward direction
bAvgBulkRate	The average number of bulk rate in the backward direction
sflow_fpacket	The average number of packets in a sub-flow in the forward direction
sflow_fbytes	The average number of bytes in a sub-flow in the forward direction
sflow_bpacket	The average number of packets in a sub-flow in the backward direction
sflow_bbytes	The average number of bytes in a sub-flow in the backward direction

min_active	The minimum time a flow was active before becoming idle
mean_active	Meantime a flow was active before becoming idle
max_active	The maximum time a flow was active before becoming idle
std_active	Standard deviation time a flow was active before becoming idle
min_idle	The minimum time a flow was idle before becoming active
mean_idle	Meantime a flow was idle before becoming active
max_idle	The maximum time a flow was idle before becoming active
std_idle	Standard deviation time a flow was idle before becoming active
Init_Win_bytes_forward	The total number of bytes sent in the initial window in the forward direction
Init_Win_bytes_backward	The total number of bytes sent in the initial window in the backward direction
Act_data_pkt_forward	Count of packets with at least 1 byte of TCP data payload in the forward direction
min_seg_size_forward	Minimum segment size observed in the forward direction