Distributed Denial-of-Service attack Detection Using ML Based Classification Tools

A project report submitted in partial fulfillment of the requirement

for the award of the degree of

Master of Computer Applications

by

Sarad Kumar Yadav

(Regd.No.-2019PGCACA63)

Under the supervision of

Dr. Alekha Kumar Mishra

(Internal Supervisor)

NIT Jamshedpur

Department of Computer Science & Engineering

National Institute of Technology Jamshedpur
Jamshedpur (INDIA)

DECLARATION

I hereby declare that dissertation of the work titled, **"Distributed Denial-of-Service Attack Detection using ML Based Classification tools"**, submitted towards requirements of project work for partial fulfillment of Master in Computer Applications is an original work of mine and the report has not formed the basis for the award of any other degree, associate ship, fellowship or similartitles. I also declare that wherever I have used materials such as data, theoretical analysis, and text from other sources, I have given due credit to them by citing source of the work in the thesis.

Sarad Kumar Yadav

Reg.No:2019PGCACA63

Dept. of Computer Science & Engineering

NIT Jamshedpur

Place: Azamgarh, Uttar Pradesh
Date: 09/05/2022

Department of Computer Science and Engineering
National Institute of Technology, Jamshedpur
(An Institution of National Importance under MHRD, Govt. of India, New Delhi)

CERTIFICATE

This is to certify that the thesis titled "**Distributed Denial-of-service Attack Detection using ML Based Classification tools**", submitted by **Sarad Kumar Yadav (Reg. No.: 2019PGCACA63)** towards partial fulfillment of the requirements for the award of degree of Master of Computer Applications, is a bona fide work carried out under the supervision and guidance of mine. I have worked on my project from 10/01/2022 to 10/05/2022.

Dr. Alekha Kumar Mishra

(Supervisor)

ACKNOWLEDGEMENT

It gives me immense pleasure to express my deep sense of gratitude to my supervisor **Dr. Alekha Kumar Mishra** for his valuable guidance, motivation, constant inspiration and above all for their ever-cooperating attitude that enable me in bringing up this thesis in the present form.

My heartfelt gratitude also goes to **Dr. Sanjay Kumar**, Head of Department of Computer Science and Engineering. for providing me the opportunity to avail the excellent facilities and infrastructure.I am equally thankful to all other faculty members and non-teaching staffs of Computer Science and Engineering Department for their guidance and support.

I am also thankful to all my family members whose love, affection, blessings and patience encouraged me to carry out this thesis successfully. I also extend my gratitude to all my friends for their cooperation.

I thank Almighty God, my lord for giving me the will power and strength to make it happen. Lastly, I thank myself for putting for sheer hard work, dedication and perseverance.

Sarad Kumar Yadav

ABSTRACT

Denial of Service attacks (DOS) are among the most dangerous cyber security threats affecting server platforms. Such attacks target a server, app or service platforms by sending a huge amount of malicious traffic with the aim to overload its computational (e.g., CPU load level) or networks resources. The most challenging DOS type is the distributed DOS (DDOS), as a pool of multiple source attackers with different and, often, dynamic/spoofed IP addresses perform a combined attack action. Blocking these types of attacks is difficult since standard IP-address blacklist countermeasures, based on static policies, are not effective. The most utilized DDOS attacks are typically grouped in the following categories: TCP SYN flood, UDP flood, ICMP flood and HTTP flood. In this thesis, we select two ML methods to predict DDOs attack
Use different Machine learning algorithm to detect DDOS Attack like Logistic Regression and MLP Classifiers. Finally, the DDoS detection scheme was tested by simulation experiment. The test results showed that the method could effectively detect DDoS, with an average success rate.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1Introduction

A distributed denial-of-service (DDoS) attack is a malicious attempt to disrupt the normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic. An attack can have devastating results. For individuals, this includes unauthorized purchases, the stealing of funds, or identify theft[1].

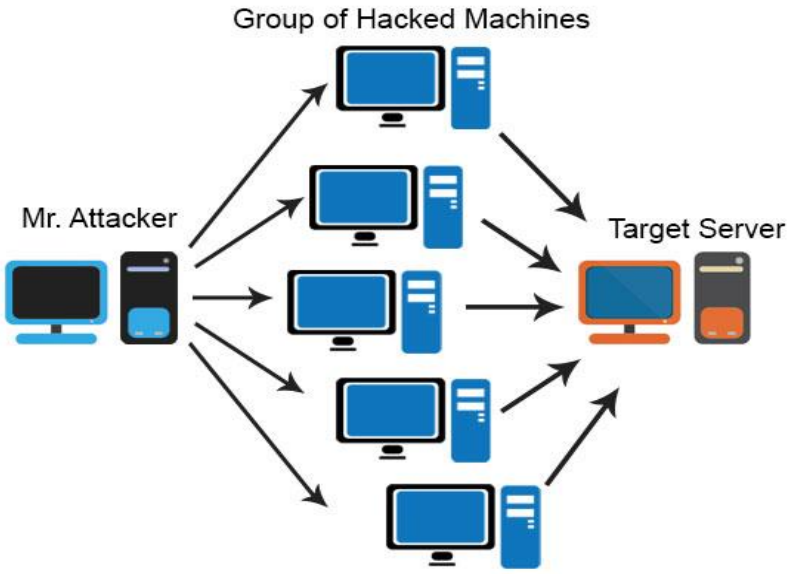


Figure 1: Overview of DDOS attack Detection[1]

DDOS attacks are carried out with networks of Internet-connected machines. These networks consist of computers and other devices (such as IoT devices) which have been infected with malware, allowing them to be controlled remotely by an attacker. These individual devices are referred to as bots (or zombies), and a group of bots is called a botnet [1]. Once a botnet has been established, the attacker is able to direct an attack by sending remote instructions to each bot.

When a victim's server or network is targeted by the botnet, each bot sends requests to the target's IP address, potentially causing the server or network to become overwhelmed, resulting in a denial-of-service to normal traffic. Because each bot is a legitimate Internet device, separating the attack traffic from normal traffic can be difficult.

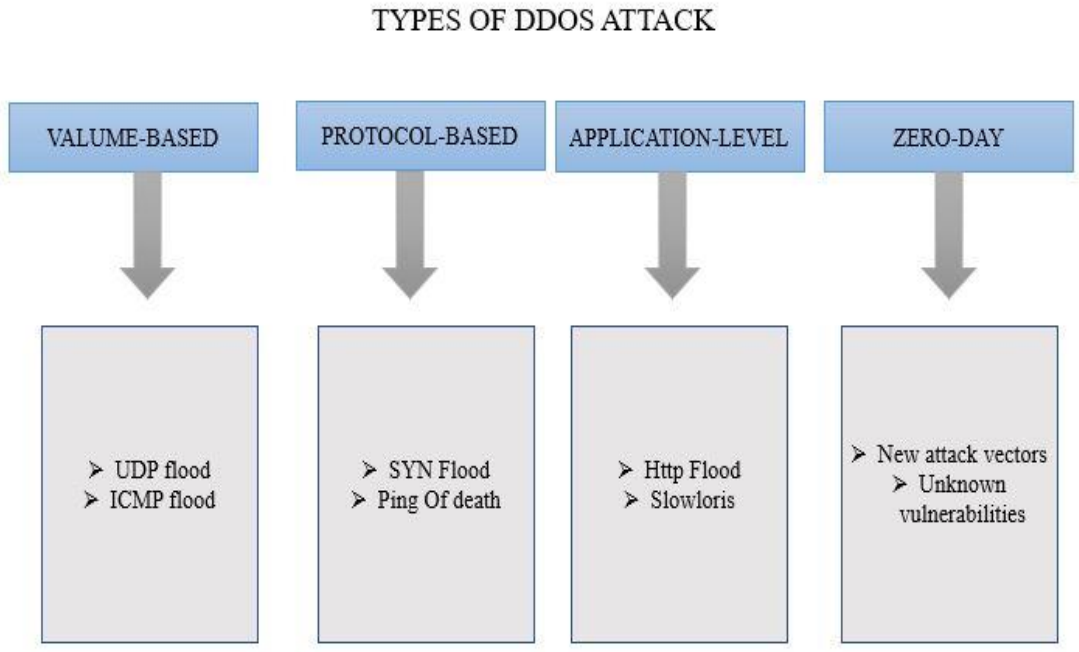


Figure 2: Types of DDOS attack Detection [3]

Broadly speaking, DoS and DDoS attacks can be divided into three types[1,2]:

1.1.1 Volume Based Attacks :

Includes UDP floods, ICMP floods, and other spoofed-packet floods.

A UDP flood, by definition, is any DDoS attack that floods a target with User Datagram Protocol (UDP) packets. The goal of the attack is to flood random ports on a remote host. This causes the host to repeatedly check for the application listening at that port, and (when no application is found) reply with an ICMP 'Destination Unreachable' packet. This process saps host resources, which can ultimately lead to inaccessibility[2].

1.1.2 Protocol Attacks:

Includes SYN floods, fragmented packet attacks, Ping of Death, Smurf DDOS and more.

A SYN flood DDOS attack exploits a known weakness in the TCP connection sequence (the "three-way handshake"), wherein a SYN request to initiate a TCP connection with a host must be answered by a SYN-ACK response from that host, and then confirmed by an ACK response from the requester. In a SYN flood scenario, the requester sends multiple SYN requests, but either does not respond to the host's SYN-ACK response, or sends the SYN requests from a spoofed IP address. Either way, the host system continues to wait for acknowledgement for each of the requests, binding resources until no new connections can be made, and ultimately resulting in denial of service[1].

1.1.3 Application Layer Attacks:

Includes low-and-slow attacks, GET/POST floods, attacks that target Apache. In an HTTP flood DDOS attack, the attacker exploits seemingly-legitimate HTTP GET or POST requests to attack a web server or application. HTTP floods do not use malformed packets, spoofing or reflection techniques, and require less bandwidth than other attacks to bring down the targeted site or server. The attack is most effective when it forces the server or application to allocate the maximum resources possible in response to every single request.

1.2 Problem Statement

Preventing DDOS attack is very difficult as attackers are targeting server from different location of world . And To track the IP address is very difficult . There are different types of hidden dynamic features . With the massive work exists for DDOS detection task, there is no set of features that has been determined as the best to detected Attack. Moreover, the same non deterministic scenario is applied for the underling classification algorithm. Finally, there is a need to keep on enhancing the accuracy of the detection techniques. Overall the problems carried out in this research are as following:

1. How to determine the best set of features to be used with Attack detection.
2. How to select the best classification algorithm to be used for Attack detection.
3. How to enhance the performance of the best selected features and classifiers.

How to integrate multiple classification algorithms for DDOS detection and to evaluate such integration

1.3 Motivation

DDOS attacks are quickly becoming the most prevalent type of cyber threat, growing rapidly in the past year in both number and volume according to recent market research. The trend is towards shorter attack duration, but bigger packet-per-second attack volume. DDOS attack are almost 10 times more prevalent than in 2000 and are still the most common cybercrime in 2022. Trending attacks are uses Volume attack like UDP and ICMP floods . so I am trying to extract Features that invloves Most in attack.

1.4 Objective

The objectives of this research are as follows:

1. Determine and evaluate the best set of features to be used for Volume attack.
2. To determine the best classification algorithm for DDOS detection.
3. To fine tune the classification algorithm for best performance.
4. Compare the performance metrics of all the models.

1.5 Scope and Limitation

The scope of this research is DDOS attack detection, where features are extracted from the network. Moreover, Naive Bayes, Random Forest, Logistic Regression etc. top classification ML algorithms were used for DDOS attack detection. This research also target to develop an integration of best performing classifier for better prediction.

For the limitation, this research will not cover the types of Protocols attack, moreover the experiments will not cover all the available classification algorithms. However, this study will evaluate experimentally the most well-known algorithms.

1.6 Outline of Chapters

The thesis is consists of five chapters organized as the follows:

1. **Chapter One:** (Introduction) It gives an overview of DDOS attack and types, problem statement, the objective of the study, the motivation, the scope and limitation, thesis contribution and finally outline of chapters
2. **Chapter Two:** (Literature review) this chapter provides an overview of the related works in DDOS detection and summary of articles that published by other researchers.
3. **Chapter Three:** (Methodology) this chapter provides an outline of the research methodology which used in this thesis. Overview of the software that used for the evaluation of the proposed method and the dataset were used in this research.
4. **Chapter Four**: (Implementation) this chapter describes the implementation details of experiment and the results that were obtained for all the proposed scenarios and comparison of the results.
5. **Chapter Five**: Conclusion and future work.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

Detection of DDOS has received a lot of attention recently due to their impact on users' security. Therefore, many techniques have been developed to detect DDOS attack .Many DDOS attack detection techniques have been proposed to solve the problem. The DDOS attack itself has also frequently changed in order to evade current solutions for DDOS attack defense. The DDOS attack detection can be divided into different categories namely filtering mechanism, routers function, network flow, statistical analysis and learning machine. Some of them focus on cloud computing, software define networking (SDN), backbone web traffic [3] and big data strategies.

2.2 DDOS Detection Techniques

There are Four types of attack[2,3,8] detection have been proposed to solve the Problem.

1. Filtering mechanism
2. Routers function
3. Network flow
4. Statistical analysis and learning machine

Form above 4 Methods we discussed about statistical analysis and learning machine. According to, three defense strategies are typically employed to mitigate DDoS attacks, classified based on the location of the detection engine:

1. Source-based detection, implemented at the attacking hosts
2. Destination-based detection, implemented at the victim hosts
3. Network-based detection, implemented at the network intermediate nodes (e.g., switches, routers)

2.3 Machine Learning Methods

This method applies automated classifiers that rely on machine learning. These classifiers work beside the server and predicting attack by examining different features.[2,4,5]

2.3.1 LOGISTIC REGRESSION

The logistic regression is a widely-used method due to its easily-interpretable and practical results. This model is functional in predicting binary data (0/1 response) as it relies on statistical data and applies a generalized linear model.

Despite of this method's simplicity, it has three shortcomings; first, it requires more statistical assumptions before being applied. Second, it its more functional with variables that have linear relation than those with a complex relation. Last, the accurateness of the predication rate is sensitive to the completeness of the data [5].

2.3.2 MLP CLASSIFIER

MLPs are suitable for classification prediction problems where inputs are assigned a class or label. They are also suitable for regression prediction problems where a real-valued quantity is predicted given a set of inputs.

MLP is a type of artificial neural network (ANN). Simplest MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. In the world of deep learning, TensorFlow, Keras, Microsoft Cognitive Toolkit (CNTK), and PyTorch are very popular. It consists of three types of layers—the input layer, output layer and hidden layer, as shown in.

The input layer receives the input signal to be processed.[4]

Chapter 3

METHODOLOGY

3.1 Introduction

This chapter presents detailed project work on DDOS Attack detection. The work contribute to the field by developing an ensemble integration model by combining top 4 fine-tuned classifications techniques to enhance the detection accuracy. Moreover, a comparative assessment between various classification algorithms is proposed[1].

3.2 Project Approach

The project work starts by going through the various research paper and in estimating the Various types of DDOS Attack. Then, determining set of features that are prominent in the Attack. The project start by creating ten topologies in Mininet in which switches are connected to single Ryu controller. Network simulation runs for benign TCP, UDP and ICMP traffic and malicious traffic which is the collection of TCP Syn attack, UDP Flood attack, ICMP attack. Total 23 features are available in the data set in which some are extracted from the switches and others are calculated. and a CSV dataset is constructed by extracting the feature set values from the Network. The dataset is now subjected to various visualization plots and dimensionality reduction techniques. Feature Importance is also plotted to show case the important features in the process of classification. Various classification algorithms are then implemented upon the dataset. Top 4 classification algorithms are selected on the basis of the cross validation error. The selected models are then integrated to form an ensemble model to get better performance[3].

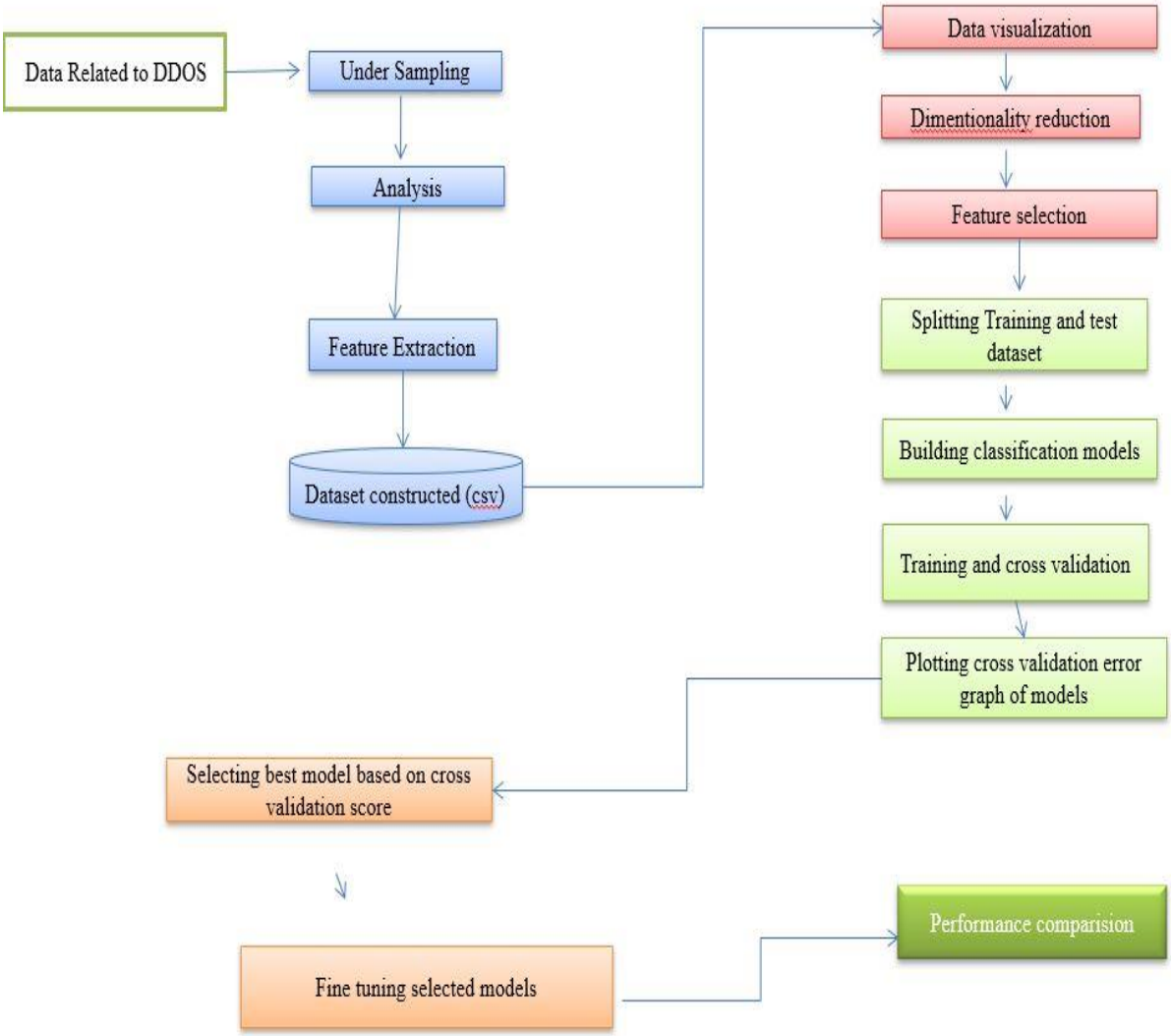


Figure 3: Workflow design of project

3.2 Pre-processing

In the pre-processing step, SDN dataset is collected. Total 23 features are available in the data set in which some are extracted from the switches and others are calculated. Extracted features include Switch-id, Packet_count, byte_count, duration_sec, duration_nsec which is duration in nano-seconds, total duration is sum of duration_sec and durstaion_nsec, Source IP, Destination IP, Port number, tx_bytes is the number of bytes transferred from the switch port, rx_bytes is the number of bytes received on the switch port. dt field show the date and time which has been converted into number and a flow is monitored at a monitoring interval of 30 second. Calculated features include Packet per flow which is packet count during a single flow, Byte per flow is byte count during a single flow, Packet Rate is number of packets send per second and calculated by dividing the packet per flow by monitoring interval, number of Packet_ins messages, total flow entries in the switch, tx_kbps, rx_kbps are data transfer and receiving rate and Port Bandwidth is the sum of tx_kbps and rx_kbps. Last column indicates the class label which indicates whether the traffic type is benign or malicious. Benign traffic has label 0 and malicious traffic has label 1. Network simulation is run for 250 minutes and 1,04,345 rows of data is collected. The simulation is run for defined interval again and more data can be collected.

These feature are presented in tables below:

Feature	Description
Switch-id	feature showing the Id of Switch_id in the Network
Packet_Count	feature showing the number of Packets
byte_count	feature showing the Number of bytes count in Network
duration_sec	feature showing the duration time in seconds.
duration_nsec	feature showing duration in nano-seconds
total duration	feature showing the sum of duration_sec and durstaion_nsec
Source IP	Ip address of Source
Destination IP	Feature showing IP of destination

Feature	Description
tx_bytes	The number of bytes transferred from the switch port
rx_bytes	The number of bytes received on the switch port
dt	field show the date and time which has been converted into number
Packet per flow	packet count during a single flow
Byte per flow	byte count during a single flow
Packet Rate	number of packets send per second and calculated by dividing the packet per flow by monitoring interval,

Packet_ins	total flow entries in the switch
tx_kbps	data transfer Rate
rx_kbps	data Recieving Rate

3.3 Data Visualization

The data set formed from the features extracted from the Network would be then subjected to various visualization. The visualization of data helps in revealing hidden trends in the data which could be proved very effective during training of our model. We went with plotting the distribution of the features data. Typically, I went for plotting PCA visualization and heatmap.

3.4 Feature Selection

Feature selection , also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. It would be applied to select only the relevant features for building the model. It is very helpful in reducing the model training time and also improve the accuracy as model is not affected by unnecessary feature variables. Feature selection is implemented using 4 filters:

1. Univariate filter
2. Variance filter
3. High Correlation filter
4. Feature importance with Extra Tree Classifier

The feature importance (variable importance) describes which features are relevant. It can help with better understanding of the solved problem and sometimes lead to model improvements by employing the feature selection. We have employed Extra Tree Classifier for evaluating the feature importance. It Extracts Top K Important features which helps to increase accuracy of Model.

3.5 Algorithm Evaluation

We have selected 2 classification algorithms to train and test the accuracy of DDOS classification with the features. The reason behind selecting these algorithms is the different training strategy they use in discovering the rules and the mechanism of learning and testing, the selected algorithms are:

1. Logistic Regression [5]
2. MLP (Multilayer Perceptrons) [4]

Chapter 4

IMPLEMENTATION

4.1 Introduction

This chapter presents the dataset that was constructed from the mininet emulator . It further throws lights upon various findings of the experiments and lastly presenting the performance of the classification algorithm.

4.2 Feature Extraction and Data set Preparation

This is a SDN specific data[11] set generated by using mininet emulator and used for traffic classification by machine learning and deep learning Algorithm.The project start by creating ten topologies in mininet in which switches are connected to single Ryu controller. Network simulation runs for benign TCP, UDP and ICMP traffic and malicious traffic which is the collection of TCP Syn attack, UDP Flood attack, ICMP attack. Total 23 features are available in the data set in which some are extracted from the switches and others are calculated. The data set is created as a part of the research work at Bennett University and can be reproduced by the steps mentioned below:

1. Create topology in mininet and choosing a random topology for sending traffic between the hosts.
2. Create a python file to collect the flow and port statistics for the duration of monitoring interval.
3. Two different CSV files are generated which contain flow and port statistics.
4. These files are merged and final data set is created

The final dataset with extracted features value is prepared and stored in CSV file for usage in coming phase.

4.3 Exploratory Data Analysis

In order to reveal hidden trends in data and to get better prediction at model building phase, we must perform EDA.

4.4 Data Cleaning

As dataset is constructed [11,12]directly from a Network by extracting the features value. No null valued cell is found. But several duplicates rows were found which were then dropped from the dataset and the final dataset has:
Number of Benign class records: 18304
Number of malicious class records: 21696 Dataset shape = (40000 x 23)

4.5 Data Visualization

For data visualization I have plotted those using different graphs. They are

4.5.1 Heatmap

Heatmap is defined as a graphical representation of data using colors to visualize the value of the matrix. In this, to represent more common values or higher activities dark colors basically reddish colors and green are used and to represent less common or activity values, brighter yellowish used . Heatmap is also defined by the name of the shading matrix. Heatmaps in Seaborn can be plotted by using the seaborn.heatmap()function.

As I can interpret from the heatmap shown in Figure 4, there is a strong high correlation between features.

a. Pkt perflow and byte perflow
b. Pkt perflow and pkt rate
c. byte count and pkt count
d. packetins and tx_bytes
e. pkt rate and pktperflow

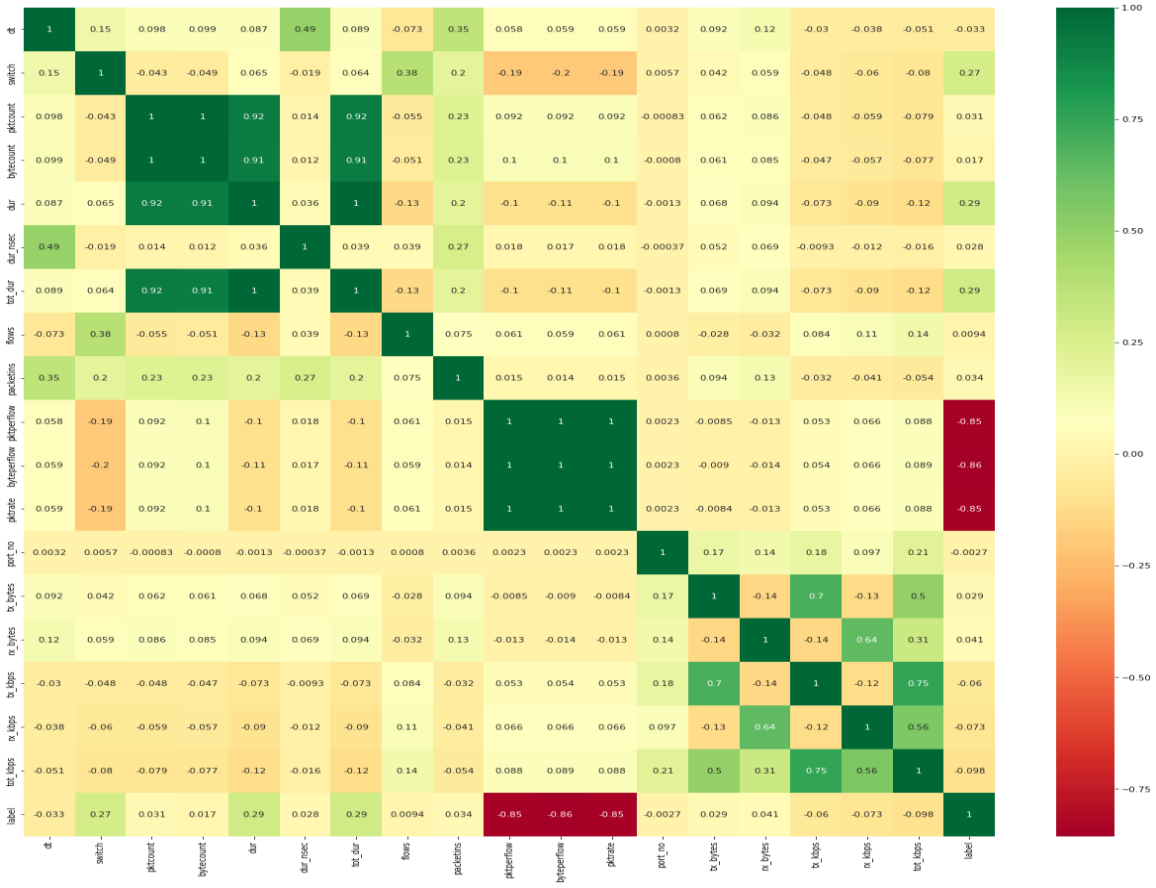


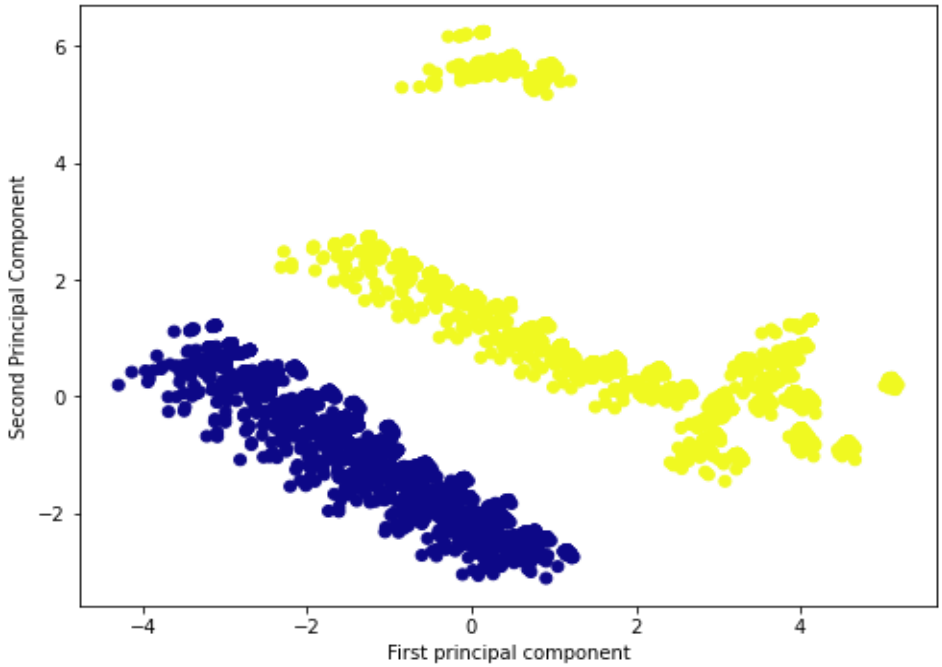
Figure 4: Heatmap

These correlations are quite obvious trend revealed in the heatmap. As number of packet count increases there is a high chance of malicious attack like Ddos attack. As in the above figure we concluded that targeted value(label) depends upon most 3 Features

1. Pkt perflow
2. byte perflow
3. pkt rate

4.5.2 PCA visualization

PCA is also used for visualization. It is a technique used to emphasize variation and bring out strong patterns in a dataset. It's often used to[8] make data easy to explore and visualize. Here,



I

Figure 5: PCA Visualization

can see that clusters are forming which means there is a separation between the both classes like malicious and benign.

4.6 Feature Selection

Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in. Having irrelevant features in your data can decrease the accuracy of the models and make your model learn based on irrelevant features. We went on implementing 3 filters to get most important features. These are:

4.6.1 Variance Filter

Variance Threshold() is a simple baseline approach to feature selection. It removes all features whose variance doesn't meet some threshold. By default, it removes all zero-

variance features, i.e. features that have the same value in all samples.

Following features are left after filteration :

```
Index(['HTML', 'body richness', 'Number of URLs', 'Malicious URL',
       'text link disparity', 'IP URLs', 'hexadecimal URL',
       'Maximum Domains Counts', 'Re: mail', 'number of dots',
       'number of dash'],
      dtype='object')
```

Figure 6: Variance filtered feature

4.6.2 Feature importance with Extra Tree Classifier

The Extra Tree Classifier algorithm has built-in feature importance which I have computed Top ten best features those are importance

for Predicting target Attribute which is label.

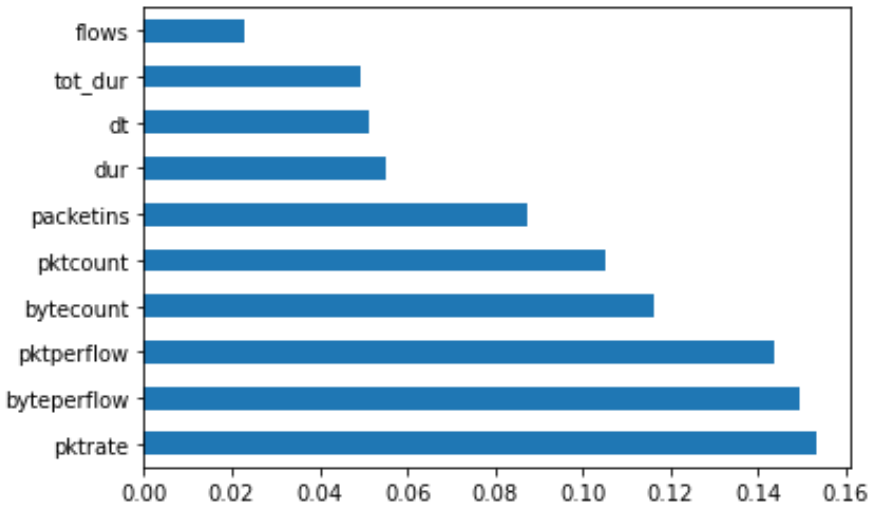


Figure 7: Features Importance with Extra Tree Classifier

4.6.3 High Correlation Filter

After filtering from above all filters, I union all the filtered features.

Resultant is 19 features are:

```
['dt', 'switch', 'pktcount', 'bytecount', 'dur', 'dur_nsec', 'tot_dur', 'flows', 'packetins', 'pktperflow', 'byteperflow', 'pkt
rate', 'port_no', 'tx_bytes', 'rx_bytes', 'tx_kbps', 'rx_kbps', 'tot_kbps', 'label']
```

Figure 8: high correlation Filter

It reveals that there is decent correlation between

a. Packet Perflow and pkt Rate

b. Byte Perflow and byte Perflow

c. packetins and tx_bytes

We as until now I have 19 features and sacrificing 4 features would be not much beneficial, we will preserve this information. Finally, we have all the 19 features selected for next phase i.e. model building. After observing mean f1 score and mean error, it is decided to select following two models for hyperparameter tuning.

a. Logistic Regression
b. MLP classifier

Following are the performances of models before hyperparameter tuning:

a) Logistic Regression

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability[5,13].

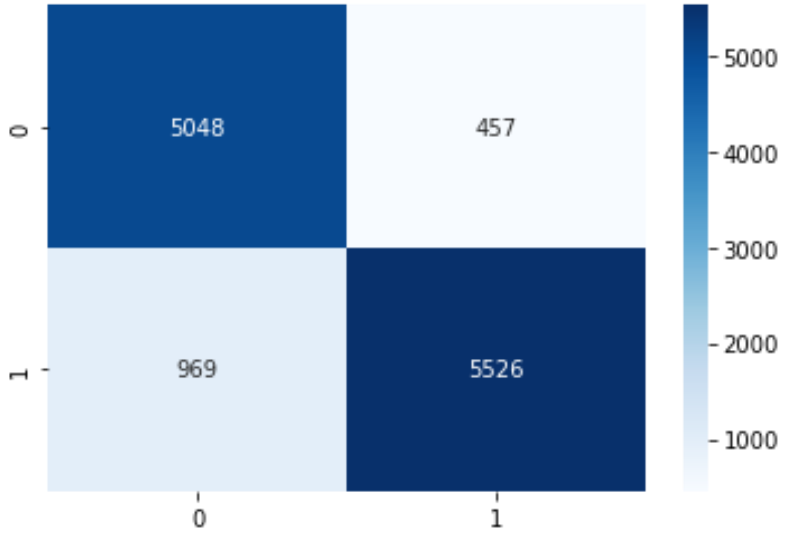


Figure 9 : Confusion Matrix of Logistic Regression Classifier

	precision	recall	f1-score	support
0	0.84	0.92	0.88	5505
1	0.92	0.85	0.89	6495
accuracy			0.88	12000
macro avg	0.88	0.88	0.88	12000
weighted avg	0.88	0.88	0.88	12000

Table 1 : Accuracy Result Table of the Logistic Regression

b) MLP classifier

It implements a multi-layer perceptron (MLP) that trains using backpropagation[4].

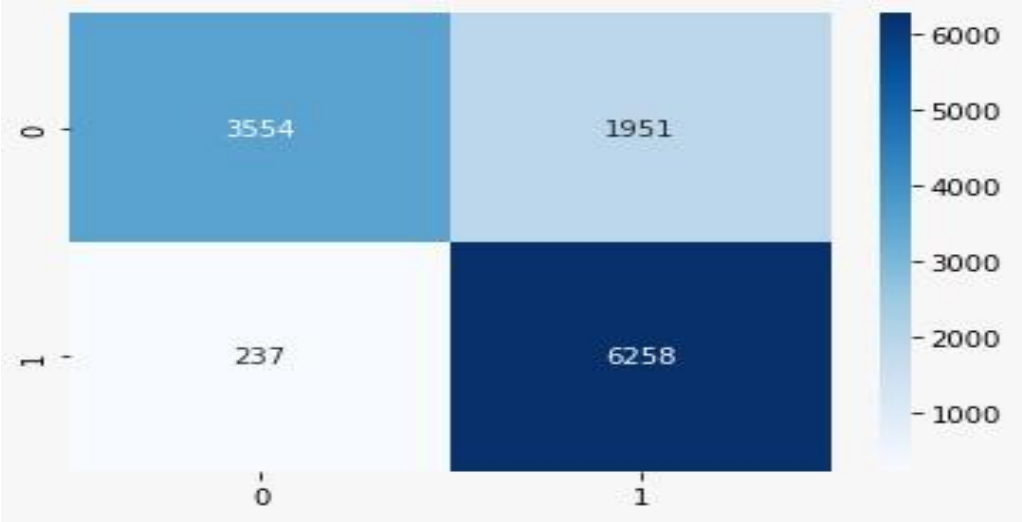


Figure 10 : Confusion Matrix of MLP Classifier

	precision	recall	f1-score	support
0	0.94	0.65	0.76	5505
1	0.76	0.96	0.85	6495
accuracy			0.82	12000
macro avg	0.85	0.80	0.81	12000
weighted avg	0.84	0.82	0.81	12000

Table 2 : Accuracy Result Table of the MLP

4.7 Hyperparameter tuning selected models

Hyper parameters tuning is crucial as they control the overall behavior of a machine learning model. A hyper parameter is a parameter whose value is set before the learning process begins. The ultimate goal is to find an optimal combination of hyper parameters that minimizes a predefined loss function to give better results.

I have used Grid Search for hyper parameter tuning. This method tries every possible combination of each set of hyper-parameters. Using this method, we can find the best set of values in the parameter search space. This usually uses more computational power and takes along time to run since this method needs to try every combination in the grid size.

Hyper parameters are crucial as they control the overall behavior of a machine learning model. The ultimate goal is to find an optimal combination of hyper parameters that minimizes a predefined loss function to give better result After hyper parameter, tuning performance of models along with their best parameters are:

4.7.1 Logistic Regression

Logistic Regression(C=1.0, solver='liblinear')

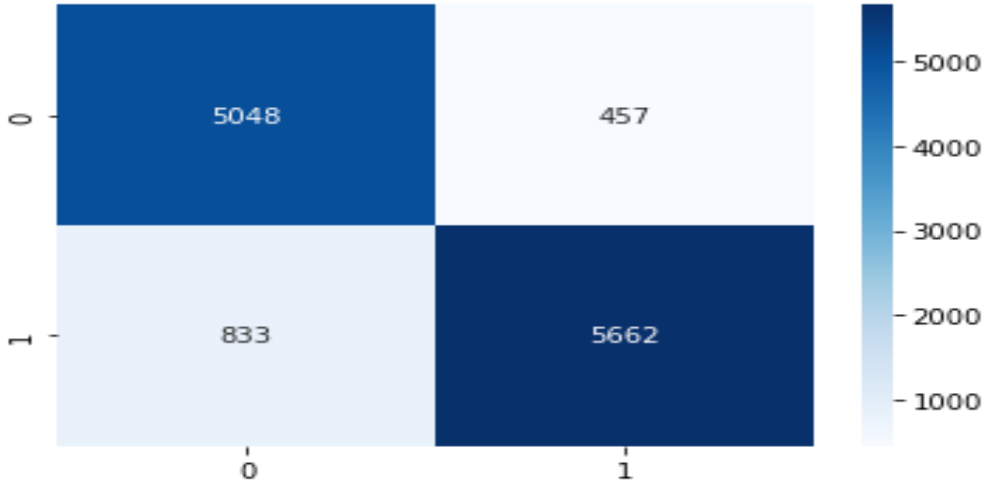


Figure 11: Confusion Matrix of Logistic Regression Classifier

	precision	recall	f1-score	support
0	0.86	0.92	0.89	5505
1	0.93	0.87	0.90	6495
accuracy			0.89	12000
macro avg	0.89	0.89	0.89	12000
weighted avg	0.89	0.89	0.89	12000

Table 3 : Accuracy Result Table of the Logistic Regression after hyperparameter tuning

4.7.2 MLP classifier

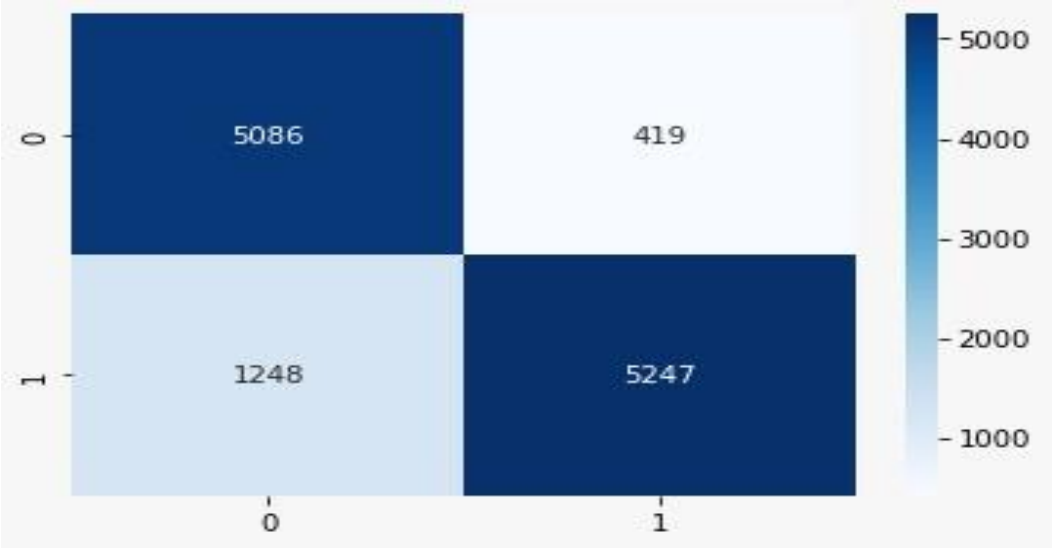


Figure 12: Confusion Matrix of MLP Classifier

	precision	recall	f1-score	support
0	0.80	0.92	0.86	5505
1	0.93	0.81	0.86	6495
accuracy			0.86	12000
macro avg	0.86	0.87	0.86	12000
weighted avg	0.87	0.86	0.86	12000

Table 4: Accuracy Result Table

4.8 Performance comparison

As our models are trained, cross validated and tested, it's time to compare them. As our Although all the models performed well after tuning up the hyper parameters. During the research both Logistic Regression and KNN were really close to each other performance wise. KNN came out as the better model to identify the DDOS attack considering the selected features than Extra Trees as it has decent balance of bias and variance. SVC would also be a really good choice . We can see every model has improved the performance after tuning up. Following inferences are made from the research . Best performing model is hyperparameter tuned KNN Classifier with accuracy of 97.469% and log loss of 0.077.and parameters are (no. of neighbors *K =3, neighbors weight=Uniform*)

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this research, the accuracy of DDOS attack detection were evaluated based on manually determining which feature to extract from data Frame and automated feature selection through various filters. Finally, comparison among the performance of models is done.

The research achieved accuracy of 97.468% by using the kNN Classifier .The log loss of KNN is much less than the other and also has a better learning curve. So Knn emerges as best performing model among all the models.

5.2 Future Work

Feature selection techniques need more improvement to cope with the continuous development of new techniques by the Attacker over the time. Therefore, it is recommended to developing a new automated tool in order to extract new features from new network-level to improve the accuracy of detecting Ddos attack and to cope with the expanding with new attack.

References:

[1] N. Agrawal and S. Tapaswi, "Defense Mechanisms Against DDoS Attacks in a Cloud Computing Environment: State-of-the-Art and Research Challenges," in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3769-3795, Fourthquarter 2019, doi: 10.1109/COMST.2019.2934468.

[2] A. Praseed and P. S. Thilagam, "DDoS Attacks at the Application Layer: Challenges and Research Perspectives for Safeguarding Web Applications," in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 661-685, Firstquarter 2019, doi: 10.1109/COMST.2018.2870658.

[3] S. S. Priya, M. Sivaram, D. Yuvaraj and A. Jayanthiladevi, "Machine Learning based DDOS Detection," *2020 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 2020, pp. 234-237, doi: 10.1109/ESCI48226.2020.9167642.

[4] T. Pinto and Y. Sebastian, "Detecting DDoS attacks using a cascade of machine learning classifiers based on Random Forest and MLP-ANN," *2021 IEEE Madras Section Conference (MASCON)*, 2021, pp. 1-6, doi: 10.1109/MASCON51689.2021.9563266.

[5] S. Yadav and S. Selvakumar, "Detection of application layer DDoS attack by modeling user behavior using logistic regression," *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, 2015, pp. 1-6, doi: 10.1109/ICRITO.2015.7359289.

[6] source : https://www.educba.com/what-is-ddos-attack/ “Overview of DDOS detection.”

[7] source : https://www.imperva.com/learn/ddos/ddos-attacks/ “types of DDOS detection.

[8] source : https://www.kentik.com/kentipedia/ddos-detection/ “DDOS Detection Technique

[9]source :https://www.researchgate.net/publication/”Systematic literature review and taxonomy for DDoS attack detection”

[10]source:https://medium.com/@sagar.rawale3/feature-selection-methods-in-machine-learning “ Feature selection

[11] source : https://data.mendeley.com/datasets/jxpfjc64kr/1 “Take DataSet “

[12] source : https://www.javatpoint.com/MLP-algorithm-for-machine-learning “about MLP algorithm “

[13]source : https://www.javatpoint.com/machine-learning-Logistic-regression-algorithm "about Logistic Regression algorithm"

[14]source: https://digital.com/best-web-hosting/introduction-to-distributed-denial-of-service-ddos- "what is DDOS detection and types of DDos."

[15] source : https://www.sciencedirect.com/ "MLP"