Efficient Detection of Denial of Service (DoS) attack using Machine Learning

MSc Internship
Cyber-Security

Somesh Saxena
Student ID: x18176895

School of Computing
National College of Ireland

Supervisor: Dr. Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Somesh Saxena		
Student ID:	X18176895		
Programme:	MSc. Cybersecurity	**Year:**	2020
Module:	MSc. Internship		
Supervisor:	Dr. Vikas Sahni		
Submission Due Date:	17th August 2020		
Project Title:	Efficient Detection of Denial of Service (DoS) attack using Machine Learning		
Word Count:	5465	**Page Count:**	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.
I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature:	Somesh Saxena
Date:	17th August, 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Efficient Detection of Denial of Service (DoS) attack using Machine Learning

Somesh Saxena
X18176895

Abstract

In this digital society internet is used almost by everyone. As the use of internet increases so as the threats are growing just proportionally. One such threat is DoS attack that uses service requests to disrupt computing and its network, leading inability to access websites and data to legitimate users. DoS attack can happen at various layers of OSI model like Network, Transport and Application layer. The paper deals with the detection of DoS attacks using Machine Learning algorithm's in an efficient way. The kind of DoS attacks that will be covered on this paper are ICMP flood attack, TCP-SYN flood attack and UDP flood attack. The project uses dataset which are separated into various segments and are been split for each algorithm. The paper deals with Decision Tree classifier, Logistic Regression model, Multi-Layer Perceptron, K Nearest Neighbors classifier and Light Gradient Boosting Machine Algorithm. Comparison between algorithm is done to evaluate precision and accuracy for the algorithm against DoS attacks. The paper emphasises on the use of Light GBM for detection of DoS attack and is shown to produce superior performance than any other Machine Learning algorithm.

1 Introduction

Since the use of new modern devices are growing significantly with the rising number of populations. The development of modern network is causing a great number of threats to the network. One such threat that is quite potential is a Denial of Service (DoS) attack. Talking of the todays internet world Denial of Service (DoS) is the greatest threat to organization and people. Denial of Service attack is a type of attack in which the targeted machine is flooded with data packets to render the network system and interrupt the device normal functioning. DoS attack can cause a high loss in terms of reputation and even unavailability of service to its end user or clients.

A DoS attack can be done in many ways. Some of the most common ways of DoS attacks are Flood attack in which the normal traffic is flooded with high number of data packets, disrupting the state of sessions such as TCP, disrupting the connections so that the user cannot access the data or information, disrupting a service so that the end user is not able to use the service. There are different types of DoS attacks. The most common types are ICMP flood attack in which the misconfigured target machine on the network is targeted forcing the network to send false or inappropriate data packets to the node, Buffer Overflow attack is an attack type in which the attacker send a high number of data packets which render the system and can crash the system or can have a full control on the system, SYN flood attack is an attack type in which the attacker sends a connection request to the server but never authenticates the

SYN-ACK back to the server and then the attacker keeps on sending multiple requests on the server until the server crashes.

The purpose and the objective of this research project is to detect of DoS attack using Machine Learning which will address the research question provided as “To detect Denial of Service attack using Machine Learning in an efficient way and to compare with the other used Machine Learning algorithm”. The type of DoS attack discussed in this research project is UDP flood attack, TCP-SYN attack and ICMP flood attack. UDP flood attack is a type of a DoS attack in which the targeted server or client is sent a large number of User Datagram Protocol (UDP) packets with the aim of device inability to process and respond to request. TCP-SYN attack is a type of DoS attack in which the attacker exploits the three-way handshake that a TCP-SYN requests make with the server, the attacker sends repeated SYN requests to server making the server unresponsive to legitimate traffic. ICMP flood attack is a type of attack in which the targeted system is flooded with ICMP echo-requests which cause the system inaccessible to normal traffic.

The Machine Learning algorithm that is discussed in this research project is Logistic Regression, Decision Tree, Multi-Layer Perceptron, K Nearest Neighbors and Light Gradient Boosting Machine. The research project emphasis on the use of Light Gradient Boosting Machine algorithm for an efficient detection of DoS attack. Light Gradient Boosting Machine model has a faster processing with high efficiency, requires less memory, can work easily with large datasets and provides better accuracy with compared to different algorithms.

Objective of this research are:

- Analyse the data for its characteristic.
- Develop Machine Learning models using Logistic Regression, Decision Tree, Multi-Layer Perceptron, K Nearest Neighbors and Light Gradient Boosting Machine.
- Evaluate the model and predict the accuracy of the model for the detection of a DoS attacks.
- Propose a model that detect the accuracy better or similar with compared to other models. Achieving a better model that can be used for detection of DoS attack in an efficient manner.

Major Contribution: The major contribution obtained from this research project is to use an efficient model that is Light Gradient Boosting Machine model for the detection of an DoS attack, as this work hasn’t been done before and there was an eagerness for a model that provides better accuracy and is light weighted which means it requires less memory, less processing and gives a better detection rate with less error. Light GBM provides fast processing with the capability to handle large datasets. And performs quite well in real time risk assessment.

2 Related Work

2.1 Previous approaches for DoS Detection

In a research done by G. Tsang et al in which the author uses UDP flood prediction using Support Vector Machine (SVM) and Radial Basis Function Neural Network (RBFNN). The author selected randomly chosen dataset and used for preparation and remainder for testing. The author used Defence Advanced Research Project Agency (DARPA) dataset for testing and found out that SVM requires more time for the new unexplained phenomena than RBFBB in the research process. The key consideration of his research work was based on consistency and any misclassifications are permitted, SVM was preferred and was recommended to use while it was also proposed to use the classification time as a major factor for RBF [1].

Research by S. Seufert and D. O'Brien proposed a method in which the authors used Neural Network to detect and filter Distributed Denial of Service Attacks. The researchers split the data into various levels and found out that the server had a low and slow response time just after the attack occurs using Neural Network method [2].

Researcher T. Subbulakshmi et al used a revised model of Support Vector Machines (SVM) to a detect DoS attack. The supervised learning algorithm used EMCSVM dealt with the kernel functions like linear, radial basis and polynomial function. The framework that was suggested uses its own dataset for network based, transport based and application-based layer DoS attack testing. EMCSVM kernel radially basis achieved the best grouping. The different kernel functions and parameter is known by calculating the output of the EMCSVM [3].

Researcher's S. Umarani and D. Sharmila uses HTTP trace for reference matrix definition. They proposed a program that will identify traffic flow of the packets as a regular traffic or will identify it as a DDoS attack. The authors used K-Nearest and Naïve Bayes Neighbourhood Classifiers in their study and found out that the results collected by PCA in a detection rate and there was an increase in the False Positive Rate (FPR) improvement by 0.9% and 4.11% respectively [4].

In a research done by Z. Ta et al. in which the author proposed a framework for the DoS attack detection in the network layer and transport layer using computer vision technique. Multivariate correlation model was used to classify traffic of the network records into respective images. The images are basically observable objects for DoS attack detection that was built on the basis of a metric which is known as Earth Movere's Distance (EMD). The dataset that was used was ISCX 2012 IDS dataset and KDD Cup99 dataset for the detection method with the help of Ten-Fold cross validations. The study depicts that the identification rate for KDD Cup99 dataset was 99.95% and for ISCX 2012 IDS dataset was 90.12% [5].

In a study by E. Nosrati et al. in the domain of Internet Multimedia Subsystem that was majorly on the identification of DoS attacks. For Session Initiation Protocol (SIP), the devices execute the operation to authenticate and give an alert to the concerned authority in the network. The study provides a variation of the Cumulative Sum (CUSUM) which is also called as adaptive Z-Score CUSUM. The main characteristic is that it can respond to changes that may occur in network traffic. The method provided the results in a low detection time and a low False Rate [6].

In a study by B. Cui Mei suggested a network and transport layer architecture for DoS attack. The study shows that Simple Network Based Protocol (SNMP) was used to obtain data rather than using direct network packets. Using SVM, the detection of the traffic is done and to determine whether an attack is actual or not with any particular kind of attack. The proposed study provides a detection rate of 99.27% and a False Positive Rate of 1.9% and a False Negative Rate of 0.73% [7].

In a study by M. Alkasassbe et al suggested a program that uses UDP flood and HTTP flood detection techniques. As the deliver capacity is quite high to provide accurate tests, the suggested method is run using a simulator called as Network Simulator (NS2). To define the dataset against DDoS attacks, machine learning algorithm such as Random Forest, MLP and Naïve Bayes were added. Mainly attack types of DDoS namely HTTP-flood, Smurf, UDP-flood and SID-DOS. The research evaluated the highest precision levels obtained by Multilayer Perceptron (MLP) [8].

Researcher D. Kshirsager et al suggested an Intrusion Detection System that was mainly focused on the signature or pattern of network layer DoS attacks such as TCP-SYN, UDP flood and ICMP flood. The suggested method senses a pattern-based flood attack like TCP-SYN and then it lowers the CPU load. The suggested DIDS also uses a methodology that uses signature to detect DDoS attacks like TCP and UDP flooding. The detection system uses client and server IDS that extends the scalability of IDS [9].

2.2 Study of an efficient algorithm for detection of DoS attacks

Researcher Kun Mo and Jian Li suggested an approach for Intrusion Detection System for a network using a Deep Auto-Encoder based Light Gradient Boosting Machine. In the research it was proposed that using Light GBM and deep Auto-Encoder was used for improving IDS performance. The performance of the model was evaluated by KDD CUP99 dataset and a series of experiments were conducted to explore different parameters. Just to avoid with the loss of transformation and feature reduction the suggested method used the immediate result of the deep auto encoder and a clean dataset. The proposed method achieved 95.3% of accuracy on the test dataset [10].

In a study by Guolin Ke, and his team suggested and GBDT algorithm known as Light GBM that contains techniques such as Gradient-based and Exclusive Bundling to cope up with a large number of data and features. The experiment shows that Light GBM can significantly outperform XGBoost and SGB in terms of speed and consumption related to memory with the help of GOSS and EFB. The study shows that the Light GBM resulted in speeding up the training process of GBDT by almost over 20 times achieving the same accuracy [11].

In a research done by Mingzhu Tang and his team, suggested an improved Light GBM model for online fault detection of wind turbines. The experiment conducted with a SCADA dataset from a farm and the detection of the fault was validated using Light GBM model. The study showed that Light GBM model showed superior performance over GBDT & XGBoost. By using Light GBM model an intelligent fault detection method was finally developed in the research. The model showed performance evaluation criteria better than other models in relation with high efficiency, fast speed, higher accuracy of the model. The accuracy of the detection rate was about 98.67% of the Light GBM model [12].

In a study by Marcos, Salma & Ivaldo presented with a study to effectively predict customer loyalty using Light GBM method. The study shows the comparison between XGBoost algorithm and Light GBM algorithm. The results showed that Light GBM algorithm outperform other GBDT algorithms and provided with good results when compared with XGBoost. The model provided with higher accuracies and is a efficient model in lieu to prediction of any data [13].

3 Research Methodology

The proposed research methodology is a discussion for prediction and detection of a vulnerability whether the user will be attacked or not, using Machine Learning approach. The tests have been performed on datasets that are been acquired from online sources. The research methodology will cover significant aspects of Data Mining and Analytics. And the use of different Machine Learning models to predict the accuracy of the attack.

It consists of the following steps:

- Data selection
- Data pre-processing
- Model Building & training
- Evaluation

3.1 Data Selection

The research required a dataset that contains information that are required to detect a DDoS attack. One such dataset is available on public domain is KDD-CUP99 dataset[1]. The dataset is well known for Intrusion detection techniques. The dataset was prepared and managed by MIT Lincoln Labs. The dataset has about 42 attributes which is used in this study. The attributes were classified into four different types of classes which are as follows:

- Basic (B) features of TCP connections.
- Content (C) features within a connection that are suggested by domain knowledge.

[1] Dataset: https://datahub.io/machine-learning/kddcup99

- Traffic (T) features that computed using two-second time window.
- Host (H) features that are designed to assess attacks that last for more than two seconds.

3.2 Data Pre-processing

After the selection and importing of the dataset, now the data needs to be split into two types i.e. Train data and Test data. So that the data can be used in an efficient way. There are multiple things that are done to the data so that it can be used to transform the raw data in some useful and efficient format. The aim of pre-processing is to examine the data before mining. The steps that are included in pre-processing are as follows:

- Split the data into Test and Train Data.
- Add feature scaling to the Train data.

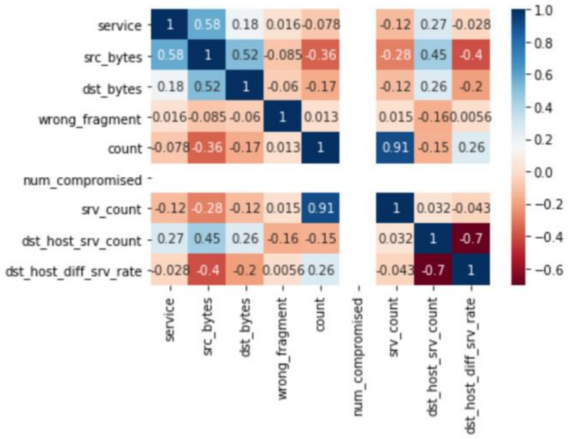


Figure 1. Correlation Matrix (UDP)

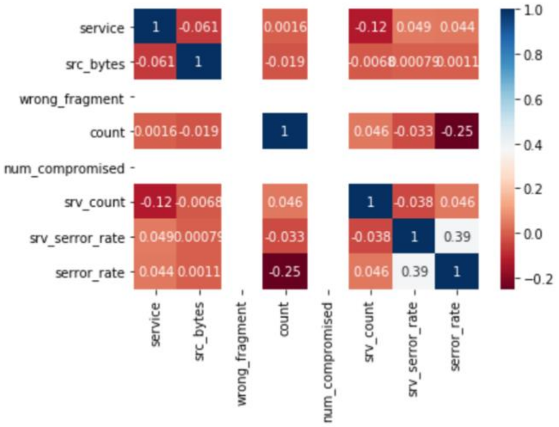


Figure 2. Correlation Matrix (TCP)

The outliners were removed, and the correlation matrix can be seen in Figure 1 and in Figure 2.

3.3 Model Building & Training

The model is built on the training data that is used to predict the accuracy of the Light Gradient Boosting Machine, Logistic Regression, K Nearest Neighbors Classifier, Multi-Layer Perceptron & Decision Tree. The models are compared to come up with the algorithm that have the best predictions of the attack efficiently. Accuracy totally depends upon what parameters you assign for a specific model. In order to perform predictions on the test data, the main parameters that are used to train the model depends upon the minimum amount of data in leaf, number of threads.

3.4 Evaluation/Interpretation

The metrics used in evaluation to analyse the models are Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R^2, Confusion Matrix and Receiving Operating Characteristic (ROC) curve.

- **Root Mean Squared Error (RMSE)** – It is the evaluation metrics that is used in regression analysis. RMSE is the squared root of Mean Squared Error (MSE). RMSE is the squared root of the difference between the expected value and the predicted value of the sample data. The formula to evaluate RMSE is shown in Equation 1.

$$\text{RMSE} = \sqrt{\frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(\boldsymbol{y_i} - \boldsymbol{p_i})^2} \qquad (1)$$

Where y represents the actual value and p represents the predicted value of the ith iterations. n is the number of sample data. The error of prediction will be minimum when there is a low value of RMSE, which means that the model is built better with less error.

- **Mean Absolute Error (MAE)** – The average of the absolute difference between the actual value and the predicted value. MAE measures the average of magnitude error in prediction of the certain model. The formula to evaluate MAE is given by Equation 2.

$$\text{MAE} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}|\boldsymbol{y_i} - \boldsymbol{p_i}| \qquad (2)$$

Where y is the actual value, p is the predicted value and n is the sample size.

- **R^2 score** – R^2 score is the measure of the accuracy performance of the regression model. It is the amount of difference between the actual value and the predicted value of the model. R^2 is used to check how well-observed the results are evaluated from a certain model. The formula to obtain R^2 is given by Equation 3.

$$\text{R}^2 = 1 - \frac{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(\boldsymbol{y_i}-\boldsymbol{p_i})^{\mathbf{2}}}{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(\boldsymbol{y_i}-\boldsymbol{\mu})^{\mathbf{2}}} \qquad (3)$$

Where y is the actual value, p is the predicted value, μ is the mean value and n is the number of the data in the sample.

- **Confusion Matrix** – It is the summary of the prediction results that are obtained by the model. The main benefit of confusion matrix is that it not only provides with the errors that are made by the classifier but also tells about the types of errors that are made by the classifier. The formula is given by Equation 4.

$$\text{Confusion Matrix} = \begin{bmatrix} \boldsymbol{TP} & \boldsymbol{FN} \\ \boldsymbol{FP} & \boldsymbol{TN} \end{bmatrix} \qquad (4)$$

Where TP = True Positive, FN = False Negative, FP = False Positive, TN = True Negative.

Using the values of the confusion matrix, different other values can be computed as follows:

$$\text{Accuracy} = \frac{\boldsymbol{TP + TN}}{\boldsymbol{TP + TN + FP + FN}} \tag{5}$$

$$\text{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \tag{6}$$

$$\text{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \tag{7}$$

$$\text{F} - \text{Measure} = \frac{\boldsymbol{2 \times Recall \times Precision}}{\boldsymbol{Recall+Precision}} \tag{8}$$

- **Receiver Operating Characteristic (ROC) Curve** – ROC curve is a plot of the False Positive rate on the x-axis versus True Positive rate on the y-axis of the data in a threshold value between 0.0 to 0.1. The Area Under the Curve (AUC) is used as the summary of the model.

$$\text{True Positive Rate} = \frac{\boldsymbol{True\ Positive}}{\boldsymbol{True\ Positive+False\ Positive}} \tag{9}$$

$$\text{False Positive Rate} = \frac{\boldsymbol{False\ Positive}}{\boldsymbol{False\ Positive+True\ Positive}} \tag{10}$$

4 Design Specification

For this research project the dataset is not created explicitly but a dataset that is available on public domain is used. Figure 3 shows that the initial step was to select a dataset which was taken from KDD CUP99, it is a dataset which performs good in Intrusion Detection System. Then the major step after the data selection is the data pre-processing due to which the entire results of the module depends. Python is used to develop models and the entire code is written in python. An open source tool known as Anaconda is used to manage and deploy python scripts. The scripts are written in Jupyter Notebook, which is used to load the data, pre-process the data, build models and evaluate the results. The models are built using Light Gradient

Boosting Machine, Logistic Regression, K Nearest Neighbors Classifier, Multi-Layer Perceptron & Decision Tree. The models are evaluated using various methods and the results are provided in visual format for a clear understanding.

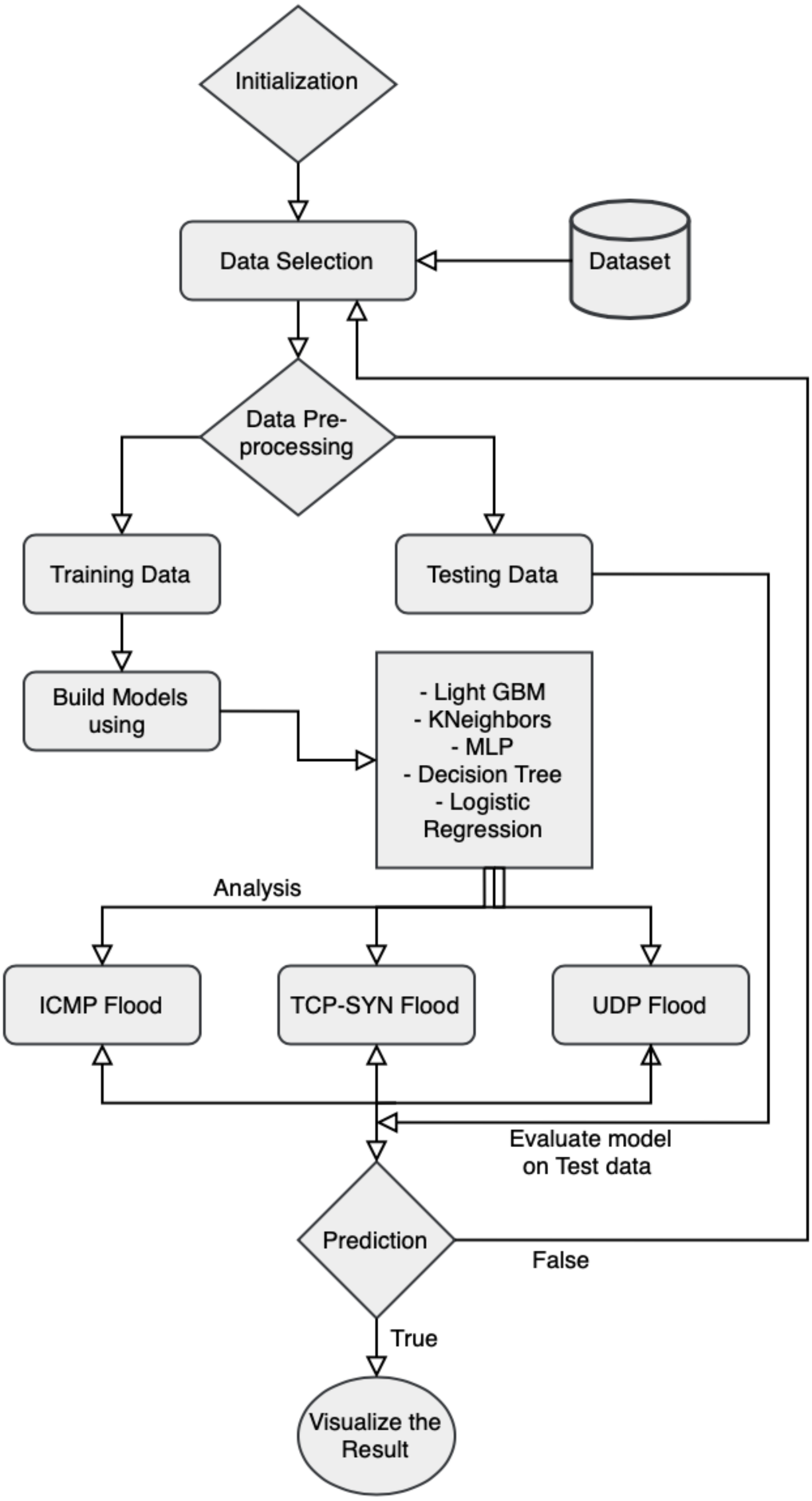


Figure 3. Design Architecture

5 Implementation

For the implementation of this research project dataset has been taken from a public domain from KDD CUP99 dataset. The code is written and debugged using python. The tools include Anaconda which is an open source python distribution which comes bundled with a variety of tools that was used in this project. The python code is written in Jupyter Notebook where the dataset is loaded and then pre-processing and model building of the data is done.

5.1 Light Gradient Boosting Machine (LGBM)

The sklearn Light Gradient Boosting Machine is used to implement the model. Light Gradient Boosting Machine model is an open source framework for gradient boosting machines. The model is fast and can handle large datasets and require less memory. The main parameters include num_leaves where increasing the number of leaves increases the accuracy, max_bin a larger number provides higher accuracy, learning_rate, max_depth. There has been a deep focus on learning task parameters which include RMSE, MAE, R^2, ROC and AUC.

5.2 Logistic Regression

The sklearn Logistic Regression model is used to implement the model. Logistic Regression is used when there is a prediction that needs to be made in Yes (1) or No (0). The main parameters include learning_rate, no_iterations, fit_intercept, normalize and n_jobs. The parameter fit_intercept has an output as boolean which evaluate the intercept, normalize has an output as Boolean in which True signifies the regressor will be normalized and False signifies that the fit_intercept will be ignored. The data is normalized before building the model.

5.3 K Nearest Neighbors Classifier

The sklearn K Nearest Neighbors classifier is used to implement the model. K Nearest Neighbors is used in estimation and prediction. The main parameters include n_neighbors which tells the number of neighbors to use its 5 by default, n_jobs tells the number of jobs to run in parallel, weight which is used in prediction, algorithm which tells which algorithm to compute nearest neighbors, leaf_size, metric. n_neighbor = 3 is taken and all other values are taken by default.

5.4 Multilayer Perceptron (MLP)

The sklearn Multi-layer Perceptron is used to implement the model. Multi-layered Perceptron include parameters such as activation, solver, hidden_layer_sizes, learning_rate, learning_rate_init, max_iter and alpha. Mostly the values are used that are default but alpha = 0.005. The model is normalized before building. learning_rate = ‘adaptive’, learning_rate_init = 0.001.

5.5 Decision Tree

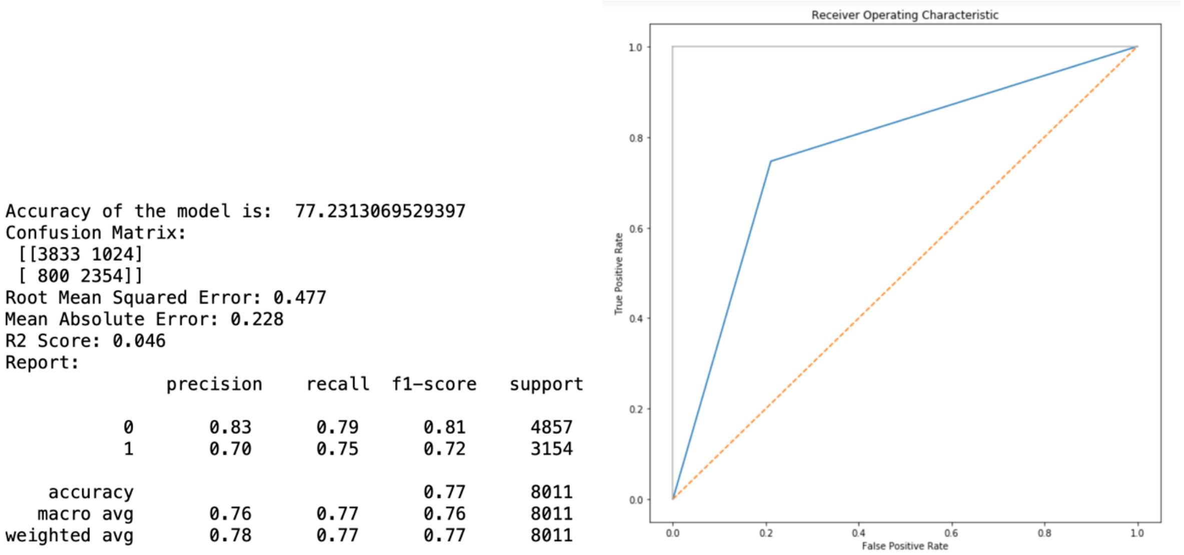
The sklearn Decision Tree is used to implement the model. Decision Tree is used for prediction of the model. The parameters used are criterion, splitter, max_depth, min_samples_split, max_feature, random_state, class_weight. The default values of the model are used in the project to compute the accuracy and prediction rate of the model. Splitter = ‘best’, max_depth = ‘none’, max_feature = ‘auto’.

6 Evaluation

This section consists of evaluation of results that are achieved to meet the objective of this research project. There are three cases that are taken which means there are three different split data which is used for the detection of UDP flood, ICMP flood and TCP-SYN flood attack. For the evaluation of the data using the model there are evaluation metrics provided that evaluate the model and predict the accuracy. The evaluation metrics used are RMSE, MAE, R^2, ROC, Confusion Matrix and AUC. Using Confusion Matrix values certain other metrics such as Accuracy, Precision, Recall & F-Measure are evaluated for the model. Since the sample size is high, visualization is done using graphs and tables for easy understanding.

6.1 Case Study with UDP – Flood attack

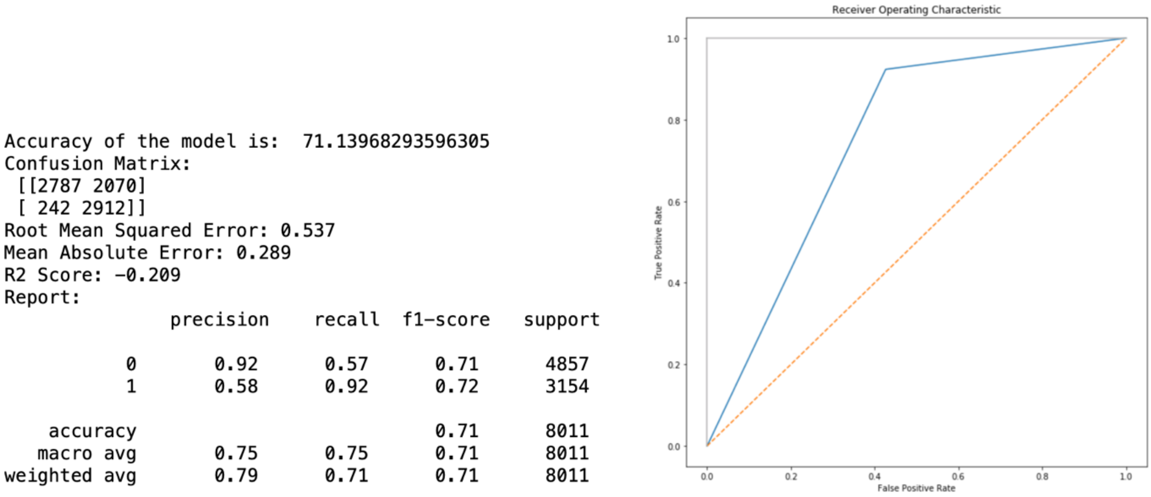
6.1.1 Light Gradient Boosting Machine



```
Accuracy of the model is:  77.2313069529397
Confusion Matrix:
 [[3833 1024]
 [ 800 2354]]
Root Mean Squared Error: 0.477
Mean Absolute Error: 0.228
R2 Score: 0.046
Report:
               precision    recall  f1-score   support

           0       0.83      0.79      0.81      4857
           1       0.70      0.75      0.72      3154

    accuracy                           0.77      8011
   macro avg       0.76      0.77      0.76      8011
weighted avg       0.78      0.77      0.77      8011
```

Figure 4. Detailed Evaluation using Light Gradient Boosting Machine

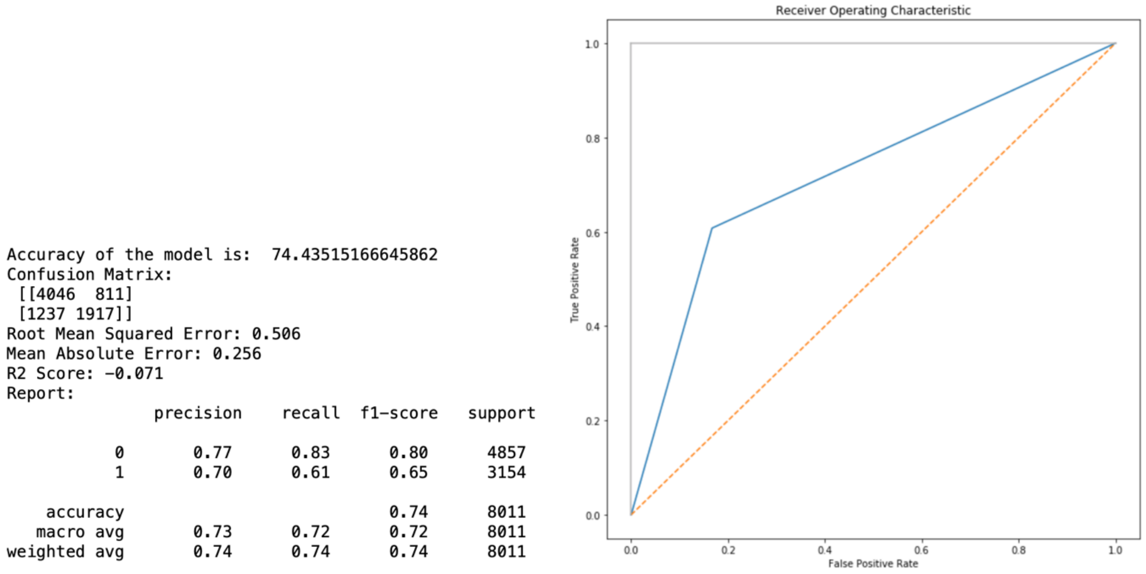
6.1.2 Logistic Regression



```
Accuracy of the model is:  71.13968293596305
Confusion Matrix:
 [[2787 2070]
 [ 242 2912]]
Root Mean Squared Error: 0.537
Mean Absolute Error: 0.289
R2 Score: -0.209
Report:
               precision    recall  f1-score   support

           0       0.92      0.57      0.71      4857
           1       0.58      0.92      0.72      3154

    accuracy                           0.71      8011
   macro avg       0.75      0.75      0.71      8011
weighted avg       0.79      0.71      0.71      8011
```

Figure 5. Detailed Evaluation using Logistic Regression

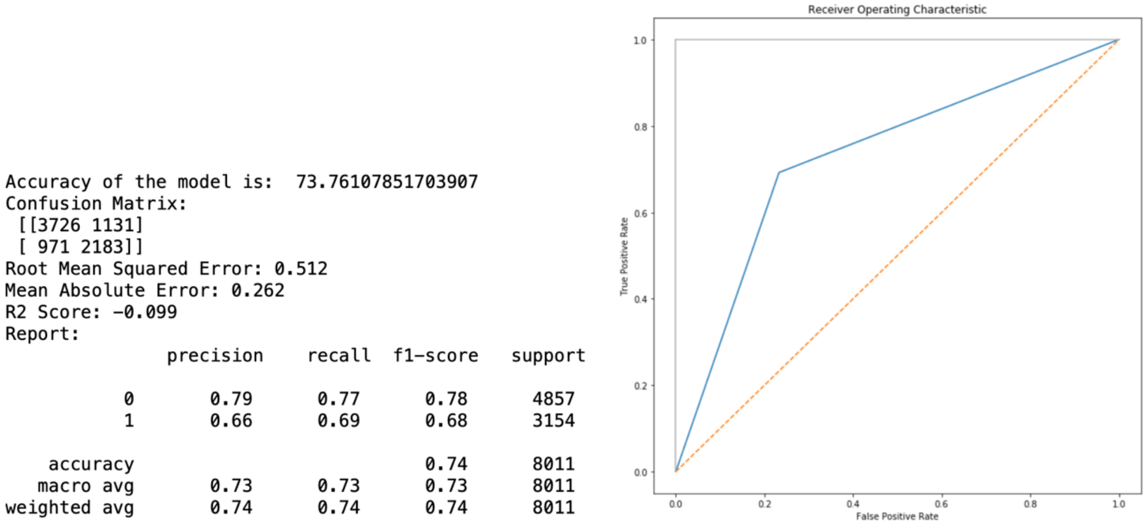
6.1.3 K Nearest Neighbors Classifier



```
Accuracy of the model is:  74.43515166645862
Confusion Matrix:
 [[4046  811]
 [1237 1917]]
Root Mean Squared Error: 0.506
Mean Absolute Error: 0.256
R2 Score: -0.071
Report:
              precision    recall  f1-score   support

           0       0.77      0.83      0.80      4857
           1       0.70      0.61      0.65      3154

    accuracy                           0.74      8011
   macro avg       0.73      0.72      0.72      8011
weighted avg       0.74      0.74      0.74      8011
```

Figure 6. Detailed Evaluation using K Nearest Neighbors

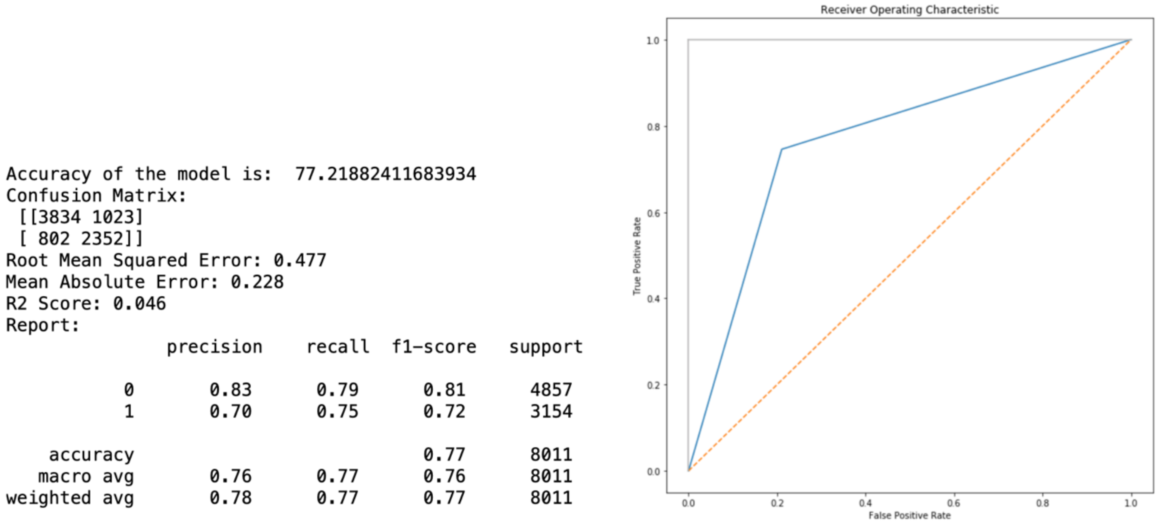
6.1.4 Multilayer Perceptron



```
Accuracy of the model is:  73.76107851703907
Confusion Matrix:
 [[3726 1131]
 [ 971 2183]]
Root Mean Squared Error: 0.512
Mean Absolute Error: 0.262
R2 Score: -0.099
Report:
              precision    recall  f1-score   support

           0       0.79      0.77      0.78      4857
           1       0.66      0.69      0.68      3154

    accuracy                           0.74      8011
   macro avg       0.73      0.73      0.73      8011
weighted avg       0.74      0.74      0.74      8011
```

Figure 7. Detailed Evaluation using Multilayer Perceptron

6.1.5 Decision Tree

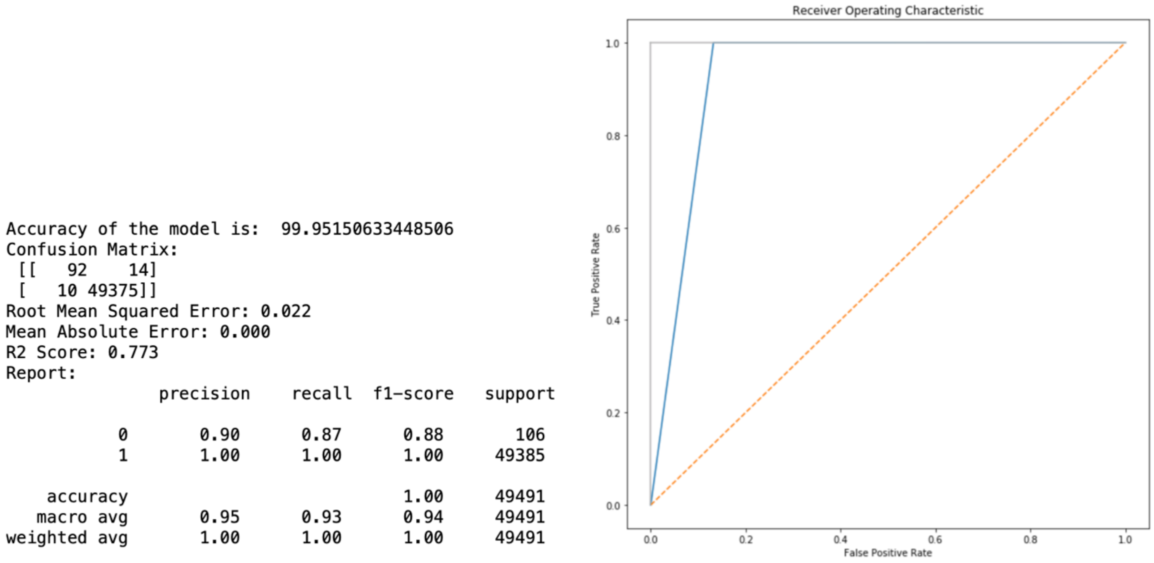


```
Accuracy of the model is:  77.21882411683934
Confusion Matrix:
 [[3834 1023]
 [ 802 2352]]
Root Mean Squared Error: 0.477
Mean Absolute Error: 0.228
R2 Score: 0.046
Report:
               precision    recall  f1-score   support

           0       0.83      0.79      0.81      4857
           1       0.70      0.75      0.72      3154

    accuracy                           0.77      8011
   macro avg       0.76      0.77      0.76      8011
weighted avg       0.78      0.77      0.77      8011
```

Figure 8. Detailed Evaluation using Decision Tree

6.2 Case Study with ICMP – Flood attack

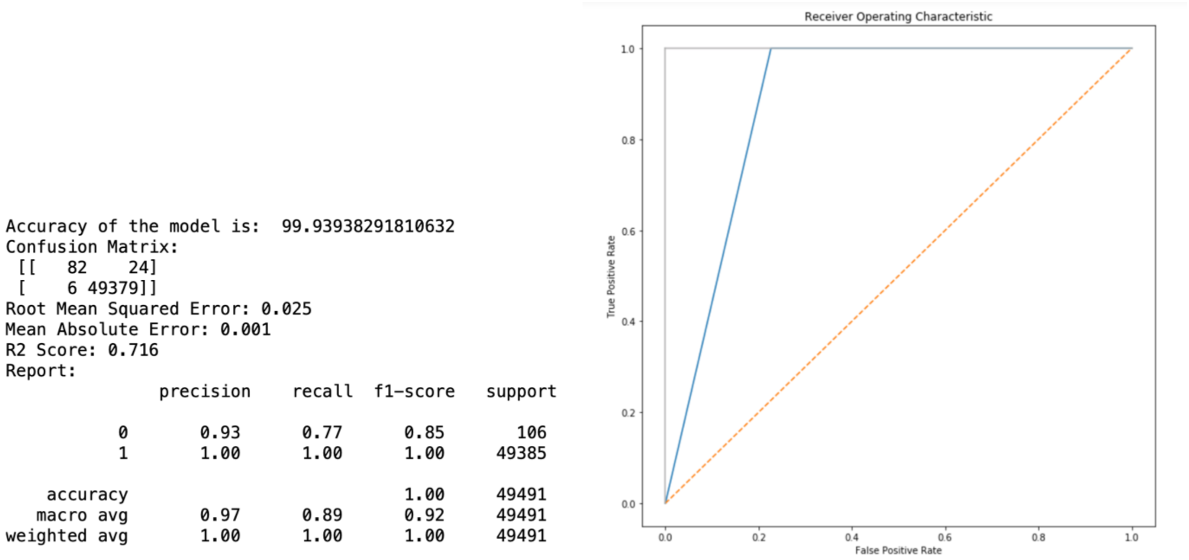
6.2.1 Light Gradient Boosting Machine



```
Accuracy of the model is:  99.95150633448506
Confusion Matrix:
 [[   92    14]
 [   10 49375]]
Root Mean Squared Error: 0.022
Mean Absolute Error: 0.000
R2 Score: 0.773
Report:
               precision    recall  f1-score   support

           0       0.90      0.87      0.88       106
           1       1.00      1.00      1.00     49385

    accuracy                           1.00     49491
   macro avg       0.95      0.93      0.94     49491
weighted avg       1.00      1.00      1.00     49491
```

Figure 9. Detailed Evaluation using Light Gradient Boosting Machine

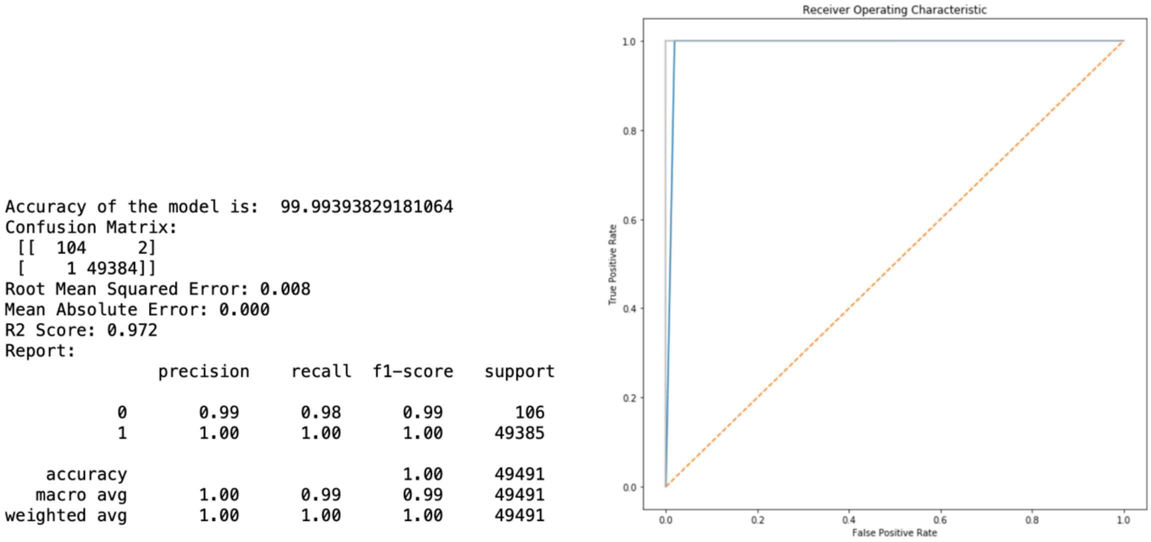
6.2.2 Logistic Regression



```
Accuracy of the model is:  99.93938291810632
Confusion Matrix:
 [[   82    24]
 [    6 49379]]
Root Mean Squared Error: 0.025
Mean Absolute Error: 0.001
R2 Score: 0.716
Report:
               precision    recall  f1-score   support

           0       0.93      0.77      0.85       106
           1       1.00      1.00      1.00     49385

    accuracy                           1.00     49491
   macro avg       0.97      0.89      0.92     49491
weighted avg       1.00      1.00      1.00     49491
```

Figure 10. Detailed Evaluation using Logistic Regression

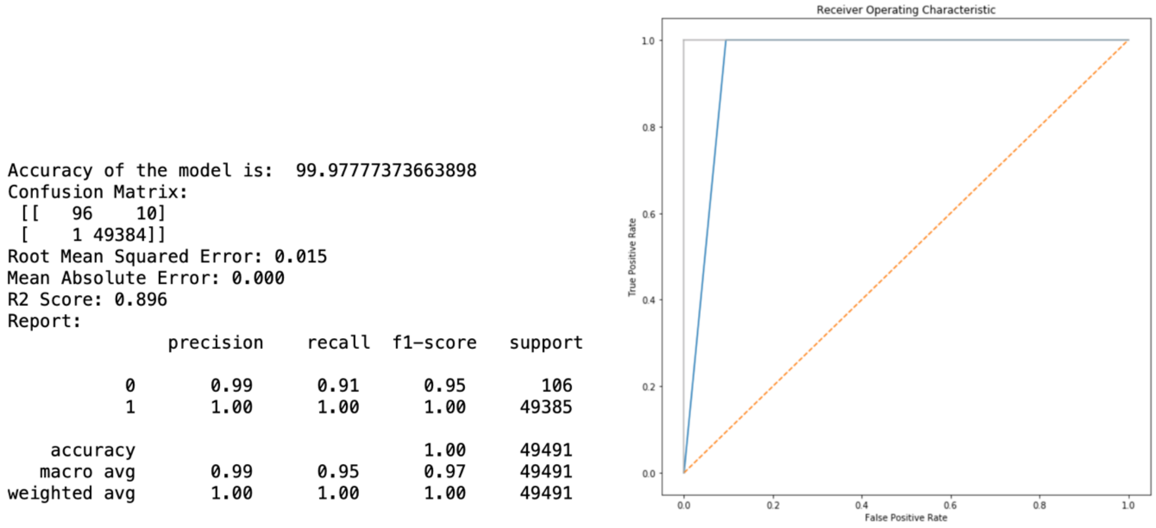
6.2.3 K Nearest Neighbors Classifier



```
Accuracy of the model is:  99.99393829181064
Confusion Matrix:
 [[  104     2]
 [    1 49384]]
Root Mean Squared Error: 0.008
Mean Absolute Error: 0.000
R2 Score: 0.972
Report:
               precision    recall  f1-score   support

           0       0.99      0.98      0.99       106
           1       1.00      1.00      1.00     49385

    accuracy                           1.00     49491
   macro avg       1.00      0.99      0.99     49491
weighted avg       1.00      1.00      1.00     49491
```

Figure 11. Detailed Evaluation using K Nearest Neighbors

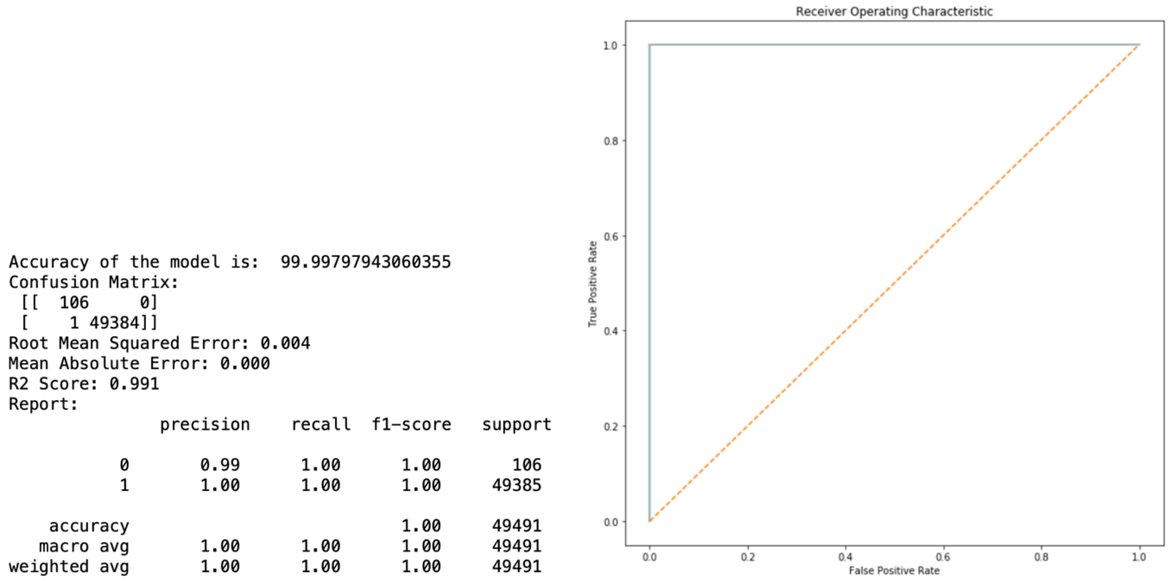
6.2.4 Multilayer Perceptron



```
Accuracy of the model is:  99.97777373663898
Confusion Matrix:
 [[   96    10]
 [    1 49384]]
Root Mean Squared Error: 0.015
Mean Absolute Error: 0.000
R2 Score: 0.896
Report:
               precision    recall  f1-score   support

           0       0.99      0.91      0.95       106
           1       1.00      1.00      1.00     49385

    accuracy                           1.00     49491
   macro avg       0.99      0.95      0.97     49491
weighted avg       1.00      1.00      1.00     49491
```

Figure 12. Detailed Evaluation using Multilayer Perceptron

6.2.5 Decision Tree



```
Accuracy of the model is:  99.99797943060355
Confusion Matrix:
 [[  106     0]
 [    1 49384]]
Root Mean Squared Error: 0.004
Mean Absolute Error: 0.000
R2 Score: 0.991
Report:
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       106
           1       1.00      1.00      1.00     49385

    accuracy                           1.00     49491
   macro avg       1.00      1.00      1.00     49491
weighted avg       1.00      1.00      1.00     49491
```

Figure 13. Detailed Evaluation using Decision Tree

6.3 Discussion

This section deals with the discussion of the results that are achieved to meet the objective of this research project. There are three cases that are taken which means there are three different split data which is used for the detection of UDP flood, ICMP flood and TCP-SYN flood attack. For the evaluation of the data using the model there are evaluation metrics provided that evaluate the model and predict the accuracy. The models are built using Light Gradient Boosting Machine, Logistic Regression, K Nearest Neighbors Classifier, Multi-Layer

Perceptron & Decision Tree. Different analysis has been done accuracy and other evaluation metrics have been evaluated.

For the final review graphs and tables are been made for a better understanding. Figure 14. Plot of UDP datasetFigure 14 shows the graph of the UDP attack data type, the graph shows the detection rate of the different algorithm used we can clearly see that Light Gradient Boosting Machine model performs the best as compared to another model. Figure 15 shows the graph of the ICMP attack data type, the graph shows the detection rate of the different algorithm used we can see that Light Gradient Boosting Machine model performs equivalent as compared to another model. Figure 16 shows the graph of the TCP-SYN attack data type, the graph shows the detection rate of the different algorithm used we can see that Light Gradient Boosting Machine model performs equivalent as compared to another model.

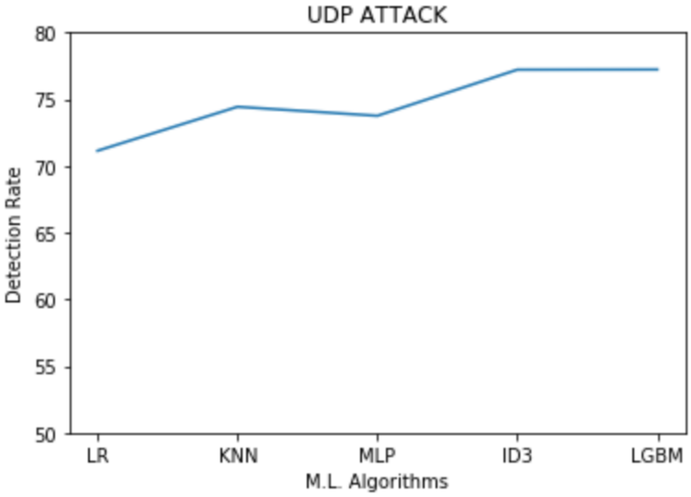


Figure 14. Plot of UDP dataset

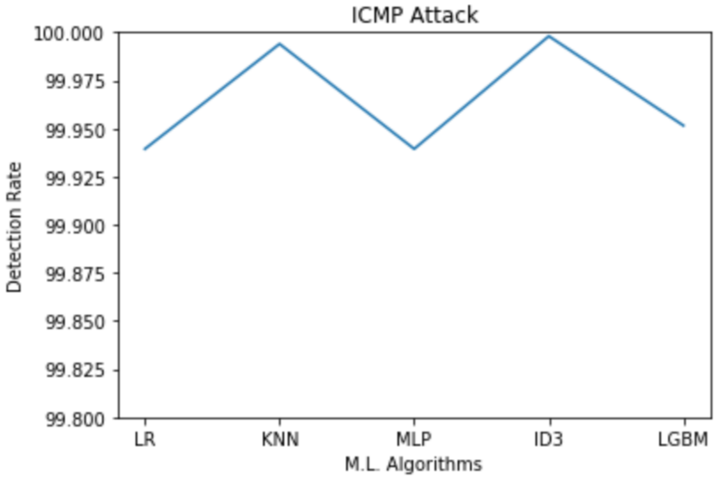


Figure 15. Plot of ICMP dataset

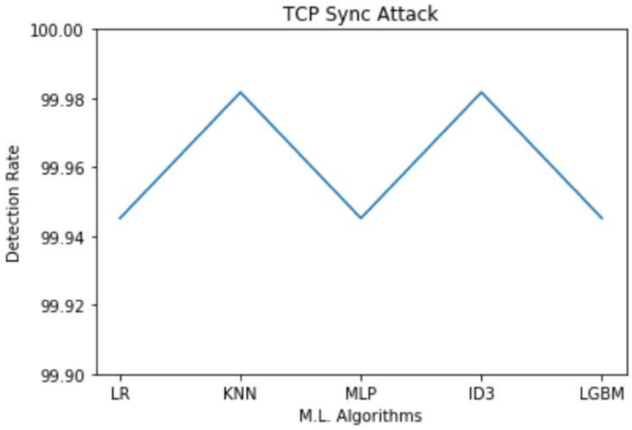


Figure 16. Plot of TCP-SYN dataset

Tables have been generated to a clear understanding of various other evaluation metrics that have been considered for evaluation of the models which include RMSE, MAE, R^2 and Accuracy. Table 1 shows the results obtained of ICMP attack type. Table 2 shows the results obtained of TCP-SYN attack type. Table 3 shows the results obtained of UDP attack type.

Models used	RMSE	MAE	R^2	Accuracy
Light Gradient Boosting Machine	0.022	0.000	0.773	99.9515
Logistic Regression	0.025	0.001	0.716	99.9393
K Nearest Neighbors	0.008	0.000	0.972	99.9939
Multi-Layer Perceptron	0.015	0.000	0.896	99.9777
Decision Tree	0.004	0.000	0.991	99.9979

Table 1. Results obtained of ICMP attack

Models used	RMSE	MAE	R^2	Accuracy
Light Gradient Boosting Machine	0.023	0.001	-0.001	99.9451
Logistic Regression	0.023	0.001	-0.001	99.9451
K Nearest Neighbors	0.014	0.000	0.666	99.9817
Multi-Layer Perceptron	0.023	0.001	-0.001	99.9451
Decision Tree	0.014	0.000	0.666	99.9817

Table 2. Results of TCP-SYN attack

Models used	RMSE	MAE	R^2	Accuracy
Light Gradient Boosting Machine	0.477	0.228	0.046	77.2310
Logistic Regression	0.537	0.289	-0.209	71.1396
K Nearest Neighbors	0.506	0.256	-0.071	74.4351
Multi-Layer Perceptron	0.512	0.262	-0.099	73.7610
Decision Tree	0.477	0.228	0.046	77.2188

Table 3. Results of UDP attack

7 Conclusion and Future Work

The models developed provided with promising results in predicting and detection of Denial of Service (DoS) attack. Light Gradient Boosting Machine model performed well and outperform all other models by providing a better result and predicting better for all the test cases that have been considered. The prediction of Light GBM is low when the test data considered was of ICMP attack type. As there was an eagerness for a model that provides better accuracy and requires less processing, the results obtained from this research project is to use an efficient model that is Light Gradient Boosting Machine model for the detection of an DoS attack provided better accuracy and it requires less memory, less processing and gives a better detection rate with less error. Light GBM provides fast processing with the capability to handle large datasets. And performs quite well in real time risk assessment.

There is more scope in future for the research which include real time traffic detection as it would lower the chances of Denial of Service attack and also a need for a model that performs various other tasks as well which would provide every assessment with the single model, so that the user or client don't use other model. And also providing a greater number of parameters that can be considered for a certain model which would help in a high detection of the DoS attack.

8 Acknowledgement

I would like to thank my supervisor Dr. Vikas Sahni for his support and encouragement that have made this research work possible. I would also like to thank him for his guidance that he provided since very beginning. All the tips provided was of great help in the making of this research work. I would like to thank National College of Ireland and School of Computing for providing me with the resources and giving me the opportunity to work for this research. Special thanks to my family and friends for supporting me all throughout the project.

References

[1] G. C. Y. Tsang, P. P. K. Chan, D. S. Yeung, and E. C. C. Tsang, 'Denial of service detection by support vector machines and radial-basis function neural network', in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, Aug. 2004, vol. 7, pp. 4263–4268 vol.7, doi: 10.1109/ICMLC.2004.1384587.

[2] S. Seufert and D. O'Brien, 'Machine Learning for Automatic Defence Against Distributed Denial of Service Attacks', in *2007 IEEE International Conference on Communications*, Jun. 2007, pp. 1217–1222, doi: 10.1109/ICC.2007.206.

[3] T. Subbulakshmi, K. BalaKrishnan, S. M. Shalinie, D. AnandKumar, V. GanapathiSubramanian, and K. Kannathal, 'Detection of DDoS attacks using Enhanced Support Vector Machines with real time generated dataset', in *2011 Third International Conference on Advanced Computing*, Dec. 2011, pp. 17–22, doi: 10.1109/ICoAC.2011.6165212.

[4] S. Umarani and D. Sharmila, ‘Predicting Application Layer DDoS Attacks Using Machine Learning Algorithms’, *Int. J. Comput. Syst. Eng.*, vol. 8, no. 10, pp. 1912–1917, Jan. 2015.

[5] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, and J. Hu, ‘Detection of Denial-of-Service Attacks Based on Computer Vision Techniques’, *IEEE Trans. Comput.*, vol. 64, no. 9, pp. 2519–2533, Sep. 2015, doi: 10.1109/TC.2014.2375218.

[6] E. Nosrati, A. S. Kashi, Y. Darabian, and S. N. H. Tonekaboni, ‘Register flooding attacks detection in IP multimedia subsystems by using adaptive z-score CUSUM algorithm’, in *ICIMU 2011 : Proceedings of the 5th international Conference on Information Technology Multimedia*, Nov. 2011, pp. 1–4, doi: 10.1109/ICIMU.2011.6122765.

[7] C.-M. Bao, ‘Intrusion Detection Based on One-class SVM and SNMP MIB Data’, in *2009 Fifth International Conference on Information Assurance and Security*, Aug. 2009, vol. 2, pp. 346–349, doi: 10.1109/IAS.2009.124.

[8] M. Alkasassbeh, G. Al-naymat, A. B. A. Hassanat, and M. Almseidin, *Detecting Distributed Denial of Service Attacks Using Data Mining Techniques*. .

[9] D. D. Kshirsagar, S. S. Sale, D. K. Tagad, and G. Khandagale, ‘Network Intrusion Detection based on attack pattern’, in *2011 3rd International Conference on Electronics Computer Technology*, Apr. 2011, vol. 5, pp. 283–286, doi: 10.1109/ICECTECH.2011.5942003.

[10] K. Mo and J. Li, ‘A Deep Auto-Encoder based LightGBM Approach for Network Intrusion Detection System’:, in *Proceedings of the International Conference on Advances in Computer Technology, Information Science and Communications*, Xiamen, China, 2019, pp. 142–147, doi: 10.5220/0008098401420147.

[11] G. Ke *et al.*, ‘LightGBM: A Highly Efficient Gradient Boosting Decision Tree’, in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 3146–3154.

[12] M. Tang *et al.*, ‘An Improved LightGBM Algorithm for Online Fault Detection of Wind Turbine Gearboxes’, *Energies*, vol. 13, no. 4, Art. no. 4, Jan. 2020, doi: 10.3390/en13040807.

[13] M. R. Machado, S. Karray, and I. T. de Sousa, ‘LightGBM: an Effective Decision Tree Gradient Boosting Method to Predict Customer Loyalty in the Finance Industry’, in *2019 14th International Conference on Computer Science Education (ICCSE)*, Aug. 2019, pp. 1111–1116, doi: 10.1109/ICCSE.2019.8845529.