DDOS attack Classification and Prediction Method Using Machine Learning

M. Evangeline Mercy[1*], Dr. I. Felcia Jerlin[2]
[1]*PG student, Department of Computer Science and Engineering, Holycross Engineering College, Srivaikuntam, Tamil Nadu, India*
[2]*Associate Professor, Department of Computer Science and Engineering, Holycross Engineering College, Srivaikuntam, Tamil Nadu, India*

***Abstract*— In general, distributed network attacks are referred to as Distributed Denial of Service (DDoS) attacks. Attacks like this capitalize on of unique constraints that apply to every arrangement asset, such as the authorized organization site's framework. This research provides a machine learning approach for classifying and predicting DDoS attacks. In this research, the classification algorithms LR, KNN, and Decision Tree are used. The datasets are pre-processed using Standard Scalar. The mean is removed and the data is scaled to the unit variance using Standard Scalar. This suggested project created a confusion matrix to determine the performance of the model. In the initial categorization, the Logistic Regression classifier technique is used for both Precision (PR) and Recall (RE). In the second classification, the KNN classifier method is utilized to overcome the difficulties and determine the accuracy, precision, and confusion matrix. The decision tree has the substantial advantage of requiring all possible outcomes of a decision to be considered and following each path to a conclusion. Python software is used to carry out this project.**

***Index Terms*— PR-Precision, RE-Recall, DDoS-Distributed Denial of Service.**

1. INTRODUCTION

DDoS (distributed denial-of-service) attack originates from many sources scattered over multiple network locations. DoS attacks are primarily motivated by the desire to significantly degrade the performance or completely consume a certain resource, and a process to exploit a machine defect and cause failure of a processing or exhausting the system resources by exploiting a system flaw [1]. Yet another method of assaulting the target system is to flood the network and monopolise it, so preventing anyone else from utilising it. DoS attacks are defined and classified by the prohibition of access to the victim machine or network, whereas DDoS attack is the use of a large number of systems from distributed environment to launch the attack, which is defined and classified by the use of many computer systems or services [2]. Keep in mind that attack agents can be any vulnerable devices or resource that has the capability of running the suspicious code, such as Internet of Things devices, networked PCs, servers, and armed mobile devices, among other things. DDoS attacks almost always include forged IP packets, making it nearly impossible to identify the source of an attack. DDoS attacks are becoming increasingly sophisticated. In addition, the length of an attack has dropped in recent years to around 4 minutes, down from previously [3]. The affected machine crashes as a result, preventing any defence solution from detecting the attack. As a result, acquiring complete information on distributed denial of service (DDoS) assaults is extremely difficult. It is impossible to directly compare defensive devices with their competitors on the market because there is no common benchmark for DDoS defence filters in the computer sector [4]. Denial-of-service (DDoS) attack refers to the use of client/server technology to combine multiple computers as an attack platform to launch attacks on one or more targets to increase the power of the attack. Distributed denial-of-service attack has changed the traditional peer-to-peer attack mode, so there is no statistical rule for attack behavior, in addition, common protocols and services are used in the attack [5]. It is difficult to distinguish attack or normal behavior only through the types of protocols and services. The distributed denial-of-service attack is not easy to detect. At present, the research on defence technology against DDoS attack at home and abroad is mostly based on the method of network intrusion detection. According to the characteristics of many-to-one attack in the process of DDoS attack, three characteristics including the number of source IP addresses, the number of destination ports and the flow density were used to describe the characteristics of attack [6]. These methods can distinguish whether most of the attack flows are rational, but only use less message information, most of which only use the source IP address and destination port information, and cannot determine the specific attack type, so the detection rate is not high. Machine learning plays an important role in prediction. DDoS attack detection based on machine learning also has made some progress [7]. The machine learning algorithms used for DDoS attack detection mainly include naive Bayesian algorithm, hidden Markov model and support vector machine. Tama's team used the method of anomaly detection to model the network data stream according to the header attribute, and used the naive Bayesian algorithm to score each arriving data stream to evaluate the rationality of the message [8]. The methods in the above literature improve the detection accuracy to a certain extent, but do not make full use of the context of the data stream. This paper proposes a DDoS attack detection method based on machine learning. Based on the previous research, through the analysis of the principle of DDoS attack, the three common attack packets obtained by operating the DDoS attack tool are grouped in the feature extraction stage [9]. Through the analysis of normal flow data, the characteristics of attack flow are obtained. The characteristics of the attack traffic obtained in the model detection phase are trained in the training model based on the random forest algorithm. Finally, the test model is validated by the DDoS attack, and the SVM method in the machine learning is compared in terms of detection accuracy. The results show that the DDoS attack detection method based on machine learning proposed in this paper has a good detection rate for the current popular DDoS attack [10].

2. RECENT WORKS

Kshira Sagar Sahoo et al [2020] [11] proposed a Software-Defined Network (SDN) has emerged as a promising network architecture that gives network operators greater control over network infrastructure. The goal of this paper is to detect the attack traffic, by taking the centralized control aspect of SDN. Kernal Principle component analysis (KPCA) is used in the proposed SVM model to reduce the dimension of feature vectors, and GA is used to optimise various SVM parameters. An improved kernel function (N-RBF) is proposed to reduce the noise caused by feature differences. The experimental results show that the proposed model achieves more accurate classification with better generalisation than single-SVM. Furthermore, the proposed model can be embedded within the controller to define security rules that will prevent potential attacker attacks. Creating more intriguing algorithms by combining kernel functions with other classification methods is not carried in this paper.

Ankit Agarwal et al [2021] [12] developed a Deep learning algorithms are appropriate and effective for categorising both normal and attacked data. As a result, a novel feature selection-whale optimization algorithm was developed. This study proposes a deep neural network (FS-WOA-DNN) method for effectively mitigating DDoS attacks. The input dataset is first pre-processed, with the min-max normalisation technique used to replace all of the input within a specified range. Later, the normalised data is fed into the proposed FSWOA, which selects the best set of features to make the classification process easier. Because of the severity of DDoS attacks in large scale multinational corporations, conducting research in attack prevention models is advantageous. Rather than individual instantiations, IDS schemes for detecting novel attacks will not be considered in this paper.

Wu Zhijun et al [2020] [13] proposed a SDN uses centralised control logic, it is vulnerable to various types of Distributed Denial of Service (DDoS) attacks. This paper investigates the mechanism of low-rate DDoS attacks against the SDN data layer in order to improve detection accuracy, and then proposes a multi-feature DDoS attack detection method based on Factorization Machine (FM). The FM algorithm can detect fine-grained low-rate DDoS attacks, providing a reliable condition for defence against such attacks. By dynamically adjusting the timeout, the load-aware method controls the growth of the number of flow rules. The weight of attribute value SDCC must be set manually according to the algorithm in the real situation, this may lead to some errors in the detection results.

Jin Ye et al [2018] [14] described a emergence of software-defined networks (SDN) introduces some novel methods to this topic, in which a deep learning algorithm is used to model the attack behaviour based on data collected from the SDN controller. In this paper, an SDN environment is built using the minuet and floodlight simulation platforms, 6-tuple characteristic values from the switch fowl table are extracted, and a DDoS attack model is built by combining SVM classification algorithms. The SOM algorithm is used to detect DDoS attacks by extracting DDoS attack statistics. This method has a low consumption rate and a high detection rate. Disadvantage of this method is that the detection has some hysteresis and the attack behaviour is not found in a timely and accurate manner.

Jesus Arturo Pérez-Díaz et al [2020] [15] explained a versatile modular architecture that enables the detection and mitigation of LR-DDoS attacks in SDN environments. Specifically. Use six machine learning (ML) models to train the intrusion detection system (IDS) in architecture and evaluate its performance using the Canadian Institute of Cybersecurity (CIC) DoS dataset. The evaluation results show that the approach is effective. Despite the difficulty in detecting LR-DoS attacks. In testing topology, the intrusion prevention detection system mitigates all attacks previously detected by the IDS system. This demonstrates the utility of the architecture in identifying and mitigating LR-DDoS attacks. The goal of such techniques is to avoid blocking legitimate users when the false positive rate increases is drawback of this analysis.

3. PROPOSED WORK

Distributed Denial-of-Service (DDoS) attacks have drawn extensive attention in the cyberspace during the last few years. In the recent years, the concepts and the techniques of the Software Defined Networking (SDN) have been introduced and widely researched. The DDoS attacks can threaten the availability of the SDN due to the difference in the architecture between the SDN network and the traditional network. Especially, the SDN controller is the most vulnerable part to be affected by the DDoS attacks. In general, the DoS attack is an attempt to make the resources of a network unavailable for legitimizing users. In general, the DoS attack is an attempt to make the resources of a network unavailable for legitimizing users. DoS attack on an SDN using separated logic of the SDN in the control-data planes and developed a network scanning tool that could identify an SDN network. In their method, since the data path had different values in the flow response times for the existing and the new flows due to the querying of the controller, the time values were gathered based on the header field by the scanner which could scan the network in order to change the network header fields. Once the network was found to be considered as a SDN network, the flow requests were transmitted to the target network, which were forwarded by the data path to the controller. However, increasing the number of the flows in the data path will make the switches suffer from flow setup requests on the controller and hence eventually cause it to be broken a DDoS attack on the SDN controller is where an attacker continuously sends IP packets with random headers to disrupt the controller. However, a DDoS detection mechanism was required since the secondary controller could also be vulnerable to the DoS or the DDoS attacks. Hence, the use of multiple controllers still could not completely resolve the problem of the DDoS attacks since it could lead to cascading fault of multiple controllers.

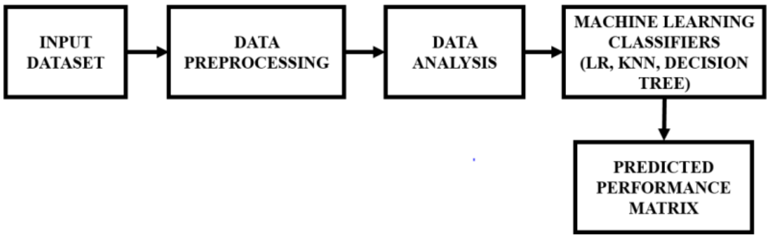

INPUT DATASET
DATA PREPROCESSING
DATA ANALYSIS
MACHINE LEARNING CLASSIFIERS (LR, KNN, DECISION TREE)
PREDICTED PERFORMANCE MATRIX

Figure 1 Block Diagram for Proposed System

This proposed work uses a machine learning method to classify and predict the different kinds of DDoS attacks. SDN Dataset is used in this project. The initial stage of data pre-processing involves an analysis of the input dataset. The datasets are pre-processed using Standard Scalar. The dataset's useless data is handled using data pre-processing techniques. Following data pre-processing, the stage of data analysis includes researching the observed data. For the data, the following machine learning classifier technique is allowable. The machine learning classifier technique uses the LR, KNN, Decision Tree algorithms. The KNN classifier algorithm is used in the first classification for both Precision (PR) and Recall (RE). A significant advantage of a decision tree is that it forces the consideration of all possible outcomes of a decision and traces each path to a conclusion. Finally, the data are evaluated to achieve a prediction output.

3.1 Input Dataset

To ensure the accuracy of intrusion detection systems, one must take into account the dataset used. Today's exponential growth of networks and applications makes secure network resilience necessary. It could be accomplished by selecting the proper learning and testing datasets.

3.2 Data Preprocessing

Data preprocessing, a component of data preparation, describes any type of processing performed on raw data to prepare it for another data processing procedure. It has traditionally been an important preliminary step for the data mining process. The need for data preprocessing is there because good data is undoubtedly more important than good models and for which the quality of the data is of paramount importance. Therefore, companies and individuals invest a lot of their time in cleaning and preparing the data for modeling. The data present in the real world contains a lot of quality issues, noise, inaccurate, and not complete. It may not contain relevant, specific attributes and could have missing values, even incorrect and spurious values. To improve the quality of the data preprocessing is essential. The preprocessing helps to make the data consistent by eliminating any duplicates, irregularities in the data, normalizing the data to compare, and improving the accuracy of the results.

The data preprocessing techniques in machine learning can be broadly segmented into two parts:

- Data cleaning
- Data transformation

3.2.1 Data Cleaning

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled.

As we have seen, the real-world data is not all complete, accurate, correct, consistent, and relevant. The first and the primary step is to clean the data. There are various steps in this stage, it involves:

- Making the data consistent across the values, which can mean:
- The attributes may have incorrect data types and are not in sync with the data dictionary. Correction of the data types is a must before proceeding with any type of data cleaning.
- Making the format of the date column consistent with the format of the tool used for data analysis.
- Check for null or missing values, also check for the negative values. The relevancy of the negative values depends on the data. In the income column, a negative value is spurious though the same negative value in the profit column becomes a loss.
- Smoothing of the noise present in the data by identifying and treating for outliers.

3.2.2 Data Transformation

Data transformation is defined as the technical process of converting data from one format, standard, or structure to another – without changing the content of the datasets – typically to prepare it for consumption by an app or a user or to improve the data quality.

3.3 Data Analysis

Although many groups, organizations, and experts have different ways of approaching data analysis, most of them can be distilled into a one-size-fits-all definition. Data analysis is the process of cleaning, changing, and processing raw data and extracting actionable, relevant information that helps businesses make informed decisions. The procedure helps reduce the risks inherent in decision-making by providing useful insights and statistics, often presented in charts, images, tables, and graphs.

3.3.1 Data Analysis Process

The process of data analysis, or alternately, data analysis steps, involves gathering all the information, processing it, exploring the data, and using it to find patterns and other insights.

The process of data analysis consists of:

Data Requirement Gathering: Ask yourself why you're doing this analysis, what type of data you want to use, and what data you plan to analyse.

Data Collection: Guided by your identified requirements, it's time to collect the data from your sources. Sources include case studies, surveys, interviews, questionnaires, direct observation, and focus groups. Make sure to organize the collected data for analysis.

Data Cleaning: Not all of the data you collect will be useful, so it's time to clean it up. This process is where you remove white spaces, duplicate records, and basic errors. Data cleaning is mandatory before sending the information on for analysis.

Data Analysis: Here is where you use data analysis software and other tools to help you interpret and understand the data and arrive at conclusions. Data analysis tools include Excel, Python, R, Looker, Rapid Miner, Chartio, Metabase, Redash, and Microsoft Power BI.

Data Interpretation: Now that you have your results, you need to interpret them and come up with the best courses of action based on your findings.

Data Visualization: Data visualization is a fancy way of saying, “graphically show your information in a way that people can read and understand it.” You can use charts, graphs, maps, bullet points, or a host of other methods. Visualization helps you derive valuable insights by helping you compare datasets and observe relationships.

3.4 Machine Learning Classifiers

For classifications we have used two types of classifiers. They are:

- LR classifiers
- KNN classifiers
- Decision Tree.

3.4.1 Logistic Regression (Lr) Classifiers

Logistic regression is an example of supervised learning. Logistic regression is a simple and very effective classification algorithm. It is used to calculate or predict the probability of a binary (yes/no) event occurring. The logistic regression classification is used to find the values of precision (PR), F1 score, and recall (RE). Logistic regression is used in various fields, including machine learning, most medical fields, and social sciences. Logistic regression can also be prone to over fitting, particularly when there is a high number of predictor variables within the model. Regularization is typically used to penalize parameters large coefficients when the model suffers from high dimensionality.

3.4.2 Knn Classifiers

KNN is one of the simplest forms of machine learning algorithms mostly used for classification. It classifies the data point on how its neighbor is classified. KNN classifies the new data points based on the similarity measure of the earlier stored data points. 'K' in KNN is a parameter that refers to the number of nearest neighbors to include in the majority of the voting process. KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

3.4.2.1 K-Data Mining Concept

KNN stands for k nearest neighbor classifications, identifying new records by a combination of K's most recent historical records. KNN is a well-known statistical method that has been studied intensively in pattern recognition over the past 40 years. KNN has been applied to text categorization in early research strategies and is one of the highly operational methods of the benchmark Reuter’s body. Other methods, such as LLSF, decision trees, and neural networks The idea of KNN is as follows: First, calculate the distance between the new sample and the training sample, find the nearest K neighbors; then, according to the category to which the neighbor belongs, determine the category of the new sample, if they all belong to the same category, then The new sample also falls into this category; otherwise, each post-selection category is scored and the new sample category is determined according to certain rules. Take the K neighbors of the unknown sample X, and look at which category the K neighbors belong to, and classify X into which category. That is, among the K samples of X, K neighbors of X are found. The KNN grows from the test sample X, continuously expanding the area until it contains K training samples, and classifies the test sample X as the most frequently occurring category among the most recent K training samples. For example, in the case of K=6 in Fig. 4.2, the test sample X is classified into a black category according to the decision rule.

The neighborhood classification is a lazy learning method based on the eyeball, that is, it stores all the training samples and knows that the new samples need to be classified to establish the classification. This is in stark contrast to decision numbers and back propagation algorithms, which need to construct a general model before accepting new samples to be classified. Lazy learning is faster in training than in eager learning, but slower in classification because all calculations are postponed until then.

3.4.2.2 Mathematical Model for Knn Algorithm

The reason for prediction using the nearest neighbor method is based on the assumption that objects of neighbors have similar prediction values. The basic idea of the nearest neighbor algorithm is to find k points nearest to the unknown sample in the multidimensional space Rn, and judge the class of the unknown sample according to the categories of the k points. These k points are the k-nearest neighbors of the unknown samples. The algorithm assumes that all instances correspond to points in dimensional space. The nearest neighbor of an instance is defined according to the standard Euclidean distance. Let the eigenvector of x be:

$$<a_1(X), a_2(X), \ldots, a_n(X)> \qquad (3.1)$$

Where,

a_r(x) Represents the rth attribute value of instance x.

The distance between the two instances X_i and X_j is defined as$d(X_i, Y_j)$,

$$d(X_i, Y_j) = \sqrt{\sum_{r=1}^{n} \left(ar(Xi) - ar(xj)\right)2} \quad (3.2)$$

3.4.3 Decision Tree Algorithms

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. Decision trees in machine learning provide an effective method for making decisions because they lay out the problem and all the possible outcomes. Also, the Decision Tree is used for estimating the precision (PR), F1 score, and recall values (RE). Decision trees help you to evaluate your options. Decision trees are excellent tools for helping you to choose between several courses of action. They provide a highly effective structure within which you can lay out options and investigate the possible outcomes of choosing those options.

3.5 Model Evaluation

This technique of Evaluation helps us to know which algorithm best suits the given dataset for solving a particular problem. Likewise, in terms of Machine Learning it is called as "Best Fit". It evaluates the performance of different Machine Learning models, based on the same input dataset. The method of evaluation focuses on accuracy of the model, in predicting the end outcomes. Out of all the different algorithms we use in the stage, we choose the algorithm that gives more accuracy for the input data and is considered as the best model as it better predicts the outcome. The accuracy is considered as the main factor, when we work on solving different problems using machine learning. If the accuracy is high, the model predictions on the given data are also true to the maximum possible extent. There are several stages in solving an ML problem like collection of dataset, defining the problem, brainstorming on the given data, preprocessing, transformation, training the model and evaluating. Even though there are several stages, the stage of Evaluation of a ML model is the most crucial stage, because it gives us an idea of the accuracy of model prediction. The performance and usage of the ML model is decided in terms of accuracy measures at the end.

3.6 Model Prediction

Data analysis have variation from company to company depending upon the needs, so various data model has been designed to meet the requirements. Predictive modeling is the subpart of data analytics that uses data mining and probability to predict results. Each model is built up by the number of predictors that are highly favorable to determine future decisions. Once the data is received for a specific predictor, an analytical model is formulated.

There are basically two types of predictive modeling;

- Parametric Model
- Non- Parametric Model

3.6.1 Parametric Model

Assumptions are the crucial part of any data model, it not only makes the model easy also improves predictions, so the algorithms that consider assumptions and make the function simple are known as parametric ML algorithms, and a learning model that compiles data with different parameters of a predetermined size, independent to number of training variables, is termed as parametric model.

3.6.2 Non- Parametric Model

ML algorithms that enable to make strong assumptions in terms of the mapping function are called non-parametric Ml algorithms and without worth assumptions, ML algorithms are available to pick up any functional form training data. Non-parametric models are a good fit for the huge amount of data with no previous knowledge.

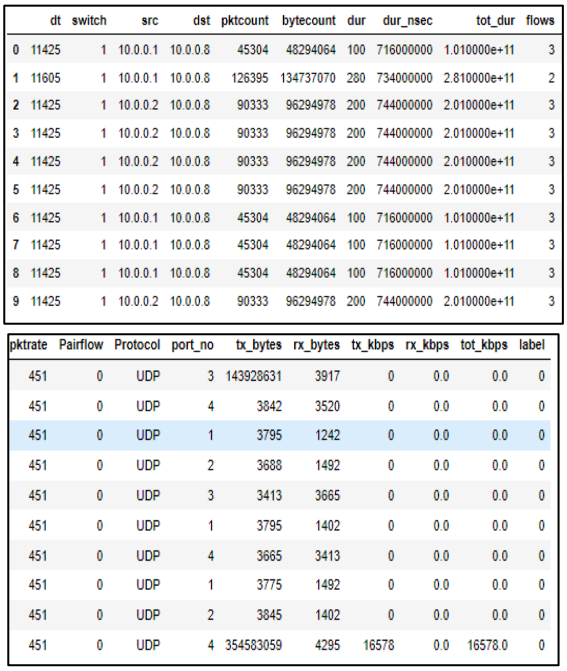
3.7 Predicted Performance Matrix

The module predicts the classified datasets and evaluates the values of accuracy, recall, time, F1 score, precision, and confusion matrix. Finally compared the performance of the three classification methods.

4. RESULT AND DISCUSSION

This project is implemented using python software. Anaconda software helps you create an environment for many different versions of Python and package versions. Anaconda is also used to install, remove, and upgrade packages in your project environments. Furthermore, you may use Anaconda to deploy any required project with a few mouse clicks. The Jupiter Notebook application allows you to create and edit documents that display the input and output of a Python or R language script. Once saved, you can share these files with others.

4.1 Input Dataset



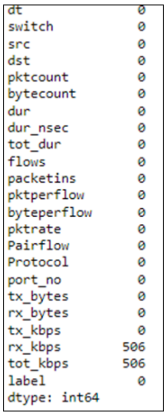
	dt	switch	src	dst	pktcount	bytecount	dur	dur_nsec	tot_dur	flows
0	11425	1	10.0.0.1	10.0.0.8	45304	48294064	100	716000000	1.010000e+11	3
1	11605	1	10.0.0.1	10.0.0.8	126395	134737070	280	734000000	2.810000e+11	2
2	11425	1	10.0.0.2	10.0.0.8	90333	96294978	200	744000000	2.010000e+11	3
3	11425	1	10.0.0.2	10.0.0.8	90333	96294978	200	744000000	2.010000e+11	3
4	11425	1	10.0.0.2	10.0.0.8	90333	96294978	200	744000000	2.010000e+11	3
5	11425	1	10.0.0.2	10.0.0.8	90333	96294978	200	744000000	2.010000e+11	3
6	11425	1	10.0.0.1	10.0.0.8	45304	48294064	100	716000000	1.010000e+11	3
7	11425	1	10.0.0.1	10.0.0.8	45304	48294064	100	716000000	1.010000e+11	3
8	11425	1	10.0.0.1	10.0.0.8	45304	48294064	100	716000000	1.010000e+11	3
9	11425	1	10.0.0.2	10.0.0.8	90333	96294978	200	744000000	2.010000e+11	3

pktrate	Pairflow	Protocol	port_no	tx_bytes	rx_bytes	tx_kbps	rx_kbps	tot_kbps	label
451	0	UDP	3	143928631	3917	0	0.0	0.0	0
451	0	UDP	4	3842	3520	0	0.0	0.0	0
451	0	UDP	1	3795	1242	0	0.0	0.0	0
451	0	UDP	2	3688	1492	0	0.0	0.0	0
451	0	UDP	3	3413	3665	0	0.0	0.0	0
451	0	UDP	1	3795	1402	0	0.0	0.0	0
451	0	UDP	4	3665	3413	0	0.0	0.0	0
451	0	UDP	1	3775	1492	0	0.0	0.0	0
451	0	UDP	2	3845	1402	0	0.0	0.0	0
451	0	UDP	4	354583059	4295	16578	0.0	16578.0	0

Figure 4.1 Input Dataset

The SDN dataset includes several types of DoS attacks that can be driven in different OSI model layers. SDN dataset also includes several DDoS attacks scenarios such as TCP-SYN Flood, UDP Flood, and ICMP Flood attacks. In figure 4.1, the input dataset is displayed.

4.2 Data Preprocessing



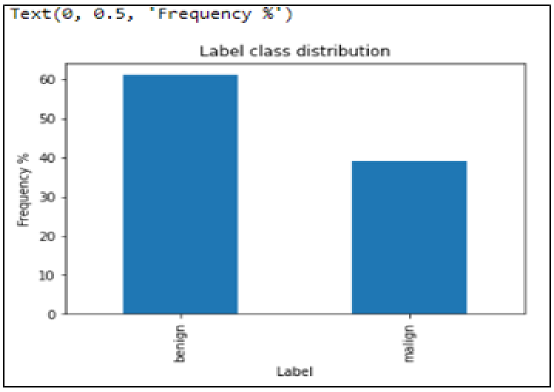
```
dt             0
switch         0
src            0
dst            0
pktcount       0
bytecount      0
dur            0
dur_nsec       0
tot_dur        0
flows          0
packetins      0
pktperflow     0
byteperflow    0
pktrate        0
Pairflow       0
Protocol       0
port_no        0
tx_bytes       0
rx_bytes       0
tx_kbps        0
rx_kbps      506
tot_kbps     506
label          0
dtype: int64
```

Figure 4.2 Null Values

Figure 4.2 displays the null values. Data preprocessing, a component of data preparation, describes any type of processing performed on raw data to prepare it for another data processing procedure. To improve the quality of the data, preprocessing is essential. In preprocessing, the collected SDN datasets are given as input to the machine, which is then trained accordingly.

4.3 Data Analysis

Data analysis is the process of systematically applying statistical techniques to describe and illustrate, condense and recap, and evaluate data. Data analysis is used to extract useful information from data and make a decision based on the analysis.



```
Text(0, 0.5, 'Frequency %')
```

Figure 4.3 Label Class Distribution

The label class distribution is shown in figure 4.3

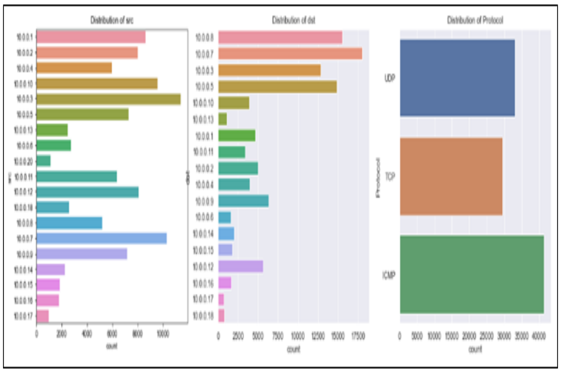


Figure 4.4 Distributions of SRC, DST and Protocol. Distributions of SRC, DST, and Protocol are illustrated in figure 4.4.

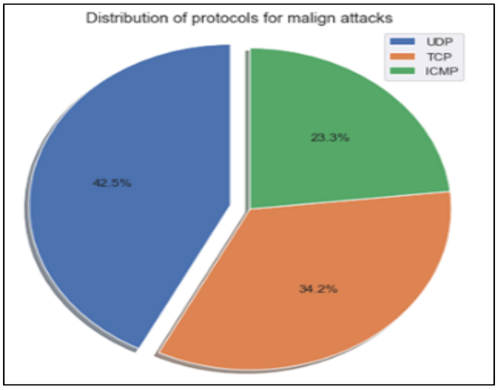
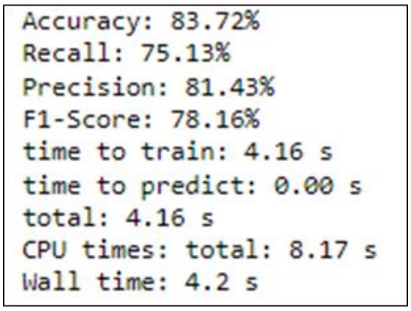


Figure 4.5 Distribution of Protocol for Malign Attacks

Figure 4.5 demonstrates the spatial distribution of the protocol for malicious attacks.

4.4 Logistic Regression



```
Accuracy: 83.72%
Recall: 75.13%
Precision: 81.43%
F1-Score: 78.16%
time to train: 4.16 s
time to predict: 0.00 s
total: 4.16 s
CPU times: total: 8.17 s
Wall time: 4.2 s
```

Figure 4.6 Performance matrix for Logistic Regression

The figure 4.6 displays a performance matrix for logistic regression. Logistic regression is an example of supervised learning. Logistic regression is a simple and very effective classification algorithm. It is used to calculate or predict the probability of a binary (yes/no) event occurring. The logistic regression classification is used to find the values of precision (PR), F1 score, and recall (RE).

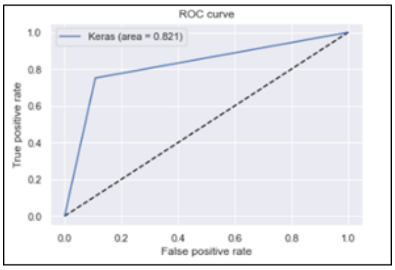
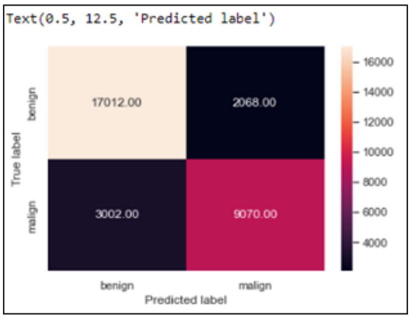


Figure 4.7 ROC Curve for Logistic Regression

Figure 4.7 represents a Logistic Regression ROC Curve. An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate. False Positive Rate.



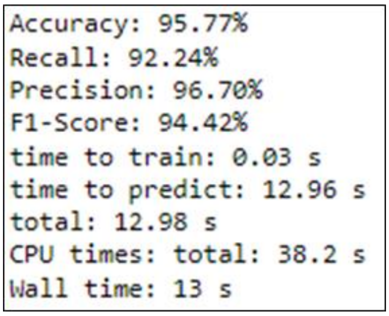
```
Text(0.5, 12.5, 'Predicted label')
```

Figure 4.8 Confusion Matrix for Logistic Regression

Confusion matrix for logistic regression are given in Figure 4.8. One common way to evaluate the quality of a logistic regression model is to create a confusion matrix, which is a 2×2 table that shows the predicted values from the model vs. the actual values from the test dataset.

4.5 K-Nearest Neighbors Algorithm

KNN is one of the most basic types of machine learning algorithms, and it is usually used for categorisation. It categorises the data point based on the classification of its neighbours. KNN classifies new data points based on their similarity to previously stored data points. The KNN classification (RE) is used to calculate the precision (PR), F1 score, and recall values.



```
Accuracy: 95.77%
Recall: 92.24%
Precision: 96.70%
F1-Score: 94.42%
time to train: 0.03 s
time to predict: 12.96 s
total: 12.98 s
CPU times: total: 38.2 s
Wall time: 13 s
```

Figure 4.9 Performance matrix for KNN

Metrics like accuracy, precision, recall are good ways to evaluate classification models for balanced datasets, but if the data is imbalanced then other methods like ROC/AUC perform better in evaluating the model performance. Performance for KNN is shown in a matrix in figure 4.9.

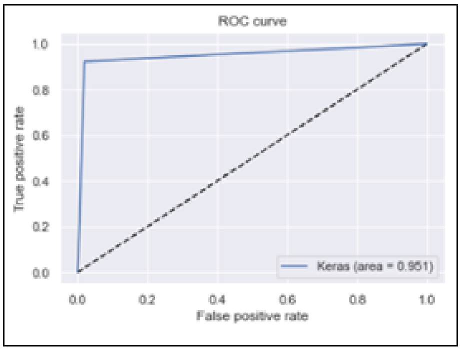
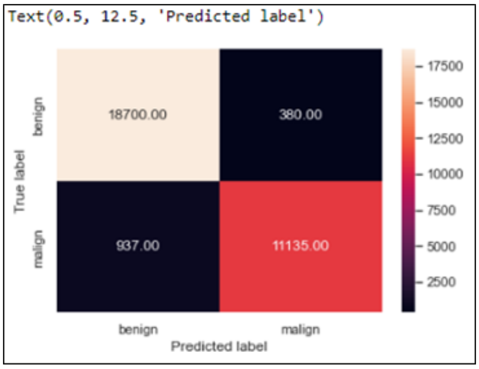


Figure 4.10 ROC Curve for KNN

The ROC curve for KNN is depicted in figure 4.10.



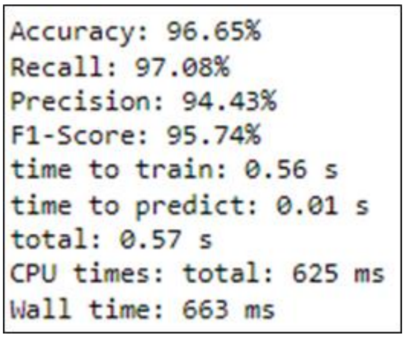
```
Text(0.5, 12.5, 'Predicted label')
```

Figure 4.11 Confusion Matrix for KNN.

Confusion matrix for KNN is demonstrated in Figure 4.11. A confusion matrix is a summary of predictions of the classification problem. The correct and incorrect predictions are totalled and broken down by class using count values.

4.6 Decision Tree

A decision tree is a non-parametric supervised learning approach that can be used for classification as well as regression applications. Because they lay out the problem and its possible outcomes, decision trees in machine learning give an excellent technique for making decisions. The Decision Tree is also used to calculate the precision (PR), F1 score, and recall values (RE). Figure 4.12 depicts the decision tree's performance matrix.



```
Accuracy: 96.65%
Recall: 97.08%
Precision: 94.43%
F1-Score: 95.74%
time to train: 0.56 s
time to predict: 0.01 s
total: 0.57 s
CPU times: total: 625 ms
Wall time: 663 ms
```

Figure 4.12 Performance matrix for Decision Tree

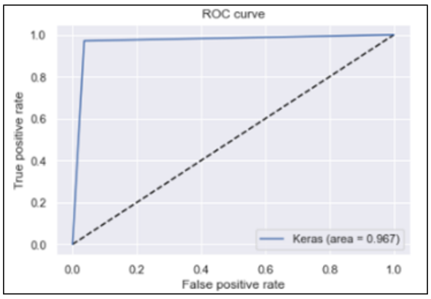
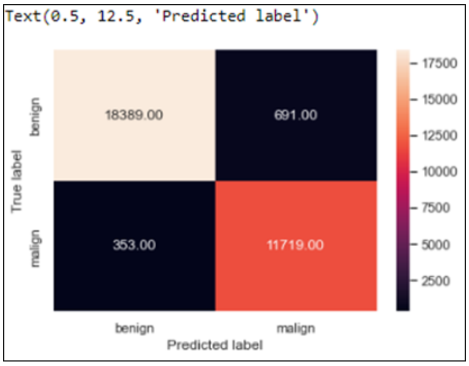


Figure 4.13 ROC Curve for Decision Tree

Figure 4.13 indicates the ROC Curve for the Decision Tree. An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. True Positive Rate and False Positive Rate, two parameters, are plotted on this curve.



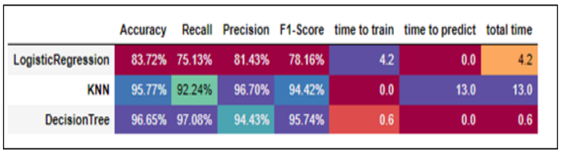
```
Text(0.5, 12.5, 'Predicted label')
```

Figure 4.14 Confusion Matrix for Decision Tree

The Confusion Matrix for a Decision Tree is presented in Figure 4.14. A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the total number of target classes. The matrix compares the actual target values with those predicted by the machine learning model.

4.7 Predicted Performance Matrix

The module predicts the classified datasets and evaluates the values of accuracy, recall, time, F1 score, precision, and time. Finally compared the performance of the three classification methods. Figure 4.15 represents the comparison of the classifications.



	Accuracy	Recall	Precision	F1-Score	time to train	time to predict	total time
LogisticRegression	83.72%	75.13%	81.43%	78.16%	4.2	0.0	4.2
KNN	95.77%	92.24%	96.70%	94.42%	0.0	13.0	13.0
DecisionTree	96.65%	97.08%	94.43%	95.74%	0.6	0.0	0.6

Figure 4.15 Comparison of Classification

5. CONCLUSION

The DDoS attack is one of the most well-known and serious cyber-attacks in recent memory. The goal of conducting a DDoS attack is to consume the victim's resources. The attacker sends a massive amount of traffic to the victim. As a result, these services would not be used for a specified period of time, and so the service would be unavailable to legitimate clients. This research proposes a detailed simulation of a machine learning algorithm for categorising and forecasting various types of DDoS attacks. The modules that comprise our suggested technique are as follows: pre-processing, data analysis and categorization, and performance matrix. To begin, all incoming traffic attributes are normalised on a standard scale in order to use various machine learning algorithms. In this project, the classification algorithms logistic regression, KNN, and decision tree work admirably. Finally, assesses the comparison to see how well the three classification techniques performed.

REFERENCE

[1] Mohmand, Muhammad Ismail, Hameed Hussain, Ayaz Ali Khan, Ubaid Ullah, Muhammad Zakarya, Aftab Ahmed, Mushtaq Raza, Izaz Ur Rahman, and Muhammad Haleem. "A machine learning-based classification and prediction technique for DDoS attacks." IEEE Access, Vol. 10, pp. 21443-21454. 2022.

[2] Bagyalakshmi, C., and E. S. Samundeeswari. "DDoS attack classification on cloud environment

using machine learning techniques with different feature selection methods." Int. J, Vol. 9, no. 5, 2020.

[3] Damodhar, Ch Meghana, Ch Samyuktha, and Ch Rashmitha. "Prediction Approach Against Ddos Attack Based On Machine Learning Multiclassfier." Turkish Journal of Computer and Mathematics Education (TURCOMAT), Vol. 14, No. 2, pp. 251-257, 2023.

[4] Pei, Jiangtao, Yunli Chen, and Wei Ji. "A DDoS attack detection method based on machine learning." In Journal of Physics: Conference Series, Vol. 1237, no. 3, p. 032040. IOP Publishing, 2019.

[5] Li, Qian, Linhai Meng, Yuan Zhang, and Jinyao Yan. "DDoS attacks detection using machine learning algorithms." In Digital TV and Multimedia Communication: 15th International Forum, IFTC 2018, Shanghai, China, September 20–21, 2018, Revised Selected Papers 15, pp. 205-216. Springer Singapore, 2019.

[6] Sahoo, Kshira Sagar, Amaan Iqbal, Prasenjit Maiti, and Bibhudatta Sahoo. "A machine learning approach for predicting DDoS traffic in software defined networks." In 2018 International Conference on Information Technology (ICIT), pp. 199-203. IEEE, 2018.

[7] Alduailij, Mona, Qazi Waqas Khan, Muhammad Tahir, Muhammad Sardaraz, Mai Alduailij, and Fazila Malik. "Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method." Symmetry, Vol. 14, no. 6, pp. 1095, 2022.

[8] Sambangi, Swathi, and Lakshmeeswari Gondi. "A machine learning approach for ddos (distributed denial of service) attack detection using multiple linear regression." In Proceedings, Vol. 63, no. 1, p. 51. MDPI, 2020.

[9] Idhammad, Mohamed, Karim Afdel, and Mustapha Belouch. "Semi-supervised machine learning approach for DDoS detection." Applied Intelligence, Vol. 48, pp. 3193-3208, 2018.

[10] Najafimehr, Mohammad, Sajjad Zarifzadeh, and Seyedakbar Mostafavi. "A hybrid machine learning approach for detecting unprecedented DDoS attacks." The Journal of Supercomputing, Vol. 78, no. 6, pp. 8106-8136, 2022.

[11] Aljuhani. (2021), "Machine Learning Approaches for Combating Distributed Denial of Service Attacks in Modern Networking Environments," in IEEE Access, Vol. 9, pp. 42236-42264.

[12] Agarwal. A, Khari. M., & Singh. (2021), "R. Detection of DDOS attack using deep learning model in cloud storage application," Wireless Personal Communications, pp. 1-21.

[13] B. Nugraha and R. N. Murthy. (2020), "Deep Learning-based Slow DDoS Attack Detection in SDN-based Networks," 2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Leganes, Spain.

[14] H. A. Alamri and V. Thayananthan. (2020), "Bandwidth Control Mechanism and Extreme Gradient Boosting Algorithm for Protecting Software-Defined Networks Against DDoS Attacks," in IEEE Access, Vol. 8, pp. 194269-194288, 2018.

[15] J. A. Pérez-Díaz, I. A. Valdovinos, K. -K. R. Choo and D. Zhu. (2020), "A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning," in IEEE Access, Vol. 8, pp. 155859-155872, 2020.