DDoS Attack Identification - ML Project

Divyam Choudhary
MS2022011
International Institute of Information Technology
Bangalore
divyam.choudhary@iiitb.ac.in

Butani Prince Nileshbhai
MS2022006
International Institute of Information Technology
Bangalore
butaniprince.nileshbhai@iiitb.ac.in

***Abstract*—Internet service consumption has lately increased due to the advancement of cutting-edge digital technology. To be successful, these internet enterprises must be able to regularly and successfully deliver their services. As a result of the DDoS attack, internet resources suffer in terms of availability and computing power. DDoS assaults are useful for cyber-attackers since there is no practical way to detect them. In recent years, academics have been experimenting with various cutting-edge techniques, such as machine learning (ML) approaches, to see whether they can develop efficient ways for identifying DDoS attacks. In this project, various models employ machine learning and big data to detect DDoS attacks.**

***Index Terms*—DDoS attack, Machine Learning, Big Data**

I. Introduction

A DDoS attack occurs when a large number of hostile computers attack the victim's resources at the same time. Assault programmes like Slowloris, GoldenEye, and others make it simple for anybody to launch a DDoS attack on a target, wreaking havoc on their resources or rendering their bandwidth unreachable to others. DDoS attacks can take many forms, making it challenging for the detection filter to keep up. TCP flooding occurs when an attacker sends many SYN packets to the victim's end to overflow the connection table.

II. Data and Preprocessing

A. Data set

The data set utilised for this project is the Intrusion Detection Evaluation Data set, and it serves as the foundation for the work done. The specific data set was previously established to replace numerous data sets used in cybersecurity literature for studies to enhance intrusion detection systems. The data set was built by simulating multiple assaults in a realistic testbed architecture over a week. The simulated network includes four attackers and ten victim PCs, with 25 people accessing the network using protocols such as HTTPS, HTTP, FTP, SSH, and email. The attacking computers attempt various assaults attempted by the attacking computers during the simulation of a DDoS attack.

The enormous data set contains 78 features assessed over 2.8 million network flow events. A flow is defined as a succession of packets with the same Source IP, Destination IP, Source Port, Destination Port, and Protocol (TCP or UDP) variables. The data set captures a substantial variety of extra variables utilising CICFlowMeter and fundamental network flow statistics such as the destination port, flow time, and the total number of packets. Statistical measurements include the maximum, minimum, mean, and standard deviation of packet length, the flow's active/idle duration, the inter-arrival time of packet length, the flow's active/idle duration, and the inter-arrival time between flows are among these properties.

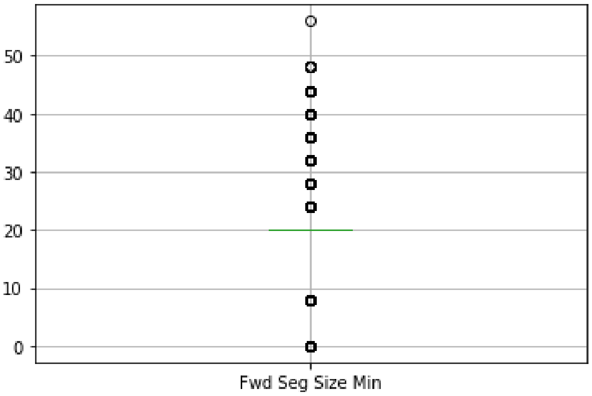

Fig. 1. Box plot of 'Fwd Seg Size Min' showing the data distribution.

B. Pre-processing

The data set had 78 features, among which many were not required for the implementation. The column features such as Bwd URG Flags, Bwd PSH Flags, Fwd URG Flags, Fwd PSH Flags, Pkt Len Min, FIN Flag Cnt, SYN Flag Cnt, URG Flag Cnt, CWE Flag Count, Fwd Byts/b Avg, Fwd Pkts/b Avg, Fwd Blk Rate Avg, Bwd Byts/b Avg, Bwd Pkts/b Avg and Bwd Blk Rate Avg have a lot of zero values and hence are neglected. Many other column features have infinite values, and hence those values are removed. Box plots were plotted for features to understand the data distribution in a better way. Since the data set is vast, every column feature was reduced in size to int32, float32, int16 or float16, whichever was applicable for that particular column. The duplicate values were checked for every column and were removed from the data set.

III. Models

Different Machine learning models were implemented to train and test the data set and measure the performance of each model.

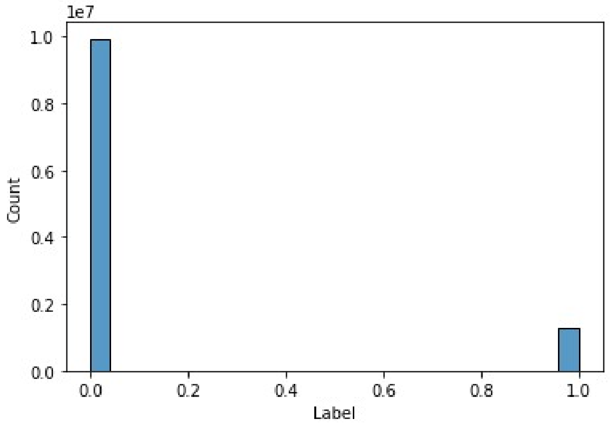

Fig. 2. Histogram plotted using the seaborn library, depicting the imbalanced nature of the data set

A. *Logistic Regression Model*

After the pre-processing steps, a histogram plot was used to check whether the data was imbalanced. The data had 11.36% of Malicious values and 88.64% Benign values. After this, random under-sampling was carried out. The data was then divided into train and test data in the ratio of 4:1 with a test size of 0.2. After fitting the model, it didn't converge with the max iteration of 1000 and was increased to 10,000. After this, the model converged, generating a confusion matrix. The accuracy, as well as the F1 score obtained here, is 74%.

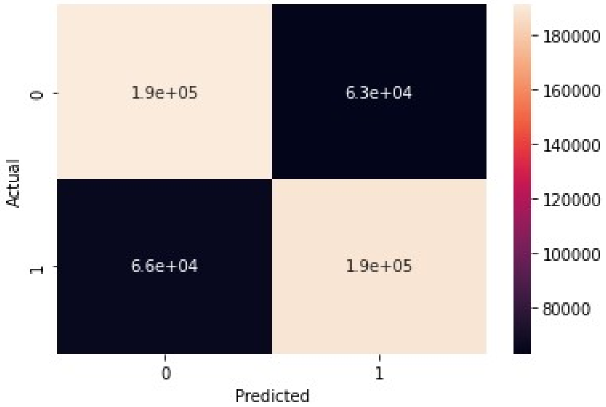

Fig. 3. Confusion Matrix showing the actual and predicted values.

B. *Random Forest*

The data was divided into a train and test split in the ratio of 4:1, and a random forest classifier was introduced. Hyper-parameters with two n-estimators of 50 and 100 were implemented, i.e. 50 decision trees and 100 decision trees for both types of random forest. Grid search was implemented, and the number of cross-validation was 3, i.e. two training and one test for the models. In total, out of 6 fits, the model which estimates the best is to be selected. Since the large data set, all the available CPU cores were used, reducing the load on a single core. Later, the accuracy and F1 score was calculated and found to be 95%, which is a good accuracy.