DDoS Attack Detection using Supervised Machine Learning Techniques

Bhavana G[1], Dr. Tanuja R[2]

[1] *Student in Master of Technology, Department of Computer Science Engineering, University Visvesvaraya College of Engineering, Bangalore University, Karnataka, India*
[2] *Associate Professor, Department of Computer Science Engineering, University Visvesvaraya College of Engineering, Bangalore University, Karnataka, India*

ABSTRACT

There has been a massive growth in cyberspace in the last few decades which has demanded the implementation of efficient networks for communication. There are various types of cyber-attacks, and they grow in congruence with every new technology where each has a varying level of threat impact. In a simple network sniffing attack, the target might not experience any consequences while the impact of a DDoS attack is on the contrary. DDoS attacks are simple to perform but the effects of it result in production downtime, financial losses, customer dissatisfaction and loss of reputation of the targeted businesses therefore becoming an important security concern. This study proposes an effective solution for the detection of DDoS attacks using four different Supervised Machine Learning techniques including Random Forest, K-Nearest Neighbor, AdaBoost and Logistic Regression. These algorithms are trained and tested with a subset of the CICDoS 2019, 2018 and 2017 datasets; and the classification accuracy scores are 99.99%, 99.92%, 99.97% and 98.06% respectively. The dataset cluster consists of about 5,00,000 records and 8 features were selected as necessary from a total of 80 features. The training time and data class classification metrics are considered for the determination of the most appropriate Supervised ML technique for the base model and Random Forest is selected as the base for validation since it performs better when compared to other techniques. Wireshark is used to gather real time network traffic and the captured packets are then given as input for the trained Random Forest model for validation purpose and it is found that the prediction values are accurate. The prediction '0' suggests that the nature of the traffic is Benign and the predicted '1' suggests that the nature of the traffic is DDoS.

Keyword: - *DDoS Attack Detection, Deep Learning, Intrusion Detection, IDS and Supervised Machine Learning*

1. INTRODUCTION

DDoS attacks are cyber-attacks in which there is a malicious attempt to interfere with the normal flow of network traffic of a network or a server, overwhelming the target or the surrounding environment with huge amounts of traffic resulting in denial of service. These attacks are usually perpetrated using compromised computer systems, distributed in nature, infected with malware allowing the attacker to have remote access to these systems. The network of these compromised systems is called a botnet, and the individual nodes are called bots. Some of the scenarios where DDoS attacks are often performed are, to pull down a competitor's website or web application, politically motivated attack, for financial gain or for revenge purposes. The repercussions include downtime of web applications and added vulnerabilities, server and hosting issues, loss of money, time, customer satisfaction, reliability, and reputation. politically motivated attack, for financial gain or for revenge purposes. The repercussions include downtime of web applications and added vulnerabilities, server and hosting issues, loss of money, time, customer satisfaction, reliability, and reputation.

In recent times, the severity of DDoS attack incidents has escalated significantly. MICROSOFT AZURE, in the latter half of 2021, identified a staggering 359,713 instances of DDoS attacks, marking a substantial 43% surge compared to the preceding six months. To undermine the detection capabilities of ISPs and firewalls, malicious actors often employ a hybrid DDoS attack strategy by combining various types of DDoS attack flows simultaneously. Hybrid DDoS attack incidents have emerged as the primary activities posing a significant threat to DDoS protection. These events introduce fresh challenges and complexities to the task of safeguarding against

DDoS attacks. Recent reports indicate that approximately 46% of DDoS threat events in 2021 involve the utilization of at least two different types of DDoS attacks.

Research indicates that despite the growing number of measures implemented to uphold network security, the evolution of DDos attacks continues, resulting in an escalating level of harm inflicted upon the network. Conversely, the conventional congestion-based DDoS attack technique has witnessed a consistent rise in peak traffic volume during attacks on an annual basis. According to a research report published by Tencent, a prominent Chinese Internet company, the volume of DDoS attacks experienced a consistent linear growth from 2013 to 2018. Notably, in March 2018, a game company faced a peak DDoS attack with a staggering magnitude of 1.7 Tbps. This study discusses the performance of various Supervised Machine Learning techniques such as Logistic regression, Adaboost, Random Forest and K-Nearest Neighbour used to detect DDoS attacks. The main objectives of this study are as follows:

- Implement a robust model to detect DDoS traffic in the network.
- Usage of Supervised Machine Learning techniques.
- Compare performance of proposed ML models.
- To improve the performance of existing models.

Fig - 1 represents the DDoS attack architecture. The attacker performs the DDoS attack using a network of malicious devices, majorly computers called as a botnet.

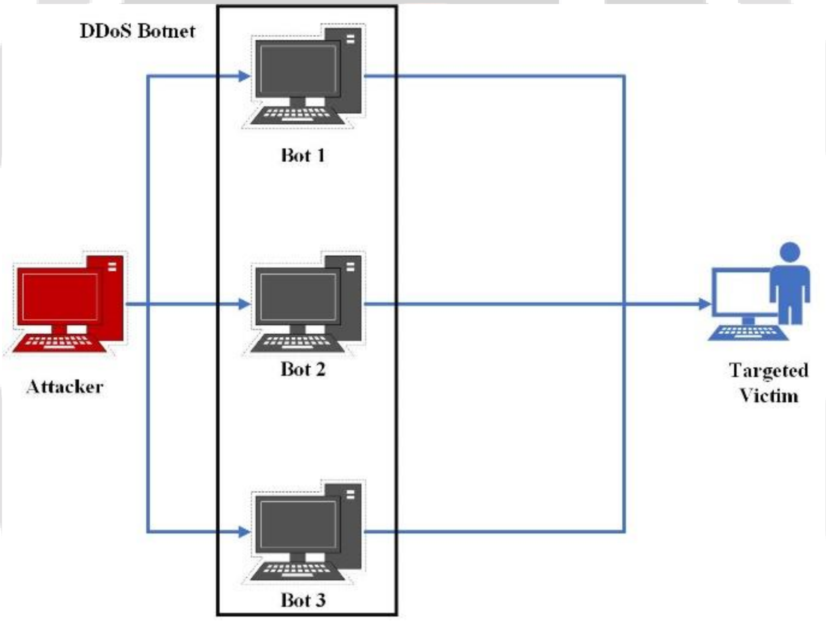


Fig -1: DDoS Attack Architecture

2. PREVIOUS WORK

Yini Chen, Jun Hou, Qianmu Li and Huaqiu Long [1] proposes a DDoS attack detection approach based on Random Forest Classification (RFC) method. This model is built on the fact that DDoS attacks are analogous to normal background traffic and hence the analysis of TCP flood attacks, UDP flood attacks and ICMP flood attacks is conducted and the characteristics of Data Stream Information Entropy (DISE) to characterize the attack behavior is also defined. The goal of this model is to predict if the network traffic is normal traffic or otherwise. It was found that 90% of the flood attacks were TCP flood attacks and about 6% of the flood attacks were UDP and ICMP. The simulation was divided into three parts, one for each attack type. The detection results are compared with HMM and SVM algorithms for verification. The proposed RFC model shows better performance compared to the other two models.

Yanchao Sun, Yuanfeng Han, Yue Zhang, Mingsong Chen, Shui Yu and Yimin Xu [2] proposes a two-stage DDoS attack detection algorithm combining time series-based multi-dimensional sketch and machine learning technologies. They construct the model with limited space cost by storing elephant flow information with the Boyer-Moore voting algorithm and hash index. For the first stage, they adopt a CNN model to produce sketch-level DDoS attack detection results from the time series-based multidimensional sketch. For the next stage, RNN model is used to implement flow DDoS detection using the information from the first stage. It is found that the accuracies and the computational time cost is less irrespective of the machine learning operation. This two – fold method reduces the number of calls to the machine learning function when compared with traditional flow methods.

Heena Kousar, Pooja Shettar and Narayan DG [3] proposes an Intrusion Detection technique where Apache Spark is used for the detection of DDoS attacks with NSL-KDD dataset is used as a benchmark. The advantage of using Spark over Hadoop being better processing delay. The model is trained with the NSL-KDD dataset which consists of 42 features from which only 7 features were extracted based on a correlation matrix, stored in KDD format and then converted to spark data frame since it is pre-processed faster and then stored in HDFS. For model training algorithms such as Random Forest, Decision tree, SVM, Naïve Bayes are implemented using the Spark MLlib library. Then the results such as recall, precision and accuracy are calculated. It is found that the accuracy for Random Forest and Decision Tree is 90.86% and 90.82% respectively. The proposed system also reduces the time taken to detect DDoS attacks and the efficiency is improved.

Tanut Visetbunditkum and Warakorn [4] proposes an ensemble model along with Recursive Feature Elimination (RFE) algorithm. Mixed learning is also performed using voting, bagging, boosting, and stacking. The CICIDS 2017 dataset is used which consists of more than 80 different network traffic features and Random Forest is used to select the best features. The training is performed, and the model is stored. The accuracy, precision, F1 score, and recall is calculated. The Ensemble model with Neural Network is compared with Ensemble model with Random Forest. The Neural Networks used for the comparison are CNN, RNN, LSTM. The Ensemble model with Random Forest when compared to Ensemble model with Neural Networks performs better with an accuracy and precision score of 99.96% and 99.77 respectively. However, recall and F1 score are better with CNN model. But Ensemble model with RF performs better overall since it takes significantly less time for training and testing when compared to other algorithms taking 15.51 and 0.0039 minutes respectively.

Yifan Zhang, Fenghua Li and Siyuan Leng [5] proposes a distributed and collaborative framework, DICOF for the detection of multiple types of DDoS attacks. This method addresses the detection of hybrid DDoS attacks which when compared to other models have only focused on detecting only one type of DDoS attack. The proposed model will detect and classify the DDoS attacks simultaneously. First, an entropy-based method, LSTM will identify the attack and then a Gated Recurrent Unit (GRU) based classification technique is used to identify the different types of DDoS attack. The framework consists of four different components namely detector, classifier, responder, and collector divided among three layers being. The collector work in Layer 1, detector and classifier in Layer 2 and the responder in Layer 3 but the focus is on Layer 2. The CIC-DDoS2019 dataset in which 9 reflective attacks MSSQL, SSDP, NTP, TFTP, UDP, DNS, LDAP, NETBIOS, SNMP and 2 types of direct attacks SYN, UDP-Lag are present. In this study, the accuracy, average precision, average recall and average F1 score are calculated. When an attack is detected, DICOF generates alarms between 0.3 and 0.6 seconds and notifies the attack.

Wangshu Guo, Ming Xian and Yejin Tan [6] suggest that the detection performance reduces because of the small sample size hence they propose a novel approach for detecting small-sample DDoS attacks by utilizing deep transfer learning. Initially, deep learning techniques are applied to train multiple neural networks capable of effectively transferring knowledge in the context of DDoS attacks, given a sufficient number of samples. Subsequently, a transferability metric is devised to evaluate and compare the transfer performance of these networks. By utilizing this metric, the network exhibiting the most favorable transfer performance can be identified from the four networks under consideration. Furthermore, the paper demonstrates that the utilization of deep learning techniques leads to a decline in performance when dealing with a limited number of DDoS attack samples, with the detection performance decreasing from 99.28 to 67. However, through the deep transfer of the 8LANN network in the target domain, a notable improvement of 20.8% in the detection performance is achieved. The experimental results validate the performance of the proposed deep transfer learning-based detection method in mitigating the performance deterioration associated with deep learning techniques when applied to small-sample DDoS attack detection.

An automated DDoS detection model was developed for the purpose of effectively identifying and classifying DDoS attacks in study [7]. This detector utilizes machine learning models and can be implemented on standard hardware. The classification accuracy results obtained from this detector is reported to be 98.5%. To classify DDoS packets from normal packets, three classification algorithms, namely KNN, RF, and NB, are employed. These algorithms utilize two key features, namely delta time and packet size. The detector can detect a multitude of DDoS attacks, including ICMP flood, TCP flood, and UDP flood. In contrast to older systems that can only detect certain classes of DDoS attacks and may require a larger number of features, the suggested model overcomes these limitations. Wireshark tool is utilized to capture the information about the packets such as the width of the packet, destination and source, timestamp etc. It can detect any type of DDoS attack irrespective of a specific protocol and utilizes a reduced number of features.

3. PROPOSED MODEL

3.1 Dataset

The dataset used is a subset of the combination of CSE-CIC-IDS 2018, CICIDS 2017 and CIC DoS 2017 datasets. The dataset includes both DDoS attack traffic and Benign traffic in the ratio of 1:5 respectively. The total number of records in the dataset is 500k out of which 100k are DDoS in nature. The dataset is split into training and testing set in the ratio 7:3. Chart - 1 depicts the count of the classes in the dataset.

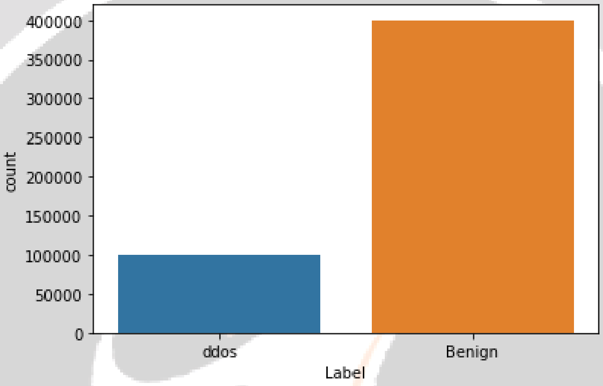
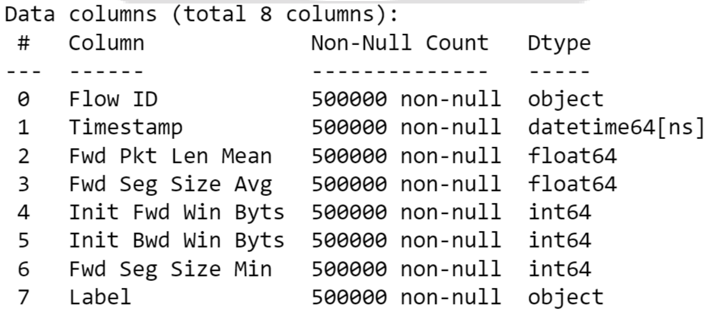


Chart - 1: Class Imbalance of Dataset

3.2 Feature Selection and Engineering

There are about 80 features in the CICDoS datasets out of which 8 features have been extracted for our purpose. The 'Flow ID' feature allows us to extract the source and destination IP Addresses and further divide these addresses. Feature Engineering is performed for IP addresses of the data flow by converting them into machine understandable format by splitting the addresses into four sub addresses, separated at the dot ".". This enables the Machine Learning models to process the IP addresses. A single IP address, for example, 192.168.4.4 is divided as 192, 168, 4, 4 and put into four separate columns for the Machine Learning models to fetch for training, testing and validation. The selected features are listed below in Fig - 2.



```
Data columns (total 8 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   Flow ID            500000 non-null  object
 1   Timestamp          500000 non-null  datetime64[ns]
 2   Fwd Pkt Len Mean   500000 non-null  float64
 3   Fwd Seg Size Avg   500000 non-null  float64
 4   Init Fwd Win Byts  500000 non-null  int64
 5   Init Bwd Win Byts  500000 non-null  int64
 6   Fwd Seg Size Min   500000 non-null  int64
 7   Label              500000 non-null  object
```

Fig -2: Dataset Features

Label encoding of the target variable is performed, 1 being DDoS and 0 being Benign. Class imbalance occurs due to the huge difference between the count of the target values, and this is corrected by random under sampling method. The essential features for our study are described in the figure, the Flow ID feature consists of the source IP address, destination IP address, destination port address and the relationship between the three is as follows:

The characteristics of DDoS attacks are described using information entropy. As per the principles of information theory, the information entropy of variable *X* is determined by the prior probability of variable *X*, denoted as $p\ (x_i)$.

$$H(X) = -\sum_i p(x_i)\, log_2\left(p(x_i)\right) \tag{1}$$

The entropy of *X* about *Z* can be given as (2), $p\ (x_i|z_i)$ would be the posterior probability of x_i with respect to z_j.

$$H(X|Z) = -\sum_i p(z_j) \sum_i p(x_i|z_i)\, log_2\ \left(p(x_i|z_j)\right) \tag{2}$$

Let the source IP address, destination IP address and the destination port of the traffic data be represented by the variables SIP, DIP and DP respectively. The traffic flow information can be generally characterized by the three characteristics, with relationships between Source IP to Destination IP, Destination IP to Destination Port and Destination Port to Destination Port. The Data Flow Information Entropy (DFIE) is used to represent the uncertainty of the above relationships.

DFIE is calculated with SIP as an example assuming that the total number of data flow collected at time is *T*, the Source IP addresses collected in time *S* is $\{s_{ii} \mid i = 1, 2, 3, \ldots N\}$ and the Destination IP address set is $\{d_{ii} \mid i = 1, 2, 3, \ldots M\}$. The number of data streams with destination address d_{ii} and the source address s_{ii} is *P[M]: A[i]* and *Q[N][M]: Q[i][j]* respectively. Using the equation (2), we arrive at (3) as follows:

$$\begin{aligned} DFIE &= -\ \sum_j p(d_{ij}) \sum_i p(s_{ii}|d_{ij})\, log_2(p(s_{ii}|d_{ij})) \\ &= -\ \sum_{j=1}^{M} \frac{P[j]}{S} \sum_{i=1}^{N} \frac{Q[i][j]}{P[j]}\, log_2\left(\frac{Q[i][j]}{P[j]}\right) \end{aligned} \tag{3}$$

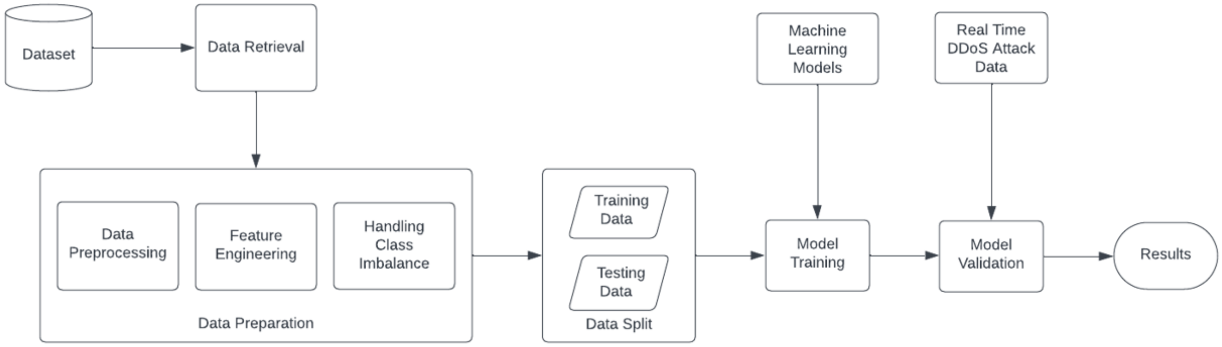


Fig - 3: System Architecture of the Proposed Model

3.3 Machine Learning Techniques

Multiple machine learning techniques have been implemented in existing studies which have achieved good performance, but the goal of our study is to improve performance of the widely used models. Fig – 3 represents the architecture of the proposed model. The algorithms used for this study include Adaboost, Logistic regression, Random Forest and K-Nearest Neighbor. The hyper parameters used for the respective algorithms are depicted in Table 1.

Table - 1: Hyper parameters used for ML Model Training

ML Algorithm	Hyper parameters
Random Forest	max_depth = 7, n_estimators = 30, max_leaf_nodes = 30
AdaBoost	n_estimators = 50, learning_rate = 1, random_state=0
K-Nearest Neighbor	n_neighbors = 2, leaf_size = 20, weights = 'uniform', algorithm = 'auto'
Logistic Regression	solver = 'liblinear', penalty = 'l1', max_iter = 500

3.4 Model Validation Technique

To achieve impartial assessment, model validation is frequently performed on datasets that were not employed during the model's training phase. Model validation allows for the refinement of model parameters and the attainment of improved outcomes. The validation data is gathered using Wireshark, which captures a series of packets (Pcap) that can be transformed into features for Machine Learning models. Wireshark can activate promiscuous mode on network interface controllers (if supported), allowing users to monitor all traffic on the interface, including unicast traffic not meant for that controller's MAC address. By capturing packets on a remote machine and sending them to a Wireshark-equipped machine via the TZSP protocol or OmniPeek's protocol, Wireshark can analyze the packets in real-time as they are intercepted.

The Pcap files extracted from Wireshark are then utilized for validation purposes. Two MS Office Excel files are employed for model validation - one as the input provider and the other for generating the output. The validation process entails feeding a set of records, which are network traffic packets captured by Wireshark, into the input file. Our model then determines if each record is classified as "Benign" or "DDoS" and outputs the corresponding result in the output file. By analyzing the output, we can accurately identify if a particular network traffic packet or a sequence of captured packets display characteristics of a DDoS attack, thus confirming the presence of such an attack.

4. RESULTS AND ANALYSIS

Four machine learning models, namely AdaBoost, Logistic Regression, K-Nearest Neighbor, and Random Forest, were employed in the research. The main goal is to determine the best-performing model to serve as the base model. The selection of the base model included assessing each algorithm's performance based on two crucial factors: the training time and the classification accuracy achieved during testing. Different performance evaluation metrics were utilized to analyze the models. The classification accuracy can be calculated using the equation (4).

$$\text{Accuracy} = \frac{\textit{Number of correct predictions made}}{\textit{Total numbe rof predictions made}}$$

$$= \frac{TP+TN}{TP+TN+FP+FN} \qquad (4)$$

The classification accuracies of all the four models are as listed in Table - 2 and the bar plot in Chart - 2.

Table - 2: Classification Accuracy

ML Algorithm	Accuracies	Training Time
Random Forest	99.99%	6 minutes
AdaBoost	99.97%	39 minutes

K-Nearest Neighbor	99.92%	20 minutes
Logistic Regression	98.06%	7 minutes

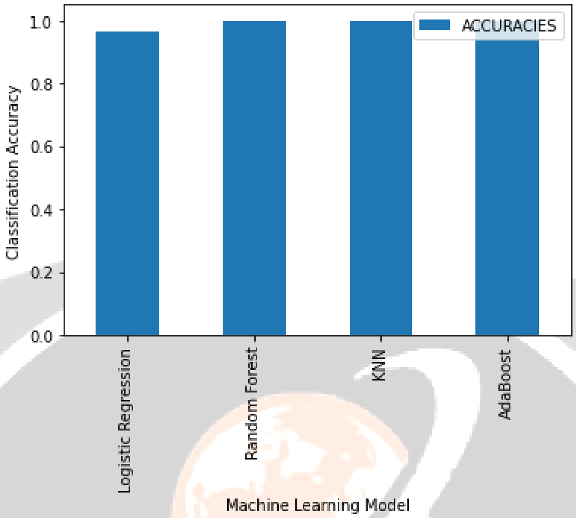


Chart – 2: Classification Accuracy of all four ML Models

Chart - 3 displays the time taken by each ML model for training with the dataset. Based on the Chart - 3 and Table - 3, the Logistic Regression model and Random Forest model have the shortest training times, at 7 minutes and 5 minutes, respectively. On the other hand, the AdaBoost model requires the longest training time among the models, taking 39 minutes to complete the training process.

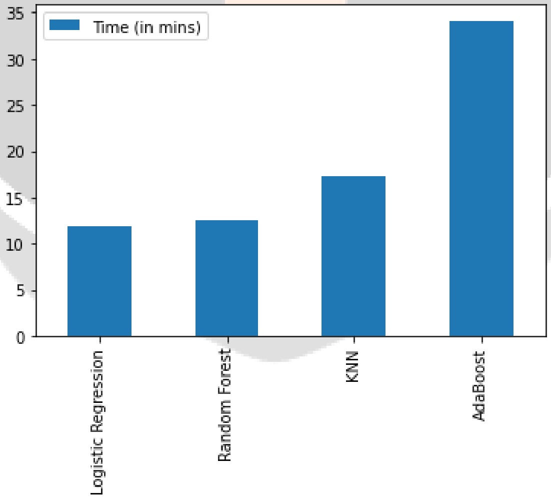


Chart – 3: Training Time of ML Models

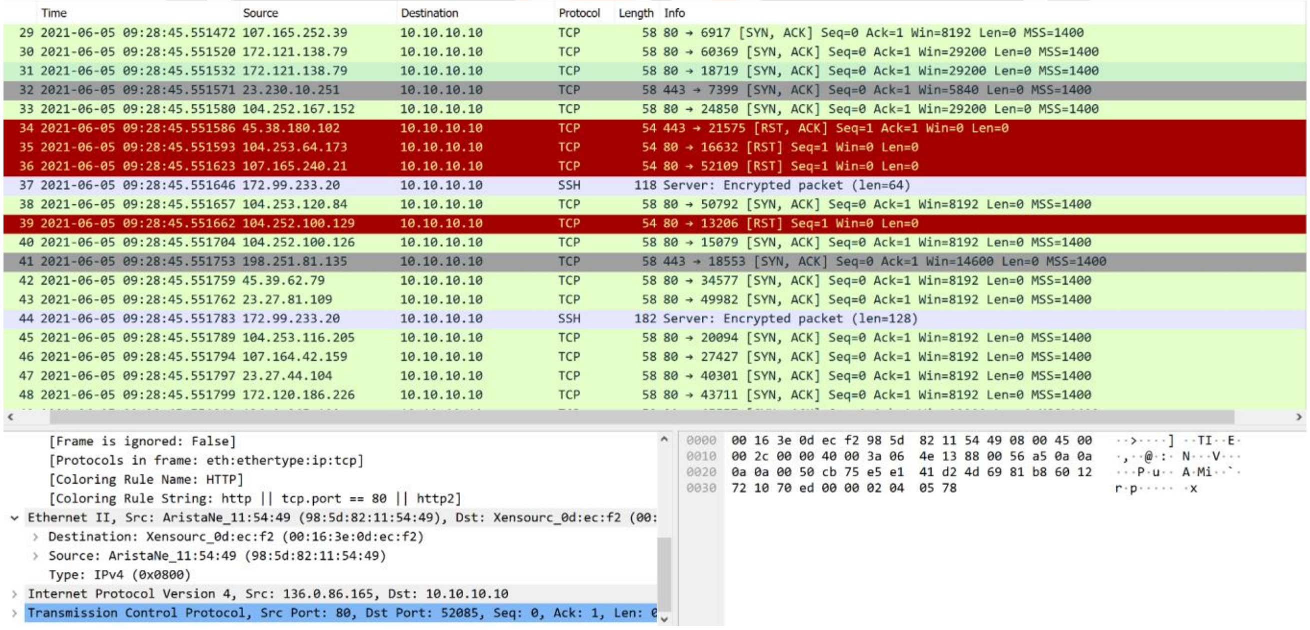
The training accuracies and time show that Random Forest and Logistic Regression models perform better compared to KNN and AdaBoost. Now, we plot the confusion matrices for all models to visualize the accuracy in binary classification. Table III represents an extraction of the confusion matrix, here, only the False Positives and False Negatives are considered since they define the total number of incorrect data misclassification. Logistic Regression has misclassified a huge chunk of records; hence it cannot be our base model. Random Forest on the other hand, has the least class misclassifications so this model is the top contender for the base model.

Table - 3: Confusion Matrices of ML Models

ML Algorithm	No. of False Positives	No. of False Negatives
Random Forest	2	0
AdaBoost	14	4
K-Nearest Neighbor	26	20
Logistic Regression	exponential	exponential

5. VALIDATION ANALYSIS

The validation data is extracted from Wireshark and the below Fig - 4 is an instance of the captured packets that is utilized for validation. Each row represents a packet, and every packet is highlighted in various colors based upon the nature of activity. The packets highlighted in green represent normal traffic flow, and the detailed color codes are represented in Fig – 5.

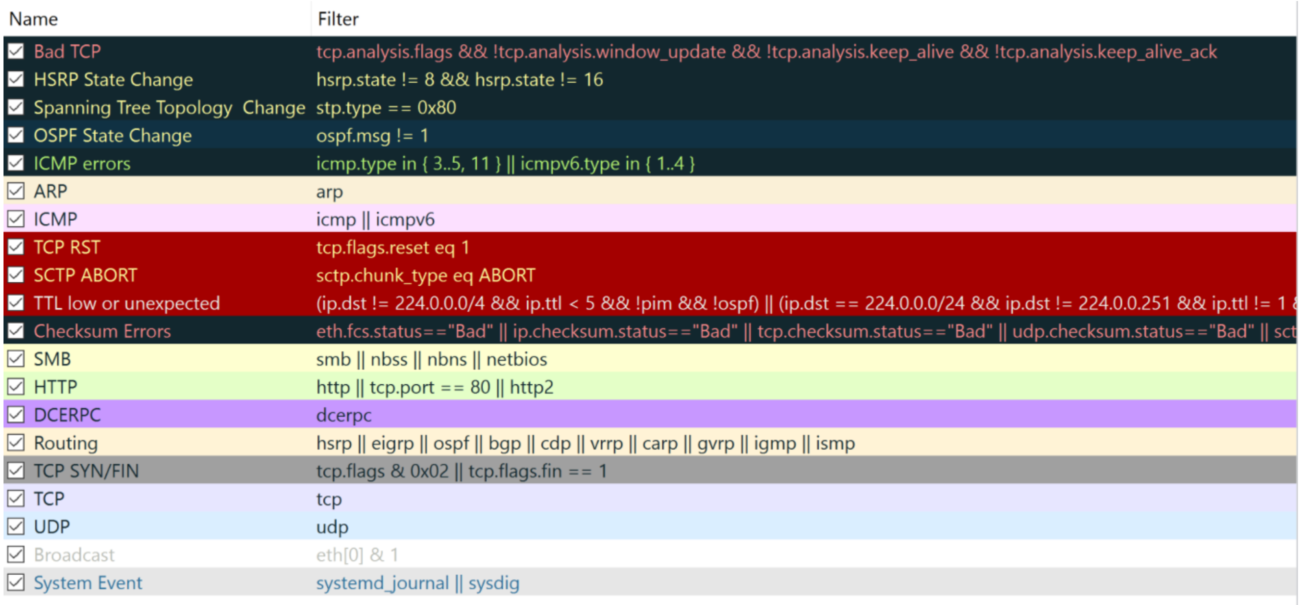


```
   Time                       Source           Destination    Protocol  Length  Info
29 2021-06-05 09:28:45.551472 107.165.252.39   10.10.10.10    TCP        58 80 → 6917 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1400
30 2021-06-05 09:28:45.551520 172.121.138.79   10.10.10.10    TCP        58 80 → 60369 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1400
31 2021-06-05 09:28:45.551532 172.121.138.79   10.10.10.10    TCP        58 80 → 18719 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1400
32 2021-06-05 09:28:45.551571 23.230.10.251    10.10.10.10    TCP        58 443 → 7399 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1400
33 2021-06-05 09:28:45.551580 104.252.167.152  10.10.10.10    TCP        58 80 → 24850 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1400
34 2021-06-05 09:28:45.551586 45.38.180.102    10.10.10.10    TCP        54 443 → 21575 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35 2021-06-05 09:28:45.551593 104.253.64.173   10.10.10.10    TCP        54 80 → 16632 [RST] Seq=1 Win=0 Len=0
36 2021-06-05 09:28:45.551623 107.165.240.21   10.10.10.10    TCP        54 80 → 52109 [RST] Seq=1 Win=0 Len=0
37 2021-06-05 09:28:45.551646 172.99.233.20    10.10.10.10    SSH       118 Server: Encrypted packet (len=64)
38 2021-06-05 09:28:45.551657 104.253.120.84   10.10.10.10    TCP        58 80 → 50792 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1400
39 2021-06-05 09:28:45.551662 104.252.100.129  10.10.10.10    TCP        54 80 → 13206 [RST] Seq=1 Win=0 Len=0
40 2021-06-05 09:28:45.551704 104.252.100.126  10.10.10.10    TCP        58 80 → 15079 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1400
41 2021-06-05 09:28:45.551753 198.251.81.135   10.10.10.10    TCP        58 443 → 18553 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1400
42 2021-06-05 09:28:45.551759 45.39.62.79      10.10.10.10    TCP        58 80 → 34577 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1400
43 2021-06-05 09:28:45.551762 23.27.81.109     10.10.10.10    TCP        58 80 → 49982 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1400
44 2021-06-05 09:28:45.551783 172.99.233.20    10.10.10.10    SSH       182 Server: Encrypted packet (len=128)
45 2021-06-05 09:28:45.551789 104.253.116.205  10.10.10.10    TCP        58 80 → 20094 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1400
46 2021-06-05 09:28:45.551794 107.164.42.159   10.10.10.10    TCP        58 80 → 27427 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1400
47 2021-06-05 09:28:45.551797 23.27.44.104     10.10.10.10    TCP        58 80 → 40301 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1400
48 2021-06-05 09:28:45.551799 172.120.186.226  10.10.10.10    TCP        58 80 → 43711 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1400
```

```
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
⌄ Ethernet II, Src: AristaNe_11:54:49 (98:5d:82:11:54:49), Dst: Xensourc_0d:ec:f2 (00:
  › Destination: Xensourc_0d:ec:f2 (00:16:3e:0d:ec:f2)
  › Source: AristaNe_11:54:49 (98:5d:82:11:54:49)
    Type: IPv4 (0x0800)
› Internet Protocol Version 4, Src: 136.0.86.165, Dst: 10.10.10.10
› Transmission Control Protocol, Src Port: 80, Dst Port: 52085, Seq: 0, Ack: 1, Len: 0
```

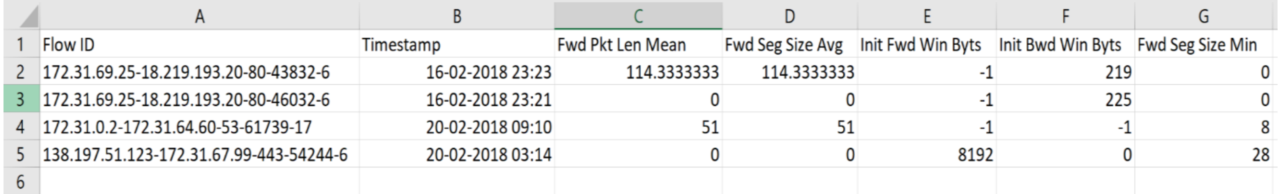
```
0000  00 16 3e 0d ec f2 98 5d  82 11 54 49 08 00 45 00   ··>····] ··TI··E·
0010  00 2c 00 00 40 00 3a 06  4e 13 88 00 56 a5 0a 0a   ·,··@·:· N···V···
0020  0a 0a 00 50 cb 75 e5 e1  41 d2 4d 69 81 b8 60 12   ···P·u·· A·Mi··`·
0030  72 10 70 ed 00 00 02 04  05 78                     r·p····· ·x
```

Fig – 4: Wireshark Packet Capture



Name	Filter
☑ Bad TCP	tcp.analysis.flags && !tcp.analysis.window_update && !tcp.analysis.keep_alive && !tcp.analysis.keep_alive_ack
☑ HSRP State Change	hsrp.state != 8 && hsrp.state != 16
☑ Spanning Tree Topology Change	stp.type == 0x80
☑ OSPF State Change	ospf.msg != 1
☑ ICMP errors	icmp.type in { 3..5, 11 } || icmpv6.type in { 1..4 }
☑ ARP	arp
☑ ICMP	icmp || icmpv6
☑ TCP RST	tcp.flags.reset eq 1
☑ SCTP ABORT	sctp.chunk_type eq ABORT
☑ TTL low or unexpected	(ip.dst != 224.0.0.0/4 && ip.ttl < 5 && !pim && !ospf) || (ip.dst == 224.0.0.0/24 && ip.dst != 224.0.0.251 && ip.ttl != 1 &
☑ Checksum Errors	eth.fcs.status=="Bad" || ip.checksum.status=="Bad" || tcp.checksum.status=="Bad" || udp.checksum.status=="Bad" || sct
☑ SMB	smb || nbss || nbns || netbios
☑ HTTP	http || tcp.port == 80 || http2
☑ DCERPC	dcerpc
☑ Routing	hsrp || eigrp || ospf || bgp || cdp || vrrp || carp || gvrp || igmp || ismp
☑ TCP SYN/FIN	tcp.flags & 0x02 || tcp.flags.fin == 1
☑ TCP	tcp
☑ UDP	udp
☑ Broadcast	eth[0] & 1
☑ System Event	systemd_journal || sysdig

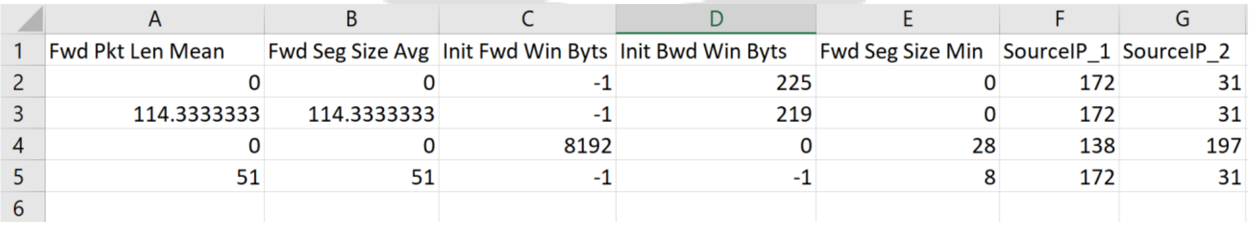
Fig - 5: Wireshark Packet Capture Color Codes



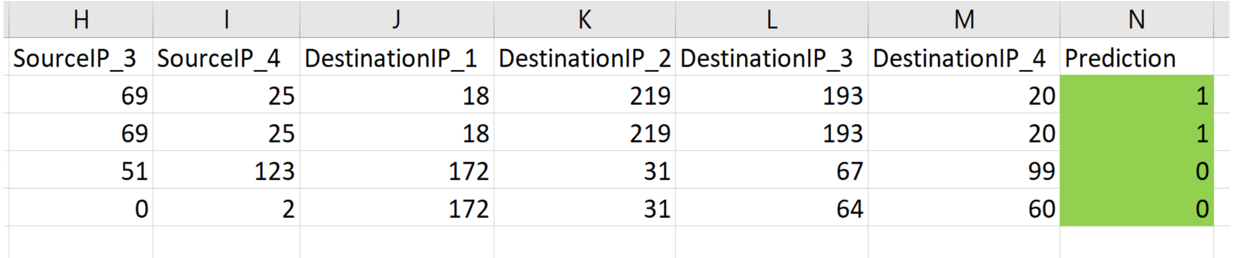
	A	B	C	D	E	F	G
1	Flow ID	Timestamp	Fwd Pkt Len Mean	Fwd Seg Size Avg	Init Fwd Win Byts	Init Bwd Win Byts	Fwd Seg Size Min
2	172.31.69.25-18.219.193.20-80-43832-6	16-02-2018 23:23	114.3333333	114.3333333	-1	219	0
3	172.31.69.25-18.219.193.20-80-46032-6	16-02-2018 23:21	0	0	-1	225	0
4	172.31.0.2-172.31.64.60-53-61739-17	20-02-2018 09:10	51	51	-1	-1	8
5	138.197.51.123-172.31.67.99-443-54244-6	20-02-2018 03:14	0	0	8192	0	28
6							

Fig - 6: Validation Input Dataset

The input dataset for the purpose of validation can be seen in Fig - 6, after the processing of data and model training and testing with the help of Random Forest, the output for the same can be seen in column N, labeled ‘Prediction’ in Fig - 7. This binary classification can predict two values, being ‘1’ and ‘0’. If the prediction is ‘0’, then the packet is Benign in nature, and it is part of the normal traffic flow. However, if the prediction is ‘1’, then the packet or the traffic record is DDoS in nature and further investigation of the traffic can be performed in Wireshark. When there’s a DDoS prediction, it is best advised to activate countermeasures to ensure that there is minimal loss of functionality.



	A	B	C	D	E	F	G
1	Fwd Pkt Len Mean	Fwd Seg Size Avg	Init Fwd Win Byts	Init Bwd Win Byts	Fwd Seg Size Min	SourceIP_1	SourceIP_2
2	0	0	-1	225	0	172	31
3	114.3333333	114.3333333	-1	219	0	172	31
4	0	0	8192	0	28	138	197
5	51	51	-1	-1	8	172	31
6							



H	I	J	K	L	M	N
SourceIP_3	SourceIP_4	DestinationIP_1	DestinationIP_2	DestinationIP_3	DestinationIP_4	Prediction
69	25	18	219	193	20	1
69	25	18	219	193	20	1
51	123	172	31	67	99	0
0	2	172	31	64	60	0

Fig – 7: Validation Output – Prediction

6. CONCLUSION

The occurrence of denial-of-service attacks has been rising, necessitating the implementation of efficient detection and prevention measures. This study presents an architectural approach that evaluates the performance of existing models and selects the optimal base model. The comparison was conducted using AdaBoost, Logistic Regression, K-Nearest Neighbor, and Random Forest models. The results revealed that the Random Forest Classifier outperformed the others in terms of accuracy, data class classification, and training time. The Random Forest classifier achieves an accuracy of 99.99% and takes 6 minutes for training with the CICDoS dataset consisting of 5,00,000 records. The confusion matrices also shows that Random Forest performs better comparatively and hence is considered as the base model for further validation. Real time traffic is extracted from Wireshark for validation and the Random Forest model provides accurate predictions, ‘0’ is predicted for Benign traffic whereas ‘1’ is predicted for DDoS traffic.

7. REFERENCES

[1] Yini Chen, Jun Hou, Qianmu Li and Huaqiu Long. “DDoS Attack Detection Based on Random Forest”. IEEE International Conference on progress in Information and Computing (PIC), 2020.

[2] Yanchao Sun, Yuanfeng Han, Yue Zhang, Mingsong Chen, Shui Yu and Yimin Xu. “DDoS Attack Detection Combining Time Series-based Multi-dimensional Sketch and Machine Learning”. The 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS) 2022.

[3] Heena Kousar, Mohammed Moin Mulla, Pooja Shettar, Narayan D. G. “DDoS Attack Detection System using Apache Spark”. 2021 International Conference on Computer Communication and Informatics (ICCCI -2021), Jan. 27 – 29, 2021.

[4] Tanut Visetbunditkum, Warakorn Srichavengsup. “DDoS Attack Detection Using Ensemble Machine Learning Models with RFE Algorithm”. 7th International Conference on Business and Industrial Research (ICBIR2022), 2022.

[5] Siyuan Leng, Yingke Xie, Yifan Zhang, Yunchuan Guo, Liang Fang, Fenghua Li. “DICOF: A Distributed and Collaborative Framework for Hybrid DDoS Attack Detection”. IEEE Symposium on Computers and Communications (ISCC), 2022.

[6] Wangshu Guo, Ming Xian and Yejin Tan, Jiawei He. “A Small Sample DDoS Attack Detection Method Based on Deep Transfer Learning”. International Conference on computer Communication and Network Security (CCNS), 2020.

[7] S.Shanmuga Priya, M.Sivaram, D.Yuvaraj, A. Jayanthiladevi. "Machine Learning based DDOS Detection". International Conference on Emerging Smart Computing and Informatics (ESCI) AISSMS Institute of Information Technology, Pune, India. Mar 12-14, 2020

[8] Xin Cheng, Zhiliang Wang, Shize Zhang, Jia Li, Jiahai Yang, and Xinran Liu. "Slider: Towards Precise, Robust and Updatable Sketch-based DDoS Flooding Attack Detection". IEEE Global Communications Conference, 2021.

[9] D. Li and Q. Li "Adversarial Deep Ensemble: Evasion Attacks and Defenses for Malware detection". IEEE Transaction on Information and Security, 2020.

[10] X. Liu, J. Ren, H. He, Q. Wang, and Song, "Low rate ddos attacks detection method using data compression behavior divergence measurement" Computers & Security, vol. 100, p. 102107, 2021.

[11] A. Mubarakali, K. Srinivasan, R. Mukhalid, S. C. Jaganathan, and N. Marina, "Security challenges in internet of things: Distributed denial of service attack detection using support vector machine-based expert systems" Computational Intelligence, vol. 36, no. 4, pp. 1580-1592, 2020.

[12] S. Lakshminarasimman, S. Ruswin, and K. Sundarakanthan, "Detecting ddos attacks using decision tree algorithm" in proc. Og Interbatinal Conference of Signal Processing, Communication and Networking (ICSCN), 2017, pp. 1-6.

[13] X. Xie, J. Li, X. Hu, H. Jin, H. Chen, X. Ma, and H. Huang, "High performance ddos attack detection system based on distribution statistics," in proc. of IFIP International Conference on Network and Parallel Computing (NPC), 2019, pp. 132–142.

[14] A. V. Kachavimath, S. V. Nazare, and S. S. Akki, "Distributed denial of service attack detection using na¨ıve bayes and k-nearest neighbor for network forensics," in proc. of International conference on innovative mechanisms for industry applications (ICIMIA), 2020, pp. 711–717.

[15] P. S. Saini, S. Behal, and S. Bhatia, "Detection of ddos attacks using machine learning algorithms," in proc. of International Conference on Computing for Sustainable Global Development (INDIACom), 2020, pp. 16–21.