NFSU JOURNAL OF CYBER SECURITY & DIGITAL FORENSICS

NFSU – Journal of Cyber Security and Digital Forensics
Volume – 1, Issue – Special, June 2024
E – ISSN – 2583-7559

An Explainable Ensemble Learning Model for Detection of Low-rate and High-rate DDoS Network Traffic

M. Raghupathi
Research Scholar, Department of CSE, Jawaharlal Nehru Technological University, Hyderabad
Department of Information Technology, VNR VJIET, Hyderabad

N. Sai Pranay Teja
Research Scholar, Department of CSE, Jawaharlal Nehru Technological University, Hyderabad
Department of Information Technology, VNR VJIET, Hyderabad

V. Radha krishna
Research Scholar, Department of CSE, Jawaharlal Nehru Technological University, Hyderabad
Department of Information Technology, VNR VJIET, Hyderabad

P. Rukmini
Research Scholar, Department of CSE, Jawaharlal Nehru Technological University, Hyderabad
Department of Information Technology, VNR VJIET, Hyderabad

M. Sai Manish
Research Scholar, Department of CSE, Jawaharlal Nehru Technological University, Hyderabad
Department of Information Technology, VNR VJIET, Hyderabad

A Akanksha
Research Scholar, Department of CSE, Jawaharlal Nehru Technological University, Hyderabad
Department of Information Technology, VNR VJIET, Hyderabad

***Abstract*— With the increasing prevalence of network attacks specifically low-rate and high-rate Distributed Denial of Service (DDoS) attacks, there coins an emerging and immediate need to arrive with computationally efficient but simpler methods and those that can yield better classifier evaluation parameters. Although existing deep learning techniques have shown promising results in identifying such attacks, but the lack of interpretability and explainability in such models is the major limitation and this consequently hinders their widespread adoption in security critical domains by industry. Also, the current need for industry is to have explainable model at hand to make better decision making. This leads to the design of explainable AI systems as addressed by MIT, USA in recent times. Researchers across the globe and the IT industry are now looking to come out with explainable AI models which form the basic motivation for the current work. The objective of this research is to propose an ensemble machine learning system to detect both low rate and high-rate network attacks with better detection rates when compared to state-of-the-art learning algorithms and studies that are employed for network intrusion detection.**

Keywords: *low-rate traffic, high-rate traffic, explainability, XAI, intrusion detection, anomaly detection, ensemble learning.*

I. INTRODUCTION

One of the main difficulties that arise when dealing with DDoS and DoS attacks [1, 11] is to analyze the situation to decide whether the server is under an attack, or the traffic is legitimate but in huge volume unintentionally. Attackers are always changing their techniques to avoid being detected. They might use strategies to imitate legitimate traffic or change the attack patterns. Attackers frequently hide their identities by using methods like IP spoofing. Because of this, it is difficult to identify the origin of attack traffic and distinguish between legitimate and malicious sources.

Defending from DDoS cannot be done by just detecting a certain IP address which seems suspicious, as botnets of several infected machines are used very frequently to perform attacks on the victim servers. Low-rate attacks produce less suspicious traffic, which makes them difficult to identify. On the other side, high-rate attacks overwhelm the victim with a tremendous amount of traffic, but their intensity may make them simpler to see. High-rate attacks are more likely to be discovered due to anomalous traffic patterns, whereas low-rate attacks may concentrate on avoiding detection by remaining below predetermined criteria. DDoS attackers can use a variety of strategies, such as application-layer attacks, protocol attacks and amplification attacks. It takes a broad range of detection techniques and knowledge of different network protocols to

identify these distinct attack methods. DDoS attacks can be classified as volumetric, protocol, or application-layer. Detection techniques vary depending on the type, making it difficult to provide a universally applicable response.

Many systems are developed to detect network attacks. Intrusion detection systems (IDS) play an important role in network security. An intrusion detection system includes a monitoring system designed to identify suspicious actions. IDS observes network and system activity for suspicious behavior. They are used in identifying network attacks. Attackers come up with new strategies to bypass the existing IDS. IDS generates security threat alerts in real time. This allows the user to reduce the effect of attack. The working of IDS includes analyzing network traffic, identifying security threats, and recording network activity. The IDS inspects the network packets whether they match with known attack patterns or signatures. Additionally, packet headers may be checked for anomalous properties. The IDS provides an alarm when it spots a potential intrusion or anomaly. These notifications may contain details about the threat's kind, its severity, the system, or network that was compromised, and its timestamp.

The objective of this research is to propose an ensemble machine learning system for detection of modern low rate and high-rate network attacks with improved accuracy and detection rates when compared to state-of-the-art machine learning algorithms that are employed for network intrusion detection system.

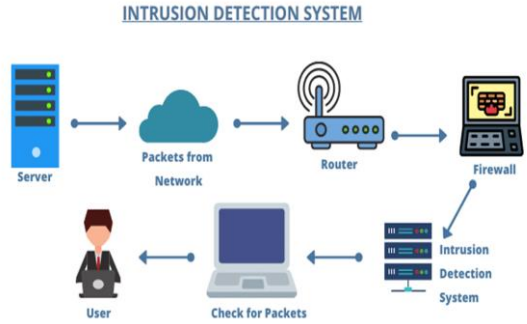

Fig 1. A diagram depicting an IDS.

A. *Research Issues and Challenges*

(i) The collection and usage of a complete and diverse dataset are essential for building an efficient ML-based DDoS attack detection system. To properly train the model to detect anomalies, the dataset used should include a variety of network traffic types, including both legitimate and attack data traffic.

(ii) It can be difficult to distinguish between legitimate traffic and Distributed Denial of Service (DDoS) attack traffic because DDoS attacks sometimes exhibit patterns that resemble normal network behavior.

(iii) The issue with detecting low-rate DDoS attacks is severe. In contrast to high-rate DDoS attacks, these attacks frequently produce traffic at a rate that doesn't stand out as attack traffic, making them more difficult to identify.

(iv) The model's capability to differentiate between high-rate and low-rate DDoS attacks is a crucial feature. This distinction is crucial because different prevention measures may be used and because low-rate attacks may be more undetectable and subtle.

(v) The goal is to detect DDoS attacks with the highest possible balanced accuracy, which requires an effective balance between sensitivity and specificity. When comparing the effectiveness of the ML-based IDS to industry-standard ML models, this must be done.

B. *Research Gap*

Existing intrusion detection systems for network attacks detection are not designed by considering network traffic dataset which satisfies standard properties of a quality dataset. The 11 properties of a quality dataset are: (i) Complete Network Configuration (ii) Complete Traffic (iii) Labelled dataset (iv) Complete interaction (v) Complete capture (vi) Available protocols (vii) Attack diversity (viii) Heterogeneity (ix) Anonymity (x) Feature Set (xi) Metadata. ML-based IDS are currently being built using benchmark network traffic datasets (KDD99, NSL-KDD, CAIDA 2007, DARPA), the majority of which have majority class as normal traffic and the minority class as attack traffic. This work uses the most recent CICDDOS2019 dataset [6] that satisfies all the above-mentioned criteria.

C. *Problem statement*

Formally, the problem is stated as follows: "***Given a network traffic flow of benign, high rate, and low-rate DDoS attacks, the problem is to design an explainable system and method that detects both high rate and low-rate DDoS attack traffic with the best detection rate (sensitivity), better specificity, and high balanced accuracy rate".***

II. DATASET DESCRIPTION

The dataset used to build the model was CICDDOS2019 dataset [6] which is a complete and diverse dataset for detecting DDoS attacks. It satisfies all the 11 properties of a quality dataset [5]. The most recent and common DDoS attacks are included in CICDDoS2019; these attacks resemble actual real-world data. It includes labeled flows according to time stamps, source and destination IP addresses, source and destination ports, protocols, and attacks; additionally, it includes the results of a CICFlowMeter-V3 network traffic analysis (CSV files). In this dataset, 12 DDoS attacks were carried out on the training day, consisting of NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, and TFTP; On the testing day, 7 attacks were conducted, comprising of PortScan,

NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, and SYN. The dataset included both low-rate and high-rate DDoS attacks. Depending on the amount or intensity of the attack traffic, DDoS attacks are commonly categorized as "high-rate DDoS attacks" or "low-rate DDoS attacks." A brief description of these attacks is mentioned below.

(i) UDP flood: User Datagram Protocol (UDP) floods can produce a lot of traffic by attacking the target with many UDP packets, overloading its infrastructure and bandwidth.

(ii) DNS Amplification attack: The attacker uses a DNS server that is weak to increase traffic in this kind of high-rate attack. They acquire large responses from the DNS servers by sending short DNS requests with a false source IP address, which causes a huge amount of traffic directed at the target.

(iii) NTP Amplification: By exploiting the weak NTP servers, attackers can amplify their requests, causing huge NTP responses to their target, resulting in a high traffic volume.

(iv) UDP-Lag: UDP-Lag attacks can be categorized in many ways. They may involve transmitting a lot of UDP traffic, although the rate will vary depending on the strategy for attack and the intended target.

(v) MSSQL Amplification: MSSQL amplification attacks use vulnerable instances of Microsoft SQL Server to amplify and generate a lot of traffic. MSSQL based attacks can be both high-rate and low-rate depending on method.

(vi) LDAP Amplification: To create DDoS traffic, LDAP amplification attacks leverage LDAP servers that are vulnerable to attack. Although it may change, the rate is often lower than some volumetric attacks.

(vii) NetBIOS: Attacks based on NetBIOS can use low-rate traffic and frequently target vulnerabilities in NetBIOS services.

(viii) PortMap: PortMap attacks are categorized as low-rate DDoS attacks. The attacks target services, such as the Portmapper service. To overwhelm the Portmapper service and obstruct its regular operation, PortMap attacks include sending a high number of malicious requests to it. As opposed to high-rate DDoS attacks, which overwhelm the target with a large amount of traffic, portmap attacks typically create traffic at a lower rate.

(ix) WebDDoS: Layer 7 application attacks (e.g., WebDDoS) and HTTP GET/POST floods are examples of attacks that target specific application layer resources, they create traffic at a lesser rate than volumetric attacks.

III. NETWORK TRAFFIC FLOW ACQUISITION FOR EXPERIMENT ANALYSIS

The process of capturing and collecting data from the traffic that flows in a computer network is known as network traffic flow acquisition. For monitoring, and security applications, this data is crucial. To collect this traffic data, network managers and security experts frequently use packet capture techniques. Data from network packet captures is frequently stored in the PCAP file format. These files include a history of each network packet that has travelled through a network. The source and destination addresses, timestamps, protocol specifics, and the actual packet content are all included in each packet capture. PCAP packet capture files are obtained from the CIC-DDoS2019 dataset where diverse DDoS attack traffic was generated. We have used CICFlowmeter to extract features from traffic data.

TABLE I. Attack-Benign distribution counts of original train dataset.

Training data			
Name	**DDoS Attack samples**	**Benign samples**	**Total samples**
DNS	5071011	3402	5074413
LDAP	2179930	1612	2181542
MSSQL	4522492	2006	4524498
NETBIOS	4093279	1707	4094986
NTP	1202642	14365	1217007
SNMP	5159870	1507	5161377
SSDP	2610611	763	2611374
UDP	3134645	2157	3136802
SYN	1582289	392	1582681
TFTP	20082580	25247	20107827
UDPLAG	366461	3705	370166
WEBDDOS	439	0	439
Total	50006249	56863	50063112

TABLE II. Attack-Benign distribution counts of original test dataset.

Testing data				
Name	**Attacks**	**Number of samples**	**Benign traffic samples**	**Total**
LDAP	NETBIOS	202919	5124	2113234
	LDAP	1905191		
MSSQL	LDAP	9931	2794	5775786
	MSSQL	5763061		
NETBIOS	NETBIOS	3454578	1321	3455899
PORTMAP	PORTMAP	186960	4734	191694
SYN	SYN	4284751	35790	4320541
UDP	UDP	3754680	3134	3782206
	MSSQL	24392		
UDPLAG	UDPLAG	1873	4068	725165
	UDP	112475		
	SYN	606749		
	total	20307560	56965	20364525

CICFlowmeter-3.0: CICFlowMeter is a network traffic flow analysis tool designed for cybersecurity and network monitoring. It can extract useful data from network flows and transform PCAP files (Packet Capture files) into a more structured and understandable format, like CSV.

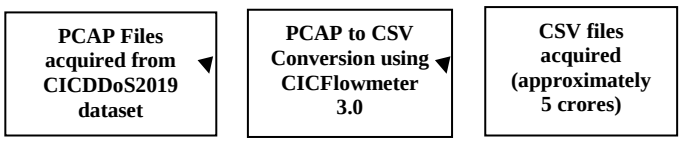

Fig 2. Network Traffic flow acquisition.

Conversion of PCAP to CSV: PCAP files keep track of specific network packets. To create network flow records, CICFlowmeter reads PCAP files and processes them. The tool retrieves relevant information from each packet, including timestamps, protocol, protocol ports, source, and destination IP addresses, and more.

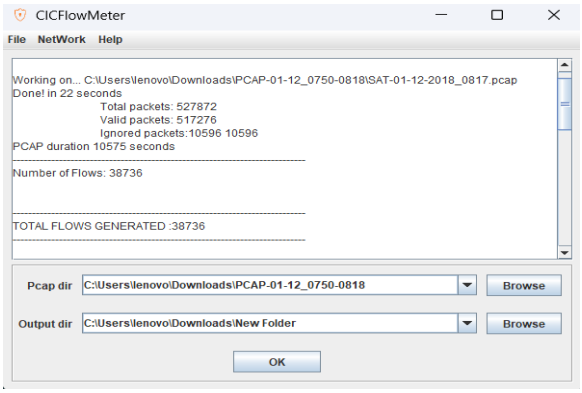

Fig 3. PCAP to CSV Conversion

After being extracted, the data is organized and saved in the CSV format, which is a simpler and more structured format to store network flow data. The acquired dataset contains 5 crore records with 84 features.

A. Network Traffic sampling.

The sampled data is subsequently used for a variety of things such as creating models, doing analysis, or coming to conclusions [7, 8]. When working with big datasets that would be expensive or time consuming to process entirely, data sampling can be especially helpful.

Train Dataset: We have generated a 10 Lakh dataset for training, which includes all the 12 types of DDoS attacks and benign traffic. The total number of benign records in the original csv are 56863. So, we include all the records in the training sample. To take the remaining 943137 records with equal distribution of all types of attacks we need to take 78594 records, but we have only 439 records in WebDDoS. So, we took all of them. As per our calculations, we need to take 85699 records from remaining attacks to get the dataset with equal attack distribution. Therefore, we included 85699*10 from all the attacks except TFTP and took (85699+9) records from TFTP as it has the highest attack records. In this way, we generated 10L dataset for training. All the data records were collected in a sequential manner from the start in the order of timestamp by doing so we implicitly take time as a feature. Training dataset information is depicted in table III.

Test Dataset: We have generated four datasets for testing of size 40K. To generate these test datasets with 40K records, we have included 20k benign records from the total number of benign records that are 56965. For the remaining 20k records for attack, we have taken 1873 records in UDP-lag, 3022 records from mssql and 3021* 5 from remaining attacks to get 20K. For generating these test datasets, we have followed two strategies, sequential sampling, and random Sampling. Table IV shows the testing dataset information.

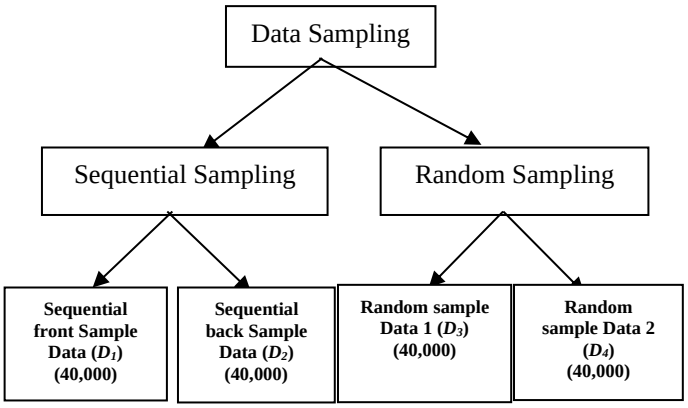

Fig 4. Diagram depicting data sampling.

TABLE III. Attack-Benign distribution counts of sample training dataset.

Training dataset used for experimentation	
Name	**Count**
DNS	85699
LDAP	85699
MSSQL	85699
NETBIOS	85699
NTP	85699
SNMP	85699
SSDP	85699
UDP	85699

SYN	85699
TFTP	85708
UDPLAG	85699
WEBDDOS	439
BENIGN	56863
Total	1000000

Sequential Sampling: In sequential sampling, the data points are chosen sequentially in the order of timestamp, by doing so we implicitly take time as a feature. Two sequential samples D_1 and D_2 each for testing are chosen with 40K size respectively as shown in Fig.4.

Random Sampling: Random sampling is a typical strategy in which data points are chosen at random from a dataset. The sample is more unbiased and representative of the entire population through this technique. Two random samples D_3 and D_4 were generated in each of which training and testing datasets of sizes 10L and 40k respectively are taken as shown in Fig.4.

TABLE IV. Attack-Benign distribution counts of sample testing dataset.

Testing dataset used for experimentation	
Name	**Count**
NETBIOS	3021
LDAP	3021
UDP	3021
SYN	3021
MSSQL	3022
UDPLAG	1873
PORTMAP	3021
BENIGN	20000
Total	40000

B. Network Traffic Pre-processing

In data preprocessing, feature selection, data transformation and data cleaning are performed. Data cleaning focuses on identifying and addressing issues in the raw data. The goal is to ensure that the data is accurate, consistent, and reliable.

Feature Selection:

Six static features ('Timestamp', 'Source_Port', 'Destination_IP', 'Flow_ID', 'Source_IP', 'Destination_Port') were removed by definition as they do not contribute towards the information held by the dataset, then 12 single valued features were eliminated and after that 8 redundant features respectively were also removed by calculating the correlation between each feature.

Data Transformation:

On the resultant dataset, dummy encoding was performed on protocol feature. Protocol feature includes values '0', '6' and '17'. Protocol 0 corresponds to the "HOPOPT" or "IPv6 Hop-by-Hop Option" header. Protocol 6 corresponds to TCP (Transmission Control Protocol). Protocol 17 corresponds to UDP (User Datagram Protocol). Protocol feature is encoded in this study during pre-processing. All these pre-processing step by step procedure depicted in the Fig. 5.

Data Cleaning:

After inspecting samples for infinite values and NaN values we have found that there were two features which had infinite values, (i) Flow Bytes/s (15640 records) and (ii) Flow Packets/s (31546 records) and one feature containing Nan values (i) Flow Bytes/s (15906 records). The NaN values were replaced by 0 and inf were replaced by 99999999. This value is chosen by considering the maximum value which occurred in the columns that have 'inf'.

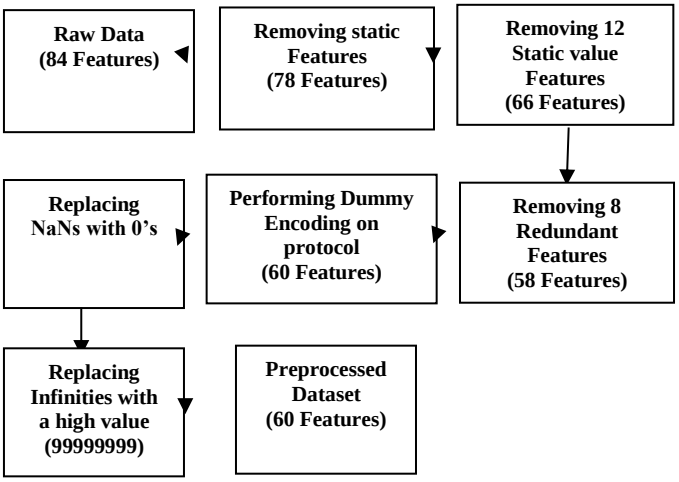

Fig 5. Network traffic data preprocessing.

The finalised feature list after removing above mentioned features is:

'Idle_Max', 'min_seg_size_forward', 'Max_Packet_Length', 'Backward_Packet_Length_Max', 'Forward_IAT_Min', 'Init_Win_bytes_forward', 'Backward_Packet_Length_Min', 'Flow_IAT_Std', 'Protocol_17', 'Backward_IAT_Total', 'Active_Min', 'Forward_IAT_Std', 'Packet_Length_Variance', 'Flow_IAT_Min', 'Packet_Length_Std', 'URG_Flag_Count', 'Backward_IAT_Std', 'Backward_IAT_Max', 'Active_Mean', 'Backward_IAT_Mean', 'Protocol_6', 'Total_Forward_Pckts, 'Total_Length_of_Backward_Pckts, 'Forward_Pckts /s', 'Protocol_0', 'act_data_pkt_Forward', 'Flow_Bytes/s', 'Forward_Packet_Length_Std', 'Init_Win_bytes_backward', 'Forward_IAT_Max', 'Idle_Mean', 'Backward_Header_Length', 'Forward_IAT_Total', 'Forward_Packet_Length_Min', 'RST_Flag_Count', 'Backward_IAT_Min', 'ACK_Flag_Count', 'Backward_Packet_Length_Std', 'Forward_Header_Length.1', 'Backward_Packet_Length_Mean', 'Backward_Pckts /s', 'Min_Packet_Length', 'Idle_Std', 'Active_Max', 'SYN_Flag_Count', 'Down/Up_Ratio', 'Idle_Min', 'Active_Std', 'Average_Packet_Size', 'Forward_Packet_Length_Mean', 'Total_Backward_Pckts, 'CWE_Flag_Count', 'Forward_Packet_Length_Max', 'Forward_IAT_Mean', 'Total_Length_of_Forward_Pckts, 'Flow_IAT_Max', 'Flow_IAT_Mean', 'Flow_Duration', 'Flow_Pckts /s', 'Packet_Length_Mean', 'Label'.

IV. PROPOSED ENSEMBLE MODEL

Fig.6 shows the system architecture depicting connection between different components. The User Interface component here consists of HTML/CSS as frontend and Django as Backend. The External tools used here is the CICFlowMeter software. The machine learning module component has the prediction ML model that is deployed in the system. The File system component contains all the .sav files and the .csv files that are used for the implementation for the system.

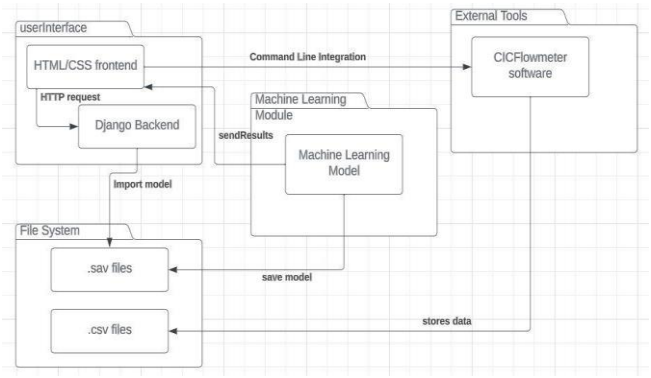

Fig 6. System Architecture of Proposed IDS

Fig.7 depicts the proposed ensemble model along with the sensitivity (SN), specificity (SP), accuracy (Acc) values obtained for test data which is unseen during training of the proposed XAI model which is an ensemble of Gaussian Naïve Bayes classifier, Linear discriminant Analysis Classifier and Multilayer Perceptron.

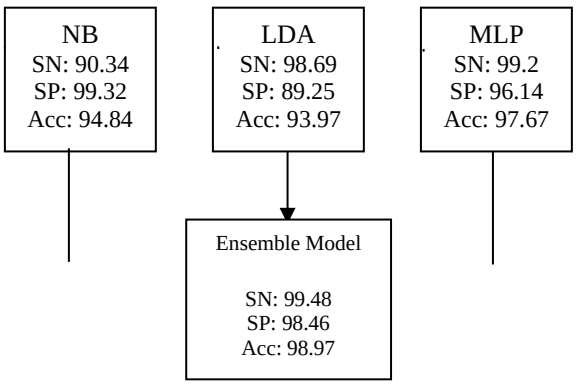

Fig 7. Proposed XAI Ensemble Model.

The main design of the model is based on ensemble technique. Here, we have chosen the models based on their performances according to various classification metrics, which are mentioned in the table V. To create an ensemble model, we have chosen are Gaussian Naïve Bayes, Linear Discriminant Analysis, and Multi-Layer Perceptron learning models. The Naïve Bayes classifier is observed to detect the Normal traffic with a high performance, LDA classifier observed to detect the Attack traffic with a high performance whereas multi-layer perceptron is a balanced model which predicts both attack and normal traffic with a good performance, this technique has resulted in an ensemble model whose performance is seen to be more robust, enhanced and balanced compared to all the base classifiers.

V. EXPERIMENTATION

Pre-processed data, which is discussed in Section III, is used for experimentation. The experimentation was done in two scenarios. The first scenario is baseline experimentation. In this experimentation, eight supervised machine learning classifiers were used on each of 15 features [7], 20 features [12], 25 features [4], and our proposed 60 feature dataset. All these experimental results are depicted in Table V. The second scenario is ensemble model experimentation, with 60 features, a 10L dataset used for training and a 40K dataset for testing, and all these results are depicted in Table XI.

VI. VISUALIZATION OF LOW RATE AND HIGH-RATE NETWORK ATTACKS AND RESULT ANALYSIS

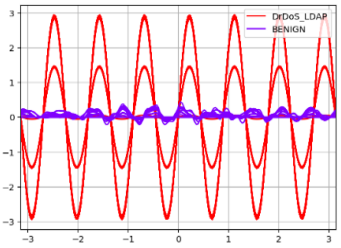

Fig 8(a) Visualization of LDAP high-rate attack

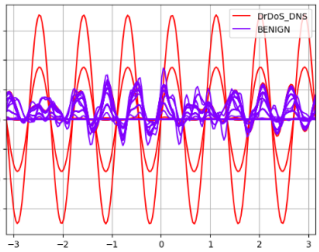

Fig 8(b) Visualization of DNS high-rate attack

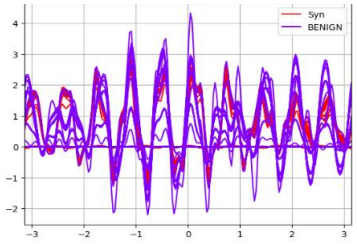

Fig 8(c) Visualization of SYN low-rate attack Fig 8(d) Visualization of WebDDoS low-rate attack

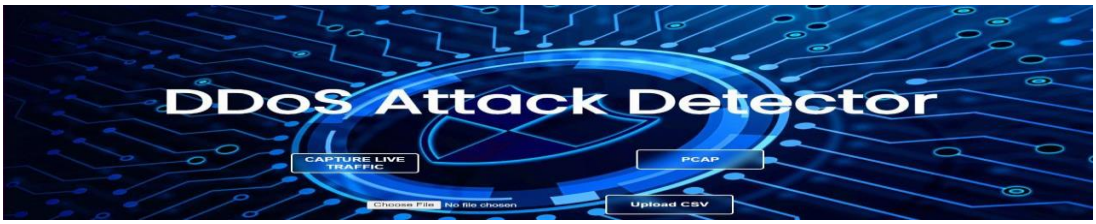

Fig 8(e) IDS system for attack detection

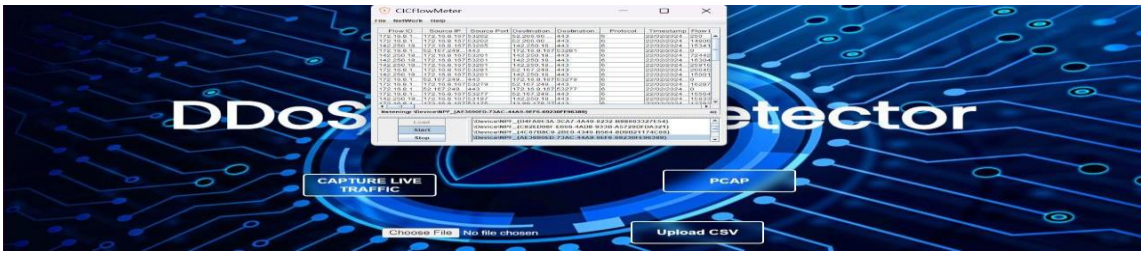

Fig 8(f) Acquiring real time data network traffic data

TABLE V. Accuracy of classifiers using different feature subsets of D_1 dataset

Machine Learning Classifiers	15 Features [7]		20 Features [12]		25 Features [4]		60 Features	
	Validation Accuracy (%)	Testing Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)
Logistic Regression	97.91	77.57	98.94	96.06	94.76	56.18	99.29	96.14
Decision Tree	99.69	78.05	99.99	42.14	99.6	86.41	99.99	88.66
Naive Bayes	96.34	78.61	89.05	89.67	92.63	56.88	97.36	94.83
Random Forest	99.66	86.81	99.98	97.19	99.34	85.66	99.99	97.29
LDA	96.54	78.39	95.59	40.48	94.68	56.15	99.18	93.97
MLP	99.19	54.57	99.64	**98.14**	98.26	82.41	99.79	97.67
QDA	96.8	40.38	96.01	44.73	94.53	58.63	98.85	50
XGB	99.67	**91.87**	99.98	97.86	99.62	**88.79**	99.99	**97.71**

TABLE VI. Experimental results of 15 features of CICDDoS2019 dataset

Machine Learning Classifiers	15 Features [7]				
	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)	F1-Score
Logistic Regression	93.51	61.63	70.91	77.57	0.8065
Decision Tree	70.38	85.73	83.14	78.06	0.7623
Naive Bayes	89.37	67.86	73.55	78.62	0.8069
Random Forest	81.26	92.37	91.41	86.81	0.8604

LDA	93.42	63.37	71.83	78.39	0.8121
MLP	18.77	90.39	66.13	54.58	0.2924
QDA	12.54	68.24	28.3	40.39	0.1738
XGB	89.5	94.26	93.97	91.88	0.9168

TABLE VII. Experimental results of 20 features of CICDDoS2019 dataset

Machine Learning Classifiers	20 Features [12]				
	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)	F1-Score
Logistic Regression	98.62	93.51	93.83	96.06	0.9616
Decision Tree	18.38	65.92	35.03	42.15	0.2411
Naive Bayes	87.42	91.94	91.56	89.68	0.8944
Random Forest	98.02	96.37	96.42	97.19	0.9722
LDA	17.22	63.76	32.2	40.49	0.2244
MLP	98.38	97.92	97.93	98.15	0.9815
QDA	15.16	74.32	37.12	44.74	0.2153
XGB	98.94	96.79	96.86	97.87	0.9789

TABLE VIII. Experimental results of 25 features of CICDDoS2019 dataset

Machine Learning Classifiers	25 Features [4]				
	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)	F1-Score
Logistic Regression	99.33	13.03	53.32	56.18	0.6939
Decision Tree	94.03	78.81	81.61	86.42	0.8738
Naive Bayes	97.05	16.72	53.82	56.88	0.6924
Random Forest	98.85	72.48	78.22	85.67	0.8733
LDA	99.3	13.01	53.3	56.15	0.6937
MLP	93.44	71.4	76.56	82.42	0.8416
QDA	94	23.28	55.06	58.64	0.6944
XGB	93.03	84.57	85.77	88.8	0.8925

TABLE IX. Experimental results of proposed 60 features of CICDDoS2019 dataset

Machine Learning Classifiers	60 Features				
	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)	F1-Score
Logistic Regression	98.68	93.6	93.91	96.14	0.9623
Decision Tree	85.86	91.48	90.97	88.67	0.8834
Naive Bayes	90.35	99.33	99.26	94.84	0.9459
Random Forest	96.7	97.88	97.85	97.29	0.9727
LDA	98.7	89.25	90.18	93.97	0.9424
MLP	99.2	96.15	96.26	97.67	0.9770
QDA	0.01	100	50	50	0.00009
XGB	99.81	95.63	95.81	97.72	0.9776

TABLE X. Ensemble model experimental results on 4 test datasets

	Ensemble Model performance			
	D_1 Test dataset	D_2 Test dataset	D_3 Test dataset	D_4 Test dataset
Sensitivity (%)	98.66	99.09	99.48	99.45
Specificity (%)	98.615	99.53	98.46	98.43
Precision (%)	98.615	98.53	98.48	98.44
Accuracy (%)	98.63	98.81	98.97	98.93
F1-Score	0.9863	0.9881	0.9897	0.9894

Low-rate attack and high-rate attack visualization are compared to Benign using Andrew's curve plot. Fig 8(a) and Fig 8(b) depict LDAP and DNS high-rate attacks plotted against benign traffic. Fig 8(c) and Fig 8(d) represent the WebDDoS and SYN low-rate attacks plotted against benign traffic. It is clear from 8(a) to 8(d) that high-rate attacks are less overlapping with benign traffic when compared to low-rate traffic. Hence, it becomes highly challenging to identify low-rate distributed network attacks when compared to high-rate attacks. Fig 8(e) and 8(f) shows the GUI of the Web application we have developed for detection of low rate and high-rate network attacks. Fig 9 below shows the performance of the proposed ensemble model for the test data consisting of 40000 network traffic instances. The proposed model has achieved 98.62% balanced accuracy for the test dataset which consists of unseen network traffic w.r.t train data used to train the model which is an appreciable detection rate when compared to results obtained in some of the recent contributions [2][3][4][5][6].

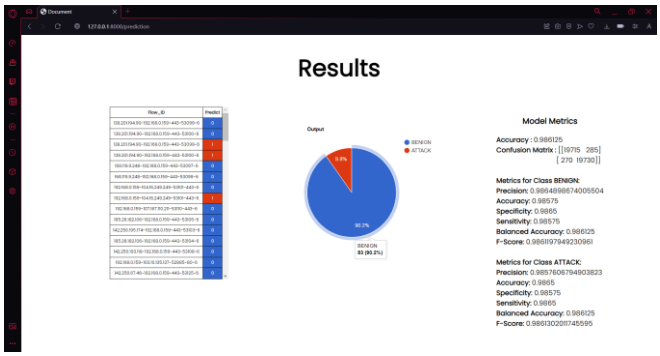

Fig 9. GUI of the IDS application depicting evaluation metrics

Compared to ResNetDDoS [7] model and XGB [4] model, our model has performed well. Whereas compared to Auto encoders [12] model, our proposed model has deteriorated 0.03%.

TABLE XI. Model performance compared with existing works

Existing Works	**Number of Features**	**Approached Model**	**Accuracy (%)**
F. Hussain [7]	15	ResNetDDoS	98.89
Salahuddin [12]	20	Auto encoders	99
J. Halledy [4]	25	XGB	98.58
Proposed Model	**60**	**Ensemble Model**	**98.97**

VII. CONCLUSION

Any IDS must satisfy two properties (i) Completeness and (ii) Correctness. Further, IDS model should be deterministic in nature to be considered as complete and correct. Existing deep learning-based models [9, 10] lack interpretability and explainability [11] which hinders industry adoption in health sector and security-critical related domains. Lack of interpretability and explainability thus forms the basis for design of explainable systems as addressed by MIT, USA in the recent times. Researchers across the globe and the IT industry are now looking to come out with explainable AI models which form the basic motivation for the present research work. In this work, we design, analyze, and build an ensemble IDS model for detection of low rate and high-rate network attacks. The aim of this work is to develop a simple, explainable, and interpretable machine learning system and method that can effectively detect low rate and high-rate DDoS attacks in the incoming network traffic with better computational performance. The dataset chosen for developing the model is provided by Canadian Institute of Cyber Security which was released in the year 2019 (CICDDoS-2019 Dataset). The dataset consists of various types of DDoS attacks which concentrate more on the attack volume rather than benign volume, which is lacking in previous datasets like KDD, CICDDoS-2017. We aim to develop an attack detection system rather than an anomaly detection system taking advantage of the CICDDoS-2019 dataset. The proposed IDS model achieved 98.97% detection rate for normal and attack traffic.

References

1. Hoque, N., Bhattacharyya, D. K., and Kalita, J. K. (2016) FFSc: a novel measure for low-rate and high-rate DDoS attack detection using multivariate data analysis. Security Comm. Networks, 9: 2032– 2041. doi: 10.1002/sec.1460.
2. Iman Sharafaldin, Arash Habibi Lashkari, Saqib Hakak, and Ali A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy", IEEE 53rd International Carnahan Conference on Security Technology, 2019.

3. F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad and G. A. Shah, "IoT DoS and DDoS Attack Detection using ResNet," 2020 IEEE 23rd International Multitopic Conference (INMIC), 2020, pp. 1-6, doi: 10.1109/INMIC50486.2020.9318216
4. J. Halladay et al., "Detection and Characterization of DDoS Attacks Using Time-Based Features," in IEEE Access, vol. 10, pp. 49794-49807, 2022, doi: 10.1109/ACCESS.2022.3173319.
5. Ali Shiravi, Hadi Shiravi, Mahbod Tavallaee, Ali A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection", Computers & Security, Volume 31, Issue 3, May 2012, Pages 357-374, ISSN 0167-4048, 10.1016/j.cose.2011.12.012.
6. I. Sharafaldin, A. H. Lashkari, S. Hakak and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," *2019 International Carnahan Conference on Security Technology (ICCST)*, Chennai, India, 2019, pp. 1-8, doi: 10.1109/CCST.2019.8888419.
7. F. Hussain et al., "A Two-Fold Machine Learning Approach to Prevent and Detect IoT Botnet Attacks," in IEEE Access, vol. 9, pp. 163412-163430, 2021, doi: 10.1109/ACCESS.2021.3131014.
8. Nimisha Pandey, Pramod Kumar Mishra, "Detection of DDoS attack in IoT traffic using ensemble machine learning techniques", Networks and Heterogeneous Media, vol.18, no.3, pp.1393, 2023.
9. M. Y. Aldarwbi, A. H. Lashkari, and A. A. Ghorbani. "The sound of intrusion: A novel network intrusion detection system," Computers and Electrical Engineering, vol. 104, part A, 108455, 2022.
10. F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad and G. A. Shah, "IoT DoS and DDoS Attack Detection using ResNet," 2020 IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan, 2020, pp. 1-6, doi: 10.1109/INMIC50486.2020.9318216.
11. S. Mahdavifar and A. A. Ghorbani, "CapsRule: Explainable Deep Learning for Classifying Network Attacks," in IEEE Transactions on Neural Networks and Learning Systems, Early Access, pp. 1-15, Apr. 2023.
12. M. A. Salahuddin, M. Faizul Bari, H. A. Alameddine, V. Pourahmadi and R. Boutaba, "Time-based Anomaly Detection using Autoencoder," 2020 16th International Conference on Network and Service Management (CNSM), Izmir, Turkey, 2020, pp. 1-9, doi: 10.23919/CNSM50824.2020.9269112.