NTU Journal of Engineering and Technology (2023) 2 (2) : 29-47
DOI: https://doi.org/10.56286/ntujet.v2i2.436

IRAQI Academic Scientific Journals

P-ISSN: 2788-9971 E-ISSN: 2788-998X
NTU Journal of Engineering and Technology
Available online at: https://journals.ntu.edu.iq/index.php/NTU-JET/index

RF-RFE-SMOTE: A DoS And DDoS Attack Detection Framework

1st Nora Rashid Najam[1], 2nd Razan Abdulhammed Abduljawad [2]
1. Department of Computer engineering technology, Engineering Technical College, Northern Technical University, Iraq, nora.rashid@ntu.edu.iq , 2. Department of Computer engineering technology, Engineering Technical College, Northern Technical University, Iraq, rabdulhammed@ntu.edu.iq .

Article Informations

Received: 23-01- 2023,
Revised:29-04-2023,
Accepted: 24-05-2023,
Published online: 17-10-2023

Corresponding author:
Name: Nora Rashid
Affiliation: Department of Computer engineering technology, Engineering Technical College, Northern Technical University, Iraq
Email: nora.rashid@ntu.edu.iq

Key Words: DDoS Detection; Machine Learning; Imbalanced Dataset; Feature Selection; CIC-IDS-2018; CIC-DDoS2019.

A B S T R A C T

Denial of service and Distributed denial of service (Dos/DDos) attacks continue to be one of the most significant dangers in cybersecurity. Many efforts are being put into developing defenses against these types of attacks. The tools used by attackers to perform these types of attacks increase day-to-day. Thus, a countermeasure is necessary. For this reason, this thesis utilized one of the most recent datasets (CSE-CICIDS2018 and CIC-DDoS2019) containing most Dos/DDoS attacks. This study proposed a framework based on Machine Learning for detecting denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks. The framework comprises three main modules: feature selection method using Random Forest—Recursive Feature Elimination (RF-RFE), handling the Imbalanced class distributions using Synthetic Minority Oversampling Technique (SMOTE), and classification. This study used five classifiers to make comparisons that include Random Forest (RF), Naive Bayes (NB), Logistic Regression (LR), and Linear and Quadratic Discriminant Analysis (LDA, QDA). Framework empirical findings reveal that the RF-RFE_SMOTE_RF outperformed all other models by obtaining an accuracy of 100% for CSE-CIC-IDS2018 and 0.99% for CIC-DDoS2019.

1. Introduction

To this day, DoS attacks remains an intractable issue facing the field of network security. The danger posed by DoS and DDoS attacks continues to occur, and the number of these attacks rises annually despite the existence of a multitude of investigative and preventative schemes [1]. DDoS attacks are a real threat nowadays and in the future as users and IoTs end nodes continue to grow exponentially in numbers. Moreover, DDos are becoming more common every day. According to a report by Kaspersky, DDoS attacks in 2022 spotted 57,116 attacks in third quarter of 2022 Year [2]. And it was 45.95% in second quarter of 2022 Year, by the third quarter of that year, it had dropped to 39.60%. This ratio rose from 38.690% in Q3 2021 to 53.533% in Q3 2022 throughout the globe. This ratio grew from 38.690% in Q3 2021 to 53.533% in Q3 2022 on a global scale. There was a 60% rise in malicious DDoS assaults in first half of 2022 compared to the same time in 2021 as reported of Secure List by Kaspersky [3]. This report demonstrates the rising demand for efficient frameworks and the necessity for an enhancement to the security mechanisms of DDoS counter measure. Performing DDoS attacks have become much easier and more cost-effective despite their increasing complexity [4]. DDoS assaults may be quickly organized at a minimal cost by the attackers simply by inputting the target addresses, and the organization mechanisms can be easily disabled. DDoS attacks are so easy and cheap to do, yet they pose a big threat to businesses on the internet. DoS attacks are used to block access to a system by users who are legitimately authorized to use it. In DDoS, the attackers use a variety of sources of dispersed attacks to achieve the same goal which is block accesses.

In this study, the authors suggest a machine learning-based framework for identifying DoS and DDoS attacks with network traffic. Protecting a system from attackers by manually monitoring network traffic is very time-consuming; hence, an intelligent security framework that can identify attacks is required. This study aims to improve the performance of the detection of these attacks and achieve better accuracy. So, this study suggests using SMOTE to resolve class imbalance and using the warped filter method based on RF-RFE to select the best features to hel the model perform well and reduce dimensionality.

The paper is organized into five sections. Related work is presented in Section 2. Whereas, Methodology are explained along with Dataset Description, Data preprocessing, Feature Selection, SMOTE,Normalization, and Performance Evaluation. Section 4 discuss Results, Finally, Section 5 Concludes the framework as a countermeasure.

2. Related works

The researcher, M. Alkasassbeh et al. [5], gathered a novel dataset which consists of 27 features and 5 different classes. This dataset was intended to be employed in various types of network attacks. The algorithms MLP (Multi-Layer Perceptron), NB, and RF were used for classification of DDOS attack. confusion matrix used to figure out how well the models did. achieved an accuracy of 98.63% for MLP algorithm, 98.02% for Random Forest algorithm, and 96.91% for Naive Bayes algorithm.

The researcher, V. Sharma et al. [6], deployed the machine learning methods of SVM, NB, and RF to the Snort haven dataset, classifying the data based on its characteristics and of four categories implemented in the WEKA. confusion matrix used to evaluate the study provided an accuracy of 99.7% for the SVM, 97.6% for the RF, and 98.0% for the NB.

W. Bhaya and M. Ebadymanaa, in 2017 [7], the researchers, utilized many unsupervised data analysis methods. The technique employed in this work to identify a DDoS attack is windowing on incoming packets utilizing DM algorithms combining CURE with a clustering model. Used the CAIDA2008, CAIDA2007, and DARPA2000 datasets in their implementation. The results showed a detection rate of 96.29%, an accuracy of over 99%, and FAR 0%.

Abdurrahman and M. K. Ibrahim [8], suggested a hybrid intrusion detection system for the detection of DDoS attacks in 2018. Based on the CICIDS2017 dataset, this dataset contains both DDoS attacks and normal traffic. RF, C5.0, NB, and SVM algorithms were used for the classification of DDoS attacks. The confusion matrix is used to determine which models have the best accuracy (86.80% for the RF, 86.45% for the C5.0, and 99% precision for both the
RF and the C5.0), but the lowest FAR is 0.05% for the RF, 0.046% for the C5.0, and the highest FAR is 75% for the SVM.

In 2021, N. M. Yungaicela-Naula et al. [9], utilize machine learning and deep learning algorithms such as GRU, RF, and LR for DDoS attack detection through depending on the CICDoS2017 and CICDDoS2019 datasets and

achieved an accuracy of 0.99% on new test data.CICDDoS2019 datasets and achieved an accuracy of 0.99% on new test data.

3. The Description of the Adopted Dataset

This subsection describes the datasets CSE-CIC-IDS2018 and CIC-DDoS2019 that were select to reflect the performance of the proposed framework.We utilized this dataset to train and test on DDoS detection approach. CSE-CIC-IIDS 2018 was created by Sharafaldin et al. [10], in2018. It provides many attack behaviors that represent common attack families. The attacks include Botnet attacks, DoS attacks, Brute-force attacks, DDoS attacks, Web attacks, and infiltration. The CSE-CIC-IDS2018 database includes 84 attributes created by the CICFlowMeter tool [11]. Sharafaldin et al. [12] also created CICDDoS2019 dataset, in 2019. It is a DDoS attack dataset. The total number of CICDDoS2019 instances is 500 63112 out of these are 56863 instances of normal class and 50006249 DDoS attack instances. The training version of the dataset carried out twelve different DDoS attacks, including NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, and TFTP. While, the training version of the dataset carried out seven different DDoS attacks, including PortScan, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, and SYN. The CICDoS2019 dataset includes 88 attributes, 84 of which were created using the CICFlowMeter tool, while Sharafalding et al created the other four. The paper focuses on two categories of Dos and DDos attacks.

In this paper , we utilized three files of CSE-CIC-IDS2018 dataset while in CIC-DDoS2019 dataset we utilized six files because content on the DoS/DDoS attacks. In CSE-CIC-IDS2018, we merges three files in a single combined file and feeds the combined file to the RF-RFE-SMOTE framework. After merges process the files, CSE-CIC-IDS2018 become contains 3,145,724 records, including 1,342,042 instances for DoS/DDoS attacks and 1,803,682 instances for Benign as shown in table 1. The attacks included DDoS-HOIC, DoS-Hulk, DoS SlowHTTPTest, DoS-GoldenEye, DoS-Slowloris, and DDoS-LOIC-UDP. Whereas for CIC-DDoS2019 dataset, we merges six files in a single combined file. After merges process the files, CIC-DDoS2019 become contains and contains 5111159 instances, including 5065529 DDoS attack instances and 45630 Benign instances as shown in table 2 .Hence, the combined file contains Syn, NetBIOS, UDP, LDAP, Portmap, MSSQL, UDPLag types of attacks. Figures 1 and 2 highlight the merging process for the two datasets.

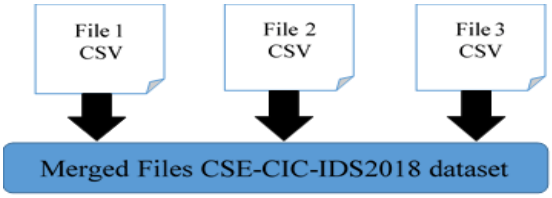


Figure 1. Merged Files CSE-CIC-IDS2018 dataset

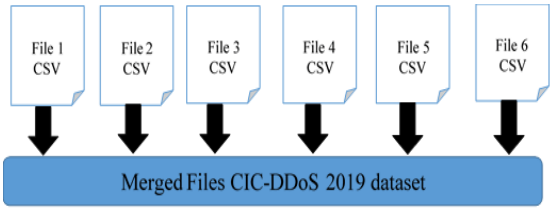


Figure 2. Merged Files of CIC-DDoS 2019 dataset

Table 1. Details of instances in the CSE-CIC-IDS2018 dataset

Classes	Instances
Benign	1803682
DDOS attack-HOIC	686012
DoS attacks-Hulk	461912
DoS attacks-SlowHTTPTest	139890
DoS attacks-GoldenEye	41508
DoS attacks-Slowloris	10990
DDOS attack-LOIC-UDP	1730
Total	3145724

Table 2. Details of instances in the CIC-DDoS2019 dataset

Classes	Instances
Benign	45630
LDAP	841586
MSSQL	24392
NetBIOS	1251410
Portmap	186960
Syn	1624663
UDP	1134645
UDPLag	1873
Total	5111159

4. Methodology of the Proposed Frame Work

This section presents a brief description of the framework. Figure 3 highlights the DoS/DDoS attack detection framework. The framework has three phases. The first phase takes care of feature

selection by applying RF-RFE for feature selection and feature reduction, the second phase handled the Imbalance class using SMOTE, and the third phase is the classification phase.

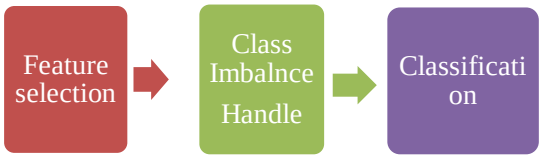


Figure3. Proposed DoS/DDoS attack detection framework

In which the framework tests various classifiers for the best classification performance evaluation. Figure 4 details the proposed framework along with all the adopted approaches .

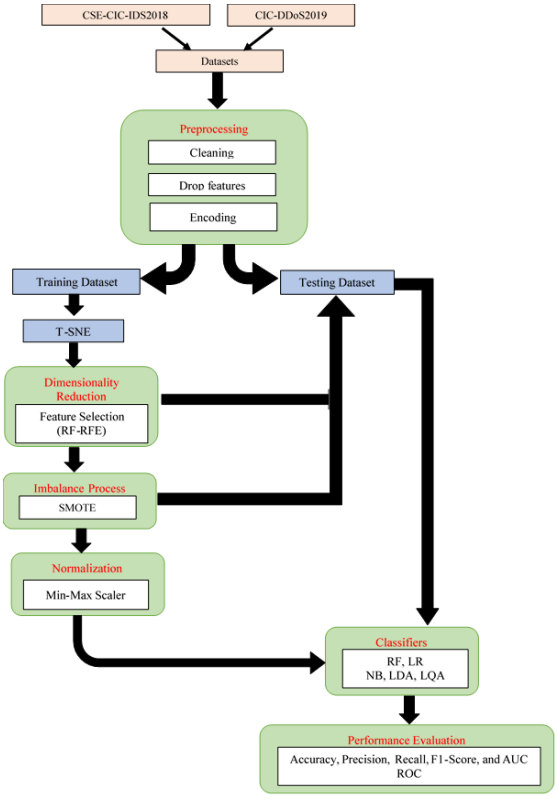


Figure 4. Brief details of Proposed DoS/DDoS attack detection Framework

4.1 Preprocessing

Data preprocessing is considered one of the essential steps in any machine-learning approach. This step is usually done at the very first stage. In general, to enrich the data quality and supports precise decision-making, cleaning, dropping, encoding, and splitting are included in the preprocessing of a particular dataset for better training of the machine-learning module.

- Cleaning: it is used to fix and remove any incomplete information in a certain dataset. The adopted datasets contain a large amount of missing (NaN) and infinity (Inf) values. Thus, the proposed framework cleans these by removing (NaN) and infinity (Inf) values.

- Dropping: for each CSE-CIC-IDS2018 and CIC-DDoS2019 dataset we drop features such as "Timestamp" which are of little help in training our neural network, and Some of the CSV files contain the Features "Unnamed: 0", "Flow ID", "Source IP", "Source Port" and "Destination IP", these features are not accessible in any of the other CSV files, hence they have been deleted from the files. The Flow ID, Source IP, and Destination IP are non-numeric data types. Thus, they are not suitable for machine learning algorithms in their current form, but the loss of the model. The features in CSE-CIC-IDS2018 dataset fall after dropping from 83 to 77 features and the features and CIC-DDoS2019 dataset become after dropping from 87 to 81 features.

- Encoding: is frequently employed to deal with categorical variables. Each label is given a unique integer. The CSE-CIC-IDS2018 dataset includes seven various types of attacks labeled, six attacks and normal. To prepare it for machine learning, it is numbered from0 to 6.in the other CIC-DDoS2019 dataset has been labelled with eight different types of attacks, seven attacks and benign, it is numbered from 0 to 7.

- Splitting: it is used to split dataset into (Train and Test) to evaluate the performance of the model. The proposed framework import The propoosed framework imported the Train test split function from the "Sklearn Library" in python and used70 percent of data for training and 30 percent for the test. This percentage (70:30) is considered standard among machine learning developers.

4.2 t-distributed stochastic neighbor embedding (t-SNE)

To give deep insight into the datasets, we utilize t-SNE [13] to visualize both CSE-CIC-IDS2018 and CIC-DDoS2019. As plotted in figures 5 and 6, the attack instances in CSE-CIC-IDS2018 are less than the normal instances. Thus, it is easier for specific attack behavior to remain hidden. Furthermore, for CIC-DDoS 2019, the normal instances are less than the attack instances, which causes confusion among them and makes it increasingly difficult for traditional intrusion detection technology to detect attacks.

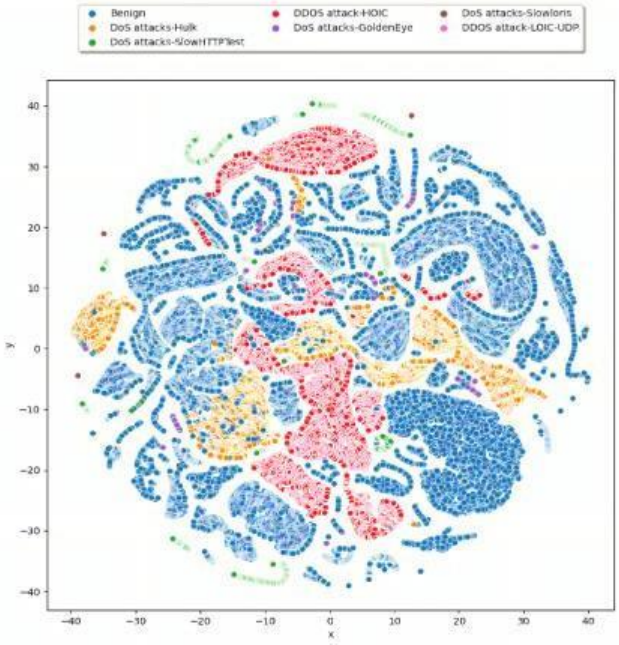


Figure 5. CSE-CIC-IDS2018 Visualization Using T-SNE

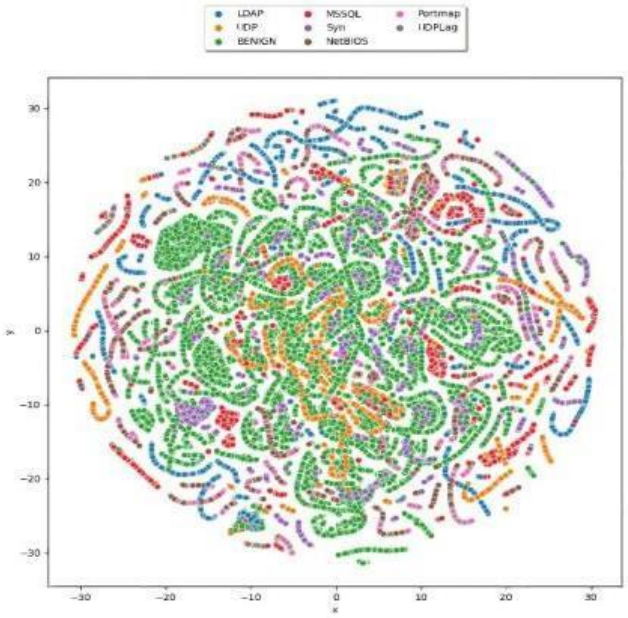


Figure 6. CIC-DDoS2019 Visualization Using T-SNE

3.4 Feature Selection

The primary FS task is finding the original dataset's most critical features or groups. Some attributes (features) are necessary for DoS/DDoS detection, while some others may just be noise, harming the training speed and accuracy. Training the classifier using the whole set of features has been demonstrated to reduce model performance [14]. In this work, decreased the feature space dimension by combining RF algorithm with RFE "Random Forest—Recursive Feature Elimination algorithm (RF-RFE)". It is assumed tthat data redundency is eliminateed and produces moree compact featuree subsets. The steps of the RF-RFE approach are shown in figure 7. First, we utilised the training data to train the model with RF algorithm, and We determined an importance per each feature depending on to its classification contribution.. Next, the features were ranked from most important to least important. At this step, feature rankings were determined. Lastly, we eliminated a lowest important feature and retrained RF model with the updated features, and acquired classification results with the new feature set. This process is implemented in an iterative procedure until the feature set is empty. After RF-RFE, a list of performance measurement values corresponding to each subset was produced. Based on the list of values, we explored the decision variant used for subset selection. Based on this, 39 best features Group Set (39-RF-RFE) was selected in CSE-CIC-IDS2018 and 40 best features Group Set (40-RF-RFE) in CIC-DDoS2019 as shown in tables 3 and 4.

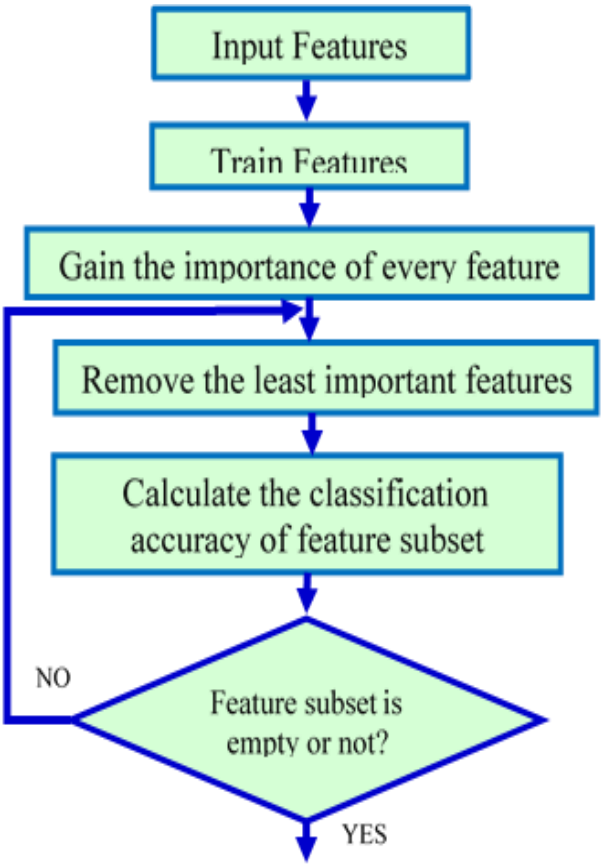


Figure 7. The Main Procedure of the Recursive Feature Elimination (RFE) Method

Table 3. The 39 Features Set CSE-CIC-IDS2018

No	Feature Name	No	Feature Name
1	Dst Port	21	Bwd Header Len
2	Flow Duration	22	Fwd Pkts/s
3	Tot Bwd Pkts	23	Bwd Pkts/s
4	TotLen Fwd Pkts	24	Pkt Len Max
5	TotLen Bwd Pkts	25	Pkt Len Mean
6	Fwd Pkt Len Max	26	Pkt Len Std
7	Fwd Pkt Len Mean	27	Pkt Len Var
8	Bwd Pkt Len Max	28	PSH Flag Cnt
9	Bwd Pkt Len Mean	29	ACK Flag Cnt
10	Bwd Pkt Len Std	30	Pkt Size Avg
11	Flow Byts/s	31	Fwd Seg Size Avg
12	Flow Pkts/s	32	Bwd Seg Size Avg
13	Flow IAT Mean	33	Subflow Fwd Pkts
14	Flow IAT Max	34	Subflow Fwd Byts
15	Flow IAT Min	35	Subflow Bwd Pkts
16	Fwd IAT TotFwd	36	Subflow Bwd Byts
17	IAT Mean	37	Init Fwd Win Byts
18	Fwd IAT Max	38	Init Bwd Win Byts
19	Fwd IAT Min	39	Fwd Seg Size Min
20	Fwd Header Len		

Table 4. The 40 Features Set CIC-DDoS2019

No	Feature Name	No	Feature Name
1	Destination Port	21	Bwd Packets/s
2	Protocol	22	Min Packet Length
3	Flow Duration	23	Max Packet Length
4	Total Fwd Packets	24	Packet Length Mean
5	Total Length of Fwd Packets	25	Packet Length Std
6	Fwd Packet Length Max	26	Packet Length Variance
7	Fwd Packet Length Min	27	ACK Flag Count
8	Fwd Packet Length Mean	28	URG Flag Count
9	Fwd Packet Length Std	29	CWE Flag Count
10	Flow Bytes/s	30	Average Packet Size
11	Flow Packets/s	31	Avg Fwd Segment Size
12	Flow IAT Mean	32	Fwd Header Length.1
13	Flow IAT Std	33	Subflow Fwd Packets
14	Flow IAT Max	34	Subflow Fwd Bytes
15	Flow IAT Min	35	Subflow Bwd Packets
16	Fwd IAT Total	36	Init_Win_bytes_forward
17	Fwd IAT Mean	37	Init_Win_bytes_backward
18	Fwd IAT Max	38	act_data_pkt_fwd
19	Fwd IAT Min	39	min_seg_size_forward
20	Fwd Header Length	40	Inbound

3.5 Synthetic Minority Oversampling Technique (SMOTE)

Imbalance datasets can lead to misclassification problems. Thus, it will affect machine-learning models and degrade their performance. To overcome this problem, developers oversample the minority class[15]. The proposed framework applies SMOTE in the training set instances. Tables 5 and 6 show the number of training instances before and after using SMOTE and testing instances, respectively. Figures 8 and 9 shows the difference between the training phase without SMOTE and with the SMOTE. The training set before SMOTE in CIC-IDS 2018 contains 2,202,006 instances, whereas after SMOTE contains 8839628 instances, and the testing set consists of 943718 instances. The training set before SMOTE in CIC-DDoS 2019 contains 3577811 instances, the training set after SMOTE contains 9100096 instances, and the testing set contains 1533348 instances.

Table 5. Distribution of the classes in the CIC-IDS 2018 dataset before and after SMOTE

Category	Attack Type	Sample train before SMOTE	Test	Sample train after SMOTE	Test
Benign	Benign	1262804	540878	1262804	540878
DoS attack	DoS-Hulk	479781	206231	1262804	206231
	DoS-SlowHTTPTest	29121	12387	1262804	12387
	DoS-GoldenEye	323087	138825	1262804	138825
	DoS-Slowloris	98309	41581	1262804	41581
DoS attack	DDOS-LOIC-UDP	7709	3281	1262804	3281
	DDoS-HOIC	1195	535	1262804	535
Total	/	2202006	943718	8839628	943718

Table 6. Distribution of the classes in the CIC-DDoS2019 dataset before and after SMOTE

Category	Attack Type	Sample train before SMOTE	Test	Sample train after SMOTE	Test
Benign	Benign	31876	13754	1137512	13754
DDoS attack	LDAP	588798	252788	1137512	252788
	MSSQL	16994	7398	1137512	7398
	NetBIOS	875638	375772	1137512	375772
	Portmap	131117	55843	1137512	55843
	Syn	1137512	487151	1137512	487151
	UDP	794550	340095	1137512	340095
	UDPLag	1326	547	1137512	547
Total	/	3577811	1533348	9100096	1533348

Figure 8. SMOTE on training CIC-IDS 2018 dataset

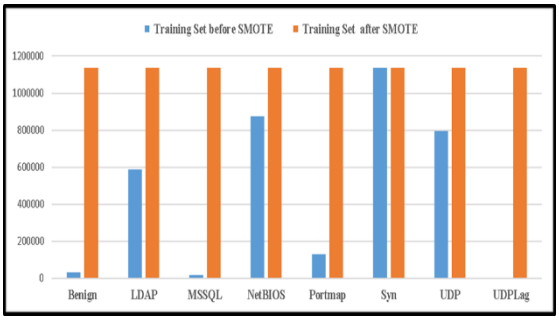


Figure 9. SMOTE on training CIC-DDoS2019 dataset

3.6 Normalization

In this subsection, we employ normalization to re-scale thee dataset's features depending on each feature's minimum and maximum values. A data was normalized to smaller range using a MinMaxScaler between (0, 1). The machine learning models' calculations become overly time- and space-intensive since each column in the data includes a varied range of data. The data is represented in a standard scale to reduce this burden (computations and time-consuming) by changing the values from the original scale to the (0, 1) scale. X"scale" computes as:

$$Xscale = \frac{Xi - Xmin}{Xmax - Xmin} \quad (1)$$

Where Xi = feature value, Xmin = minimum feature value, and Xmax = maximum feature value.

3.7 Performance Evaluation

To evaluate the performance of the proposed DoS/DDoS Attack detection framework the following metrics were used: accuracy, precision, recall, F1-Score, and ROC-AUC.

- Accuracy: indicates the number of correct produced predictions over the entire dataset. or it can be defined as, how many positive is properly predicted by the model over the whole dataset. Accuracy computes as[16]:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (2)$$

- Precision: indicates how accurate the model is in terms of positive results. It calculates how many positive values are predicted actually positive among all positive (positive case that are correctly classified as a positive over all case are classified as a positive). Precision computes as:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

- Recall: is the potential of a model to correctly predict the correct positives (positive case that are correctly classified as a positive over all actual positives). The recall computes as:

$$\text{Recall} \frac{TP}{TP+FN} \quad (4)$$

- F1-Score: It is the harmonic mean of precision and recall. It is a metric for determining how accurate a model is because it considers both how well the model makes true predictions that

are actually true and how many of all true predictions the model correctly anticipated. F1-score computes as:

$$F1Score = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \qquad (5)$$

Where TP: means both the ground truth and the network output are positives, TN: means both the network output and the ground truth are negatives, FP: means the ground truth is negative while the network output is positive, and FN: means the ground truth is positive while the network output is negative.

Receiver Operating Characteristics (ROC) curve: is analysis derives from the signal processing technique. Its usefulness is not limited to the model alone, but spans several practical fields [17]. When it comes to classification tasks, FAR (1-Specificity) and Sensitivity are presented as a compromise. The ROC curve demonstrates this compromise. ROC curves frequently serve as a way to evaluate the model's performance. The ROC curve contains Sensitivity on the (Y-axis) and 1-Specificity on the (X-axis) [18] (a larger area under the ROC curve indicates that the classifier is better able to distinguish between the two unique categories) [123]. On the opposing hand, ROC curve is used in the binary classification issue. The model is successful when the AUC (Area Under the ROC Curve) value is close to 1.

In addition, confusion matrix is a useful tool for accurately assessing classification models. In general, a confusion matrix is a matrix that consists of C × C (C here refers to the number of classes). This matrix is used to show the amount of the data samples that the model classified it correctly, and the amount of the data samples that the model classified it incorrectly. In the case of C=2 (two classes), the confusion matrix divides prediction results of the classifier into four categories , True-Positive (TP), True-Negative (TN), False-Positive (FP) and False-Negative (FN). The confusion matrix for two classes classification is shown in figure 10 [16].

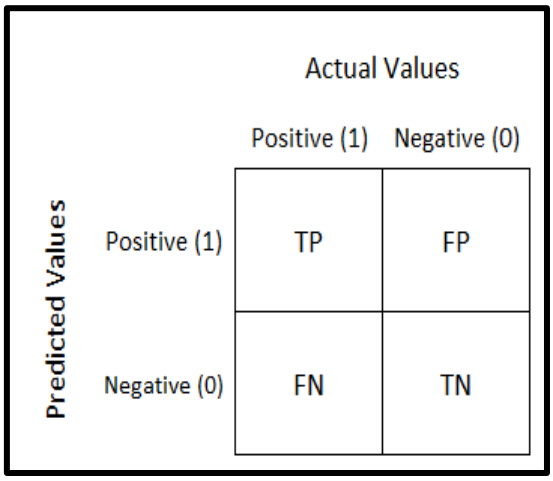


Figure 10. 2 × 2 Confusion Matrix [16]

4. Results

This section discusses the experiment's results aimed at detecting Dos/DDos attacks. All models were evaluated using the CSE-CIC-ID-2018 test set and a CIC-DDOS-2019 test set. Accuracy, AUC-ROC and confusion matrix have been used to show the performance, and classification reports (Precision, Recall, and F1-Score) have been calculated to evaluate the performance of framework.

4.1 Results for the CSE-CIC-IDS2018 Dataset

Table 7 displays the summary of the average of the results, and the classification report shows the specific results for each type of attack is illustrated in table 8. In this table, precision, recall, and F1-score respectively are calculated. Figure 11shows compared performance models; RF-RFE_SMOTE_RF achieves better performance for all classes of attacks.

The results of the test on CSE-CIC-DS2018 show that the RF-RFE_SMOTE_RF model produced the best estimation based on the accuracy, precision, recall, and f1-score criteria, with values of 1.00, 1.00, 1.00, and 1.00, respectively. In addition, the model produced the highest results in terms AUC, with value of 1.00.The RF-RFE_SMOTE_RF model achieves the highest performance of all classes.

Whereas RF-RFE_SMOTE_LDA and RF-RFE_SMOTE_LR models perform slightly poorly. So, where observed nearly identical accuracy, 0.9669 and 0.96086, with AUC-ROC of 0.96 and 0.99. However, precision, recall, and F-Score are all a little low, indicating that accuracy is often not beneficial with balanced data. While RF-

RFE_SMOTE_QDA and RF-RFE_SMOTE_NB clearly show that methodology is the worst performer. According to the results, the RF-RFE_SMOTE_RF, and RF-RFE_SMOTE_LDA models can recognize all classes.

The RF-RFE_SMOTE_QDA model seems to be the most unsuccessful model in classifying some classes of attack such as DoS-Hulk, DoS-GoldenEye, and DDoS-LOIC-UDP. The recall value for DoS-Hulk achieved 0.76 while DoS-GoldenEye, and DDoS-LOIC-UDP achieved the lowest value 0.15, and 0.46 respectively. The precision value for DoS-Hulk achieved the lowest value 0.49 while DoS-GoldenEye, and DDoS-LOIC-UDP achieved the greatest value 0.75, and 0.96 respectively. The f1-score value for DoS-Hulk, DoS-GoldenEye, and DDoS-LOIC-UDP achieved the lowest value 0.60, 0.25, and 0.62 respectively. The RF-RFE_SMOTE_NB model was not able to correctly classify DDoS-LOIC-UDP class and the worst results were with SlowHTTPTest class has low recall, precision, and F-score values. The RF-RFE_SMOTE_LR model is capable of distinguishing all classes but the worst results were with DoS-SlowHTTPTest class has low values for Precision and F-score. On the other hand, it achieved a fair recall value with 0.74.

Figure 12 depict the AUC-ROC curves of each class and figure 13 shows the confusion matrix that indicates how well the classes were predicted, as well as which classes were wrongly predicted. In accordance with the area under the ROC curves in figure 12, RF-RFE_SMOTE_RF has higher accuracy due to their 100% success rate in detecting all attacks. All of the areas under the ROC curves for all classes are nearly equal to one. While RF-RFE_SMOTE_LDQ also has higher accuracy and achieve a 97%, 98%, and 99% detection rate in the some attacks and detection rates for other classes are nearly equal to value one. In case of the RF-RFE_SMOTE_QDA, the area under the ROC curve is nearly equal to one for the 2, and 5 classes, and detection rates for other classes 1, 4, and 6 are very weak. Whereas 0, and 3 classes have nearly close to 80% detection rates. RF-RFE_SMOTE_LR model, the area under the ROC curve is nearly equal to one for the 1, and 5 classes only, and the detection rates for 0, 2, and 4 classes are close to 90%, while 3, and 6 classes have close to 80% detection rates. In the case of RF-RFE_SMOTE_NB, the area under the ROC curve is nearly equal to one for the 1, and 5 classes only, and the detection rates for 0,2, 4,and 6 classes are close to 90% and 80%, while 3 has close to 50% detection rates.

Table 7. Averaged Evaluation of Methods for Multi-class Classification after RF-RFE with SMOTE on CSE-CIC-IDS2018

Model	Recall	Precision	F1-score	accuracy
RF-RFE_SMOTE_RF	**1.00**	**1.00**	**1.00**	**1.00**
RF-RFE_SMOTE_LDA	0.874	0.882	0.878	0.9669
RF-RFE_SMOTE_QDA	0.70142	0.83857	0.72428	0.75534
RF-RFE_SMOTE_LR	0.91857	0.84142	0.86428	0.96086
RF-RFE_SMOTE_NB	0.79428	0.72571	0.74428	0.93830

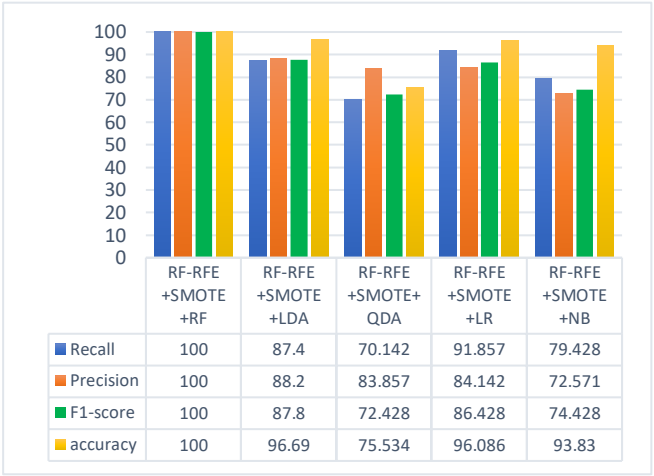
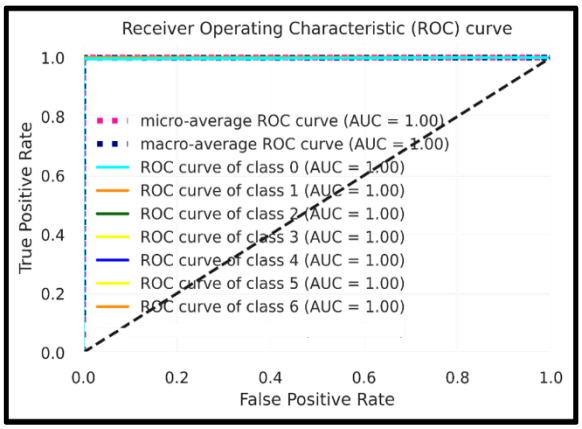


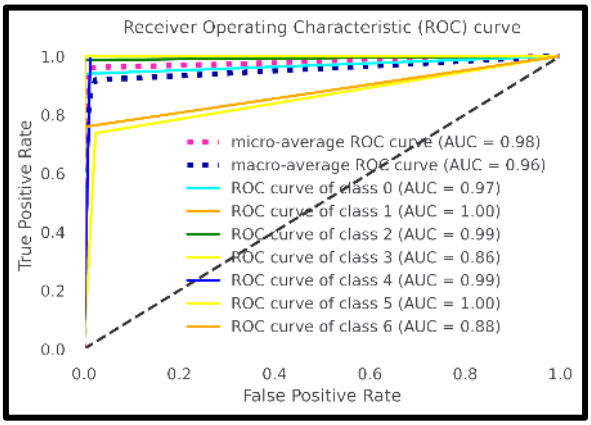
Figure 11. Performance Comparison of All Models

Table 8. Results of Testing Performance Evaluation on Methodology for Each Class of CSE-CIC-IDS2018

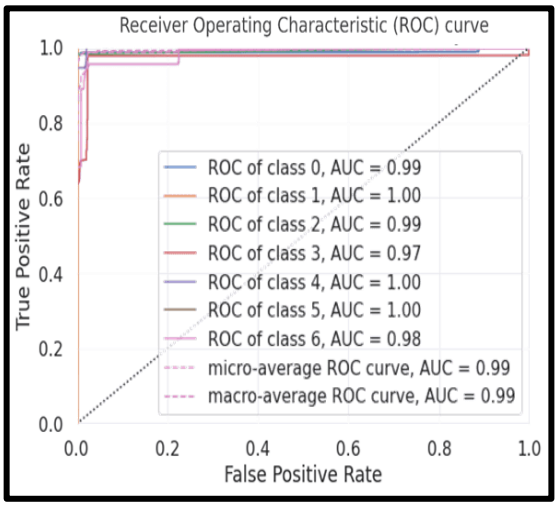
Model	Classes	Recall	Precision	F1-score
RF-RFE_SMOTE_RF	Benign	**1.00**	**1.00**	**1.00**
	DoS-Hulk	**1.00**	**1.00**	**1.00**
	DoS-HOIC	**1.00**	**1.00**	**1.00**
	DoS-SlowHTTPTest	**1.00**	**1.00**	**1.00**
	DoS-GoldenEye	**1.00**	**1.00**	**1.00**
	DoS-Slowloris	**1.00**	**1.00**	**1.00**
	DDOS-LOIC-UDP	**1.00**	**1.00**	**1.00**
RF-RFE_SMOTE_LDA	Benign	0.97	1.00	0.99
	DoS-Hulk	1.00	0.96	0.98
	DoS-HOIC	0.98	1.00	0.99
	DoS-SlowHTTPTest	0.65	0.57	0.61
	DoS-GoldenEye	0.95	0.89	0.92
	DoS-Slowloris	0.90	0.99	0.94
	DDOS-LOIC-UDP	0.67	0.77	0.72
RF-RFE_SMOTE_QDA	Benign	0.89	0.89	0.89
	DoS-Hulk	0.76	0.49	0.60
	DoS-HOIC	1.00	0.99	1.00
	DoS-SlowHTTPTest	0.65	0.81	0.72
	DoS-GoldenEye	0.15	0.75	0.25
	DoS-Slowloris	1.00	0.98	0.99
	DDOS-LOIC-UDP	0.46	0.96	0.62
RF-RFE_SMOTE_LR	Benign	0.94	1.00	0.97
	DoS-Hulk	1.00	0.99	1.00
	DoS-HOIC	0.99	1.00	0.99
	DoS-SlowHTTPTest	0.74	0.29	0.41
	DoS-GoldenEye	1.00	0.93	0.96
	DoS-Slowloris	1.00	0.98	0.99
	DDOS-LOIC-UDP	0.76	0.70	0.73
RF-RFE_SMOTE_NB	Benign	0.93	0.96	0.95
	DoS-Hulk	1.00	0.99	1.00
	DoS-HOIC	0.99	1.00	0.99
	DoS-SlowHTTPTest	0.04	0.06	0.05
	DoS-GoldenEye	0.94	0.85	0.89
	DoS-Slowloris	1.00	1.00	1.00
	DDOS-LOIC-UDP	0.66	0.22	0.33



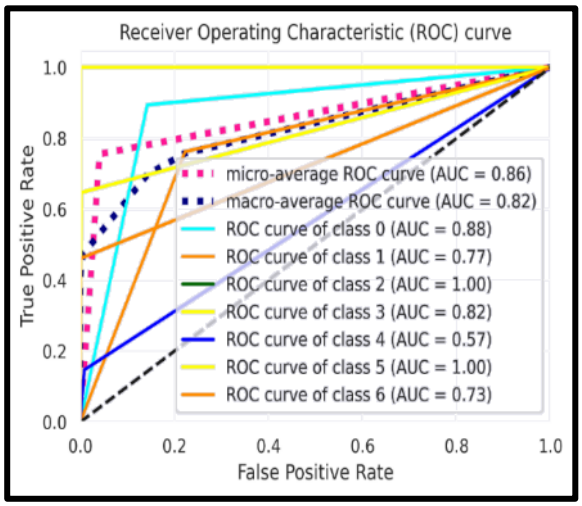
a



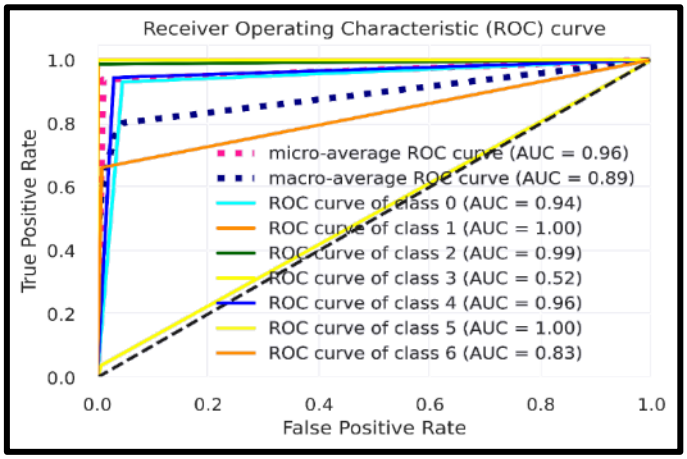
b



c

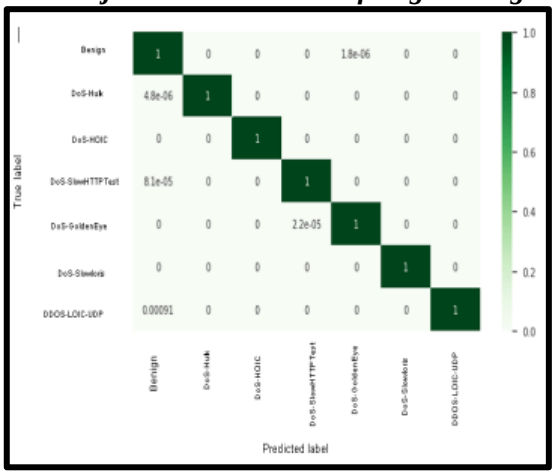


d



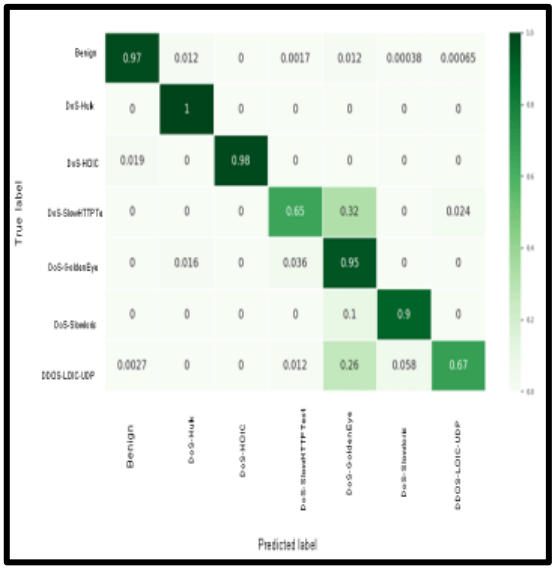
e

Figure 12. ROC Curve based on CSE-CIC-IDS2018dataset for (a) RF-RFE_SMOTE_RF (b) RF-RFE_SMOTE_LR (c) RF-RFE_SMOTE_LDA (d) RF-RFE_SMOTE_QDA (e) RF-RFE_SMOTE_NB

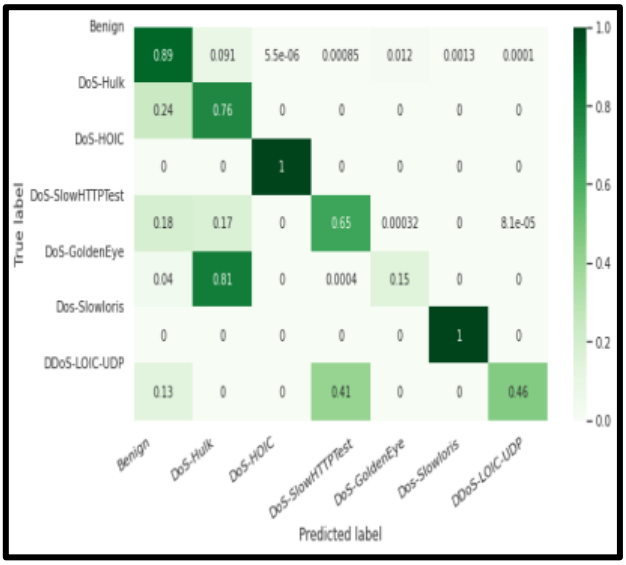


a

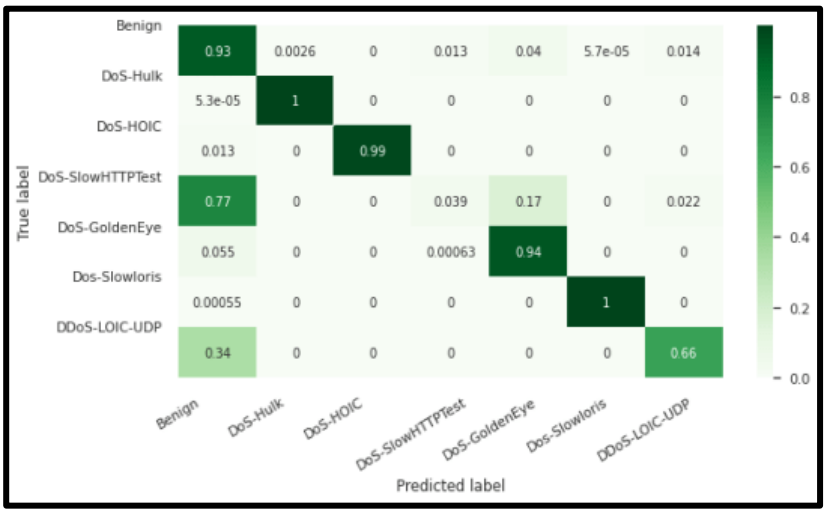
d



c



b



e

Figure 13. Confusion Matrix based CSE-CIC-IDS2018 dataset for (a) RF-RFE_SMOTE_RF (b) RF-RFE_SMOTE_LR (c) RF RFE_SMOTE_LDA (d) RF-RFE_SMOTE_QDA (e) RF-RFE_SMOTE_NB

According to the Confusion Matrix in figure (4.3), The RF-RFE-SMOTE_LR model predicted the'DoS SlowHTTPTes'attack with an accuracy of 74% and classified it as 26% 'Dos-GoldenEye' and predicted the 'DDoS-LOIC-UDP' attack with an accuracy of 76% and classified it as 21% 'DoS-Slowloris'.

The RF-RFE-SMOTE_LDA model predicted the'DoS-SlowHTTPTes' attack with an accuracy of 65%, classified it as 32% 'Dos-GoldenEye', predicted the 'DoS-Slowloris' attack with an accuracy of 90%, and classified it as 10% 'Dos-GoldenEye'. in addition, the RF-RFE-SMOTE_LDA classified the 'DDoS-LOIC-UDP'attack so, with an accuracy of 67%, and classified it as 26% 'Dos-GoldenEye', and 5%' DoS-Slowloris'.

The RF-RFE-SMOTE_QDA model predicted 'DoS-Hulk' attack with an accuracy of 76% and classified it as 24% 'Benign' and predicted the 'DoS-SlowHTTPTest' attack with an accuracy of 65% and classified it as 18% 'Benign' and 17% 'DoS-Hulk'. in addition, the RF-RFE-SMOTE_QDA model misclassified the 'Dos-GoldenEye' and 'DDoS-LOIC-UDP' it classified as 81%, 13% and 41% 'DoS-Hulk' , 'Benign' and 'DoS SlowHTTPTes' attacks respectively.

The RF-RFE-SMOTE_NB model succeed predicted all classes except 'DoS SlowHTTPTes' was unable to accurately classify and misclassified it as regular traffic. and The RF-RFE-SMOTE_NB model predicted 'DDoS-LOIC-UDP' attack with an accuracy of 66% and classified it as 34% 'Benign' While the RF-RFE-SMOTE_RF model obtained the best classification results among the other classification algorithms. The performance evaluation metrics for different techniques trained on the CSE-CIC-IDS2018 dataset in terms of the time to build and test the model is presented in table (4.3).

From table 9, it can be distinguished that RF-REF-SMOTE_NB takes the minimum build time, but RF-REF-SMOTE_RF takes the maximum build time and test time, while in the testing state, RF-REF-SMOTE_LR takes the lowermost time. The lowest times to test the model were achieved by RF-REF-SMOTE_LR with 0.008 seconds. The RF-REF-SMOTE_NB classifier takes the minimum build time but that has the worst detection performance.

Here, the RF-REF-SMOTE_RF classifier that has the best detection performance. The RF-REF-SMOTE_RF classifier comes with the highest overhead in terms of the time to build and test the model but RF-REF-SMOTE_NB classifier comes with the lowest overhead in terms of the time to build and test the model.

Table 9. Time to build and test the models for the CSE-CIC-IDS2018 dataset

Classifier	Time to Build the Model (Sec.)	Time to Test the Model (Sec.)
RF-RFE_SMOTE_RF	**733.352**	**10.305**
RF-RFE_SMOTE_LDA	114.633	0.025
RF-RFE_SMOTE_QDA	5.983	2.497
RF-RFE_SMOTE_LR	103.936	**0.008**
RF-RFE_SMOTE_NB	**1.046**	0.386

4.2 Results for the CIC-DDoS2019 Dataset

Table 10 provides a summary of the average findings, while table 11 illustrates the classification report's individual outcomes for every type of attack. In this table, the accuracy, recall, and F1-score are computed separately. Figure14 shows compared performance models; RF-RFE SMOTE_ RF achieves better performance for all classes of attacks.

Table 10. Averaged Evaluation of Methods for Multi-class Classification on CIC-DDoS2019

Model	Recall	Precision	F1-score	accuracy
RF-RFE_SMOTE_RF	**0.98875**	**0.9975**	**0.9925**	**0.99990**
RF-RFE_SMOTE_LDA	0.8575	0.81125	0.82875	0.99111
RF-RFE_SMOTE_QDA	0.55375	0.68875	0.45375	0.5891
RF-RFE_SMOTE_LR	0.90375	0.94	0.92	0.99619
RF-RFE_SMOTE_NB	0.58	0.59375	0.5825	0.953723

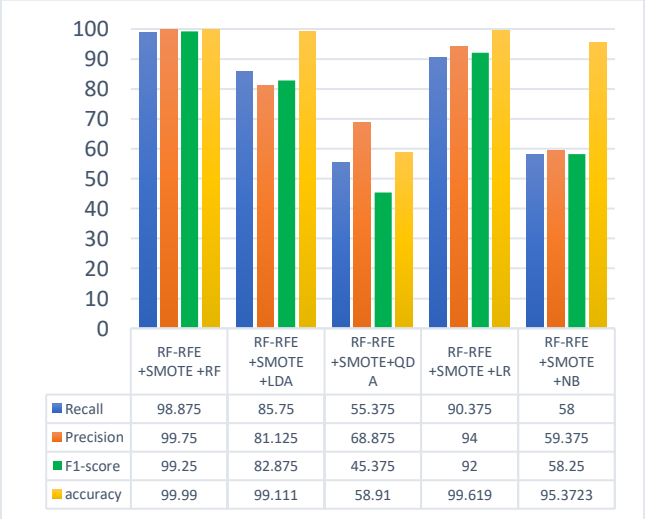


Figure 14. Performance Comparison of All Models

Table 11. Results of testing Performance evaluate on methodology for each class of CIC-DDoS2019

Model	Classes	Recall	Precision	F1-score
RF-RFE_SMOTE_RF	Benign	**1.00**	**1.00**	**1.00**
	LDAP	**1.00**	**1.00**	**1.00**
	MSSQL	**1.00**	**1.00**	**1.00**
	NetBIOS	**1.00**	**1.00**	**1.00**
	Portmap	**1.00**	**1.00**	**1.00**
	Syn	**1.00**	**1.00**	**1.00**
	UDP	**1.00**	**1.00**	**1.00**
	UDPLag	0.91	0.98	0.94
RF-RFE_SMOTE_LDA	Benign	0.97	0.96	0.96
	LDAP	1.00	1.00	1.00
	MSSQL	0.62	0.44	0.52
	NetBIOS	0.99	1.00	1.00
	Portmap	0.99	0.97	0.98
	Syn	1.00	1.00	1.00
	UDP	0.98	0.99	0.99
	UDPLag	0.31	0.13	0.18
RF-RFE_SMOTE_QDA	Benign	1.00	0.88	0.93
	LDAP	0.90	1.00	0.95
	MSSQL	0.97	0.01	0.02
	NetBIOS	0.00	1.00	0.00
	Portmap	0.00	0.57	0.00
	Syn	0.99	1.00	1.00
	UDP	0.50	1.00	0.67
	UDPLag	0.07	0.05	0.06
RF-RFE_SMOTE_LR	Benign	0.97	0.95	0.96
	LDAP	1.00	1.00	1.00
	MSSQL	0.69	0.81	0.75
	NetBIOS	1.00	1.00	1.00
	Portmap	0.99	1.00	1.00
	Syn	1.00	1.00	1.00
	UDP	1.00	0.99	0.99
	UDPLag	0.58	0.77	0.66
RF-RFE_SMOTE_NB`	Benign	0.64	0.91	0.75
	LDAP	1.00	1.00	1.00
	MSSQL	0.00	0.00	0.00
	NetBIOS	1.00	0.87	0.93
	Portmap	0.00	0.00	0.00
	Syn	1.00	0.99	0.99
	UDP	1.00	0.98	0.99
	UDPLag	0.00	0.00	0.00

The test results on CIC-DDOS2019, the best model is the RF-RFE_SMOTE_RF, which only achieved 0.99990 % accuracy, AUC-ROC of 99, precision of 0.9975, recall of 0.98875, and an F-score of 0.9925. The RF-RFE_SMOTE_RF model recognizes all classes. On the other side, The RF-RFE_SMOTE_LDA and the RF-RFE_SMOTE_LR models have an accuracy of 0.99111 and 0.99619, respectively, and the AUC-ROC of 0.93 and 0.95, respectively, achieve the highest. While the worst methodologies are RF-RFE_SMOTE_QDA and RF-RFE_SMOTE_NB. The RF-RFE_SMOTE_NB and RF-RFE_SMOTE_QDA models were not able to correctly classify the MSSQL, Portmap, UDPLag, NetBIOS, and, UDP classes. According to the results, there is a problem with detecting classes like Portmap, UDPLag, MSSQL, and NetBIOS, UDP. However, the detection accuracy for the other class types is rather high.

Figure 15 depict the AUC-ROC curves of each class and figure 16 shows the confusion matrix that indicates how well the classes were predicted, as well as which classes were wrongly predicted. In accordance with the area under the ROC curves in figure 15, RF-RFE_SMOTE_RF has higher accuracy due to their 100% success rate in detecting all attacks. All of the areas under the ROC curves for all classes are nearly equal to one. The RF-RFE_SMOTE_LR model, the area under the ROC curve is nearly equal to one for the 1, 3, 4, 5, and 6 classes, and detection rates for other classes 0, and 2 are close to 90% and 80%, while 7 class has close to 70% detection rates. While RF-RFE_SMOTE_LDQ model, the area under the ROC curve is nearly equal to one for the 1, 3, 4, and 5 classes only, and the detection rates for 2, and 7 classes are close to 80% and 60%, while 0, and 6 classes have close to 90% detection rates. In the case of RF-RFE_SMOTE_QDA and RF-RFE_SMOTE_NB, the area under the ROC curve is nearly equal to one for the 1, 5, and 6 classes only, and the detection rates for 0, and 3 classes are close to 80% and 90%, while 2, 4, and 7classes have close to 50% detection rates.

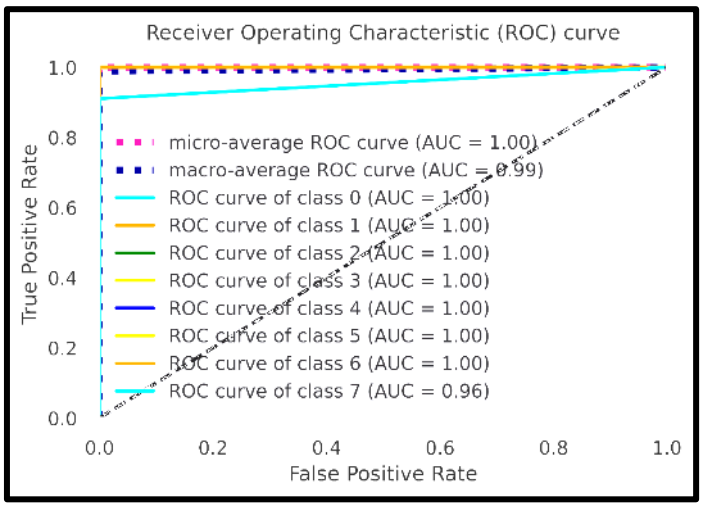
According to the Confusion Matrix in figure (4.6), The RF-RFE_SMOTE_LR model predicted the'MSSQL attack with an accuracy of 69%, classified it as 26% 'UDP', predicted the 'UDPLag' attack with an accuracy of 58%, and classified it as 27% 'UDP'.

In addition, The RF-RFE_SMOTE_LDA model predicted the'MSSQL'attack with an accuracy of 62% and classified it as 31% 'UDP'. On the other hand, the RF-RFE_SMOTE_LDA failed classified'UDPLag' attack so, with an accuracy of 31%, and classified it as 55% 'UDP'. The RF-RFE_SMOTE_QDA model failed classified all attacks except Bening, LDAP, MSSQL,Syn attacks that RF-RFE_SMOTE_QDA model succeed predicted. The RF-RFE_SMOTE_NB model succeed predicted all classes except 'MSSQL', 'PORTMAP', and 'UDPLag' was unable to accurately classify. and The RF-RFE_SMOTE_NB model predicted 'Bening' with an accuracy of 64% and classified it as 33% 'syn' attack While the RF-RFE_SMOTE_RF model obtained the best classification results among the other classification algoritms.The performance evaluation metrics for different techniques trained on the CSE-CIC-IDS2019 dataset in terms of the time to build and test the model is presented in table 12.

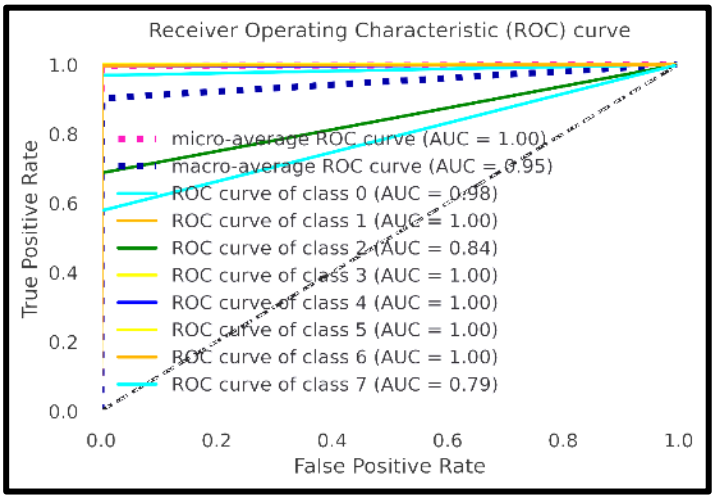
Table 12. Time to build and test the models for the CSE-CIC-IDS2019 dataset

Classifier	Time to Build the Model (Sec.)	Time to Test the Model (Sec.)
RF-RFE_SMOTE_RF	**836.714**	**15.179**
RF-RFE_SMOTE_LDA	52.572	0.768
RF-RFE_SMOTE_QDA	20.864	9.743
RF-RFE_SMOTE_LR	233.53	0.014
RF-RFE_SMOTE_NB	**3.484**	**0.009**

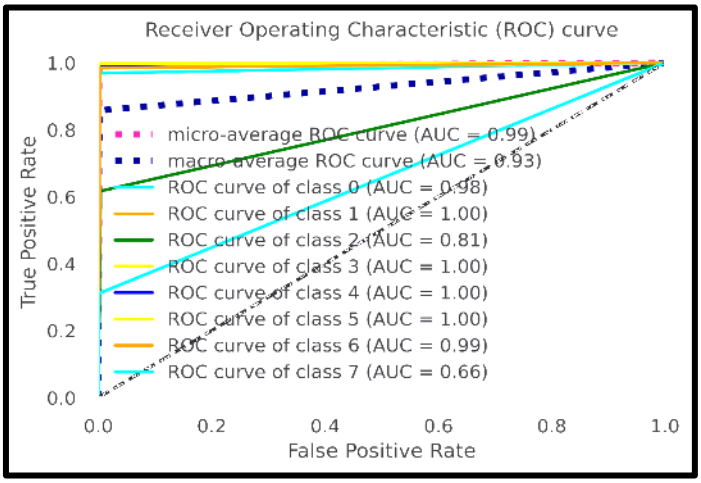
From table (4.6) it can be distinguished that RF-RFE_SMOTE_NB takes the minimum build time, but RF-RFE_SMOTE_RF takes the maximum build time and test time, while in the testing state, RF-RFE_SMOTE_NB takes the lowermost time. The lowest times to test the model was achieved by RF-RFE_SMOTE_NB with 0.009 seconds. The RF-RFE_SMOTE_NB classifier takes the minimum build time and test time but that has the worst detection performance. Here, the RF-RFE_SMOTE_RF classifier that has the best detection performance.



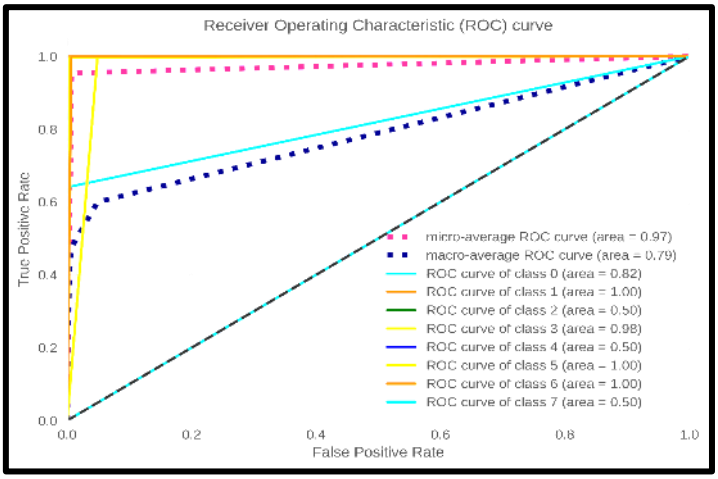
a



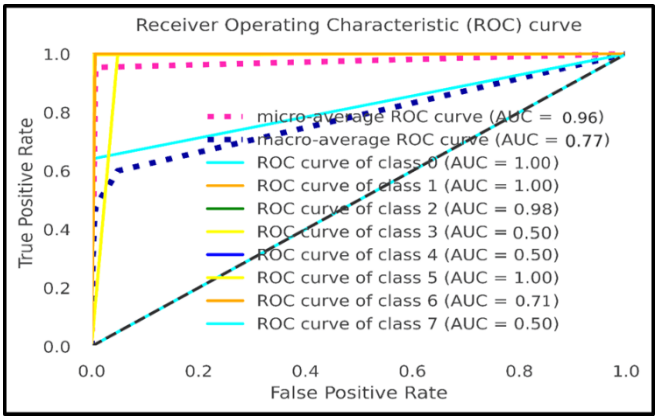
b



c

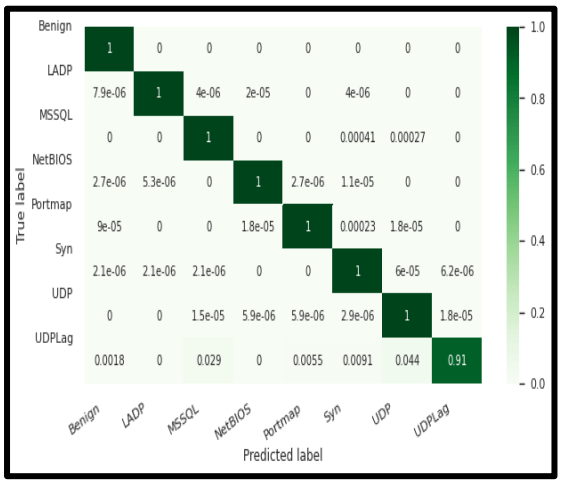


d

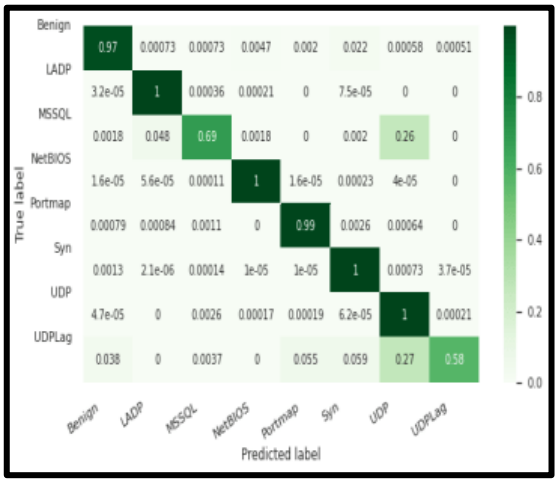


e

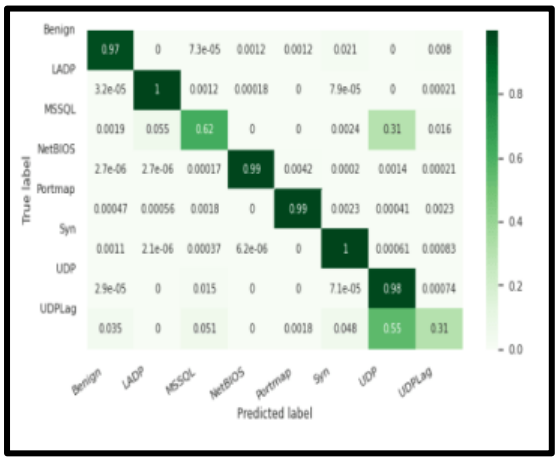
Figure 15. ROC Curve based on CIC-DDoS2019 dataset for (a) RF-RFE_SMOTE_RF (b) RF-RFE_SMOTE_LR (c) RF-RFE_SMOTE_LDA (d) RF-RFE_SMOTE_QDA (e) RF-RFE_SMOTE_NB



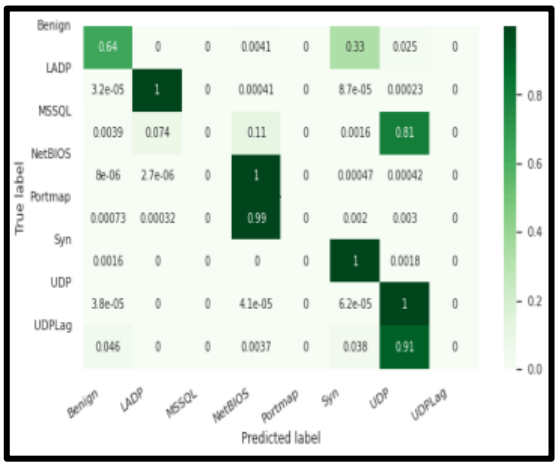
a



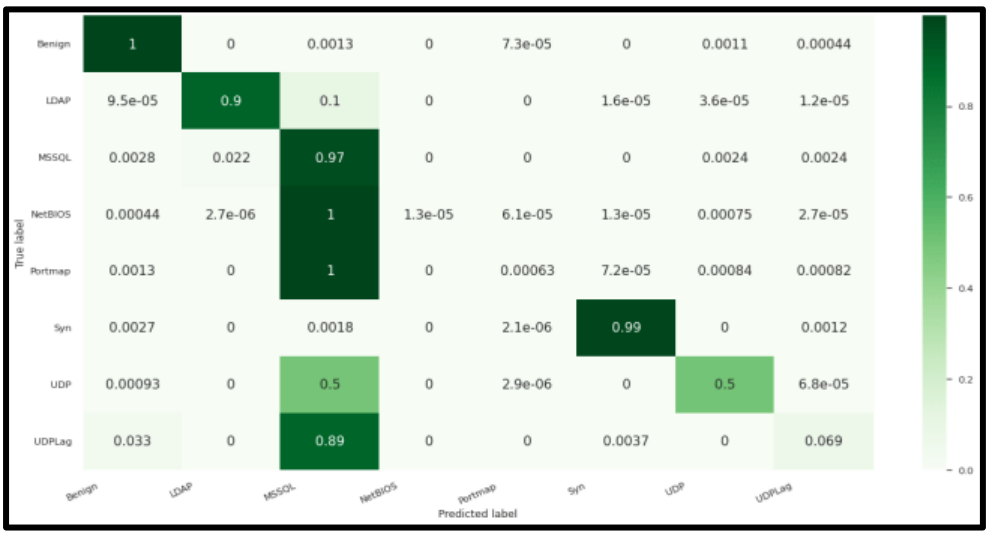
b



c



d



e

Figure 16. Confusion Matrix based on CIC-DDoS2019dataset for (a) RF-RFE_SMOTE_RF (b) RF-RFE_SMOTE_LR (c) RF-RFE_SMOTE_LDA (d) RF-RFE_SMOTE_NB (e) RF-RFE_SMOTE_QDA

5 Conclusion

DoS and DDoS attacks happen all the time on the Internet, and their number has grown exponentially over the past few years. Even though there are advanced and sophisticated ways to countermeasure these attacks, these attacks are still a severe issue in network security and a problem today.

The analysis and visualization using t-SNE for CSE-CIC-IDS2018 and CIC-DDoS2019 led to the following conclusions: the classes intertwined and were imbalanced in some classes. For this reason, the framework adopts SMOTE. Broadly translated, our findings indicate that combining RF-RFE with SMOTE can reduce feature dimensionality and reduces the impact of data imbalances. Based on this, the 39 best features of Group Set (39-RF-RFE) were selected from CSE-CIC-IDS2018, and the 40 best features of Group Set (40-RF-RFE) in CIC-DDoS2019. The present thesis findings confirmFurthermore, propose a machine-leaning-based framework to detect DoS and DDoS attacks.

The paper feeds these two group sets of features to the classification process of RF, LR, NB, LQA, and LDA as classifiers. The finding decreases the preprocessing time and complexity of the model and increases accuracy.

Results from the experiment show that RF-RFE_SMOTE_RF outperformed all other models by obtaining an accuracy of 100% for CSE-CIC-IDS2018 and 0.99% for CIC-DDoS2019.

Acknowledgments

The project presented in this article is supported by Dr. Razan Abdulhammed. The authors would like to acknowledge the infrastructure support provided by the computer networking lab, Dept. of computer engineering technology, NTU.

References

[1] U. Islam et al., "Detection of distributed denial of service (DDoS) attacks in IOT based monitoring system of banking sector using machine learning models," Sustainability, vol. 14, no. 14, p. 8374, 2022.

[2] F. Rustam, M. F. Mushtaq, A. Hamza, M. S. Farooq, A. D. Jurcut, and I. Ashraf, "Denial of Service Attack Classification Using Machine Learning with Multi-Features," Electronics, vol. 11, no. 22, p. 3817, 2022.

[3] "Hacktivism and DDOS Attacks Rise Dramatically in 2022," GovTech, Aug. 21, 2022.

[4] "Famous DDoS attacks | Biggest DDoS attacks | Cloudflare."

[5] M. Alkasassbeh, G. Al-Naymat, A. B. Hassanat, and M. Almseidin, "Detecting distributed denial of service attacks using data mining techniques," Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 1, 2016.

[6] V. Sharma, V. Verma, and A. Sharma, "Detection of DDoS attacks using machine learning in cloud computing," in International Conference on Advanced Informatics for Computing Research, 2019, pp. 260–273.

[7] W. Bhaya and M. EbadyManaa, "DDoS attack detection approach using an efficient cluster analysis in large data scale," in 2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT), 2017, pp. 168–173.

[8] A. A. Abdulrahman and M. K. Ibrahem, "Evaluation of DDoS Attacks Detection in a CICIDS2017 Dataset Based on Classification Algorithms," Iraqi J. Inf. Commun. Technol. IJICT, vol. 1, no. 3, 2018.

[9] N. M. Yungaicela-Naula, C. Vargas-Rosales, and J. A. Perez-Diaz, "SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning," IEEE Access, vol. 9, pp. 108495–108512, 2021.

[12] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," ICISSp, vol. 1, pp. 108–116, 2018.

[13] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE.," J. Mach. Learn. Res., vol. 9, no. 11, 2008.

[14] Y. Wu and A. Zhang, "Feature selection for classifying high-dimensional numerical data," in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., 2004, vol. 2, p. II–II.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

[16] S. Alla and S. K. Adari, Beginning anomaly detection using python-based deep learning. Springer, 2019 2019.

[17] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods,

systems and tools," Ieee Commun. Surv. Tutor., vol. 16, no. 1, pp. 303–336, 2013.

[18] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Commun. Surv. Tutor., vol. 18, no. 2, pp. 1153–1176, 2015.