Mathematical Statistician and Engineering Applications
ISSN: 2094-0343
2326-9865

HPDDoS: A Hyper Parameter Model for Detection of Multiclass DDoS Attacks

Ms. Vimal Gaur (Research Scholar),

Department of CSE, MM Engineering College, Maharishi Markandeshwar Deemed to be University, Mullana, Ambala, Haryana, India and Assistant Professor, Maharaja Surajmal Institute of Technology, Janakpuri, Delhi- 110 058) ORCID ID: 0000-0003-4097-1859

Dr. Rajneesh Kumar (Professor),

Department of CSE, MM Engineering College, Maharishi Markandeshwar Deemed to be University, Mullana, Ambala, Haryana, India ORCID ID: 0000-0002-8139-3533

Article Info
Page Number: 1444-1470
Publication Issue:
Vol. 71 No. 3s2 (2022)

Abstract
In the modern era, network attacks are increasing at a breakneck pace. However, providing proper security measures to mitigate the different attacks is still a significant challenge. Distributed denial of service attacks are most dangerous as they bring whole network down. Recent work has relied on black-box optimization with hyperopt and machine learning models for handling DDoS attacks. In this paper, a hyperparameter optimization model to detect DDoS attacks has been proposed to enhance the efficiency of DDoS detection in terms of accuracy and training time. We demonstrate the feasibility of this model by comparing the performance measures of machine learning algorithms with and without feature selection methods. The feature selection methods (Chi-Square, Extra Tree, ANOVA, and Mutual Information) have been applied for reducing features and machine learning algorithms (Random Forest, Decision Tree, XGBoost, KNN, SNN, and DNN) for classification. Results show that the Random Forest classifier combined with Mutual Information feature selection achieves 96.77% as maximum accuracy in 750.94 seconds of training time. Hyperparameter tuning of the random forest classifier raises the accuracy value to 96.81%. Later, with a 95% confidence level, the confidence intervals of the default and hyperparameter values of Random Forest were calculated to be 98.58% and 98.78%, respectively. The results show that the hyperparameter model shows a 2.01% improvement in accuracy with multiclass data.

Keywords: ANOVA, Chi-Square, DDoS, Extra Tree, Hyper Parameter Optimization, Mutual Information.

Article History
Article Received: 28 April 2022
Revised: 15 May 2022
Accepted: 20 June 2022
Publication: 21 July 2022

1. INTRODUCTION

With the growth of internet, network attacks are also evolving at a great pace. DDoS attack is the most common network attack as these attacks modifies, damages data and they are also called as active attacks. With the passage of time, attacker have come across new tools for performing these kinds of attacks. Nowadays, it has become easier for an attacker to compromise victim machine. Usually, DDoS attacks occur on IoT networks. This is due to lack of security mechanisms in IoT devices. IoT devices have limited resources and memory.

Several methods for resolution against DDoS attacks in IoT devices have been analyzed by researchers.

The fundamentals of security are different for every service [1]. DDoS attacks, which are the most common attack take advantage of the three-way handshake process in a TCP connection [2].DDoS attacks attack the server from multiple points and consume all the resources of the sever [3]. Thus, recent research aims at contemporary solutions for implementing machine learning algorithms to detect DDoS attacks [4]. With the enormous demands of data science and the increase of algorithms, researchers have been confronted with a decision about the algorithm's performance. Currently, choosing the best algorithms for a dataset is a significant challenge for researchers [5].

Furthermore, the machine learning model cannot learn effectively from data unless the hyper-parameters are tuned [6]. Therefore, researchers must focus on predicting whether optimization techniques will improve performance by comparing the performance of optimized models with default hyperparameter values [7]. Hyper-parameter tuning techniques usually produce a good set of values, but the computation cost is high. So it is practically impossible to try all the combinations of hyperparameters as parameter space keeps increasing. With the increase in the number of hyperparameters, the complexity of the model also increases [8]. Therefore, automatic estimations for tuning hyperparameters should be an alternative to the manual selection of hyperparameters. Aiming at the early detection of DDoS attacks, tuning of hyperparameters is done, which combines the Scikit-learn method and automated searching for hyper-parameters. Thus, this work aims at contemporary solutions that implement machine learning algorithms. We utilized feature selection approaches on the CICDDoS2019 dataset, which helps in reducing data inputs for DDoS detection. Some of the challenges existing researchers face for detecting DDoS attacks have been listed in the literature section. The majority of existing DDoS detection methods achieved good accuracy but with older datasets. These datasets have limited DDoS attack types.

A few works on recent datasets are also available, but they focused on the binary classification of data (0-Benign, 1-Attack) and did not consider the entire dataset. These methods lack the best features for the early detection of DDoS attacks. Hyperparameter Tuning was not part of any existing research. The difference between the confidence intervals of default and hyperparameter at a certain confidence level has not been considered so far. These issues motivate the consideration of the entire CICDDoS2019 dataset for experimentation purposes.

We propose a hyperparameter model for the detection of DDoS attacks using multiclass data. The significant contribution of our paper is as follows:

(1) For experimentation purposes, we have considered the CICDDoS2019 dataset (70% data for training and 30% data for testing). The results obtained are with a complete dataset and multiclass classification.

(2) A series of iterations with varying numbers of features have been performed using different feature selection algorithms (Chi-Square, Extra Tree, ANOVA, and Mutual Information).

(3) Thereafter, machine learning algorithms (Random Forest, Decision Tree, XGBoost, KNN, SNN and DNN) have been applied for classification

(4) The main objective was to find the best possible combination of machine learning algorithm and feature selection method. Random Forest with mutual Information performs best. Random Forest was found to give the highest accuracy of 96.77% with a 43.03% feature reduction ratio.

(5) Since RF outperforms other classifiers, so hyperparameter tuning of RF is done. The difference between default and hyperparameter values at 95% confidence level.

The key contribution of this paper includes finding the best classifier along with feature selection method and tuning hyperparameters to calculate the difference between default parameters and hyperparameters.

The structure of this paper is as follows: Section 2 presents a literature survey. This section describes the methodologies used by researchers on the CICDDoS2019 dataset. Section 3 describes the materials and methodology to be followed in this paper. Results and discussions have been discussed in Section 4. Finally, section 5 highlights the conclusion and future work.

2. RELATED WORK

This section explicitly describes the technicalities proposed by esteemed researchers for detecting Distributed Denial of Service attacks on IoT devices. Experimental results of several machine learning algorithms have been presented on the CICDDoS2019 dataset, along with their merits and demerits.

Gaur et al. [9] proposed a hybrid methodology for distributed denial of service attack detection. In this methodology, machine learning algorithms (Random Forest, Decision Tree, KNN, and XGBoost) have been applied to feature selection methods. As a result of using these algorithms, ANOVA with XGBoost achieves an 82.5% feature reduction rate with an accuracy of 98.34%.

Odumuyiwa et al. [10] proposed unsupervised learning algorithms: Autoencoder, Restricted Boltzmann Machine, K-means Clustering algorithm, and Expectation-Maximization Clustering Algorithm. These algorithms have worked for two DDoS attacks: SYN-Flood and UDP-Lag. Accuracy and Normalized Mutual Information are calculated when these algorithms are applied to CICDDoS2019. The accuracy rate is 89.45% and 86.17% for SYN-Flood and UDP-Lag, respectively. The Normalized Mutual Information value is 53.63% and 42.165% for SYN-Flood and UDP-Lag, respectively.

Abbas et al. [11] take into account the MIX dataset. This dataset handles two attacks (PORTMAP and LDAP). Random Forest has a 99.9764% accuracy and a 0% false alarm rate. Performance evaluation of the dataset has been done for benign data and different types of attack data.

Alamri et al. [12] proposed multiple classifiers for detecting DDoS attacks. These classifiers (Logistic Regression, Random Forest, and XGBoost) have been implemented as binary classes and multiclass on the CICDDoS2019 and NSL-KDD datasets. When implemented for

CICDDoS2019, XGBoost gives a maximum accuracy of 99.7% for binary type and 91.3% for multiclass classification. Also, it provides 100% precision for binary class and 93% for multiclass. In addition, XGBoost offers a close to zero value for the false-positive rate for both datasets.

Parfenov et al. [13] investigated network traffic to detect attacks using binary and multiclass classification approaches. The algorithms utilized are Gradient Boosting, AdaBoost, and CatBoost. CatBoost gives the highest precision at 97.1% with full features. After that, extra tree feature selection has been applied, and CatBoost achieves 97% precision with 25 features. The Feature Importance Score has also been shown graphically for the top 25 features. AdaBoost shows the worst results for all the parameters.

Shurman et al. [14] proposed methodology results in 99.85% training accuracy and 99.19% testing accuracy. This methodology is for detecting reflection-based attacks. Furthermore, a hybrid intrusion detection system and a long-term, short-term memory have been proposed. Later, DNN models were implemented based on these methodologies. Rahman et al. [15] used different machine learning (ML) approaches such as Logistic Regression (LR), Decision Trees, and Support Vector Machines (SVMs) in their system for detection of DDoS attacks from benign attempts. For example, SVM gives 97.1% accuracy, and false positive and false negative values are rare. Steve Chesney et al. [16] assigned labels to benign, LDAP, and NetBIOS attacks for intrusion detection on IoT devices. Later, the Linear Regression algorithm was deployed for Multiclass Classification. After successive iterations, the F1 Score comes out to be 0.73, 0.99, and 1.00 for Benign, LDAP, and NetBIOS, respectively.

Mohamed Amine Ferrag et al. [17] worked on CNN, RNN and DNN for binary and multiclass data. These models have been trained and tested on Agriculture 4.0 and IoT parameters for DDOS attack prevention and identification. For implementation purposes, two datasets (CICDDoS2019 and TON_IoT) have been taken. CNN gives the best results with an accuracy of approximately 90% on multiclass and close to 99% on binary classification for the CICDDOS2019 dataset while it is closer to 95% for TON_IoT.

Sanchez et al. [18] performed exhaustive hyperparameter optimization and achieved 99% accuracy for the random forest with the binary classification of data. Hyperparameters have been listed for NB, LR, KNN, RF, DT, MLP, and SVM. In addition, exhaustive tuning operations have been performed using grid search.

Elsayed et al. [19] proposed a combination of RNN and an autoencoder for binary classification. The autoencoder helps in anomaly detection of the CICDDOS2019 dataset. The deep learning model so formed has implemented Naive Bayes, Decision Trees, Boosters, Random Forest, SVM, and Linear Regression algorithms. The proposed model, DDoSNet, turns out to be the best, with 99% accuracy.

Assis et al. [20] evaluated the efficiency of different detection techniques when SDNs have been applied. Further, author deployed machine learning algorithms (GRU, CNN, LSTM, DNN, SVM, LR, KNN, and GD) for anomaly detection scheme on flow analysis. Additionally,

feasibility tests have been done, and the gated recurrent unit turns out to be the best with all the parameters near 100%.

Manikumar et al. [21] implemented three machine learning algorithms (KNN, Decision Tree, and Random Forest) for finding the best classification. The top 15 features have been chosen using an additional tree classifier. The Random Forest algorithm gives the highest value of all the parameters, accuracy = 95.19%, precision 95.10%, and recall 94.47%.

Li et al. [22] proposed a new parameter (temporal false omission rate) for finding attack packets on two datasets (1999 DARPA Intrusion detection Evaluation Dataset and CICDDoS2019) result. The rate of omitted positive samples in real-time volumetric data has been calculated using these temporal variables.

Jia et al. [23] gathered a large dataset of DDoS simulators and combined it with the CICDDoS2019 dataset. Later, a model was prepared to guard the flow. Afterward, features have been selected using a filtration procedure. Once selected features are obtained, basic machine learning algorithms, viz., LR, NB, RF, ID3, LSTM, have been applied for classification. LSTM achieves a maximum value of 99% for precision, recall, and F1 Score.

Sharafaldin et al. [24] proposed a real-time CICDDoS2019 Intrusion Detection Evaluation Dataset. This dataset has been gathered over two days and includes DDoS attacks (UDP, UDP Lag, NetBIOS, SYN, PortMap, and MSSQL). The author has applied machine learning algorithms to training and testing day attack types. The majority of these attacks are reflexive. Again, ID3 performs best, with 78% accuracy.

Vuongl et al. [25] introduced a DDoS Intrusion Detection system. Machine learning algorithms when applied to Intrusion detection systems increases accuracy and decreases the false positive rate of the model for identification of new attacks. The data labels used in the training process are highly imbalanced e.g., UDP-Lag and LDAP attacks are sporadic.

Table 1 focuses on proposed methodologies for the CICDDoS2019 dataset by eminent researchers.

Table 1. Proposed Methodologies on CICDDoS2019 Evaluation Dataset

S. no.	Author	Year	Machine Learning Algorithms	Performance Parameters	Scope of Work
1.	Gaur et al. [9]	2021	Random Forest, Decision Tree, XGBoost, SNN, DNN	98.34% accuracy for ANOVA with XGBoost. We have applied three feature selection methods.	Further accuracy can be improved with more feature selection methods.

2.	Odumuyiwa et al. [10]	2021	Unsupervised Machine learning Algorithms	Autoencoder	89.45%	86.17%	This approach didn't use an entire dataset, the first result is for SYN_Flood, and the second is for UDP_Lag.
				Restricted Boltzman machine	56.51%	50.89%	
				K-means Clustering Algorithm	75.38%	71.39%	
				Expectation-Maximization Clustering Algorithm	70.96%	67.59%	
3.	Abbas et al. [11]	2020	PCA is used for Pre-processing. MIX dataset (PORTMAP, LDAP) is used by Random Forest.	Random Forest gives 99.976% accuracy.			Data has been used in partial mode.
4.	Alamri et al. [12]	2021	LR, RF, XGBoost	Accuracy with Binary class LR= 80% RF=98.5% XGBoost=99.7% Accuracy with Multiclass LR= 35% RF=83% XGBoost=91.3%			This approach results in less accuracy value for Multiclass. The maximum value achieved is 91.3% for XGBoost.
5.	Parfenov		Gradient Boosting, AdaBoost,	The precision with full features			Gradient boost Achieves maximum precision value, but on the

	et al. [13]	2020	CatBoost implemented for binary and multiclass. Extra Tree Feature Selection has also been applied.	Gradient Boosting=97.1%, AdaBoost=61.4%, CatBoost=97.1% With Extra Tree Feature Selection Gradient Boosting=97%, AdaBoost=62.3%, CatBoost=96.7% These results are for 25 features.			application of extra tree feature selection, this precision deteriorates.
6.	Shurman et al. [14]	2020	LSTM Model (3 Variants)		Train Accuracy	Test Accuracy	These results are with full features.
				Model I	92.05%	91.54 %	
				Model II	97.27%	96.74 %	
				Model III	99.85%	99.19 %	
7.	Rahman et al. [15]	2020	Three machine Learning Algorithms: LR, DT, SVM.	SVM achieves the highest accuracy as 97.1%			A complete feature set has been used.
8.	Chesney et al. [16]	2021	Logistic Regression	Logistic regression gives an accuracy of 99.70%			The complete dataset has not been chosen for implementation. (Logistic Regression has been applied on LDAP file).
9.	Ferrag et al. [17]	2021	CNN	Binary Class- CNN = 99% Multi-Class- CNN = 90%			This approach gives less accuracy for Multiclass.
10.	Sanchez	2021	RF is used	RF gives an accuracy of 99%			This dataset is used for binary classification.

	et al. [18]				Hyperparameter Tuning is done using GridSearch.
11.	Elsayed et al. [19]	2020	RNN with Autoencoder	The proposed model DDoSNet turns out to be best with 99% accuracy	This result is for binary classification.
12.	Assis et al. [20]	2021	Gated Recurrent Units (GRU) deep learning method, CNN, LSTM, DNN, SVM, LR, KNN, and GD	GRU achieves accuracy closer to 100%	GRU is not used as a multi-label classifier.
13.	Manikumar et al. [21]	2020	Extra Tree-Based Classifier, Three Machine learning Algorithms (KNN, DT,RF)	KNN=87.34%, DT=93.83%, RF=95.19%. Random Forest gives maximum accuracy.	We have achieved 1.55% more accuracy for RF with an Extra tree classifier.
14.	LI et al. [22]	2020	Introduced a new variable for calculating Temporal False Omission Rate (TFOR)	Average Temporal False Omission Rate = 0.3447% and True positive rate is 100% and FPR is 3%.	Results are obtained with full features.

15.	Jia et al. [23]	2020	LSTM, CNN Model	LSTM Accuracy= 98.9% CNN Accuracy=99.9%	A complete feature set has been used for these results.
16.	Sharafaldin et al. [24]	2019	Machine Learning Algorithms (ID3, RF, NB, LR)	ID3 gives 78% Precision Value.	Values have been obtained with full features.
17.	Vuong l et al. [25]	2021	Random Forest Regressor has been applied for selecting 24 features	The proposed method gives 99.3% precision with a grouping of labels.	Recall, Precision and F1 Score have been calculated for individual attack types. We have calculated these values after combining all the attack types.

3. Material and Methodology

A. Hardware Specification

The model has been trained on Google Collab and has 13 GB of RAM with GPU support. In addition, the personal computer has the following specifications:

The i5-8265U processor has a clock speed of 1.60GHz and 8 cores.

RAM: 24 GB

Disk: 512GB SSD from Samsung

Intel UHD Graphics 620 is the GPU.

System environment: Windows 10.

B. Dataset

The Canadian Institute for Cybersecurity generates a new dataset, CICDDoS2019, which provides the most important distributed denial of service attacks. The dataset chosen for experimentation consists of two day log records from January 12, 2019, which started at 11:22 and ended at 13:34, and on March 11, 2019, beginning at 10:00 and completing the acquisition at 17:35. The dataset to be evaluated has 88 aggregated columns and contains benign and recent DDoS attacks viz. UDP, UDP-Lag, SYN, NetBIOS, LDAP, MSSQL, NTP, SMTP, NDP, SSDP, TFTP. On the other hand, the testing dataset includes attack categories UDP, UDP-Lag, SYN, NetBIOS, LDAP, and MSSQL which clearly resembles real-world data. For constructing this dataset, the author built the abstract behavior of 25 users on different protocols.

Intrusion detection will be more pragmatic as the testing and training data will contain different data. Since, unique attributes (Source IP, Destination IP, Source Port Number, Destination Port Number, FlowID, and Protocol) don't contribute to the detection process so they have been eliminated.

C. Data Acquisition and Data Preprocessing

In this section, data gets transformed or encoded to bring it to a state where the machine can quickly parse it. In other words, the features or independent variables are easily interpretable by the algorithm.

Steps to be followed in Data Preprocessing:

Importing the libraries: We will begin by importing the required libraries to be used by the dataset. e.g., pandas, numpy, matplotlib, sklearn, scatterplot, seaborn, etc. Then, the following steps are followed for preprocessing data.

Hypothesis Generation involves a deep understanding of the problem to consider the factors affecting the outcome.

Understanding the Data: It indicates that the number of features and data types should be known for training and testing data.

Exploratory Data Analysis (EDA): Exploratory data analysis is done to discover patterns, spot anomalies, test hypotheses, and check assumptions with the help of summary statistics and graphical representations.

Unified Analysis: It includes examining each variable individually. For example, for categorical features, we can use frequency tables or bar plots to calculate the number of each category in a particular variable. For numerical features, probability density plots are used to look at the distribution of the variable.

Treatment for Missing Value: Since model performance greatly depends upon missing values and outliers, there must be mechanisms for inputting and handling outliers.

Encoding Categorical Data: Encoding categorical data is converting categorical data into integer format so that these converted values can be provided to the models to improve the predictions. e.g., One-Hot Encoder.

Feature Scaling: Feature scaling standardizes independent features in the dataset and handles highly variable magnitudes. The following procedure can explain these steps more clearly: These steps have been defined in Algorithm 1. Initially, we need to import training and testing datasets onto an excel sheet. We implemented our methodology using the CICDDoS2019 dataset consisting of different DDoS attacks. We have enumerated the preprocessing steps in algorithm 1.

Algorithm 1: Preprocessing Process

Input: CSV Dataset

Begin

1: Read CSV Dataset

2: Load dataset into a dataframe(df)

3: Check Dataset Description.

4: Drop SimillarHTTP feature from df.

5: Replace all infinity values with NaN.

6: Replace all NaN with mean values.

7: Check for NaN values.

8: If Chi-Square then:

9: Copy all the features from the original data frame (df) except the target feature to another data frame (df1).

10: Check for all the features that have negative values.

11: Replace all the negative values in each feature with their absolute values.

12: Else

13: Map each DDoS attack to a unique integer value.

14: End

15: Segregate dependent and independent features.

X_val=Independent Features

y_val=Dependent Features

Exit

D. Machine Learning Algorithms

This section discusses how machine learning has evolved as an adaptive method for making decisions. Decisions making becomes more manageable when machines adapt to example inputs rather than following rigorous program instructions.

A brief introduction of the machine learning algorithms has been given below:

(i) Random Forest

Random Forest is a group of multiple working decision trees and they all are trained using the bagging method. It examines the best feature among a random subset of features rather than looking for the best feature during node splitting. This method results in heterogeneity, as the

resultant output gives a more accurate model [26]. The parameters used in random forest are -: number of eastimators, minimum sample leaves, minimum sample split, Gini Criterion.

(ii) Decision Tree

Decision Tree model is extensively put into operation in regression and classification-related issues. Every node in a decision tree represents a feature; each link represents a decision; finally, leaves display the outcome [27]. In comparison to other algorithms, this is simple to interpret. The steps involved in implementing a decision tree are:

To recognize the prime feature and place it at the root of the tree.

Split the dataset into the training set and testing set.

Repeat the above steps until there is a leaf node on each tree branch.

Finally, to get the value of a class variable with the help of a decision tree algorithm, we will begin the process from the tree root and compare the values with that of the instances attribute. Based on the outcomes after comparison, we decided to pick a branch and proceed to the next node of the tree. These steps will continue to iterate until the leaf node reaches the predicted value of a class variable.

(iii) XGBoost

XGBoost is a gradient boosting algorithm to improve efficiency, flexibility, and portability. It also provides a parallel tree boost, which helps solve multiple data-related problems at a higher speed [28].

(iv) KNN (K-nearest neighbors)

This is a supervised machine learning algorithm which initially stores all the available data and classifies every data point as soon as it arrives. Each time a new data point arrives, it is classified based upon the category to which it matches. K is used for measuring distance to nearest neighbors and calculated using Euclidean distance.

(iv) SNN

This neural network algorithm has multiple hidden layers and works on the principle of forward-propagation methodology [30]. The parameters used in SNN have been described below.

Layers->1 Dense layer

-Neurons->7 Neurons

-Input shape changes with each iteration (depends on the value of j).

-activation function->sigmoid

Optimizer-> Adam

-loss->loss='sparse_categorical_crossentropy'

-Metics->Accuracy

-epochs->5

(v) DNN

Deep neural networks work on the principle of feedforward neural networks (FFNNs), where data moves from one input layer to the following output layer in the forward direction. The links between layers are always forward, and every node is traversed only once. A supervised learning algorithm can obtain the results through backpropagation. The parameters used have been listed below.-:

Layers->4 Dense layer

1st Layer (Input Layer)->

- 75 Neuron
- Input shape->changes with each iteration (depends on the value of j)
- activation function-> Relu

2nd Layer->

- 50 Neurons
- activation function-> Relu

3rd Layer->

- 25 Neurons
- activation function-> Relu

4th Layer(output Layer)->

- 7 Neurons
- activation function-> Sigmoid

-Optimizer->adam

-loss->loss='sparse_categorical_crossentropy'

-Metics->Accuracy

-epochs->8

E. DDoS Attack Detection Methodology

The primitive research objective behind the proposed model is to design a machine learning model based on feature selection methods and perform hyperparameter tuning. The fundamental idea behind this model is to study the possibility of applying machine learning

algorithms to the CICDDoS2019 dataset for detecting DDoS attacks. As an initial step, the modeling process begins by using a feature selection algorithm to identify essential features since these attributes are considered valuable deliverables for the detection process. Then, hyperparamerization is done for the best classifier. Features thus selected have been considered for hyperparameter optimization, and the behavior of retained attributes has been studied and analyzed by plots. As a result, Hyperparameter optimization has effectively tackled DDoS attacks. Figure 1 summarizes the hyperparameter optimization. Next, data acquisition is made from the CICDDoS2019 dataset to detect attacks. After that, a machine learning model has been trained on training data, and testing is performed using testing data. Selecting an appropriate model is critical as it requires relevant data.

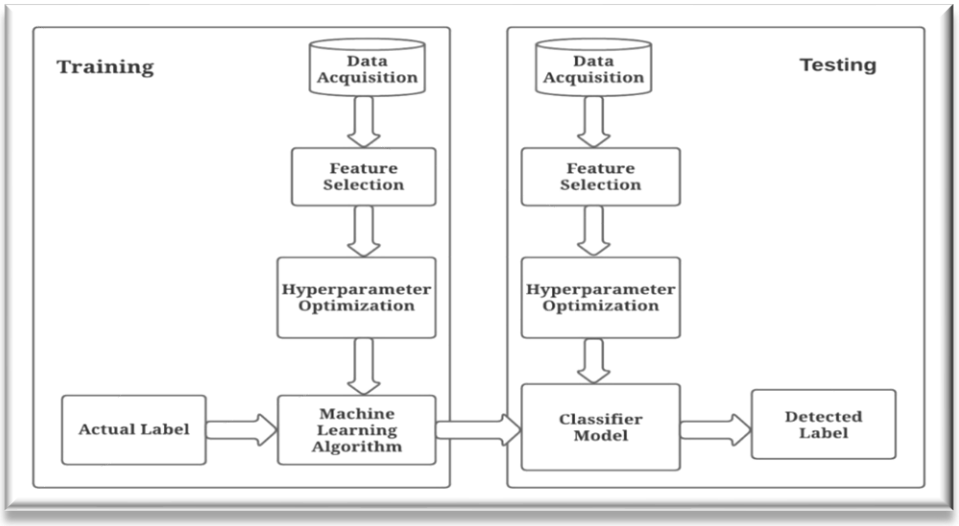

Fig. 1. HPDDOS: Hyperparameter Model for Detection of Multiclass DDoS Attacks

F. Multiclass DDoS Attacks Detection

The procedure for detection of multiclass DDoS attacks can be explained more clearly using the following algorithm.

Algorithm 2: **Multiclass DDoS Attacks Detection**

Input: Segregated dependent and independent features

X_val=Independent Features

y_val=Dependent Features

Initialization: j=0

Begin

1: While j<=75 do

2: Apply train-test split with 70-30 split assign the results to X_train, X_test, Y_train, Y_test

3: Select top j features from X_train and X_test and assign the results to X_train_fs and X_test_fs

4: Apply the Standard Scaler on X_train_fs and X_test_fs

5: Load the classifier in a variable (called model)

6: Save current time in a variable (called t1)

7: Fit X_train_fs and Y_train in the model

8: Save current time in a variable (called t2)

9: Get the training time by subtracting t2 and t1 (t2-t1)

10: Save the current model using jblib

11: Calculate the predictions on X_test_fs and save the results in a variable (called pred)

12: Calculate accuracy on y_test and pred

13: Display the accuracy

14: Display the classification report

15: Calculate True Positive Rate, True Negative Rate, False Positive Rate, and False Negative Rate and display them

16: Increment j by 5

17: End

Exit

Algorithm 2 illustrates the process of detection of Multiclass DDoS attacks. The train-test split is applied to the CICDDoS2019 dataset. 70% of the data is used for training, and 30% is used for testing.

4. Results and Discussions

We analyzed a cloud-based environment called Google Colab (an online cloud-based jupyter notebook environment). Implementation has been done on the CICDDoS2019 dataset [1].

A. Performance Evaluation

Since a hybrid methodology has been proposed in figure 1, we will focus on how features have been selected using feature selection and classifiers have been used for classification purposes.

Scenario I Classifiers Performance

In the first scenario, all the feature selection methods [31-35] have been applied to machine learning algorithms. This has been done to monitor the performance of classification algorithms, which is actually an indicator of detection of DDoS attacks.

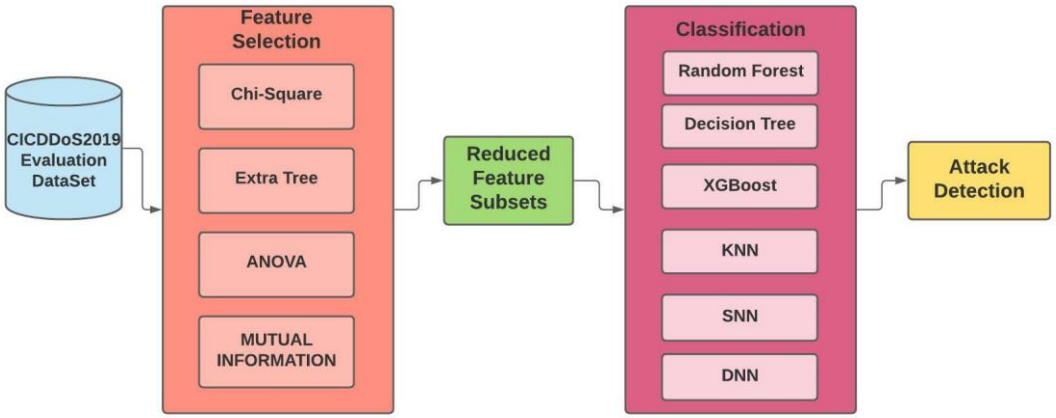

Fig. 2. Attack Detection using Feature Selection and Machine Learning Algorithms

Table 2. Performance of Classifiers without feature selection method on CICDDoS2019 Dataset

Machine Learning Algorithms	A	P	R	F	TP	TN	FP	FN
XGBoost	0.9053	0.96	0.96	0.96	0.85	0.99	0	0.14
DT	0.9041	0.93	0.91	0.92	0.84	0.98	0.01	0.15
RF	0.9073	0.97	0.97	0.96	0.86	0.99	0	0.13
KNN	0.9045	0.98	0.97	0.96	0.86	0.97	0.01	0.25
SNN	0.8514	0.02	0.79	0.80	0.56	0.96	0.02	0.43
DNN	0.8912	0.93	0.91	0.92	0.84	0.98	0.01	0.15

Performance has been evaluated using Accuracy (A), Precision (P), Recall (R), F-score (F), TPR (TP), TNR (TN), FPR (FP), FNR (FN). In addition, the number of features giving the best results has been considered, as shown in tables 3-6 below. As shown in the table, maximum accuracy of 90.73% is achieved for the random forest with no feature selection. Further iterations have been performed, and it has been noticed that random forest acquires maximum accuracy with all feature selection algorithms. Table 4-7 shows the different features on which maximum value is obtained. Random Forest gives the maximum value of performance parameters. Both XGBoost and Random Forest are independent decision trees, but they differ in tree construction. XGBoost take more training time as they build one tree at a time. This is why it is considered harder to tune them than random forest. This paper will focus on the hyper tuning of random forest, as this classifier works well for large and multiclass data.

Table 3. Performance Parameters with Chi-Square feature selection (55 Features)

Machine Learning Algorithms	A	P	R	F	TP	TN	FP	FN
XGBoost	0.9666	0.97	0.97	0.96	0.85	0.99	0	0.14
DT	0.9115	0.93	0.91	0.92	0.83	0.98	0.01	0.16
RF	0.9674	0.97	0.97	0.96	0.86	0.99	0	0.13
KNN	0.9041	0.98	0.90	0.97	0.82	0.94	0.03	0.42
SNN	0.9458	0.90	0.32	0.44	0.32	0.90	0.09	0.67
DNN	0.9621	0.95	0.87	0.87	0.53	0.97	0.02	0.46

Table 4. Performance Parameters with Extra Tree Feature Selection (35 Features)

Machine Learning Algorithms	A	P	R	F	TP	TN	FP	FN
XGBoost	0.9664	0.97	0.97	0.96	0.85	0.99	0	0.14
DT	0.8955	0.93	0.91	0.91	0.84	0.98	0.01	0.15
RF	0.9674	0.97	0.97	0.96	0.86	0.99	0	0.13
KNN	0.9144	0.91	0.97	0.96	0.84	0.94	0.05	0.24
SNN	0.9519	0.90	0.79	0.80	0.57	0.96	0.03	0.42
DNN	0.9623	0.94	0.87	0.88	0.56	0.97	0.02	0.43

Table 5. Performance Parameters with ANOVA Feature Selection (55 Features)

Machine Learning Algorithms	A	P	R	F	TP	TN	FP	FN
XGBoost	0.9666	0.97	0.97	0.96	0.85	0.99	0	0.14
DT	0.9027	0.93	0.90	0.91	0.83	0.98	0.01	0.16
RF	0.9674	0.97	0.97	0.96	0.86	0.99	0	0.13
KNN	0.9144	0.91	0.97	0.96	0.84	0.94	0.05	0.24
SNN	0.9519	0.90	0.79	0.80	0.57	0.96	0.03	0.42
DNN	0.9623	0.94	0.87	0.88	0.56	0.97	0.02	0.43

Table 6. Performance Parameters with Mutual Information Feature Selection (45 Features)

Machine Learning Algorithms	A	P	R	F	TP	TN	FP	FN
XGBoost	0.9667	0.96	0.96	0.96	0.85	0.99	0	0.14
DT	0.9129	0.93	0.91	0.92	0.841	0.981	0.01	0.15
RF	0.9677	0.97	0.97	0.96	0.86	0.99	0	0.13
KNN	0.9042	0.91	0.98	0.97	0.86	0.97	0.02	0.23
SNN	0.9514	0.02	0.79	0.80	0.56	0.96	0.02	0.43
DNN	0.9129	0.93	0.91	0.92	0.84	0.98	0.01	0.15

Since the random forest gives maximum accuracy with 45 features. So, we will evaluate the accuracy of random forest with all feature selection algorithms. On performing iterations, we found that Random Forest achieves maximum accuracy of 96.74% with Chi-Square Test, Extra Tree classifier, and ANOVA on 55, 35 and 55 features, respectively [36].

Further random forest with mutual information takes minimum training time. Further, when Mutual Information feature selection is applied, Random Forest attains the highest accuracy as 96.77% with 45 features. Thus, the number of features differentiates these feature selection algorithms. Performance has been evaluated using Accuracy (A), Precision (P), Recall (R), F-score (F), TPR (TP), TNR (TN), FPR (FP), FNR (FN). In addition, the number of features giving the best results has been considered, as shown in tables 3-7 below. Hence, Figure 3 depicts the accuracy of Random Forest with a varying number of features, and Figure 4 illustrates the training time of Random Forest with the number of features. Random Forest with Mutual Information outperforms other classifiers, presenting an accuracy of 96.77%, a false positive rate of 0, and a true negative rate of 99% with 45 features. Furthermore, mutual information sets the minimum training time as 750.94 seconds. Hence, this methodology helps in the early detection of DDoS attacks [37].

Table 7. Performance Parameters of Random Forest Classifier with different feature selection methods (55 Features)

Feature Selection Algorithms	Number of features	Accuracy	Feature Reduction Ratio (%)	Training Time (in Seconds)
Chi-Square	55	96.74	30.38	756.06
Extra Tree	35	96.74	56.70	773.35
ANOVA	55	96.74	30.38	785.58
Mutual Information	45	96.77	43.04	750.94

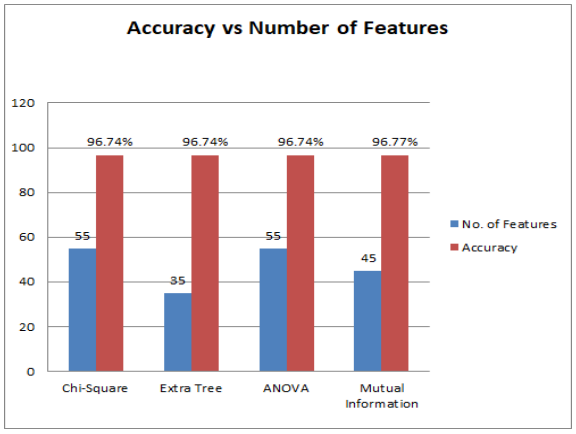

Fig. 3. Accuracy of Random Forest Classifier with

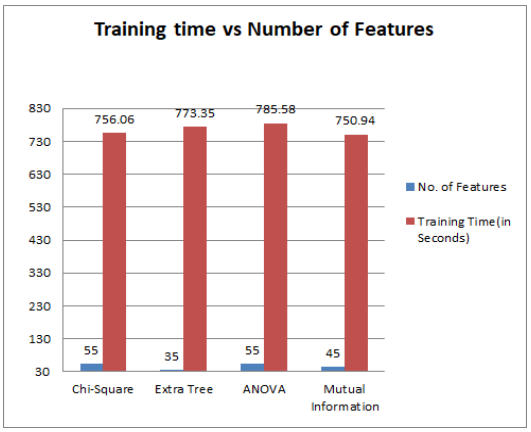

Fig. 4.Training Time of Random Forest Classifier with different Feature Selection Methods feature selection methods

Scenario II Hyperparameter Tuning

Since hyperparameter tuning can accelerate performance, in this paper we will tune parameters of random forest. While tuning hyperparameters of random forest, max_depth, min_sample_split, min_samples_leaf, n_estimators, and max_features have been optimized to increase performance parameters. To obtain the optimal parameters, we used RandomizedSearchCV and Optuna. RandomizedSearchCV is provided by scikit-learn API, and Optuna is an open-source hyperparameter optimization framework. These two methodologies are used for automating hyperparameters. Table 8 compares default and hyperparameters of Random Forest. This comparison tends to work for large spaces, and training them also requires a significant amount of time.

Table 8. Random Forest Classifier (a) Default Hyperparameter value (b) Hyperparameter with RandomSearchCV (c) Hyperparameter with Optuna

S.No	Name of parameter	Description of parameter	Hyper Parameter		
			Default Value	RandomSearchCV Value	Optuna Value
1.	max_depth	Longest Path from the root node to leaf node.	None	20	22.053
2.	min_sample_split	Minimum required several observations in any given node to split it.	2	10	10
3.	max_terminal_nodes	It sets a condition on the splitting of the node in the tree.	None	None	None
4.	min_samples_leaf	After splitting a node, the minimum number of samples should be present in the leaf node.	1	4	2
5.	n_estimators	Several trees are required for the random forest.	100	200	250
6.	max_features	The maximum number of features in every tree.	Auto	Sqrt	Sqrt

After training the Random Forest Classifier model with RandomSearchCV, the accuracy has improved by 0.04%, giving an accuracy of 96.81%.

RandomizedSearchCV helps in reducing the search space for obtaining the optimal parameters when we have a long list of parameters. Further, tuning with Optuna leads to an accuracy of 96.82% (improvement by 0.05%). The optimal parameters values obtained after applying RandomizedSearchCV and Optuna have been listed in table 8.

B. CONFIDENCE INTERVAL

Confidence intervals measure the level of uncertainty or certainty in a sampling technique. They can choose from various probability bounds, the most frequent of which are a 95% or 99% confidence level. The higher the confidence level value, the wider the confidence interval. So 99% confidence level has a wider confidence interval than 95%. Confidence intervals reveal a region where the actual value is likely to be found and the outcome variable's orientation and intensity. Besides, confidence interval allows judgments for statistical validity and clinical implications. This paper calculates the confidence interval for default and hyperparameter values of the Random Forest classifier.

Let AUC denote the sample AUC value. For large samples, the distribution of AUC is approximately normal. The AUC confidence interval calculated using standard normal distribution is as below:

$AUC \pm z\alpha/2(AUC)$

The confidence interval has a width of 2z/2SE (AUC). Hanley and McNeil (1982) provided the formula for standard error calculation for the area under the curve as SE (AUC).

$$\mathrm{SE(AUC)} = \sqrt{\frac{\mathrm{AUC(1-AUC)+(N_1-1)(Q_1-AUC^2)+(N_2-1)(Q_2-AUC^2)}}{\mathrm{N_1N_2}}}$$

Where,

$$\mathrm{Q_1} = \frac{\mathrm{AUC}}{2 - \mathrm{AUC}}$$

$$\mathrm{Q_2} = \frac{2\mathrm{AUC}^2}{1 + \mathrm{AUC}}$$

Where N1, N2 are Sample Size.

Q1 is the position of the first Quartile.

Q2 is the position of the second Quartile (Median).

AUC is Area under the ROC Curve.

Confidence Level =95%

Z- Score = 1.96.

Confidence Interval has been calculated in table 9.

Table 9. Average Confidence Interval of Random Forest Classifier

Parameters	N1	N2	AUC	Q1	Q2	SE (AUC)	Lower Bound	Upper Bound	Average Confidence Interval
Default Parameters	1180987	506138	0.9859	0.000193281509797	0.0069012539716	0.000078901916021	0.9857	0.9860	0.9858
HyperParameters	1180987	506138	0.9879	0.000142909230	0.0059401529	0.000072887237	0.9877	0.9880	0.9878

There is a 95% chance that the confidence interval of [98.57, 98.60] contains the true accuracy of finding DDoS attacks or a 5% chance that the accuracy is less than 98.57% or greater than 98.78%. The above-calculated results can be more clearly explained by the following figure 7.

This figure shows that, with a 95% confidence level, the random forest gives 98.78% accuracy. After analysis of results, the proposed model attains a maximum accuracy of 98.78% on hyperparameter tuning of random forest classifiers. Fig. 7 shows the visualizations of Random Forest with different parameters.

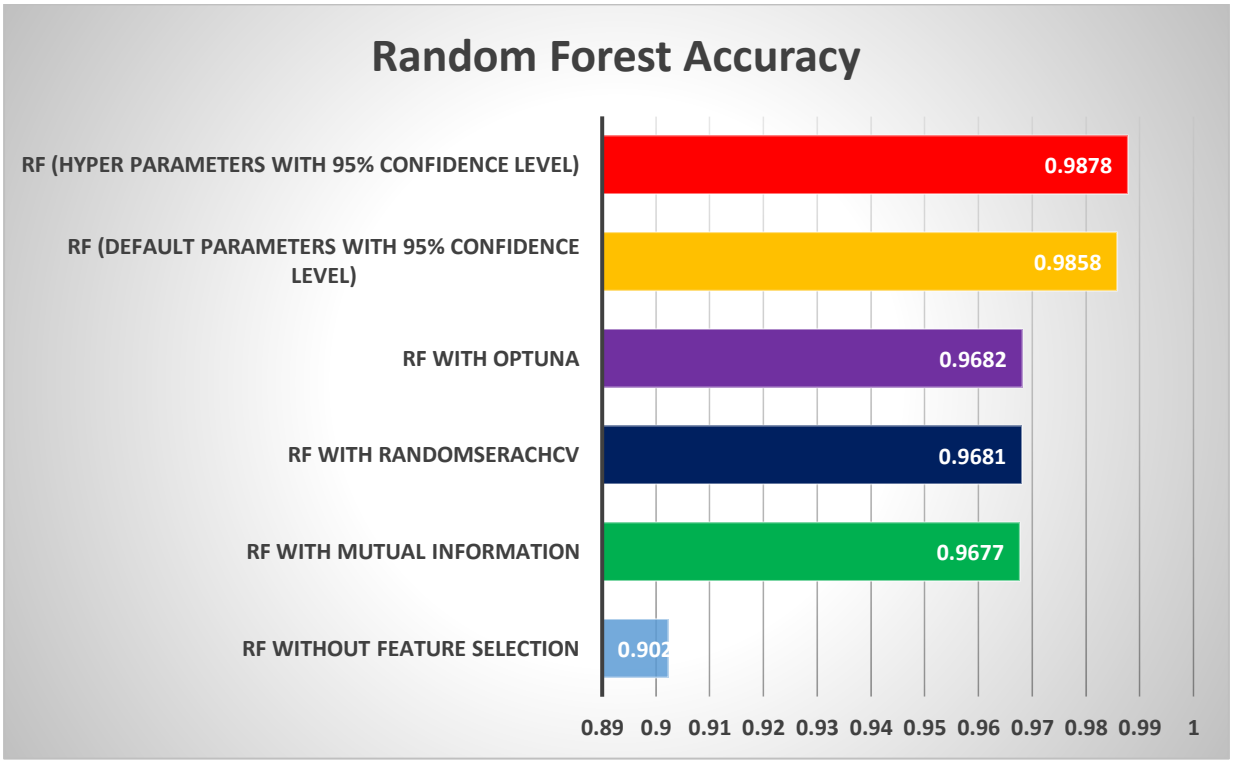

Fig. 7.Random Forest Accuracy Visualizations

Table 10. Comparison with other state-of-the-art methods on CICDDoS2019 Evaluation Dataset

Study	Year	Feature Selection	Machine Learning Classifier	Classification	Performance Parameters (Accuracy)
Gaur et al. [9]	2021	Chi-Square, Extra Tree, ANOVA	Random Forest, DT, KNN, XGBoost	Binary	XGBoost + ANOVA Accuracy = 98.34%
S. A. Abbas et al. [11]	2021	No	Random Forest	Binary	Random Forest = 99.976% Partial dataset (PORTMAP, LDAP)
H.A. Alamri et al. [12]	2021	No	LR, RF and XGBoost	Binary and Multiclass	XGBoost = 99.7% (Binary) XGBoost = 91.3% (Multiclass)
Md A. Rahman et al. [15]	2020	No	LR, DT, SVM	Binary	SVM = 97.1%
Steve Chesney et al. [16]	2021	No	LR	Binary	LR = 99.70% Partial Dataset (only LDAP file)
Mohamed Amine Ferrag et al. [17]	2021	No	CNN	Binary and Multiclass	CNN = 99%(Binary) CNN = 90%(Multiclass)
O.R. Sanchez et al. [18]	2021	No	RF	Binary	RF = 99%
M.S Elsayed et al. [19]	2020	No	RNN with Autoencoder	Binary	Proposed Model = 99%

D.V.V.S. Manikumar et al. [21]	2020	Extra Tree-Based Classifier	KNN, DT and RF	Binary	RF = 95.19% (without feature selection) RF = 96.74% (with extra tree)

Table 10 depicts the comparison of the proposed model with other state-of-the-art methods.

5. Conclusion and Future Scope

Accurate detection of Multiclass DDoS attacks is very essential to protect networks. However, due to the lack of availability of intrusion detection systems and real-time data, there are significant hindrances to the detection of DDoS attacks. This paper proposes a hyperparameter model for the detection of Multiclass DDoS attacks. The model focuses on the detection of DDoS attacks with considerably less features and good performance parameters. This has been accomplished by performing hyperparameter tuning on the latest dataset. Experimental results show that this work has a higher detection accuracy, minimum training time, and more excellent feature reduction ratio of 98.78%, 750.94 seconds, and 43.04%, respectively. We compared this model with several state-of-the-art methods on the CICDDoS2019 dataset. As a result, a 2.01% increase in accuracy has been achieved on hypertuning parameters. Further, this work can be extended using different deep learning models.

References

1. Sharafaldin, I., Lashkari, A.H., Hakak, S. and Ghorbani, A.A.: Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In: 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, pp. 1-8, 1-3 October2019, Chennai, India: IEEE
2. Silveria, F.A.F., Junior, A.d.M.B., Vargas-Solar, G., Silveria, L.F.: Smart Detection: An Online Approach for DoS/DDoS Attack detection using Machine Learning. Secur. Commun. Netw. (2019). URL 10.1155/2019/1574749
3. Ray, P.: A survey on Internet of Things architectures. J. King Saud Univ., Comp. & Info. Sci 30(3), 291-319 2018). URL 10.1016/j.jksuci.2016.10.003
4. V. Gaur and R. Kumar, "DDoSLSTM: Detection of Distributed Denial of Service Attacks on IoT Devices using LSTM Model," 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT), 2022, pp. 01-07, doi: 10.1109/IC3IOT53935.2022.9767889.
5. Nouby M. Ghazaly, M. M. A. . (2022). A Review on Engine Fault Diagnosis through Vibration Analysis . International Journal on Recent Technologies in Mechanical and Electrical Engineering, 9(2), 01–06. https://doi.org/10.17762/ijrmee.v9i2.364
6. Humayun, M.; Niazi, M.; Jhanjhi, N.; Alshayeb, M.; Mahmood, S.: Cyber security threats and vulnerabilities: a systematic mapping study. Arab. J. Sci. Eng. **45**(4), 3171–3189 (2020). https://doi.org/10.1007/s13369-019-04319-2

7. Aamir, M.; Zaidi, S.M.A.: DDoS attack detection with feature engineering and machine learning: the framework and performance evaluation. Int. J. Inf. Secur. **18**(6), 761–785 (2019). https://doi.org/10.1007/s10207-019-00434-1
8. V. Gaur and R. Kumar, "FSMDAD: Feature Selection Method for DDoS Attack Detection," 2022 International Conference on Electronics and Renewable Systems (ICEARS), 2022, pp. 939-944, doi: 10.1109/ICEARS53579.2022.9752308
9. Injadat, M.; Moubayed, A.; Nassif, A.B.; Shami, A.: Multi-stage optimized machine learning framework for network intrusion detection. IEEE Trans. Netw. Serv. Manag. (2020). https://doi.org/10.1109/TNSM.2020.3014929
10. Gaur, V., Kumar, R. Analysis of Machine Learning Classifiers for Early Detection of DDoS Attacks on IoT Devices. Arab J Sci Eng 47, 1353–1374 (2022). URL 10.1007/s13369-021-05947-3
11. Odumuyiwa, V., Alabi, R. DDoS Detection on Internet of Things Using Unsupervised Algorithms. J. Cyber Secur. Mobil..10 (3), 569-592 (2021). URL 10.13052/jcsm2245-1439.1034
12. Abbas, S.A., Alamhanna, M.S. Distributed Denial of Service attacks detection system by machine learning based on dimensionality reduction. J Phys Conf Ser.1804(1), 1-13 (2021). URL 10.1088/1742-6596/1804/1/012136
13. Ananthakrishnan, B., V. . Padmaja, S. . Nayagi, and V. . M. "Deep Neural Network Based Anomaly Detection for Real Time Video Surveillance". International Journal on Recent and Innovation Trends in Computing and Communication, vol. 10, no. 4, Apr. 2022, pp. 54-64, doi:10.17762/ijritcc.v10i4.5534.
14. Alamri, H.A., Thayananthan, V.,Yazdani, J. Machine Learning for Securing SDN based 5G network. Int. J. Comput. Appl, 174(14), 9-16 (2021). URL 10.5120/ijca2021921027
15. Parfenov, D., Zabrodina, L., Zhigalov, A., Bolodurina, I. Research of multiclass fuzzy classification of traffic for attacks identification in the networks", J Phys Conf Ser, 1679(4) (2020) URL 10.1088/1742-6596/1679/4/042023
16. Shurman, M., Khrais, R., Yateem, A. DoS and DDoS Attack Detection Using Deep Learning and IDS. Int. Arab. J. Inf. 17(4A), 655-661 (2020). URL 10.34028/iajit/17/4A/10
17. Rahman, M.A. Detection of Distributed Denial of Service Attacks based on Machine Learning Algorithms Int. J. Smart Home. 14(2) 15-24 (2020). URL 10.21742/IJSH.2020.14.2.02
18. Chesney, S., Roy, K., Khorsandroo, S. Machine Learning Algorithms for Preventing IoT Cybersecurity Attacks Arai K., Kapoor S., Bhatia R. Intelligent Systems and Applications. IntelliSys 2020. Advances in Intelligent Systems and Computing, 1252. Springer, Cham.
19. Ferrag, M.A., Shu, L., Djallel, H. Deep learning-based intrusion detection for distributed denial of service attack in Agriculture 4.0. Electron. 10(11) 1257-1283 (2021). URL 10.3390/electronics10111257
20. Tume-Bruce, B. A. A. ., A. . Delgado, and E. L. . Huamaní. "Implementation of a Web System for the Improvement in Sales and in the Application of Digital Marketing in the Company Selcom". International Journal on Recent and Innovation Trends in Computing and Communication, vol. 10, no. 5, May 2022, pp. 48-59, doi:10.17762/ijritcc.v10i5.5553.

21. Sanchez, O.R., Repetto, M., Carrega, A., Bolla, R.: Evaluating ML-based DDoS Detection with Grid Search Hyperparameter Optimizationat . In : 2021 7th International Conference on Network Softwarization (NetSoft), Tokyo, Japan. June 28 – July 2, 2021.
22. Elsayed, M.S., Khac, N.A.L., Dev, S., Jurcut, A.D.: DDoSNet: A Deep-Learning Model for detecting network attacks. In: 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Cork, Ireland, 31 August-03 September 2020, pp.391-396. Cork, Ireland: IEEE
23. Assis, M.V.O., L.F.Carvalho, L.F., Lloret, J., Proença Jr.M.L., A GRU deep learning system against attacks in software-defined networks J. Netw. Comput. Appl., 177, (2021) URL 10.1016/j.jnca.2020.102942
24. Manikumar, DVVS., Maheswari, B.U., Blockchain Based DDoS Mitigation Using Machine Learning Techniques In: 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 794-800, Coimbatore, India, July 15-17 2020.
25. Li, J., Liu, M., Xue, Z., Fan, X., He, X., Rtvd: A real-time volumetric detection scheme for DDoS in the internet of things IEEE Access, vol. 8, pp. 36191-36201. Feb 2020. URL 10.1109/ACCESS.2020.2974293
26. Garg, D. K. . (2022). Understanding the Purpose of Object Detection, Models to Detect Objects, Application Use and Benefits. International Journal on Future Revolution in Computer Science &Amp; Communication Engineering, 8(2), 01–04. https://doi.org/10.17762/ijfrcsce.v8i2.2066
27. Jia, Y., Zhong, F., Alrawais, A., Gong, B., Cheng, X., Flowguard: an intelligent edge defense mechanism against IoT DDoS attacks IEEE Internet Things J., 7(10) 9552-9562 (2020) URL 10.1109/JIOT.2020.2993782
28. Sharafaldin, A., Lashkari, H., Hakak, S., and Ghorbani, A., Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy In: 2019 International Carnahan Conference on Security Technology (ICCST), pp. 1-8, Chennai, India, Chennai, India, Oct. 1-3 2019.
29. Vuong. T.H., Thi, C.V.N, Ha, Q.T.H. In: N-tier machine learning-based architecture for DDoS attack detection In: Asian Conference on Intelligent Information and Database Systems. Springer, Cham, pp- 375-385, April 7, 2021.
30. Sadique, K.M., Rahmani, R., Johannesson, P.: Towards security on internet of things: applications and challenges in technology. Procedia Computer Science, 141, pp. 199-206. URL 10.1016/j.procs.2018.10.168
31. Wang, A.; Chang, W.; Chen, S.; Mohaisen, A.: A data-driven study of DDoS attacks and their dynamics. IEEE Trans. Dependable Secure Comput. **17**(3), 648–661 (2018). https://doi.org/10.1109/TDSC.2018.2808344
32. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B.: Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. Futur. Gener. Comput. Syst. **100**, 779–796 (2019). https://doi.org/10.1016/j.future.2019.05.041

33. Munshi, A., Alqarni, N.A., Almalki, N.A.: DDOS Attack on IoT Devices. In: 3rd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 19-21 March2020, pp. 1-5. Riyadh, Saudi Arabia: IEEE
34. Popoola, S.I.; Adebisi, B.; Hammoudeh, M.; Gui, G.; Gacanin, H.: Hybrid deep learning for botnet attack detection in the internet-of-things networks. IEEE Internet Things J. **8**(6), 4944–4956 (2020). https://doi.org/10.1109/JIOT.2020.3034156
35. Pena, M., Alvarez, X., Jadán, D., Lucero, P., Barragán, M., Guamán, R., Sánchez, V. and Cerrada, M.: ANOVA and cluster distance based contributions for feature empirical analysis to fault diagnosis in rotating machinery. In: International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), Shanghai, China pp. 69-74, 16-18 August2017, Shanghai, China IEEE. URL 10.1109/SDPC.2017.23
36. Sharma, D.: Implementing Chi-Square method and even mirroring for cryptography of speech signal using Matlab. In: International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India pp. 394-397, 4-5 September2015, Dehradun, India. IEEE.URL 10.1109/NGCT.2015.7375148
37. Alsariera, Y.A., Adeyemo, V.E., Balogun, A.O., Alazzawi, A.K.: AI meta-learners and extra-trees algorithm for the detection of phishing websites. IEEE Access, 8, 142532-142542 (2020). URL 10.1109/ACCESS.2020.3013699 38. Al Hamad, M., Zeki, A.M.: Accuracy vs. cost in decision trees: A survey. In: 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakhier, Bahrain, pp. 1-4, 18-20 November 2020, Sakhier, Bahrain: IEEE. URL 10.1109/3ICT.2018.8855780
38. P. Modiya and S. Vahora, “Brain Tumor Detection Using Transfer Learning with Dimensionality Reduction Method”, Int J Intell Syst Appl Eng, vol. 10, no. 2, pp. 201–206, May 2022.
39. Azad, M., Moshkov, M.: Classification and Optimization of Decision Trees for Inconsistent Decision Tables Represented as MVD tables. In: Proceedings of the Federated Conferenc e on Computer Science and Information Systems, Lodz, Poland, pp. 31-38, 13-16 September 2015, Lodz, Poland. IEEE. URL 10.15439/2015F231
40. V. Gaur and R. Kumar, "HCTDDA: Hybrid Classification Technique for Detection of DDoS Attacks," 2021 5th International Conference on Information Systems and Computer Networks (ISCON), 2021, pp. 1-5, doi: 10.1109/ISCON52037.2021.9702399.
41. Gao, L.; Wu, W.: Relevance assignation feature selection method based on mutual information for machine learning. Knowl. Based Syst. **209**, 106439 (2020). https://doi.org/10.1016/j.knosys.2020.106439
42. V. Gaur and R. Kumar, "ET-RF based Model for Detection of Distributed Denial of Service Attacks," 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 2022, pp. 1205-1212, doi: 10.1109/ICSCDS53736.2022.976093