Cyber Security and Applications 2 (2024) 100042

Contents lists available at ScienceDirect

Cyber Security and Applications

journal homepage: http://www.keaipublishing.com/en/journals/cyber-security-and-applications/

A novel dual optimized IDS to detect DDoS attack in SDN using hyper tuned RFE and deep grid network

Nalayini C.M. [a,*], Jeevaa Katiravan [a], Geetha S. [b], Christy Eunaicy J.I. [c]

[a] Velammal Engineering College, TamilNadu, India
[b] Government College for Women (A), TamilNadu, India
[c] Thiagarajar College, TamilNadu, India

ARTICLE INFO

Keywords:
DDoS attack
Logistic Classifier
Decision Tree
Random Forest
Recursive Feature Elimination (RFE)
Repeated Stratified K-fold

ABSTRACT

Technological advancement is one of the factors contributing to a rise of susceptible cyberattacks. Distributed denial of service (DDoS) attack reduces the efficiency of network servers by saturating them with unwanted data and preventing authorized clients from accessing them. Due to the centralized architecture of Software Defined Network (SDN), it faces a number of security vulnerabilities. In SDN, DDoS attack is one of the main strikes on the control planes. A novel Optimized Dual Intrusion Detection System is proposed to identify DDoS and Non-DDoS attack more quickly with best proposed models. Hyper Tuned parameter optimization is carried on Logistic Regression, Decision Tree and Random Forest algorithms to find the best parameters. RFE with Repeated Stratified K-fold feature selection is used using the best parameters to reduce the 77 features to 4 features. A novel Deep Grid Network combines hyper-tuned classifiers with 7 other machine learning algorithms to produce 21 models. An ensemble technique uses 6 best models from 21 models for the best prediction of DDoS attack. A new dataset is also generated through Mininet for proper validation of the model.

Introduction

Distributed denial of service (DDoS) is one of the primary security issues for SDN networks [1]. It targets the data plane, the communication channel, or all three.Today, Software Defined Network (SDN) is used by the majority of business sectors to provide a wide range of comfortable customized services. Certain requirements, such as high bandwidth, high speed, high accessibility, virtualization, and dynamic management, could not be met by traditional networks [2]. Network managers are now able to employ network services proactively by using software rather than physical infrastructure. For internal communications, document sharing, and customer service, it improves network connectivity. Additionally, it lets the public and private sectors to employ the hardware and software of multiple suppliers to build specialized network infrastructure and services [3]. Based on the appropriate platform that is compatible with cloud and virtualization, the deployment of SDN networks is also adaptable and dependable. With controllers to handle the packet forwarding process and the Open-Flow method, it provides a clever central system [4]. The user prefers SDN over the conventional network due to the simplicity of network control using programmable utilities. The incoming packets are often sent through an open interface using Open flow switches, which are designed in SDN [5].

The infrastructure layer of the SDN architecture is made up of Open Flow switches, routers, and data planes [6]. The control layer is made up of control planes, while the application layer is made up of application planes, as shown in Fig. 1a. The application layer is responsible for load balancing and routing. The control layer is really the SDN controller, which acts as the network's brain and controls traffic monitoring and deployment procedures [7]. The infrastructure layer is in charge of open switches and the data plane. Switches and routers are networking components found on the data plane. The control plane controls packet communications carried out at the data plane by networking devices using a set of rules and instructions. The Northbound and Southbound interfaces are used by utilities to code hardware that is a part of the data plane [8,9]. The SDN controller uses a northbound interface to communicate with services at the application layer and a southbound interface to communicate with devices on the data plane [10].

Due to the centralized design, if a DDoS occurs at the control plane, the entire network is shut down. DDoS would be the reason for packet loss and lack of network resources whether it occurs at the data plane or the communication channel. DDoS attack is carried out via botnets, or groups of several bots. Through the use of malicious software, botnets infiltrate normal systems and turn them into zombie or bot versions of themselves. Once genuine systems have been hacked, they have become

Peer review under responsibility of KeAi Communications Co., Ltd.
* Corresponding author.
E-mail address: nalayini@velammal.edu.in (N. C.M.).

https://doi.org/10.1016/j.csa.2024.100042
Received 16 November 2023; Received in revised form 8 January 2024; Accepted 7 February 2024
Available online 15 February 2024

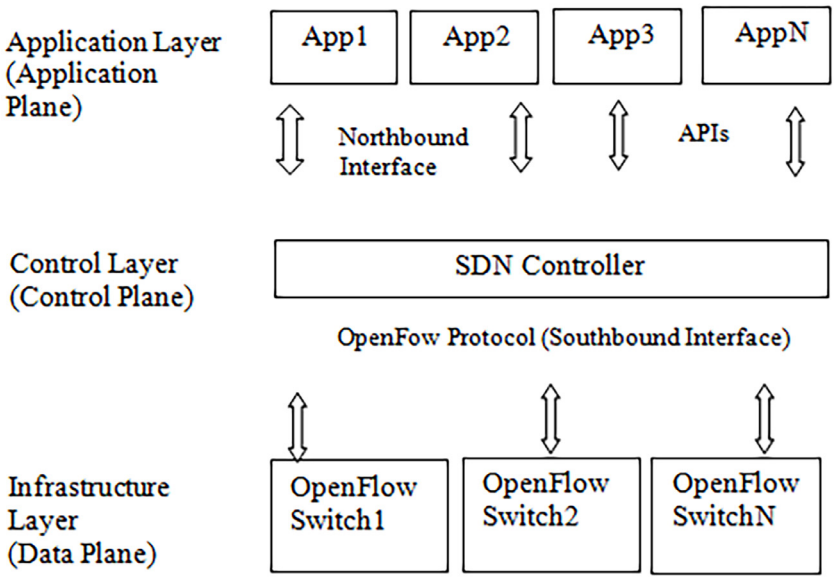

Fig. 1a. SDN Architecture.

engaged in the SDN controller's and open flow switches' communication processes [11]. SDN controllers are utilized in these assaults to establish a single point of failure and stop the resources. The SDN Controller's resources are simply consumed by the new false packets from various zombies or bots, which prevents real packets from receiving services. Automatically, the SDN controller loses its capacity to act as a central hub and shuts down, which causes the performance of the entire network to suffer as shown in Fig. 1b.

The present detection techniques for DDoS attack in the SDN environment are inefficient because of the SDN architecture's numerous quirks. Due to the increased risk of DDoS assault in the SDN environment, the SDN controller must continuously monitor the network traffic at the open flow switches. In order to exhaust the resources of the SDN controller and degrade network performance, several fictional packet flows are cloaked in real traffic consequently, thereby controller performs an additional duty. For SDN security, machine learning algorithms offer better, more precise, and dynamic solutions, as a result, the best detection models are needed to catch DDoS as soon as possible. To detect the DDoS assault extremely effectively, such models may be put either at the open flows switches or controller. This relieves the SDN Controller's responsibility of attack monitoring.

These days, intrusion detection systems (IDSs) are useful for spotting flaws or attacks that might potentially compromise networks and original data. The two IDS types that are available in this situation are host-based intrusion detection systems (HIDS) and network-based intrusion detection systems (NIDS). IDSs are used by a variety of networks, including wired networks, wireless networks, mobile networks, mobile ad hoc networks (MANETs), wireless sensor networks (WSNs), and software defined networks (SDNs) [12]. In order to make efficient decisions using network datasets, the IDSs are built by adding novel trust mechanisms, feature selection, and classification algorithms. It is necessary to observe and record the ongoing operations of the various network nodes as a dataset. On the other side, the trust mechanism is also included to examine node behaviours and rank them in accordance with the network-wide trust ratings. Additionally, by utilizing the numerous feature selection and classification techniques, the dataset is efficiently analyzed.

Today, classification is crucial for classifying documents, assaults, diseases, products, etc. In the processes of illness prediction, assault detection, product prediction, etc., the classification result is employed as a prediction result. By conducting efficient feature selection and classification, Machine Learning (ML) algorithms are playing a crucial part

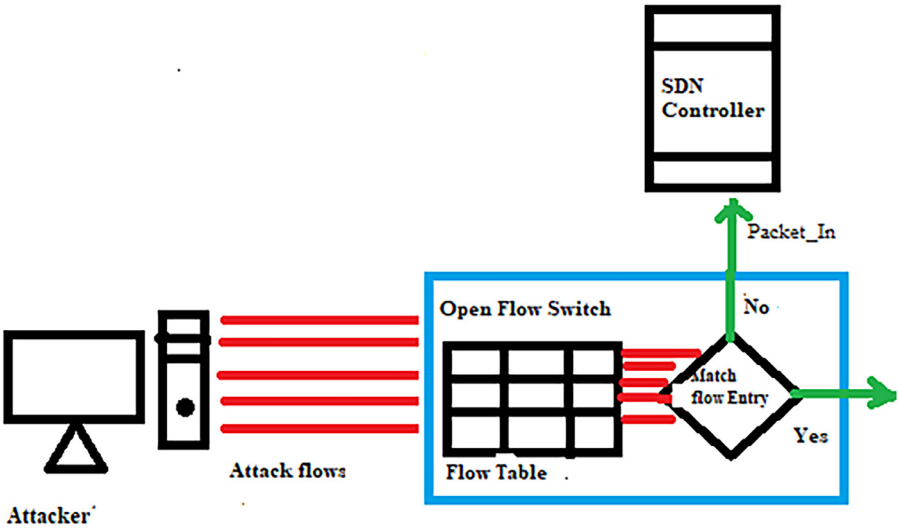

Fig. 1b. DDoS attack in SDN.

in the process of attack detection. Researchers apply various ML techniques, such as Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbour, Naive Bayes, Support Vector Machine, Linear SVM, and Non-linear SVMetc, to increase the attack detection accuracy.

The following are the main contributions of this work.

1) To put forth a fresh method of feature optimization that uses three machine learning algorithms to determine the optimal parameters via hyper tuning.
2) To create a new Repeated Stratified K-Fold RFE to cut down on the number of features in the first level optimization's optimized output.
3) To provide a brand-new Deep Grid Network to produce six best models using the enhanced feature sets.
4) To demonstrate that the proposed model is superior to current IDSs using the benchmark dataset and own dataset.

The remainder of the paper is organised as follows: Section 2 summarises the current works that are available in the direction of IDS, SDN, and DDoS attacks detection process. Section 3 explains the proposed IDS by using an overall architecture. Section 4 describes the newly developed IDS with necessary background information and algorithms. Section 5 demonstrates the performance of the newly developed IDS by considering the common evaluation parameters of the network. Section 6 concludes the work by highlighting the quantitative achievements.

Related work

Security concerns must be immediately addressed in order to fulfill the demands of daily technology evolutions. One of the latest technical developments that is crucial to its particular qualities is the SDN. Security is considered as one of the most challenging facets of SDN due to its single point of failure. The literature lists a number of security techniques, including IDS. One of them was an LDA defense system designed by Wang et al. [13] to identify a novel attack known as a link flooding DDoS attack (LFA). In order to hide its tracks, this assault begins with large-scale, slow-moving normal flows and switches the victim's connection on a frequent basis. Traditional systems struggle to detect this sort of assault as a result. First, the LDA defender software locates the victim's link with high traffic density. Next, agent software is installed at the identified victim's link to monitor the traffic, and finally, the traffic flow statistics are analyzed and rerouted to control congestion at the link. Using the appropriate flow rules created by the SDN controller, the origin is then located and removed from the network for further communication. This LDA Defender was used in Cloud Lab trials and achieved 90 % accuracy.

A detection technique was implemented in the SDN controller by Nada M et al [14] to identify Denial of Service attack. Entropy is used to keep track of IP engagement.

A survey report provided by Yan et al [15] describes how to protect the cloud environment using SDN. They omitted discussing the issues present in the SDN context.

A fresh strategy that is deployed on top of the SDN layer was provided by E. Keller et al [16]. It enables the user to create rules based on their specifications for having proper integration with the devices at the data plane. This will cause misrule and the network devices inability to make decisions.

To identify various forms of DDoS, such as HTTP, ICMP, UDP, and SIP, Dong S et al. [17] developed two techniques, namely DDoS Detection based on Degree of assault algorithm and DDoS Detection based on Machine Learning. By comparing performance criteria like accuracy, precision, F-Score, and recall values to traditional algorithms like NB and SVM, they demonstrated that their suggested algorithms offer excellent efficiency.

To identify DDoS attack, Kimmi Kumari et al. [18] presented mathematics and machine learning methods. Various characteristics, like arrival time, throughput, source and destination IP addresses, etc., are used to train the model utilizing the Weka data mining tool.

SVM was used by Dang-Van and H. Truong-u [19] to recognize DDoS attacks and fuzzy logic to identify legitimate traffic. To identify DDoS attacks, L. Linxia et al. [20] employed evolutionary techniques such as Simulated Annealing, Ant Colony Optimization, Particle Swarm Optimization, and Genetic techniques. Importance of Intrusion detection model is highly enforced to detect DDos attack in Wireless networks by C M Nalayini et al. [21].

Jia Y et al. [22] utilized CNN to categorize the assaults utilizing Flow guard mechanism and deep learning-based techniques like LSTM to detect the flooding attack and sluggish request/response attack. It comprises of programmes called flow handlers and flow filters. The flow handler records the network traffic using the flow rules produced by the deep learning models, and incoming flows are subsequently filtered in accordance with those rules. For the purpose of employing an SVM model to detect DDoS attack, Jin Ye et al. [23] gathered data from the switch flow table using six tuple values.

Nave Bayes and K-Nearest Neighbours algorithms were employed by Chakraborty and Banerjee [24] to distinguish between attack traffic and regular traffic. Connections that are detected are added to an access control list for later analysis. A methodology to identify and fight against low-rate DDoS assault was put out by Pérez-Daz et al. [25]. They employed the CIC-DoS dataset to train machine learning algorithms such SVM, Random Forest, Random Perception, Random tree, J48, and REP tree for their intrusion detection system, and they were able to achieve 95.5 detection accuracy. Additionally, they demonstrated the effectiveness of their strategy by testing it on a dataset they created using the open-source emulator Mininet.

Three threshold profiles—low, medium, and high—were created by Alamri et al. [26] to analyse regular traffic and attack traffic at regular intervals using time and byte rate parameters. The number of times the traffic exceeds the threshold limit is counted and recorded as the flow statistics that are sent as input to the XGBoost model to categorize the flow as normal or attack flow. The threshold profile is chosen depending on the timing and data flow.

Numerous machine learning techniques, including self-organizing maps, multi-pass self-organizing maps, multi-pass learning vector quantization, linear vector quantization, and hierarchical learning, were presented by Jankowski and Amanowicz [27]. To identify the DDoS assault at the data plane vector quantization is utilized.

Machine learning algorithms [28] will pave the way to identify DDoS attack that occurs at the data plane. When an attack is discovered, the SDN controller writes the appropriate flow rules and sends them to the switch to stop it. With the inclusion of multiple feature selection algorithms, classifiers, trust mechanisms, and trust based secured routing models, all the currently available IDSs are achieving a respectable accuracy on a variety of benchmark datasets and network datasets. Even yet, it might be exceedingly difficult to identify DDoS assault in a distributed setting. This paper suggests a novel, dual, and optimized IDS to properly identify DDoS and non-DDoS threat.

Background: Machine learning algorithms

In addition to providing background information and the essential equations, this part discusses the workflow of the standard ML classification algorithms RF, DT, SVM, NB, k-NN, and LC. Also underlined at the conclusion of the explanation is the part these ML algorithms play in the suggested IDS.

Logistic Classifier

It is a supervised learning technique and a statistical model suitable for binary classification. A set of independent variables is given as the input to predict the categorical dependent variable. The dependent variable should be categorical and the independent variable should not contain multi collinearity. It is divided into three types. Binomial involves yes or no type, 0 or 1 etc. Multinomial involves 3 or more unordered

dependent variables. Ordinal involves 3 or more ordered dependent variables.

Decision Tree

Both classification and regression issues may be solved using the supervised learning technique known as a decision tree. A decision tree builds a model that predicts the result using decision rules made from a set of attributes. It is composed of the root node, internal nodes, branches, and leaf nodes. The decision-making process starts at the root node, and the leaf nodes include data about features, branching, and decision-making itself. Decision trees produce binary predictions as their output.

Random Forest

The approach of supervised learning Problems involving classification and regression may both be solved with Random Forest. It benefits from the concept of group learning. After a number of decision trees have been trained, each on a different random subset of the training dataset, the remaining dataset is used as the test set. Once projections from all the decision trees are accomplished, the final result is then expected based on the high votes. Instead of using just one predictor, averages from multiple are employed to improve accuracy. It delivers high accuracy even with a bigger dataset and needs less training time. The number of trees enhances accuracy.

Recursive Feature Elimination

Recursive Feature Elimination is a wrapper style method used technically but internally it also uses filter method to recursively removes the features until the best performing feature is found. In each iteration it creates a model with remaining features to recursively remove and find the best feature until all the features of the training set are done. It is associated with the chosen machine learning model and cross validation method to automatically select the number of features to improvise the accuracy parameter.It starts with all features of the training dataset and then apply recursion to remove the poor performing features until the finite number of features remains

Repeated Stratified k-fold

It repeats the k fold 'n' times with stratified cross validation using different randomization. The number of folds should be greater than or equal to 2. After K folds are done, mean is calculated. Since K-fold Cross validation randomly splits the data, it could not address an imbalanced class. Hence it is extended to stratified k-fold cross validation to handle the class imbalance issue. For each fold, it uses same ratio to split the data.

K-Nearest Neighbor Classifier

Using supervised learning, the K-nearest neighbours (KNN) classifier may be applied to both classification and regression tasks. It doesn't take any supporting information or settings into account. Outliers will result from choosing k as low values, such as 1 or 2. K value might be as high as 5. It saves the dataset during training. Using the chosen k value, distance is determined, and several data point groupings are created based on similarity. KNN examines if a new data point and the groups of existing data points have any similarities before assigning the new data point to the appropriate group.

Naive Bayes

The supervised learning method Nave Bayes is used for binary and multi-class classification. It is a quick modelling and quick prediction method. Each characteristic is unique in comparison to other qualities, according to this system's guiding concept. Based on the attribute's likelihood, it makes a forecast. The Bayes formula is used to calculate the conditional probability when utilising a hypothesis. The probabilistic parameters H=Posterior, I=Likelihood, J=Prior, and K=Marginal are used to determine the likelihood of the contributing qualities.

$$\boldsymbol{H} = \frac{(\boldsymbol{I} \times \boldsymbol{J})}{\boldsymbol{K}} \tag{1}$$

SVM Linear versus SVM Non-Linear

Support vector machines, sometimes known as SVMs, are supervised learning methods that may be applied to both classification and regression problems. In order to divide the data points (vectors) into their corresponding classes, SVM creates a hyperplane (decision boundary). It determines which data points or vectors from the two relevant classes are closest, also known as support vectors, in order to identify the margin where the distance between the data points and the hyper plane are concerned. That allows it to determine which class each incoming data point belongs to. Default classifier is SVM Linear. SVM is considered linear if it can divide the data points (vectors) into two distinct classes; else, it is non-linear. To accommodate large dimensional space, the kernel for Non-Linear SVM may be specified as either an RBF or polynomial.

Proposed work

The proposed dual optimized IDS depicted in Fig. 2 consist of two levels of feature optimizations such as optimization 1 and optimization 2 that are applied to identify the more relevant features which are helpful for making effective decision on input datasets. Additionally, the proposed IDS includes a newly created deep grid network that employs classifiers with hyper-tuned best parameters, such as Logistic Classifier, Decision Tree, and Random Forests, to support feature selection in alignment with the standard ML algorithms with default parameters, such as Random Forest, Decision Tree, Naive Bayes, k-Nearest Neighbour, Logistic Classifier, Linear SVM, and Non-Linear SVM for carrying out effective classification. For effective prediction, an ensemble approach is finally done with best models to obtain better results. The proposed IDS is validated with our own dataset generated throughMiniNet tool and also compared with existing datasets to prove the better accuracy of our models. This section demonstrates the overall workflow of the proposed dual optimized IDS with necessary steps

SDN dataset generation

The method for creating the dataset is described in this section. In order to validate the six top models, this study uses a brand-new SDN dataset that we generated on our own using mininet. Use miniedit to first construct a new topology and configure the connecting devices. It then started the mininet-based attacks and regular traffic commands for our topology. Then, using Wireshark, normal and attack traffic is collected, and corresponding PCAP files are created. The relevant PCAP files are transformed into.CSV files using the CIC Flowmeter. As illustrated in Fig. 3, assault and the normal.csv files are combined to create the SDN dataset, which is then provided as input to the RYU controller, where our six best models are implementedfor effective detection.

Data collection process

This subsection gives a description of the data collection process. The proposed IDS uses the standard CICIDS 2019 dataset, which was produced by the University of New Brunswick's Faculty of Computer Science, for both training and testing. Additionally, the Canadian Institute created it for use in intrusion detection and prevention systems. It also contains a variety of contemporary attacks, such as those that use MSSQL, TFTP, LDAP, NetBIOS, SSDP, UDP, SNMP, DNS, SYN, WebDDoS, and UDP-Lag. Additionally, it contains the CICFlowMeter network traffic analysis findings based on source and destination IP addresses, source and destination ports, protocols, assaults, and time stamps. To create a trustworthy dataset, it took into account a number of factors,

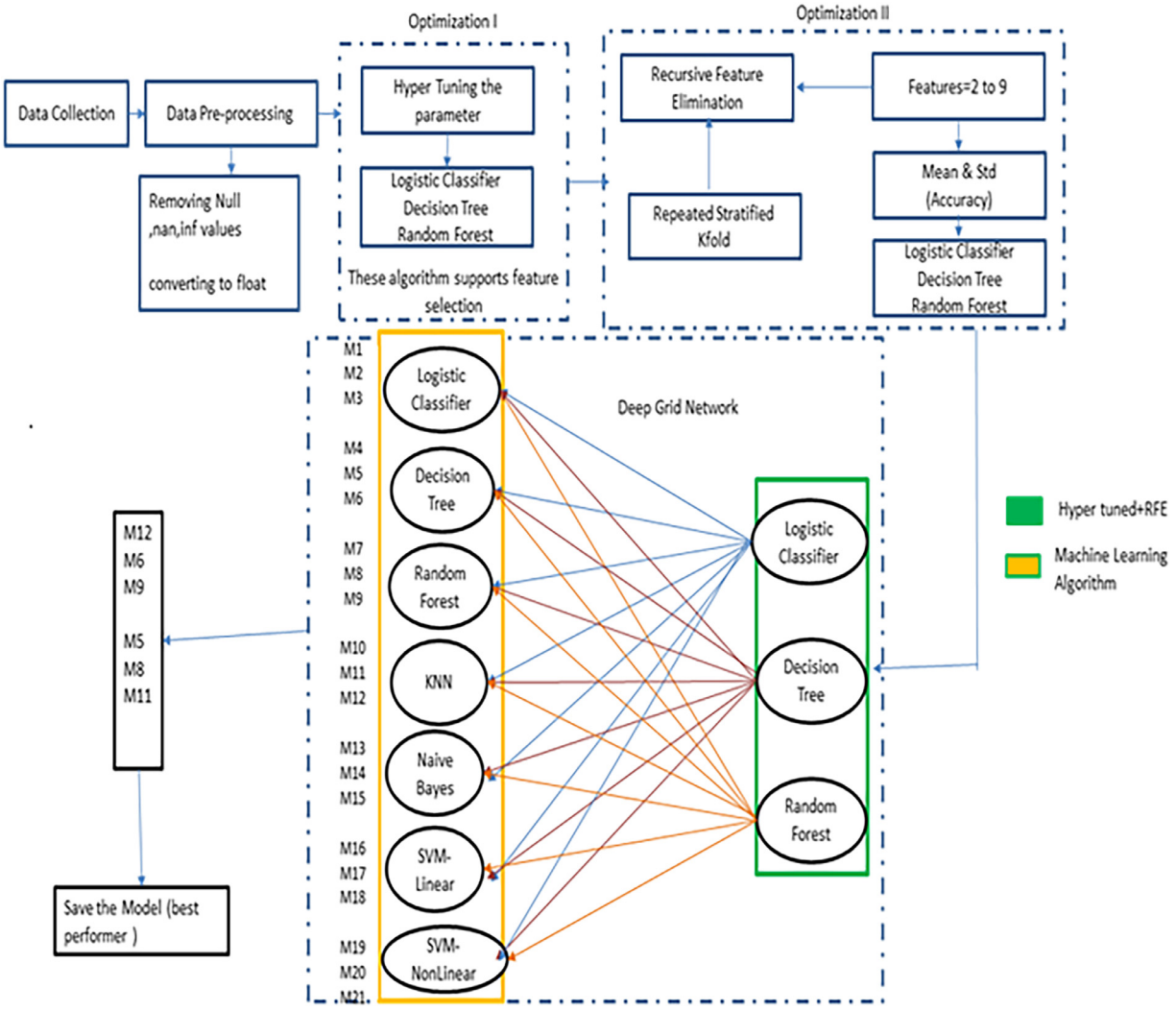

Fig. 2. System Architecture.

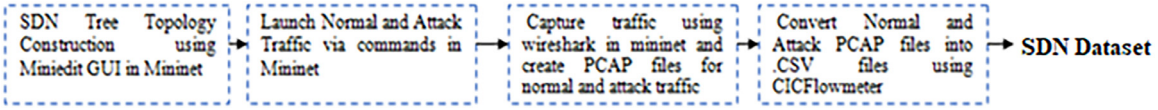

Fig. 3. SDN Dataset Generation.

including the whole network setup, all traffic, labeled data set, full interaction, full capture, available protocols, attack diversity, heterogeneity, feature set, meta data, day, date, description, and size (GB).

Data preprocessing

The standard dataset has 79 features in total, in which two columns are removed due to infinite values. Null values and Nan values are removed from 77 features. Next, the values represented in object form are converted into floats. Finally, the full featured dataset without Null and Nan values are considered as input dataset for performing feature optimization. The steps of the proposed Hyper Tuning Parameterizing and RFE aware Data Preprocessing Algorithm (HRDPA) are as follows:

Hyper Tuning Parameterizing and RFE aware Data Preprocessing Algorithm (HRDPA)

Input:Full featured dataset

Output:Dataset with optimized features

Phase 1: First level Optimization

Step 1: Read the records from standard network dataset.

Step 2: Remove the Null and Nan values and also convert into float values.

*Step 3:*In order to complete the hyper-tuned parameterized process, the following calculations of Logistic Classifier, Decision Tree and Random Forest are made.

$$LC = \frac{1}{1 + e^{-(\beta 0 + \beta 1 * x)}} \tag{2}$$

Where LC=Logistic Classifier, β_0=intercept, β_1= slope and x=input

$$DTf = \sqrt{\sum((r - \frac{\sum r}{n})^2)/n} - \sum_{C \pounds X} P(c)S(c) \tag{3}$$

Where DT_f=final decision, r=feature, n=count, p=probability and s=standard deviation

$$RF = \frac{\sum_k^n \mathrm{Nf}}{m} \tag{4}$$

Where RF = calculation of feature importance of all trees, Nf =normalized feature

Importance, m=total number of trees

Step 4: Find the best parameters which produces 100 % accuracy as the result.

Step 5: Return the best parameters

Phase 2: Second level Optimization

Step 1: Feed Phase 1 as the input to Phase 2 Optimization.

*Step2:*Take into account Dataset D and three classifiers with well-adjusted parameters.

Divide dataset d into equal portions with reference to i=1 to k divides in order to reduce errors and deviations and to enable the use of each component for training (dtrain) and testing (dtest) individually.

$$\mathrm{RSKCV} = \sum_{j=1}^{k} 1/k \tag{5}$$

where RSKCV=repeated stratified kfold cross validation

Apply recursive feature elimination with repeated stratified k fold cross validation to find optimal features n←N, where n holds the subset N for each classifier Ci do while(n!=0) evaluate feature importance place the least important feature to last location of rank array remove the least feature from the feature set N n←n-1 return optimal feature set

*Step 3:*To decrease the ideal feature set from 77 to 4, note the mean accuracy and standard deviation of the corresponding classifiers.

Applying freshly created Hyper Tuned parameterized ML classifiers, such as the Random Forest, Decision Tree, and Logistic Classifier algorithms, during Phase 1 is responsible for conducting the feature selection procedure. The standard network dataset CICIDS 2019 is utilized, and while it has a total of 79 features, following data preparation, 77 of those characteristics were employed as the input for phase I optimization. The most accurate parameters are discovered after hyper-tuning each of those three algorithms' parameters. Recursive feature elimination (RFE) is used in Phase 2 to eliminate the features or attributes that have the lowest ranking score. Errors and deviations might occur when the data are distributed in recursive feature elimination. In order to select the optimal features utilizing hyper tuned parameters, it is paired with repeated stratified cross validations for equitable distribution of data across all folds. The mean accuracy and standard deviation results for all folds and repetitions are recorded using repeated stratified K-fold repeats cross-validation, creating an accurate estimate that can be used to improve performance and evaluate a variety of various models. It guarantees that there has been the same number of observations made across all dataset folds. When compared to Logistic Classifier, Random Forest, and itself, Decision Tree Classifier performs better for 4 features. 77 attributes are therefore reduced to 4 features in phase 2 optimization. Deep Grid Network Process

Deep grid network

The Deep Grid Network (DGN) is employed in this study to carry out efficient categorization. DGN receives optimal featured dataset as input. The Logistic classifier, Decision Tree, and Random Forest with hyper-tuned optimal parameters are processed in this situation using standard parametric algorithms like KNN, Naive Bayes, SVM Linear, and SVM Non-Linear. This technique created 21 models by considering attributes ranging from 2 to 9 using RFE with Repeated Stratified K-fold. In the end, the best accuracy was attained by six of the 21 models.

From Phase III we found that six best models performed well are listed below.

Random Forest (RFE with hyper tuned parameter) →KNN (Default Parameter)

Random Forest (RFE with hyper tuned parameter) →Decision Tree (Default Parameter)

Random Forest (RFE with hyper tuned parameter) → Random Forest ((Default Parameter))

Decision Tree (RFE with hyper tuned parameter) →KNN (Default Parameter)

Decision Tree (RFE with hyper tuned parameter) →Decision Tree (Default Parameter)

Decision Tree (RFE with hyper tuned parameter) → Random Forest ((Default Parameter))

Ensemble Method for Efficient Prediction of DDoS and Benign

Ensemble approach is chosen to improve the performance of the 6 best selected models to yield the better results. Fig. 4 shows that a snippet from the dataset is fed as the input to ensemble technique which uses the six best models to obtain the most accurate results.

Fig. 5 illustrates the prediction process using the recently established ensemble technique, which includes the six top models for doing so while taking into account the optimized characteristics chosen by the

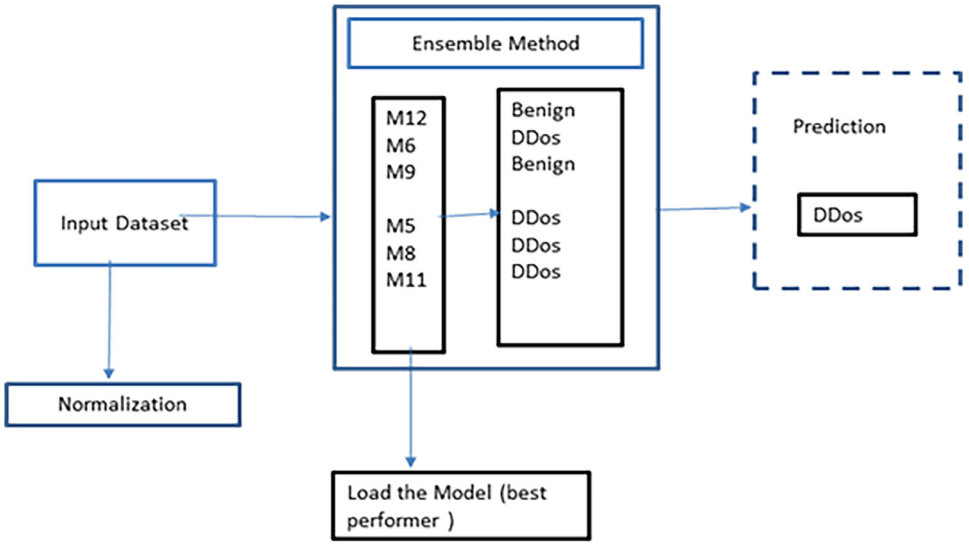

Fig. 4. Ensemble approach.

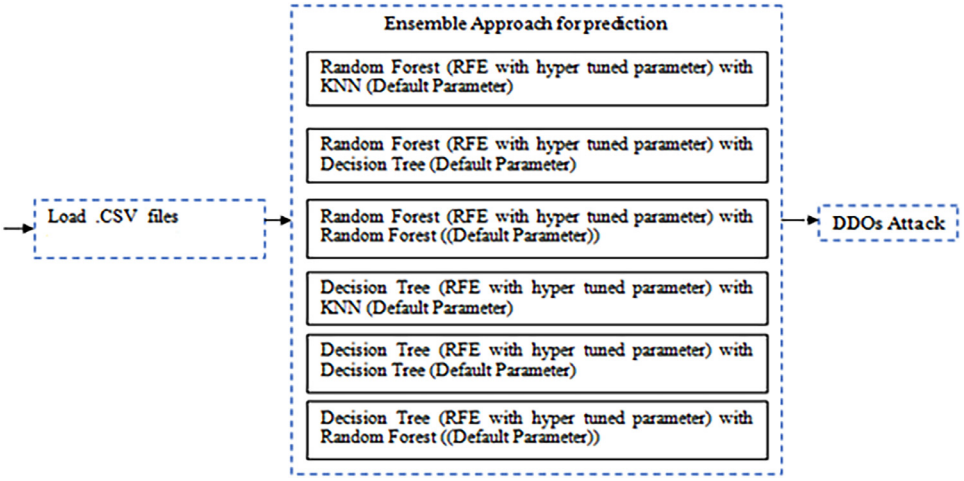

Fig. 5. Best Prediction of DDoS.

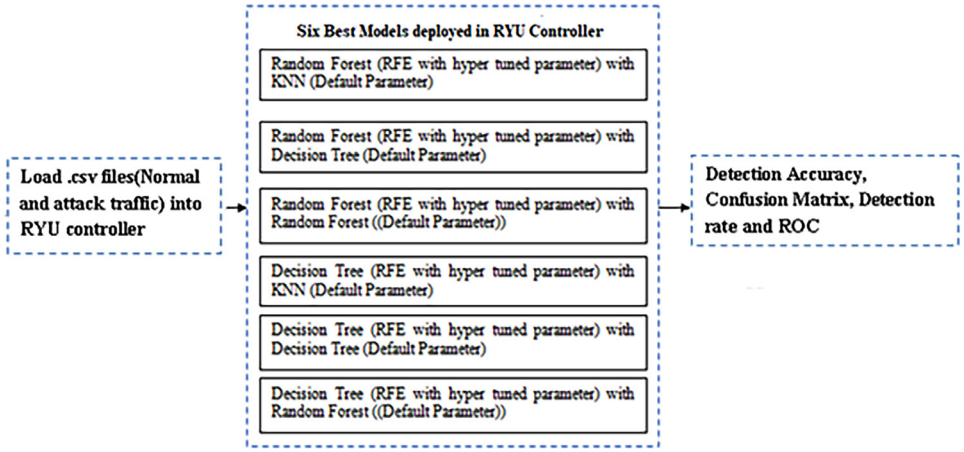

Fig. 6. Detection Accuracy, Confusion matrix, Detection rate and ROC(Our Dataset).

proposed HRDPA. To test the efficacy of our ensemble technique, we ran dataset samples from the CIC IDS 2019 dataset with and without DDoS characteristics.

For clarity, the next section explains the general procedure of the recently built IDS

Optimized dual intrusion detection system (OD-IDS)

Input: NetworkDataset
Output: Record sets with result

Step 1: Read all the record set from the standard benchmark dataset.
Step 2: Call the newly developed HRDPA for performing feature optimization.
*Step 3:*Apply the Deep Grid Network, which contains seven classifiers: Logistic Classifier, RF, DT, k-NN, Naive Bayes, Linear SVM, and Non-Linear SVM in alignment with Hyper tuned parameters classifier.
Step 4: Train and test the optimized featured dataset by using the Deep Grid Network to find the best models.
Step 5: Find the best models according to the training and testing accuracy on optimized featured dataset.
*Step 6:*For effective prediction of DDos or Benign apply ensemble method on the best models.
Step 7: Return the prediction result as record set with result.

The proposed Optimized Dual IDS works by incorporating the newly developed HRDPA and the Deep Grid Network with an ensemble method.

Proposed Models validated with our own SDN dataset generated via MiniNet

As demonstrated in Fig. 6, load the SDN dataset produced by MiniNet as input to the RYU controller, where our six top models are imple-

Table 1
Optimization-2: Recursive Feature Elimination (RFE) with Repeated Stratified KFold.

Number of features (Repeated StratifiedKFold)	Logistic Classifier		Decision Tree		Random Forest	
	mean accuracy	standard deviation	mean accuracy	standard deviation	mean accuracy	standard deviation
2	0.752	0.026	0.999	0.000	0.998	0.002
3	0.746	0.030	0.999	0.000	0.999	0.001
4	0.748	0.026	1.000	0.000	0.999	0.000
5	0.758	0.029	1.000	0.000	0.999	0.000
6	0.803	0.028	1.000	0.000	0.999	0.000
7	0.800	0.028	1.000	0.000	0.999	0.000
8	0.805	0.030	1.000	0.000	1.000	0.000
9	0.811	0.025	1.000	0.000	1.000	0.000

mented for quick and effective DDoS attack detection. For our dataset, each of the six top models obtained 100▒ % detection accuracy.

Results and discussion

This part presents the experimental setup, performance assessment metrics, experimental findings, and comparison analysis in detail, along with the outcome and comments. The experimental set up is first explained.

Experimental Setup

This work is implemented by using Mininet that is an emulator used to set up the SDN environment conveniently. First, Install the Mininet either in Ubuntu or in windows via Oracle virtual boxwith

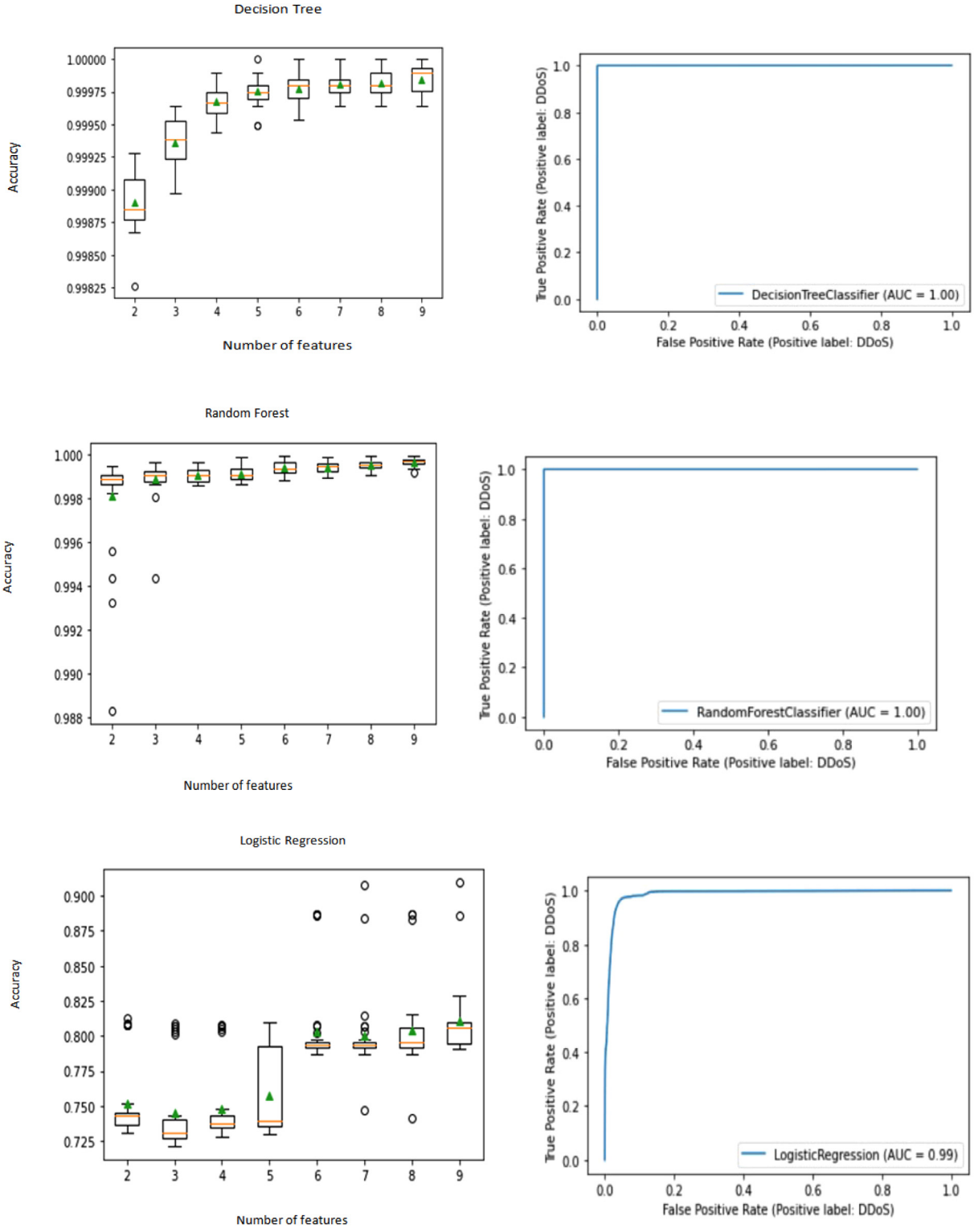

Fig. 7. Mean Accuracy.

Table 2
RFE with Repeated stratified k-fold when number of features=3.

Number of Features= 3 RFE with Repeated Stratified Kfold		Algorithm		With default parameters						
Algorithm	Feature Name			Logistic	Support Vector Machine-Linear	Support Vector Machine – Non-Linear	K-Near Neighbour	Naive Bayes	Decision Tree	Random Forest
Logistic Classifier	1. FIN Flag Count, 2. ACK Flag Count, 3. URG Flag Count,	Feature Selection Algorithm with Hyper Tuning N=3	Logistic	0.66192	0.66192	0.66192	0.59624	0.59624	0.66192	0.66192
Random Forest	1.Fwd Packet Length Max 2.Fwd Packet Length Mean 3.Fwd IAT Std		Random Forest	0.8668	0.85792	0.85712	0.99584	0.91064	0.99664	0.99664
Decision Tree	1.Destination Port 2.Total Length of Fwd Packets 3.Init_Win_bytes_forward		Decision Tree	0.77616	0.76488	0.96824	0.99904	0.74848	0.99928	0.99928

Table 3
RFE with Repeated stratified k-fold when number of features=4.

Number of Features= 4 RFE with Repeated Stratified Kfold		Algorithm		With default parameters						
Algorithm	Feature Name			Logistic	Support Vector Machine-Linear	Support Vector Machine – Non-Linear	K-Near Neighbour	Naive Bayes	Decision Tree	Random Forest
Logistic	1.FIN Flag Count, 2.ACK Flag Count, 3. URG Flag Count, 4.Down/Up Ratio,	Feature Selection Algorithm with Hyper Tuning N=4	Logistic	0.66168	0.66192	0.76336	0.6768	0.66832	0.76336	0.76336
Random Forest	1.Fwd Packet Length Max 2. Fwd Packet Length Mean 3.Fwd IAT Std 4.act_data_pkt_fwd		Random Forest	0.86696	0.8592	0.93544	0.99808	0.57152	0.99776	0.99792
Decision Tree	1.Destination Port 2.Total Length of Fwd Packets 3. Average Packet Size 4.Init_Win_bytes_forward		Decision Tree	0.83632	0.84152	0.95968	0.99952	0.5744	0.99968	0.99968

Table 4
RFE with Repeated stratified kfold when number of features=5.

Number of Features= 5 RFE with Repeated Stratified Kfold		Algorithm		With default parameters						
Algorithm	Feature Name			Logistic	Support Vector Machine-Linear	Support Vector Machine – Non-Linear	K-Near Neighbour	Naive Bayes	Decision Tree	Random Forest
Logistic	1.FIN Flag Count, 2.ACK Flag Count, 3. URG Flag Count, 4.Down/Up Ratio, 5.min_seg_size_forward	Feature Selection Algorithm with Hyper Tuning N=5	Logistic	0.75608	0.75608	0.76336	0.50456	0.59624	0.76336	0.76336
Random Forest	1.Fwd Packet Length Max 2. Fwd Packet Length Mean 3.Fwd IAT Std 4.Avg_Fwd_Segment_Size 5.act_data_pkt_fwd		Random Forest	0.86624	0.85808	0.96824	0.99816	0.91088	0.99928	0.99912
Decision Tree	1.Destination Port 2.Total Length of Fwd Packets 3. Average Packet Size 4.Init_Win_bytes_forward 5.Fwd IAT Max		Decision Tree	0.83688	0.8416	0.96	0.9996	0.9796	0.9996	0.99968

8 GB RAM, and 64GB storage.Thenetwork topology is designed via Mini Edit (GUI) in connection with Mininet. Here,a tree topologyis constructed with a controller, two switches and 10 hosts or nodes. Switch 1 and Switch 2 are connected to controller and each other. The configuration set up of all communicating devices present in the topology. Host1(IP 10.0.0.1), Host2(IP 10.0.0.2), Host3(IP 10.0.0.3), Host4(IP 10.0.0.4) and Host5(IP 10.0.0.5) are connected to Switch 1. Host6(IP 10.0.0.6), host7(IP 10.0.0.7), host8(IP 10.0.0.8), host9(IP 10.0.0.9) and Host10(IP 10.0.0.10) are connected to Switch 2. After configuration set up, need to start with CLI.Theconstructed SDN Tree topology is successfully created via Mini Edit and the respective command Sudo Python3 Mininet/Mininet/examples/miniedit.py [29] for miniedit that executes in Mininet.The successful topology creation in displayed in Mininet and execute the ovs summary from Miniedit to show the connectivity of all the devices present in the topology(controller, Bridge s1 and Bridge s2, port details, interface details).

Then, it launched Wireshark using the command SudoWireshark, and in Mininet, it started up and displayed the welcome screen where it discovered all potential interfaces. It starts the operation from the Capture menu, sets the necessary properties in the input window by

Table 5
RFE with Repeated stratified k-fold when number of features=6.

Number of Features= 6 RFE with Repeated Stratified Kfold		Algorithm		With default parameters						
Algorithm	Feature Name			Logistic	Support Vector Machine-Linear	Support Vector Machine – Non-Linear	K-Near Neighbour	Naive Bayes	Decision Tree	Random Forest
Logistic	1.FIN Flag Count, 2.ACK Flag Count, 3. URG Flag Count, 4.Down/Up Ratio, 5.min_seg_size_forward 6.Min Packet Length	Feature Selection Algorithm with Hyper Tuning N=6	Logistic	0.91616	0.91608	0.92168	0.92168	0.8832	0.92168	0.92168
Random Forest	1.Fwd Packet Length Max 2. Fwd Packet Length Mean 3.Fwd IAT Std 4.Avg_Fwd_Segment_Size 5.act_data_pkt_fwd 6.Total Length of Fwd Packets		Random Forest	0.8624	0.85776	0.96904	0.99824	0.91032	0.99928	0.99912
Decision Tree	1.Destination Port 2.Total Length of Fwd Packets 3. Average Packet Size 4.Init_Win_bytes_forward 5.Fwd IAT Max 6.Flow IAT Mean		Decision Tree	0.892	0.93544	0.97456	0.99792	0.97952	0.99984	0.99984

Table 6
RFE with Repeated stratified kfold when number of features=7.

Number of Features= 7 RFE with Repeated Stratified Kfold		Algorithm		With default parameters						
Algorithm	Feature Name			Logistic	Support Vector Machine-Linear	Support Vector Machine – Non-Linear	K-Near Neighbour	Naive Bayes	Decision Tree	Random Forest
Logistic	1.FIN Flag Count, 2.ACK Flag Count, 3. URG Flag Count, 4.Down/Up Ratio, 5.min_seg_size_forward 6.Fwd Packet Length Min 7.Total Backward Packets	Feature Selection Algorithm with Hyper Tuning N=7	Logistic	0.93872	0.94056	0.94576	0.96168	0.89928	0.96296	0.96288
Random Forest	1.Fwd Packet Length Max 2. Fwd Packet Length Mean 3.Fwd IAT Std 4.Avg_Fwd_Segment_Size 5.act_data_pkt_fwd 6.Total Length of Fwd Packets 7.Fwd Header Length.1		Random Forest	0.86552	0.85768	0.96912	0.998	0.91064	0.99928	0.99912
Decision Tree	1.Destination Port 2.Total Length of Fwd Packets 3. Average Packet Size 4.Init_Win_bytes_forward 5.Fwd IAT Max 6.Flow IAT Mean 7.Fwd Packet Length Max		Decision Tree	0.96656	0.95752	0.97408	0.99784	0.9788	0.99952	0.99992

choosing s1-eth1 for additional communication, and verifies PCAP in the output windows [30]. By using the Mininet ping all command, you may check the connection between all nodes before launching normal traffic and attack traffic. The commands dump and dpctl are used to view the status of Wireshark. Information on hosts, switches, and controllers is available from the dumps. Switch table flows are displayed by DpctlTo view the packet flows, open the switch 1 and switch 2 terminals with the commands xterm s1 and xterm s2, respectively. To find out the information about includes cookie, duration, table, number of packets, amount of bytes time, priority, icmp, port: s1-eth1, source and destination, run the dpctl dump-flows command on switches 1 and 2. We started Normal traffic and downloaded a Wire shark PCAP file. Assaults from different hosts present in switch 2 against switch 1 using DOS, UDP, and SYN. We conducted assaults using the hping3 application on the mininet. The attack traffic was gathered using Wireshark, and PCAP data were also obtained. The normal and attack traffic are combined into a single.csv file to provide the final SDN dataset. Features like source port, destination port, byte count, packet count, IP protocol, and flow time are included in the.csv file.

Performance Evaluation Metrics

The suggested model is assessed using industry-standard assessment criteria including precision, recall, f-measure, and accuracy.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{(TP + FP)} \tag{7}$$

$$F - Measure = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \tag{8}$$

$$Accuracy = \frac{TN + TP}{(TN + FP + TP + FP)} \tag{9}$$

Table 7
RFE with Repeated stratified kfold when number of features=8.

Number of Features= 8 RFE with Repeated Stratified Kfold		Algorithm		With default parameters						
Algorithm	Feature Name			Logistic	Support Vector Machine-Linear	Support Vector Machine – Non-Linear	K-Near Neighbour	Naive Bayes	Decision Tree	Random Forest
Logistic	1.FIN Flag Count, 2.ACK Flag Count, 3. URG Flag Count, 4.Down/Up Ratio, 5.min_seg_size_forward 6.Fwd Packet Length Std 7.Total Backward Packets 8.act_data_pkt_fwd	Feature Selection Algorithm with Hyper Tuning N=8	Logistic	0.85672	0.85696	0.982	0.99792	0.83768	0.99896	0.99896
Random Forest	1.Fwd Packet Length Max 2. Fwd Packet Length Mean 3.Fwd IAT Std 4.Avg_Fwd_Segment_Size 5.act_data_pkt_fwd 6.Total Length of Fwd Packets 7.Fwd Header Length.1 8.Subflow Fwd Bytes		Random Forest	0.86632	0.85792	0.96824	0.998	0.91064	0.99928	0.99912
Decision Tree	1.Destination Port 2.Total Length of Fwd Packets 3. Average Packet Size 4.Init_Win_bytes_forward 5.Fwd IAT Max 6.Flow IAT Mean 7.Fwd Packet Length Max 8. Fwd Header Length.1		Decision Tree	0.97144	0.96296	0.97952	0.99752	0.97632	0.99944	0.99984

Table 8
RFE with Repeated stratified kfold when number of features=9.

Number of Features= 9 RFE with Repeated Stratified Kfold		Algorithm		With default parameters						
Algorithm	Feature Name			Logistic	Support Vector Machine-Linear	Support Vector Machine – Non-Linear	K-Near Neighbour	Naive Bayes	Decision Tree	Random Forest
Logistic	1.FIN Flag Count, 2.ACK Flag Count, 3. URG Flag Count, 4.Down/Up Ratio, 5.min_seg_size_forward 6.Fwd Packet Length Min 7.Total Backward Packets 8.act_data_pkt_fwd 9.Subflow Bwd Packets	Feature Selection Algorithm with Hyper Tuning N=9	Logistic	0.94304	0.96944	0.98104	0.99496	0.90872	0.99544	0.99536
Random Forest	1.Fwd Packet Length Max 2. Fwd Packet Length Mean 3.Fwd IAT Std 4.Avg_Fwd_Segment_Size 5.act_data_pkt_fwd 6.Total Length of Fwd Packets 7.Fwd Header Length.1 8.Subflow Fwd Bytes 9.Average Packet Size		Random Forest	0.97632	0.9848	0.98784	0.99832	0.91064	0.99936	0.99944
Decision Tree	1.Destination Port 2.Total Length of Fwd Packets 3. Average Packet Size 4.Init_Win_bytes_forward 5.Fwd IAT Max 6.Flow IAT Mean 7.Fwd Packet Length Max 8. Fwd Header Length.1 9.Fwd IAT Total		Decision Tree	0.97408	0.96392	0.97976	0.9976	0.97632	0.99936	0.99984

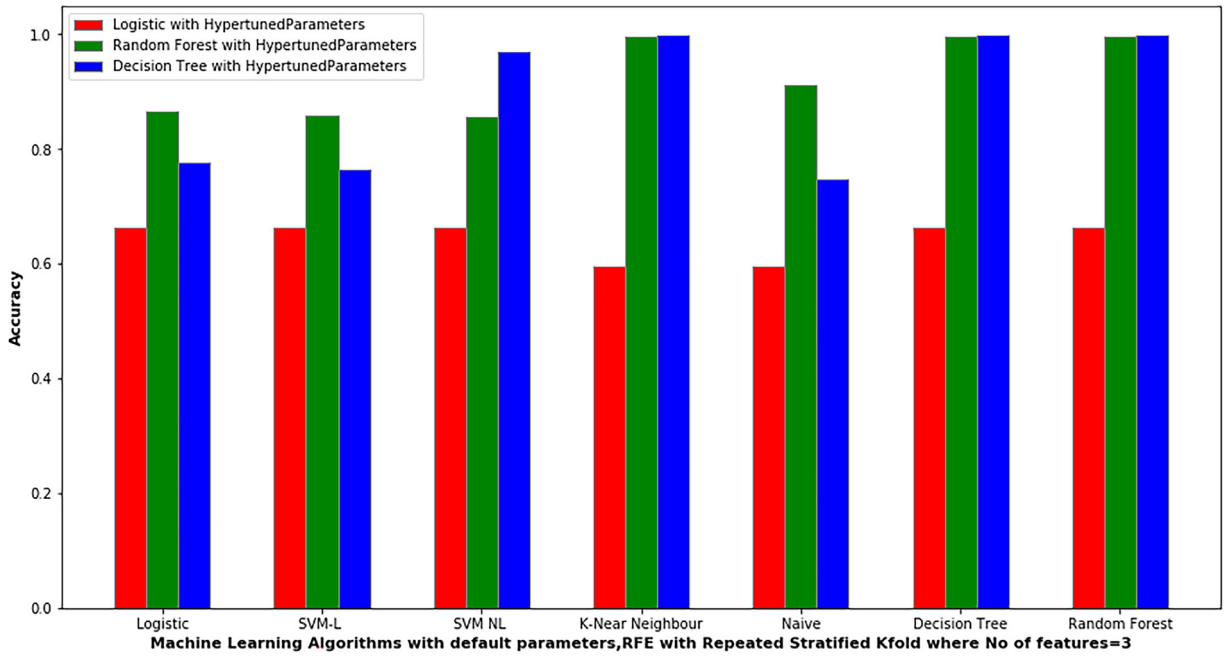

Fig. 8. Accuracy when number of features =3.

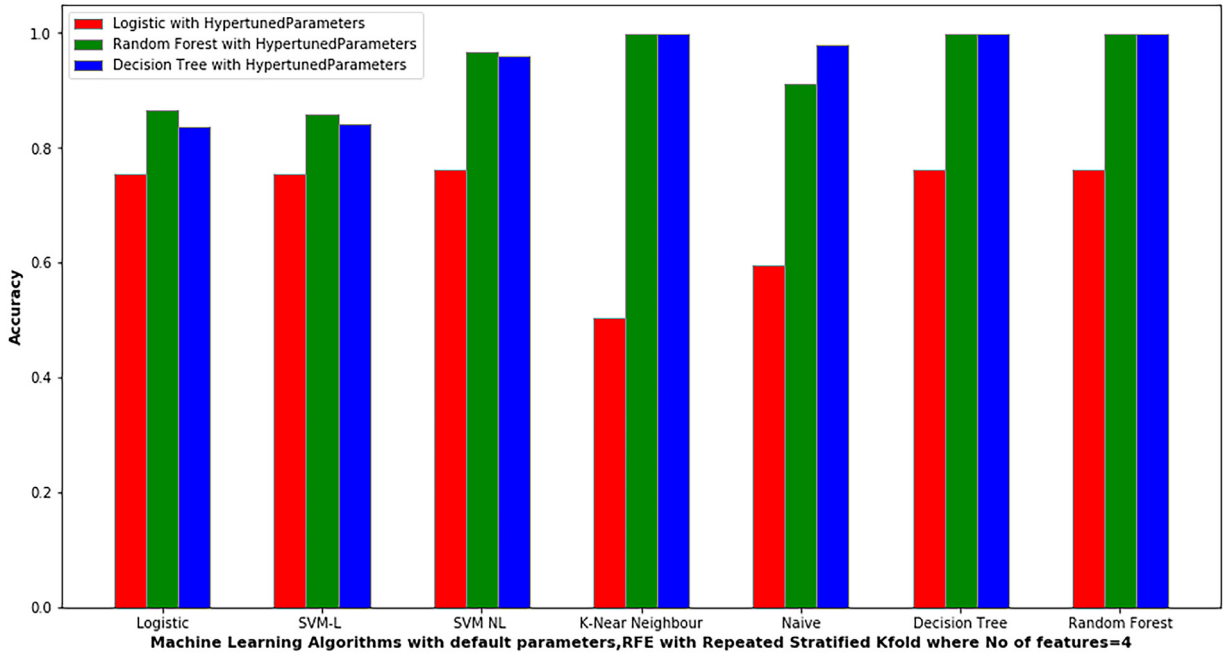

Fig. 9. Accuracy when number of features =4.

The above-mentioned metrics are used in this work for evaluating the proposed IDS. Moreover, the additional metrics such as mean value and the standard deviation values are also used for measuring the performance.

Experimental Results

With the recommended IDS, DDoS and non-DDoS attack may be effectively identified and detected. The recommended IDS is evaluated through the use of many tests. The suggested HRDPA performs the process of feature optimization. In the initial optimization step, the logistic classifier provided C: 25, the penalty was l1, and the decision tree classifier produced the optimal parameters. Gini, maximum squared features, splitter, and Random Forest classifier criteria Max_Features:auto and Bootstrap:False were the best settings that yielded 100 % accuracy. RFE with Repeated Stratified KFold is used in the second optimization step to identify the reduced optimum features and hyper-tuned parameter classifiers. Here, the parameters k, n, and r are set to 10, 3, and 1, respectively. Each classifier is trained using 2 to 9 features, as indicated in Table 1, and its mean accuracy and standard deviation are also presented. In just the 4 characteristics, we achieved the best results. As a result, the second level optimization reduces the 77 features set to 4

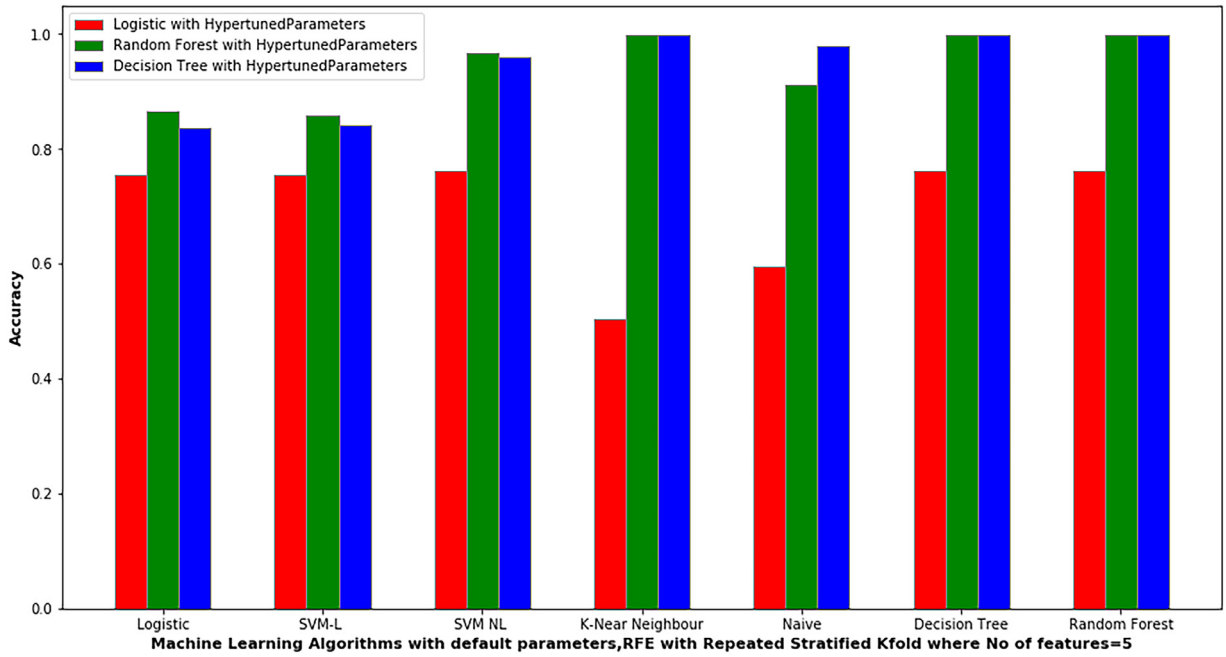

Fig. 10. Accuracy when number of features =5.

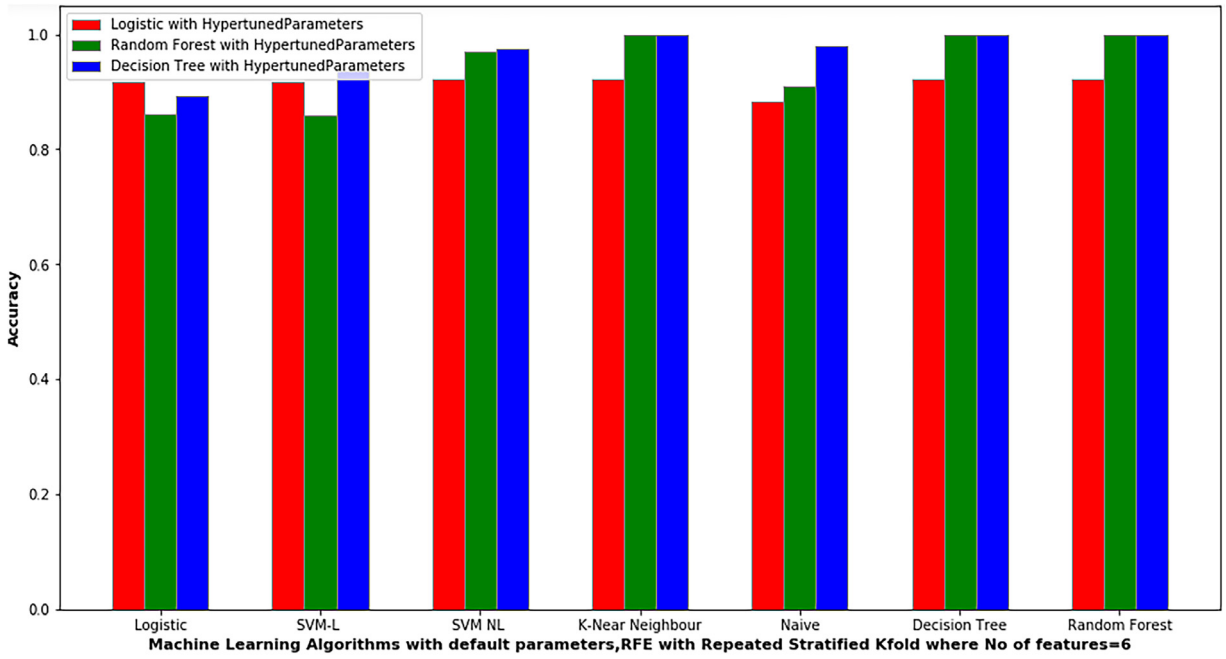

Fig. 11. Accuracy when number of features =6.

features set. For a clear understanding of the ideal feature set, the estimated mean accuracy of three classifiers is displayed in the box plot in Fig. 7.

The Deep Grid Network's input for attack detection is the optimal reduced featured dataset. Here, seven standard parametric ML algorithms are used to hyper-tuned parameters classifiers to create 21 models. For calculating the accuracy of the combined ML algorithms, we took into account a feature set of 2 to 9. Six models—Random Forest-KNN, Random Forest-Decision Tree, Random Forest-Random Forest, Decision Tree-KNN, Decision Tree-Decision Tree, and Decision Tree-Random Forest—are chosen as the top models for effective DDoS attacks detection with 99.99 accuracy based on the accuracy as shown in the Table 2 to 8 and Figs. 8 to 14.

Finally, our own SDN dataset is provided to the RYU controller as input, where our six top models are implemented for efficient DDoS attack detection.

Comparative analysis

Precision, recall, f-measure, and accuracy of the proposed IDS are compared to those of other current ML-based IDSs. Comparative exami-

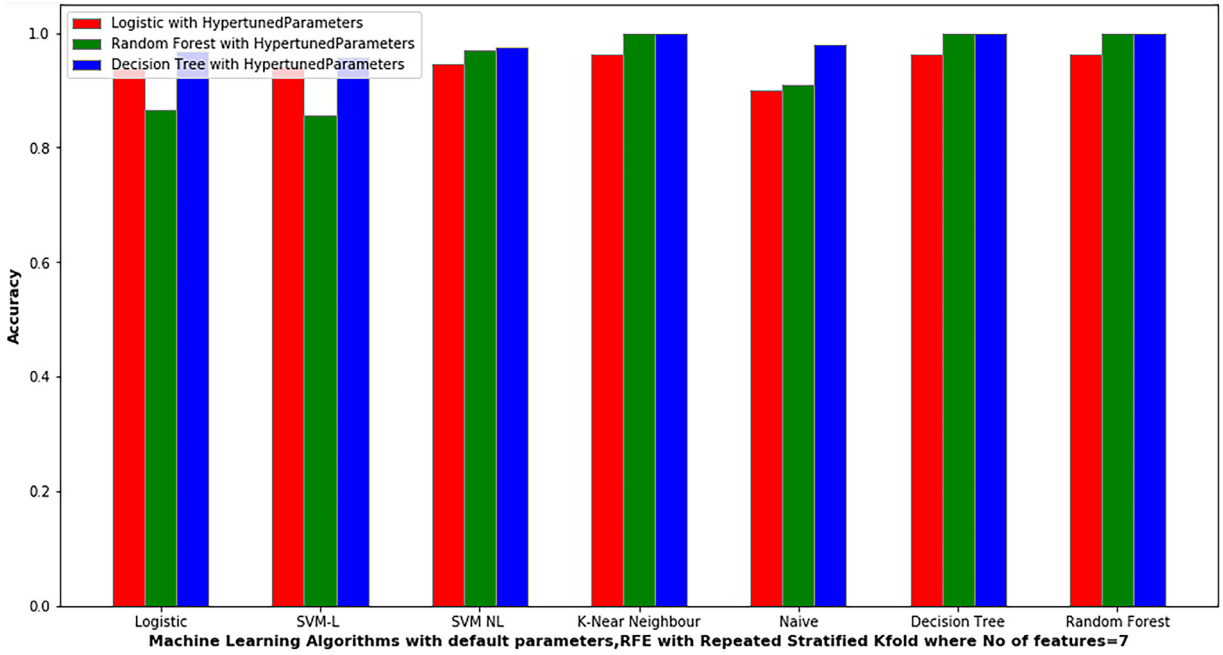

Fig. 12. Accuracy when number of features =7.

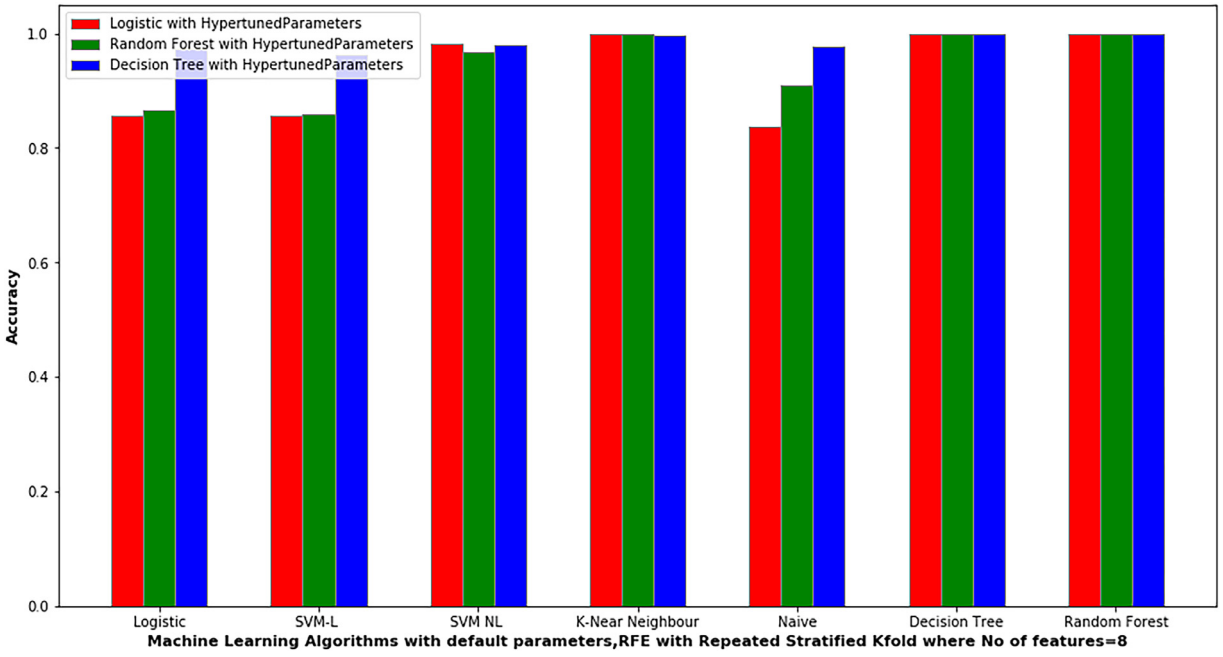

Fig. 13. Accuracy when number of features =8.

nation of the proposed Optimized Dual IDS's accuracy compared to that of the current IDSs is presented in Table 9. Furthermore, a comparison study was done by taking into account the prediction accuracies that were found using the CICIDS 2019 dataset and the newly created dataset.

The table very clearly demonstrates that when compared to the current IDSs, the proposed Optimized Dual IDS produces great accuracy. As a result, the suggested IDS is ideal for all applications to quickly identify the presence of DDoS or Non-DDoS. Because it uses effective hyper-tuned ML classifiers, the ensemble technique, the Recursive Feature Elimination method with Repeated Stratified K-Fold, Deep Grid Network, and effective model selection procedures, the recommended Optimized Dual IDS works effectively.

Fig. 15 shows the prediction accuracy analysis between the proposed Optimized Dual IDS on standard dataset and the newly generated dataset, and the various ML based IDSs that are developed by various researchers in the past [31–35]. Here, the five experiments such as E1-E5 have been conducted by considering the different number of record sets for the comparativeanalysis.

The proposed Optimized Dual IDS outperforms the existing IDSs developed by Dehkordi et al. [31], Tan et al. [32], Ye et al. [33], Sahoo et al. [34], and Muhammad et al. [35], as can be seen in the image. Because it employs effective Hyper parameter tweaking processed ML classifiers, ensemble technique, Recursive Feature Elimination method, Deep Grid Network, and efficient model selection strategy, the recommended Optimized Dual IDS works well.

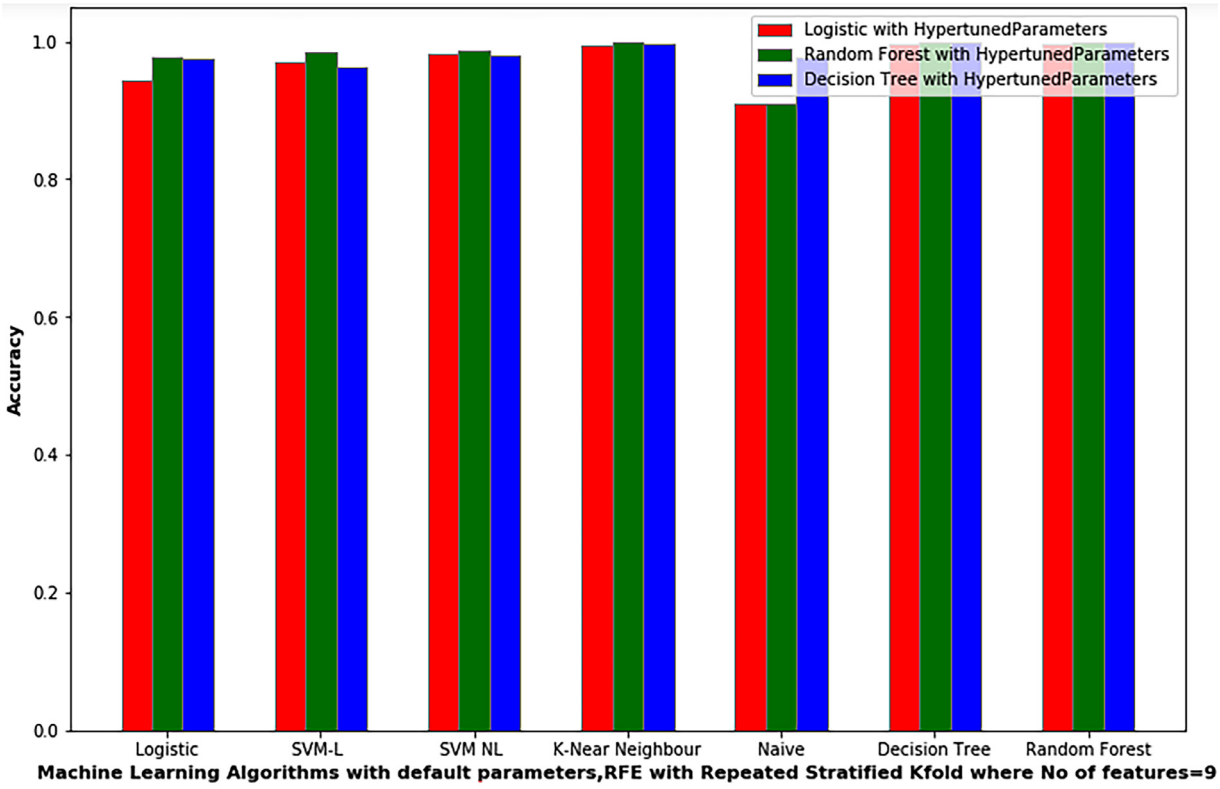

Fig. 14. Accuracy when number of features =9.

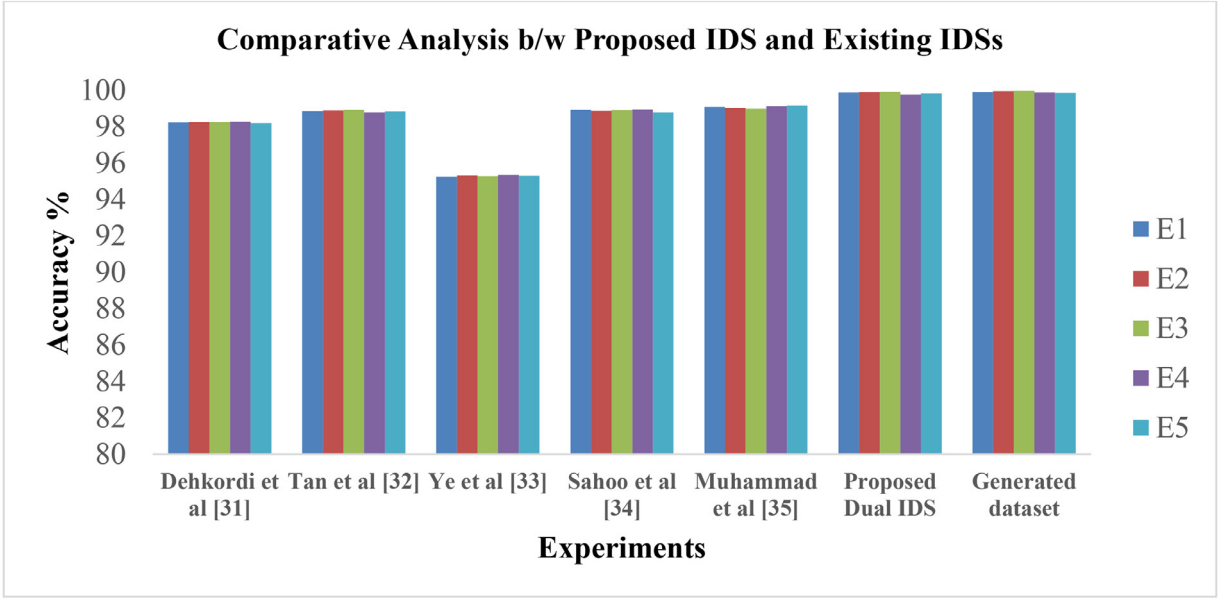

Fig. 15. Comparative Analysis.

Table 9
AccuracyAnalysis between the proposed IDS and existing IDSs.

Intrusion Detection Systems	Dataset	Accuracy
Entropy based IDS [31]	UNB-ISCX, CTU-13 and ISOT	99.85
K means & KNN based IDS [32]	NSL-KDD	98.85
SVM based IDS [33]	Dataset generated viaMininet	95.24
SVM with Kernel Principal Component Analysis using GA(Genetic Algorithm) based IDS [34]	NSL-KDD	98.90
Random Forest with RFE based IDS [35]	NSL-KDD	99.97
Proposed IDS: Dual Optimized IDS:	CICIDS 2019	99.99
HRDPA with DGN & Ensemble approach	Dataset generated via Mininet	100

Conclusion and future work

For detecting DDoS and Non-DDoS assaults, a new Optimized Dual IDS is created and put into use. It has higher accuracy compared to the current IDSs. The suggested IDS incorporates the recently developed HRDPA data preprocessing technique, which chooses the best features that are helpful for boosting the classifier's prediction accuracy using the RFE approach, Hyper tuned parameters ML classifiers, and Repeated Stratified K-Fold process. Finally, a new Deep Grid Network was developed that makes use of the machine learning classifiers LC, RF, DT, NB, Linear SVM, and Non-Linear SVM to effectively analyze the network dataset and achieve improved accuracy of 99.99 %.The suggested IDS also include a new ensemble approach for improving forecast accuracy. Due to its dual optimization characteristics, the proposed IDS additionally confirmed the accuracy of our six top models using its own dataset and in comparison to other datasets, demonstrating that these models are the most effective at detecting DDoS. The suggested approach has been shown to be the most effective for all small- and medium-scale applications. For high end applications, our proposed models struggles to find the complex patterns, it misses out unknown attacks. Also the training time is high during DGN process in our case. Deep learning algorithms can be used to identify all complex patterns to address this issue in our future research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Nalayini C.M.: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Conceptualization. **Jeevaa Katiravan:** Validation, Supervision, Software, Resources, Project administration, Formal analysis, Data curation. **Geetha S.:** Methodology, Resources. **Christy Eunaicy J.I.:** Validation, Visualization.

References

[1] C.M. Nalayini, Dr. JeevaaKatiravan, Detection of DDoS attack using machine learning algorithm, J. Emerging Technol. Innov. Res. 9 (7) (July 2022).
[2] Roberto Lubna Fayez Eliyan, Pietro Di, Hamad Bin Khalifa, DoS and DDoS attacks in software defined networks: a survey of existing solutions and research challenges, Future Gener. Comput. Syst. 122 (2021) 149–171.
[3] Y. Zhang, L. Cui, W. Wang, Y. Zhang, A survey on software defined networking with multiple controllers, J. Network Comput. Appl. 103 (3) (2018) 101–118.
[4] C M Nalayini, Jeevaa Katiravan, V Sathya, Intrusion Detection in Cyber Physical Systems Using Multichain, Malware Analysis and Intrusion Detection in Cyber-Physical Systems, IGI Global Publisher, June 2023, doi:10.4018/978-1-6684-8666-5.ch009.
[5] E. Molina, E. Jacob, Software-defined networking in cyber-physical systems: A survey, Comput. Electr. Eng. 66 (11) (2018) 407–419.
[6] A. Mondal, S. Misra, I. Maity, AMOPE: Performance analysis of openflow systems in software-definednetworks, IEEE Syst.J. 14 (1) (2019) 124–131.
[7] M. Conti, C. Lal, R. Mohammadi, U. Rawat, Lightweight solutions to counter DDoS attacks insoftwaredefinednetworking, Wireless Networks 25 (5) (2019) 2751–2768.
[8] Hamad Bin Doha, DoS and DDoS attacks in software defined networks: a survey of existing solutions and research challenges, Future Gen. Computer Syst. 122 (2021) 149–171.
[9] C.M. Nalayini, J. Katiravan, Block link flooding algorithm for TCP SYN flooding attack, in: International Conference on Computer Networks and Communication Technologies, Lecture Notes on Data Engineering and Communications Technologies, 15, 2018, pp. 1–14.
[10] Huseyin Polat, Onur Polat, Aydin Cetin, Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models, Sustainability 12 (1035) (2020) 1–21.
[11] Song Wang, Juan Fernando Balarezo, Karina Gomez Chavez, Akram Al-Hourani, Sithamparanathan Kandeepan, Muhammad Rizwan Asghar, Giovanni Russello, Detecting flooding DDoS attacks in software defined networks using supervised learning techniques, https://doi.org/10.1016/j.jestch.2022.101176, Eng. Sci. Technol..
[12] Sannasi Ganapathy, Pandi Vijayakumar, Palanichamy Yogesh, Arputharaj kannan, an intelligent CRF based feature selection for effective intrusion detection, Int. Arab J. Inf. Technol. (IAJIT) 13 (1) (2016) volumeissue.
[13] J Wang, R Wen, J Li, F Yan, B Zhao, F. Yu, Detecting and mitigating target link-flooding attacks using SDN, IEEE Trans. Dependable Secure Comput. 16 (6) (2018) 944–956.
[14] Nada M. AbdelAzim, Sherif F. Fahmy, Mohammed Ali Sobh, Ayman M. BahaaEldin, A hybrid entropy-based DoS attacks detection system for software defined networks (SDN): A proposed trust mechanism, Egypt. Inform. J. 22 (2021) 85–90, doi:10.1016/j.eij.2020.04.005.
[15] Q. Yan, F.R. Yu, Distributed denial of service (DDOS) attacks in Software-defined networking with cloud computing environments, IEEE Commun. Magazine 53 (4) (2015) 52–59.
[16] N.I. Mowla, I. Doh, K. Chae, CSDSM: Cognitive switch-based DDoS sensing and mitigation in SDN-driven CDNi word, Comput. Sci. Inf. Syst. 15 (2018) 163–185.
[17] S Dong, M. Sarem, DDoS attacks detection method based on improved KNN with the degree of DDoS attacks in software-defined networks, IEEE Access. 8 (2019) 5039–5048.
[18] Kimmi Kumari, M. Mrunalini, Detecting denial of service attacks using machine learning algorithms, J. Big. Data (2022), doi:10.1186/s40537-022-00616-0.
[19] T. Dang-Van, H. Truong-u, A multi-criteria based software defined networking system Architecture for DDoS attacks mitigation, REV J. Electr. Commun. 6 (3-4) (2016) 21–32.
[20] L. Linxia, V.C.M. Leung, L. Chin-Feng, Evolutionary algorithms in software defined networks: techniques, applications, and issues, ZTE Commun. 15 (3) (2017) 12–19.
[21] C.M Nalayini, Jeevaa Katiravan, A new IDS for detecting DDoS Attacks in wireless networks using spotted hyena optimization and fuzzy temporal CNN, J. Internet Technol. 24 (1) (Jan 2023).
[22] Y Jia, F Zhong, A Alrawais, B Gong, X. Cheng, Flowguard: an intelligent edge defense mechanism against IoTDDoS attacks, IEEE Internet. Things. J. 7 (10) (2020) 9552–9562.
[23] Jin Ye, Xiangyang Cheng, Jian Zhu, Luting Feng, Ling Song, A DDoS attacks detection method based on SVM in software defined network, Secur. Commun. Networks (2018) 1–8, doi:10.1155/2018/9804061.
[24] S Chakraborty, S. Banerjee, Proposed approach to detect distributed denial of service attacks in software defined network using machine learning algorithms, Int. J. Eng. Technol. 7 (2018) 472–476.
[25] JA Pérez-Díaz, IA Valdovinos, KK Choo, D. Zhu, A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning, IEEE Access. 8 (2020) 155859–155872.
[26] HA Alamri, V Thayananthan, Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against DDoS attacks, IEEE Access. 8 (2020) 194269–194288.
[27] D Jankowski, M Amanowicz, On efficiency of selected machine learning algorithms for intrusion detection in software defined networks, Int. J. Electr. Telecommun. 62 (2016) 247–252.
[28] NalayiniI C.M., Gayathri T, A Comparative Analysis of standard classifiers with CHDTC to detect credit card fraudulent transactions: Sivasubramanian, A., Shastry, P.N., Hong, P.C. (eds) Futuristic Communication and Network Technologies, VICFCNT 2020, Lecture Notes in Electrical Engineering, 792, Springer, Singapore, https://doi.org/10.1007/978-981-16-4625-6_99.
[29] Mininet Commands at http://mininet.org/
[30] Wireshark at https://www.wireshark.org
[31] A.B. Dehkordi, M. Soltanaghaei, F.Z. Boroujeni, The DDoS attacks detection through machine learning and statistical methods in SDN, J. Supercomput. 77 (3) (2021) 2383–2415.
[32] L. Tan, Y. Pan, J. Wu, J. Zhou, H. Jiang, A new framework for DDoS attacks detection and defense in SDN environment, IEEe Access. 8 (2020) 161908–161919.
[33] J. Ye, X. Cheng, J. Zhu, L. Feng, L. Song, A DDoS attacks detection method based on SVM in software defined network, Secur. Commun. Networks 2018 (4) (2018) 11–23.
[34] K.S. Sahoo, B.K. Tripathy, K. Naik, S. Ramasubbareddy, B. Balusamy, An evolutionary SVM model for DDoS attacks detection in software defined networks, IEEE Access. 8 (2020) 132502–132513.
[35] Muhammad Waqas Nadeem, Hock Guan Goh, Vasaki Ponnusamy, Yichiet Aun, DDoS detection in SDN using machine learning techniques, Comput. Mater. Contin. 71 (1) (2022) 1–12.