

# Empowering SDN with DDoS attack detection: leveraging hybrid machine learning based IDPS controller for robust security

Florance G.<sup>1</sup>, R. J. Anandhi<sup>2</sup>

<sup>1</sup>Department of Computer Science and Engineering, New Horizon College of Engineering, Affiliated to Visvesvaraya Technological University, Bangalore, India

<sup>2</sup>Department of Information Science and Engineering, New Horizon College of Engineering, Affiliated to Visvesvaraya Technological University, Bangalore, India

## Article Info

### Article history:

Received Sep 3, 2024

Revised Feb 26, 2025

Accepted Mar 15, 2025

### Keywords:

DDoS attack

IDPS controller

OpenDaylight controller

POX controller

Software defined network

## ABSTRACT

Software-defined network (SDN) is an innovative networking framework where a centralized controller manages networking administration and sorts out network traffic issues. It becomes difficult for the controller to identify the malicious user who is sending a large number of spoofed packets, such as in a distributed denial of service (DDoS) attack. To prevent DDoS attacks from damaging legitimate users, it is important to take steps to prevent them. The issue of preventing DDoS attacks in SDN remains unresolved despite many algorithms proposed. Methods presented in this paper employ bandwidth threshold estimation, which triggers the intrusion detection and prevention system (IDPS) controller if the threshold is exceeded. Whenever the threshold is exceeded due to network congestion, transferred packets are filtered at the server level by identifying the utilization of bandwidth in OpenDaylight (ODL) and POX. K-nearest neighbor (K-NN) and support vector machine (SVM) are used by the IDPS controller to detect and thwart DDoS attacks. Using Mininet, two SDN centralized controllers are simulated to improve performance significantly. Based on SVM in the ODL controller, this work has provided mitigation techniques for preventing DDoS attacks with an accuracy of 96.75% compared to previously published accuracy.

*This is an open access article under the [CC BY-SA](#) license.*



## Corresponding Author:

Florance G.

Department of Computer Science and Engineering, New Horizon College of Engineering

Affiliated to Visvesvaraya Technological University

Bangalore, India

Email: vijiflorance59@gmail.com

## 1. INTRODUCTION

The enormous change in network migrations and the increase in many distributed applications on the internet has become more complex in configuration and management. The conventional network infrastructure and other related evolving technologies manage the enormous amount of data. It is difficult when the user needs to change the requirements and dynamic configuration of the network infrastructure. When it is necessary to implement a new protocol, updating rules and policies for all networking devices becomes costly and time-consuming. To address these, software software-defined network (SDN) is preferable. SDN provides simplified management, a central controller, programmability, automated, and dynamic management [1].

SDN is a more popular innovative network paradigm that provides efficient management by uncoupling the data and control planes where the centralized controller is placed [2]. The data transfer

between the forwarding devices and controller by a standard protocol called OpenFlow (OF), which segregates the packets based on OF instructions. In SDN, if any updating of policies is needed, then the modification happens in the controller, like minimizing the time consumption and cost of the entire process. During data transfer, SDN switches store the incoming packets in the flow table (FT) [3]. Each entry in the FT consists of three fields: action, rule, and counter. Based on FT information, packets are getting dropped [4], [5]. The data security provided by the SDN network's controller monitors malicious traffic. SDN enhances network infrastructure management and addresses challenges in terms of security, programmability and load balancing. SDN security is a significant challenge than denying network service due to malicious traffic [6], [7]. Distributed denial of service (DDoS) attacks are service-denying attacks that exhaust the benefits of the entire network and saturate the bandwidth, which stops servicing legitimate user requests. Due to many spoofed packages, the entire network gets flooded, resulting in bandwidth depletion. This flooding leads to degrading the system's performance [8], [9].

Machine learning (ML) is an application that classifies packets using intrusion detection and prevention system (IDPS) controllers in SDN [10], [11]. Techniques such as k-nearest neighbor (K-NN), naive Bayes, k-means, and support vector machine (SVM) are employed to detect anomalous packets in IDPS. Anomaly-based IDPS systems use the concept of network behavior. ML algorithms assist set of traffics generated by the Waikato environment for knowledge analysis (WEKA) tool and provide predictions for training data derived from traffic patterns [12], [13]. Consequently, ML-based detection techniques are used to identify spoofed packets based on abnormal changes in the network, and their performance depends on characteristics considered for training [14]–[16].

Jawahar *et al.* [17] proposed ML models that use the Ethereum blockchain to list malicious IP addresses. Jebril *et al.* [18] presented a convolutional neural network-bidirectional long short-term memory (CNN-BiLSTM) model, using three deep-learning algorithms. The performance of this model is evaluated by performance criteria. Songa and Karri [19] introduced, features selection is combined as a cluster to analyze the traffic. Zhao and Yang [20] demonstrated the unfortunate flooding that spoils the network that are taken care by the adaptive resilient control design. Zhao *et al.* [21] focused on resilient event-driven dynamic feedback control systems, that is used for the impact of DDoS attack and regulate the required anticipated data transmission. Revathi and Devi [22] focused on hybrid intrusion detection method using the novel deep learning method modified hybrid deep belief network with weights (MHDBN-W). Rajan and Aravindhar [23] proposed CNN-LSTM techniques showing their superiority in spotting malicious. Mansoor *et al.* [24] proposed an approach that uses, data preprocessing by transformation, feature selection by chi-square, and detecting attacks using recurrent neural networks (RNNs) model. Yao *et al.* [25] focused on deception attacks on switched Takagi-Sugeno (T-S) fuzzy systems.

Zhou *et al.* [26] proposed a framework to handle multi-type DDoS attacks using a feature-based method. Krishnan *et al.* [27] focused on a more predictable, secured, and reliable network using the SDN, network function virtualization, and machine learning. Musumeci *et al.* [28] designed for DDoS attack detection framework with ML assistance by automatically retrieving packet information. Swati *et al.* [29] described the capability-based architecture of a transient effect ring oscillator PUF.

Ye *et al.* [30] proposed an SVM model to detect DDoS attacks by determining flow status information periodically and it extracted from the switch FT. Sumathi *et al.* [11] discussed hybrid ML techniques used to design the intrusion detection system (IDS). Bharot *et al.* [31] describe the Hellinger distance function that is used to compare the incoming and baseline traffic with probability. DDoS attack detection is based on finding entropy and standard deviation for monitoring incoming packets with two levels of the threshold value in [32].

The main contribution of this paper is to address the challenge of detecting and preventing DDoS attacks within SDN. In SDN, a centralized controller manages network administration and traffic, which makes it difficult to identify malicious users, particularly in the case of DDoS attacks involving large volumes of spoofed packets. To mitigate this issue, the paper proposes a solution that uses bandwidth threshold estimation in conjunction with an IDPS controller. The IDPS controller, in turn extracts feature and classifies packets as normal or malicious using SVM and K-NN classification algorithms.

Section 2 illustrates the proposed skeleton for mitigating DDoS attack detection strategies. Section 3 presents the simulation and performance assessment of the DDoS attack detection framework. Section 4 concludes the paper and discusses the related future enhancement of the research scope.

## 2. RESEARCH METHOD

### 2.1. Determination of bandwidth threshold

The bandwidth threshold estimation filters the traffic when the traffic occupies the bandwidth that exceeds a certain limit. This estimation uses the total bandwidth  $C$ , the bandwidth ratio is  $B_0$ , the amount of

traffic for legitimate and malicious packets, and evaluates the performance in terms of congestion when there is a fluctuation of the current bandwidth ratio  $B_0$ . Firstly, the amount of traffic calculated using (1).

$$Traf(T, t) = \sum_{i=1}^N \frac{BR_i}{B_0} \quad (1)$$

The average link delay is denoted in (2),

$$\begin{aligned} Avg_{di} &= \frac{D}{P} \\ &= \frac{(Pktrectime - pktsenttime)}{transmission\_speed} \end{aligned} \quad (2)$$

where,  $Pktrectime$  is packet receive time and  $Pktsenttime$  is packet sent time. The probability of delay is denoted as,  $y_i$ .

$$y_i = prob(D > T_i)$$

The main goal is to define the bandwidth threshold, which is 75% of the assigned bandwidth. By fine-tuning  $C$  and  $B_0$ , performance is increased. Now there are two cases as,  $L$  is the set of completely successful requests (legitimate packets) and  $M$  is the set of requests that are not completely successful (malicious packets). Consider, the network which is not congested with a set of requests, and changing current bandwidth ratio in (3) and (4).

$$\Delta B_L = + \left( \frac{\sum_{i \in L} B_i}{B_0} \right) \quad (3)$$

The network which is congested with requests and changing current bandwidth ratio.

$$\Delta B_M = - \left( \frac{\sum_{i \in M} B_i}{B_0} \right) \quad (4)$$

The number of packets received per source IP address is defined using entropy calculation in (5).

$$\begin{aligned} H(B_{SIP}) &= \sum_{i=1}^n P(B_{SIP_i}) \log_2(B_{SIP_i}) \\ P(B_{SIP_i}) &= P(B_{SIP_1}) + P(B_{SIP_2}) + \dots + P(B_{SIP_n}) \end{aligned} \quad (5)$$

The supporting value of bandwidth for each source IP address is found in (6).

$$S(B_{SIP}) = \frac{H(B_{SIP})}{\Delta B_L + \Delta B_M} \quad (6)$$

The utility function in the bandwidth threshold value of 75% for bandwidth of each source IP is defined in (7).

$$\begin{aligned} U_F(B_{SIP}) &= S(B_{SIP}) * \sum_{i=1 \text{ to } n} B_i \\ U_F(B_{SIP}) &= \Delta B_L, \\ 0.01 &\leq U_F(B_{SIP}) \leq 0.75 \\ U_F(B_{SIP}) &= \Delta B_M, \quad U_F(B_{SIP}) \geq 0.76 \end{aligned} \quad (7)$$

The average utility of each source IP address [16] is defined in (8).

$$A_{US} = \frac{U_F(B_{SIP})}{|N|} \quad (8)$$

The minimum bandwidth threshold is found in (9).

$$Min_{BT} = A_{US} = \frac{U_F(B_{SIP})}{|N|} \quad (9)$$

If the bandwidth ratio of malicious requests increases, it will consume more bandwidth of packets received from each source IP address; hence, the restriction of malicious packets is done using threshold bandwidth.

The notation used in the proposed bandwidth threshold estimation is shown in Table 1. Algorithms 1 to 4 depict the pseudocode of the proposed workflow.

Table 1. Notations used in the proposed derivation of bandwidth threshold estimation

Notation	Description
$Traf(T, t)$	Amount of traffic
$N$	Number of requests
$BR_i$	Bandwidth resource of $i^{th}$ request
$B_0$	Current bandwidth ratio
$Avg_{di}$	Average delay
$D$	Link distance
$P$	Processing delay
$y_i$	Probability of delay
$T_i$	Delay threshold
$C$	Total bandwidth
$\Delta B_L$	Bandwidth ratio for legitimate request
$B_i$	Bandwidth of $i^{th}$ request
$\Delta B_M$	Bandwidth ratio for malicious request
$H(B_{SIP})$	Entropy value of bandwidth of packets
$P(B_{SIP_i})$	Probability of bandwidth utilized by the IP address
$U_F(B_{SIP})$	Utility function defined in bandwidth threshold
$A_{US}$	Average utility of each source IP address
$Min\_BT$	Minimum bandwidth threshold

#### Algorithm 1. Bandwidth threshold estimation

```

Est_band_thres ()
1: Initially set  $Traf(T, t) = \sum_{i=1}^N \frac{BR_i}{B_0}$ 
2: Calculate  $\Delta B_L, \Delta B_M, H(B_{SIP}), S(B_{SIP})$  using (3), (4), (5) and (6)
3: Find  $U_F(B_{SIP}) = S(B_{SIP}) * \sum_{i=1 to n} B_i$ 
4: If  $(0.01 \leq U_F(B_{SIP}) \leq 0.75)$  then  $U_F(B_{SIP}) = \Delta B_L$ 
5: Else if  $(100 \geq U_F(B_{SIP}) \geq 0.76)$  then  $U_F(B_{SIP}) = \Delta B_M$ 
6: Determine  $A_{US} = \frac{U_F(B_{SIP})}{|N|}$ 
7: Return  $Min_{BT} = A_{US} = \frac{U_F(B_{SIP})}{|N|}$ 
8: Monitoring the server that exceeded the bandwidth threshold i.e.,
   If  $(100 \geq U_F(B_{SIP}) \geq 0.76)$  then invoke Algorithm 2
9: Exit

```

#### Algorithm 2. IDPS controller

```

IDPS_controller ()
1: Generated traffic using Tor's hammer tool and it's captured by the Wireshark tool is
   depicted in Table 2.
2: Select the features by calculating the variables  $NSIPA, NSP, SDR, EFP, EFB$  and  $FMR$  shown in
   Table 3
3: Invoke the Algorithm 3
4: Exit

```

#### Algorithm 3. Classification algorithms

```

Classify_Algo ()
1: SVM ()
   Construct the hyperplane by  $P1: \omega^T \cdot x + b = +1$  and  $P2: \omega^T \cdot x + b = -1$  then read the dataset
   mentioned in Table 3. Invoke Perform_metrics ();
2: K-NN ()
   - Read the dataset and define K value.
   - Obtained Euclidian distance  $ED = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$  from the dataset.
   - Decide the most frequently matching with the least K value.
3: Invoke Algorithm 4
4: Exit

```

#### Algorithm 4. Performance metrics

```

Perform_metrics ()
1: Calculate Accuracy, FAR, Recall, Precision, and F-measure.
2: Exit.

```

### 2.2. Packet filtration using hybrid machine learning model in server

The proposed IDPS controller is an application that originated to investigate malicious behaviors. It is installed in a server that describes the functions to detect DDoS attacks.

### 2.2.1. Traffic origination and packet capturing

In the SDN environment, the foremost step to detect a DDoS attack is to generate network traffic using Tor's hammer network tool as shown in Figure 1. It can cause huge traffic in network flow, making service unavailable to the legitimate user. One of the most popular tools used to capture the traffic in the network is the Wireshark tool. This tool monitors and captures network traffic in the FT via the server. When the estimated bandwidth threshold reaches a specific percentage of bandwidth, it starts maintaining the traffic logs for future analysis. This log extracts the source and destination IP address, source and destination port, protocol, time, TTL value, and information collected from packet headers like SYN, FIN, ACK, Seq, and Len.

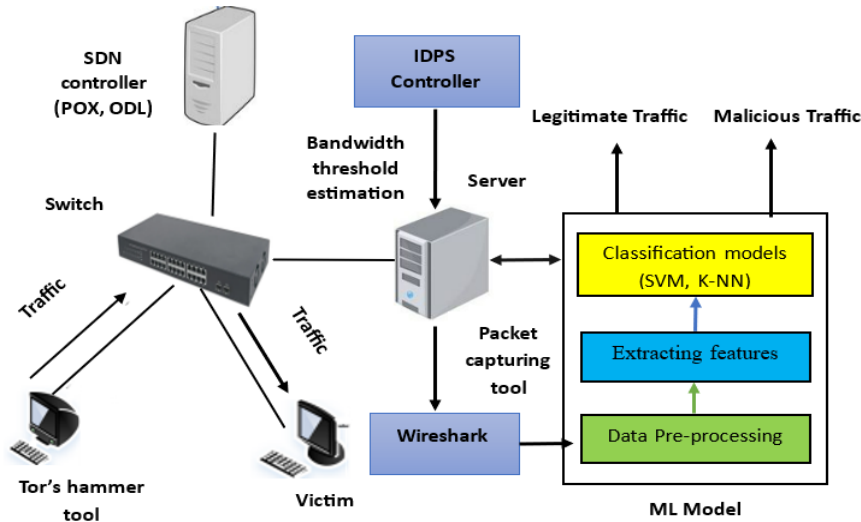


Figure 1. Proposed IDPS controller in SDN

Table 2. Generated dataset

server	src	dst	pkt count	byte count	duration	dur_nsec	tot_dur	flows	pktrate	protocol	port_no	tx_bytes	rx_bytes
1	10.0.1.1	10.0.0.15	54702	56474171	100	643200000	1.07E+11	2	278	OF	1	3619	941
2	10.0.7.1	10.0.4.13	117234	139136078	250	739000000	3.01E+11	4	591	OF	3	4871	4257
1	10.0.5.2	10.0.3.25	85734	89194991	100	689000000	1.71E+11	5	342	OF	5	3614	1521
3	10.0.9.2	10.0.0.15	92341	954131294	180	712000000	1.99E+11	2	695	OF	3	4015	1027

Table 3. Six-tuple characteristic values with its description

Parameter	Formula	Description
Number of source IP addresses (NSIPA)	$NSIPA = \frac{Tot\_Src\_IP}{T}$	The total number of source IP addresses per sampling interval is denoted in time.
Number of source port (NSP)	$NSP = \frac{Tot\_Src\_port}{T}$	The total number of ports sending packets per sampling interval is denoted in time.
Speed of data rate (SDR)	$SDR = \frac{Tot\_pkt}{T}$	The number of packets transferred per second unit.
Entropy value of flow packets (EFP)	$EFP = \sum_{i=1}^{FN} \frac{P(NPF_i) \log_2(NPF_i)}{FN}$	The size of packets entering to FT is estimated by the total number of packets in a flow (NPF) per flow number (FN).
Entropy value of flow bytes (EFB)	$EFB = \sum_{i=1}^{FN} \frac{P(NBF_i) \log_2(NBF_i)}{FN}$	The size of bytes entering to FT is estimated by the total number of bytes in a flow per flow number (FN).
FT matching ratio (FMR)	$FMR = \frac{Matched\_Flow}{NFE}$	The number of flows is getting matched with the switch FT entry.

### 2.2.2. Selection of characteristic values

Once network traffic is captured and pre-processing is done, the next is to extract features and create a dataset. The ML model converts the necessary fields from the packet header into a readable CSV format. This paper compares with the existing analysis [12] of characteristic values needed for classifying legitimate and

malicious traffic. For this purpose, six distinct tuple values are taken to detect DDoS attacks. SVM and K-NN use the features mentioned in Table 3 obtained from the network traffic to classify normal and abnormal traffic.

### 2.2.3. Classification of network traffic

This section illustrates two classification algorithms used to classify legitimate and malicious traffic: SVM and K-NN. These two algorithms will classify the network traffic using 6 tuples characteristic values.

- Support vector machine: SVM is the most popular classification technique. It collects FT entries in the switch via a server at intervals and determines the six tuple characteristics value to make a sample set Z. The sample set  $Z=(P, Q)$ , where P denotes the value obtained for six tuple characteristics called feature vectors of flow entry, and Q denotes the label marker.  $Q=0$  denotes a legitimate state, and  $Q=1$  denotes a malicious state. The SVM technique uses hyperplane construction to classify the traffic. The training data set  $D=\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  where  $X_i$  represents the input vector and  $Y_i$  represents the label, such as  $Y_i \in \{+1, -1\}$ . The hyperplane is constructed by using the equation,  $\omega \cdot x + b = 0$ , where  $\omega \in R^n, b \in R$  and two planes run parallel to the hyperplane as,  $P1: \omega^T \cdot x + b = +1$  and  $P2: \omega^T \cdot x + b = -1$ . The maximum distance between P1 and P2 is  $\frac{2}{\|\omega\|}$ . To maximize the distance of the hyperplane, need to minimize the  $\|\omega\|$  by using  $\frac{1}{2} \|\omega\|^2$ . Any trained dataset falling beyond P1 and P2 that minimizes the equation  $\frac{1}{2} \|\omega\|^2$  which satisfies the condition,  $y_i (\omega^T \cdot x_i + b) \geq 1 \forall i = 1, 2, \dots, N$ .
- K-nearest neighbor: K-NN algorithm uses the characteristics value similar to the given data query with the training dataset to evaluate classification models. It describes the following steps, i) the model reads the input as a training and testing dataset; ii) defines the K value, which represents the number of nearest points that are in scope; iii) Euclidian distance is calculated by  $ED = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$  based on datasets; iv) once the Euclidian distance is calculated, the least K point is set; and v) a final decision is obtained based on the nearest K point.

Using SVM and K-NN algorithms, a hybrid ML classification algorithm was made to collect the characteristic's value from the server to train the data and classify the normal and abnormal traffic, then use the test data to check whether the designed model is classifying properly or not.

## 3. RESULTS AND DISCUSSION

The SDN network simulation using Mininet with two SDN controllers is, ODL and POX controllers are installed on the 64-bit Ubuntu 14.04 operating system and determine the performance of the two controllers separately. The SDN network topology is created with 15 switches and 16 nodes, shown in Figure 2. ODL controller uses the ODL user experience (DLUX) application interface. This interface enables DLUX karaf features and port 8181 is linked with the local system IP address, then install l2-switch features to invoke data transfer between the nodes is shown in Figure 3. When data is transferred from nodes 1, 2, and 3 to node 15 DLUX module is getting updated is shown in Figure 4.

POX controller is a Python and Linux-based controller. It is installed with a Mininet virtual machine (VM) with miniedit.py. First, start Mininet topology for the POX controller using the command as, `sudo mn --topo single,3 --mac --arp --switch ovsk --controller=remote`. Next open GUI POX controller with created network, ODL provides the visualized representation of data transfer between the nodes and traffic is updated in DLUX module. To detect the DDoS attack, traffic will be generated and captures this traffic using the Wireshark tool as the generated dataset is shown in Table 2.

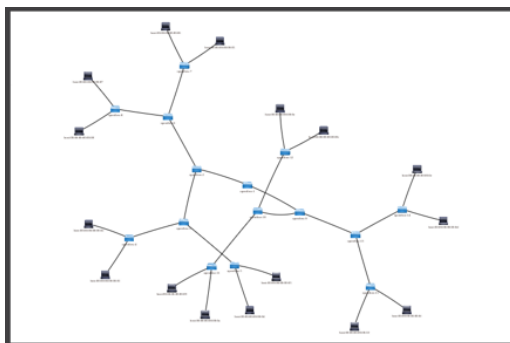


Figure 2. SDN network with 15 switches and 16 nodes

Node ID	Node Name	Node Connections	Statistics
openflow10	625	4	Flow   Node Connections
openflow9	68	4	Flow   Node Connections
openflow9	68	4	Flow   Node Connections
openflow9	68	4	Flow   Node Connections
openflow7	67	4	Flow   Node Connections
openflow4	64	4	Flow   Node Connections
openflow5	65	4	Flow   Node Connections
openflow2	62	4	Flow   Node Connections
openflow3	63	4	Flow   Node Connections
openflow1	61	2	Flow   Node Connections
openflow10	625	4	Flow   Node Connections
openflow14	614	4	Flow   Node Connections
openflow13	613	4	Flow   Node Connections
openflow12	612	4	Flow   Node Connections
openflow11	611	4	Flow   Node Connections

Figure 3. DLUX module in ODL

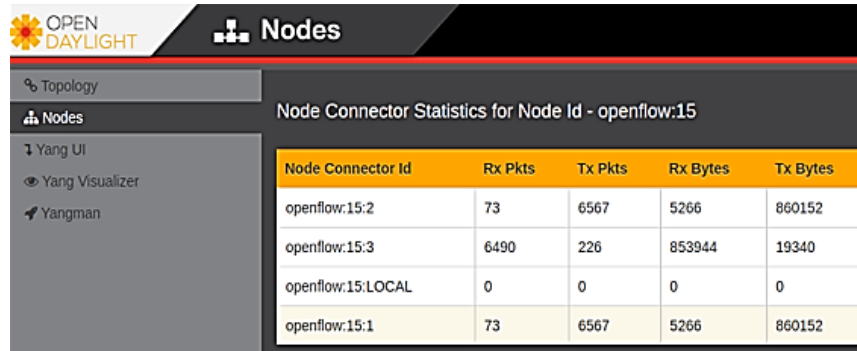


Figure 4. Flows statistics in the DLUX module

### 3.1. Estimate bandwidth threshold

To evaluate the performance of the proposed estimated threshold in two SDN controllers (Figure 5), keep sending the traffic generated by the Tor's hammer tool from nodes 1, 2, and 3 to node 15. Here two scenarios are observed. In the first scenario, the traffic was sent between the sampling period of T15 to T30. This sampling duration calculates the bandwidth utilization of the network using NSIPA, SDR, EFP, EFB, and FMR by IDPS controller. If the bandwidth utilization is less than 75% of the actual bandwidth assigned to the network, then that traffic is legitimate, is shown in Figure 5(a). Similarly, in the second scenario, the same thing will be calculated during the sampling period of T15 to T30, and the bandwidth utilization. If it's greater than 75% of the actual bandwidth then suspect that traffic is malicious, as shown in Figure 5(b).

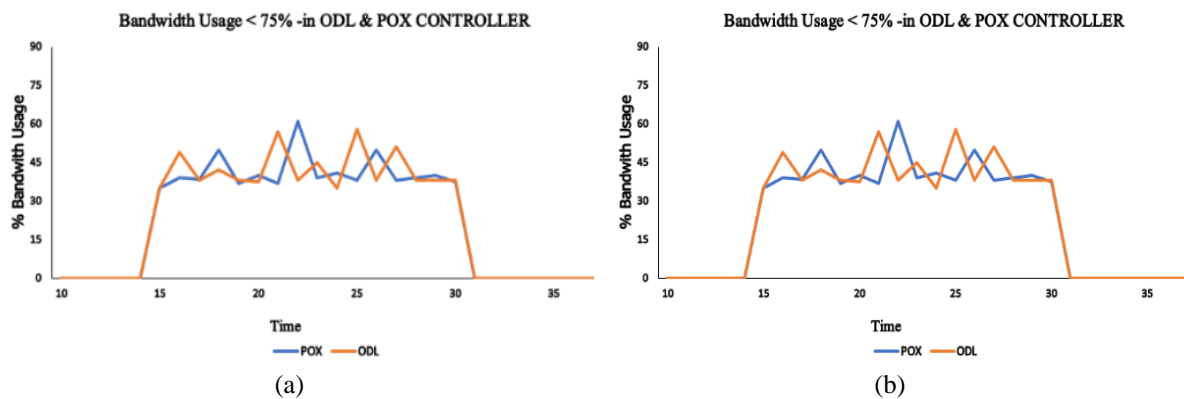


Figure 5. Bandwidth threshold estimation of (a) the bandwidth usage is less than 75% and (b) the bandwidth usage is greater than 75%

### 3.2. Packet filtration using machine learning model in server

Once the partial filtration is done, next move on to server-level filtration for suspecting traffic by calculating the six-tuple characteristics value such as NSIPA, NSP, SDR, EFP, EFB, and FMR in the sampling interval of T15 to T30 periods. During the period of T15 to T30, the six-tuple feature vector value was calculated based on the generated dataset, as shown in Figure 6. Figures 6(a) and 6(b) show the number of IP addresses and number of ports in the particular interval of T15 to T30. The number of packets transferred per second in the interval is shown in Figure 6(c). The entropy value of the flow packet is represented by taking the count of the probability of packets and its logarithmic value per flow number of T15 to T30 is shown in Figure 6(d), the count of the probability of size of packets entered into the FT and its logarithmic value per flow number in the interval of T15 to T30 is in Figure 6(e) and the FT matching ratio is determined by finding number of flows that are getting matched with FT entry per number of flows is in Figure 6(f).

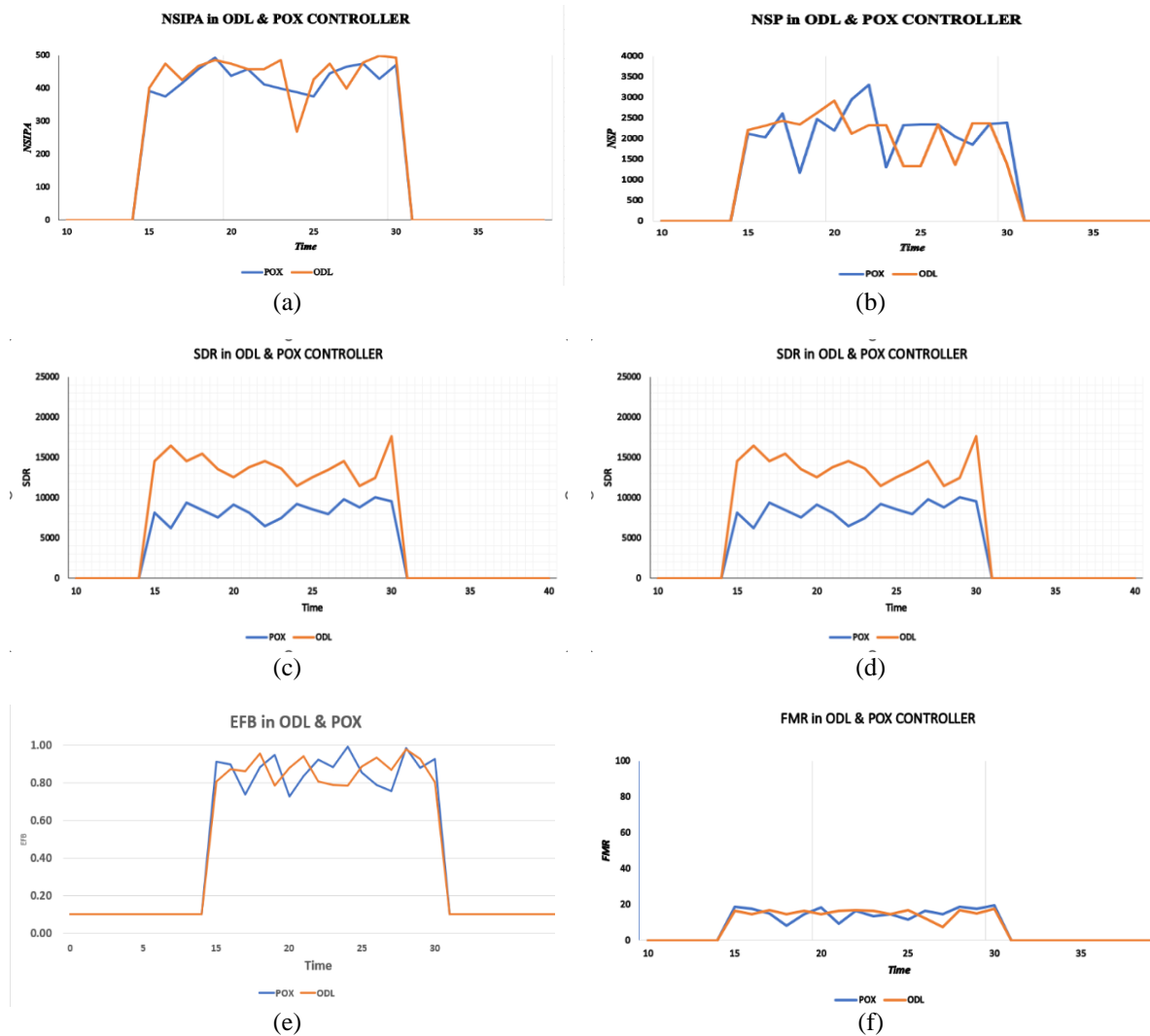


Figure 6. Six tuple characteristic values shown in two controllers as ODL and POX controller of (a) the number of source IP address, (b) number of source port, (c) speed of data rate, (d) entropy value of Flow Packets, (e) entropy value of Flow Bytes, and (f) FT matching ratio

### 3.3. Performance indicators

The SVM and K-NN models are used to train the data using the calculated characteristic values of NSIPA, NSP, EFP, EFB, and FMR and test the data using the models to predict the test data. The performance evaluation of proposed IDPS techniques in ML indicates accuracy, recall, false alarm rate (FAR), precision, and F-measure. It is based on determining true positive (TP) and false positive (FP) shows how many attacks are predicted correctly and incorrectly, true negative (TN) and false negative (FN) gives the how many normal flows has predicted correctly and incorrectly. SVM achieves good results in terms of accuracy, recall, and F-measure in the ODL controller compared to the K-NN algorithm, is shown in Table 4 and performance chart for SVM and K-NN model as shown in Figure 7, its performance of ODL controller is shown in Figure 7(a) and POX controller is shown in Figure 7(b). Table 5 depicts the ML characteristics value and its metrics. Table 6 portrays the comparative investigation taken between the proposed model and the existing model.

Table 4. Performance indicators for SVM and K-NN classifiers

Controllers	Classifier	Accuracy	False alarm rate	Recall	Precision	F-measure
OpenDaylight	SVM	96.95	0.0	96.91	1.0	98.44
	K-NN	96.64	0.0	96.61	1.0	98.18
POX	SVM	96.88	0.0	96.82	1.0	98.34
	K-NN	96.58	0.0	96.5	1.0	98.13



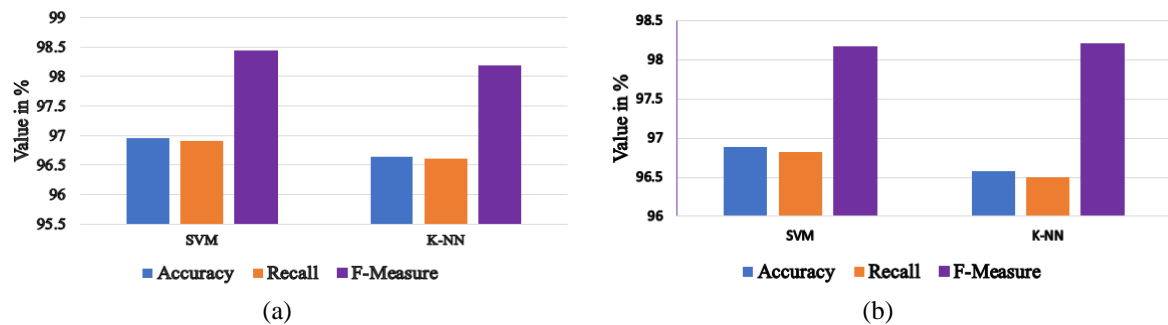


Figure 7. Performance chart for SVM and K-NN model in (a) ODL controller and (b) POX controller

Table 5. ML characteristics values and metrics

Controllers	Algorithm	TP	FP	TN	FN	Accuracy	Recall	Precision	F-measure
ODL	SVM	461	15	412	18	0.9695	0.9691	0.949	0.9844
	KNN	458	17	411	16	0.9664	0.9661	0.943	0.9818
POX	SVM	459	15	415	18	0.9688	0.9682	0.949	0.9834
	KNN	451	17	409	14	0.9658	0.965	0.943	0.9813

Table 6. Comparative analysis of the proposed model with the existing model

Model/ framework	Parameter used	Simulation/datasets	Accuracy (%)
SNORT IDS and trained ML IDS [17]	ip.length, ip.flags.df, tcp.windows_size, tcp.length, ip.frag.offset	IoT device with Ryu controller and UNSW-NB15	94
Artificial neural network (ANN), Random forest [18]	Pkt Len Max, Fwd Pkt Len Max, Pkt Len Min, Fwd Seg Size Min, Fwd Pkt Len Std	Mininet with POX controller and CICDDoS2019	72.49
Modified hybrid deep belief network with weights [23]	Available bandwidth, CPU utilization, PacketIn and PacketOut	CICIDS2018	96.62
Deep learning-RNN [25]	Pktcount, pkttrate, tx_bytes, packetins, pktperflow and protocol	Mininet with Ryu controller and DDOS attack SDN Dataset	94.98
Proposed model (IDPS_controller in POX and ODL controller)	NSIPA, NSP, SDR, EFP, EFB and FMR	Mininet with POX Controller and ODL controller -Generated data set	96.75

#### 4. CONCLUSION

This paper analyses mitigation techniques for the prevention of DDoS attacks using six-tuple characteristic values related to DDoS attacks. The accuracy of finding malicious packets is 96.75% in this study. To achieve a better accuracy bandwidth threshold estimation is derived. The traffic is segregated by using the threshold level of 75% bandwidth by the ODL and POX controllers. After defining the threshold, traffic is generated using Tor's hammed network tool and analyzed using the Wireshark capturing tool in the server, subsequently, the dataset is created. From the dataset, six-tuple feature vectors value is calculated during the sampling interval of T15 to T30 and classified the legitimate traffic and malicious traffic using SVM and K-NN algorithms in ODL and POX controllers. Among these calculated values, NISPA, NSP, EFP, and FMR were used for classifying attacks. In comparing the two controllers, the accuracy rate of SVM is relatively high accuracy in the ODL controller compared to the POX controller and achieving a 96.75% accuracy in detecting malicious packets. In the future, multiple IDPS controllers will be proposed virtually to execute DDoS attacks in multiple places to improve the performance of various classifiers in the SDN network. The dynamic threshold estimation will be embedded in the real-time scenario. Exploring the intelligence of collaborative approach to leverage the network in real-time.

#### FUNDING INFORMATION

This work is not funded by any agency or institute

#### AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Florance G.	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
R. J. Anandhi		✓				✓		✓	✓	✓	✓	✓		

C : **C**onceptualizationM : **M**ethodologySo : **S**oftwareVa : **V**alidationFo : **F**ormal analysisI : **I**nvestigationR : **R**esourcesD : **D**ata CurationO : Writing - **O**riginal DraftE : Writing - Review & **E**ditingVi : **V**isualizationSu : **S**upervisionP : **P**roject administrationFu : **F**unding acquisition

## CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

## DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [FG], upon reasonable request.




## REFERENCES

- [1] P. Kaur, M. Kumar, and A. Bhandari, "A review of detection approaches for distributed denial of service attacks," *Systems Science & Control Engineering*, vol. 5, no. 1, pp. 301–320, Jan. 2017, doi: 10.1080/21642583.2017.1331768.
- [2] B. Isyaku, M. S. Mohd Zahid, M. Bte Kamat, K. Abu Bakar, and F. A. Ghaleb, "Software defined networking flow table management of OpenFlow switches performance and security challenges: a survey," *Future Internet*, vol. 12, no. 9, Aug. 2020, doi: 10.3390/fi12090147.
- [3] Y. Guo, F. Miao, L. Zhang, and Y. Wang, "CATH: an effective method for detecting denial-of-service attacks in software defined networks," *Science China Information Sciences*, vol. 62, no. 3, p. 32106, Mar. 2019, doi: 10.1007/s11432-017-9439-7.
- [4] C. S. Khin, A. T. Kyaw, M. M. Maw, and M. Z. Oo, "Reducing packet-in messages in OpenFlow networks," *ECTI Transactions on Electrical Engineering, Electronics, and Communications*, vol. 20, no. 1, pp. 1–9, Feb. 2022, doi: 10.37936/ecti-eec.2022201.244944.
- [5] P. Ohri, A. Daniel, S. G. Neogi, and S. K. Muttoo, "Blockchain-based security framework for mitigating network attacks in multi-SDN controller environment," *International Journal of Information Technology*, Jun. 2024, doi: 10.1007/s41870-024-01933-8.
- [6] R. Swami, M. Dave, and V. Ranga, "Detection and analysis of TCP-SYN DDoS attack in software-defined networking," *Wireless Personal Communications*, vol. 118, no. 4, pp. 2295–2317, Jun. 2021, doi: 10.1007/s11277-021-08127-6.
- [7] D. Kim, B. Kim, I. Kim, J. Kim, and H. Cho, "Endpoint mitigation of DDoS attacks based on dynamic thresholding," *Information and Communications Security*, pp. 381–391, 2012, doi: 10.1007/978-3-642-34129-8\_36.
- [8] H. A. Alamri and V. Thayanathan, "Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against DDoS attacks," *IEEE Access*, vol. 8, pp. 194269–194288, 2020, doi: 10.1109/ACCESS.2020.3033942.
- [9] E. R. Jimson, K. Nisar, and M. H. bin Ahmad Hijazi, "Bandwidth management using software defined network and comparison of the throughput performance with traditional network," in *2017 International Conference on Computer and Drone Applications (ICONDA)*, Nov. 2017, pp. 71–76, doi: 10.1109/ICONDA.2017.8270402.
- [10] C.-H. Lin, J.-C. Liu, H.-C. Huang, and T.-C. Yang, "Using adaptive bandwidth allocation approach to defend DDoS attacks," in *2008 International Conference on Multimedia and Ubiquitous Engineering (mue 2008)*, 2008, pp. 176–181, doi: 10.1109/MUE.2008.23.
- [11] S. Sumathi, R. Rajesh, and K. Nagarajan, "DDoS attack detection using hybrid machine learning based IDS models," *Journal of Scientific & Industrial Research*, vol. 81, no. 03, Mar. 2022, doi: 10.56042/jsir.v81i03.58451.
- [12] K. Alemerien, S. Al-suhemat, and M. Almahadin, "Towards optimized machine-learning-driven intrusion detection for internet of things applications," *International Journal of Information Technology*, vol. 16, no. 8, pp. 4981–4994, Dec. 2024, doi: 10.1007/s41870-024-01852-8.
- [13] W. Park, J. Lee, and D. Sung, "Bandwidth optimization algorithm based on bandwidth ratio adjustment in generalized processor sharing servers," in *2006 IEEE International Conference on Communications*, 2006, pp. 699–703, doi: 10.1109/ICC.2006.254789.
- [14] E. Hikmawati, N. U. Maulidevi, and K. Surendro, "Minimum threshold determination method based on dataset characteristics in association rule mining," *Journal of Big Data*, vol. 8, no. 1, p. 146, Dec. 2021, doi: 10.1186/s40537-021-00538-3.
- [15] A. Talukder, S. F. Abedin, M. S. Munir, and C. S. Hong, "Dual threshold load balancing in SDN environment using process migration," in *2018 International Conference on Information Networking (ICOIN)*, Jan. 2018, pp. 791–796, doi: 10.1109/ICOIN.2018.8343226.
- [16] U. H. Garba, A. N. Toosi, M. F. Pasha, and S. Khan, "SDN-based detection and mitigation of DDoS attacks on smart homes," *Computer Communications*, vol. 221, pp. 29–41, May 2024, doi: 10.1016/j.comcom.2024.04.001.
- [17] A. Jawahar *et al.*, "DDoS mitigation using blockchain and machine learning techniques," *Multimedia Tools and Applications*, vol. 83, no. 21, pp. 60265–60278, Jan. 2024, doi: 10.1007/s11042-023-18028-4.
- [18] I. Jebiril *et al.*, "Deep learning based DDoS attack detection in internet of things: an optimized CNN-BiLSTM architecture with transfer learning and regularization techniques," *Infocommunications journal*, vol. 16, no. 1, pp. 2–11, 2024, doi: 10.36244/ICJ.2024.1.1.
- [19] A. V. Songa and G. R. Karri, "An integrated SDN framework for early detection of DDoS attacks in cloud computing," *Journal of Cloud Computing*, vol. 13, no. 1, Mar. 2024, doi: 10.1186/s13677-024-00625-9.




- [20] J. Zhao and G.-H. Yang, "An observer-based resilient control method against intermittent DoS attacks for nonstrict-feedback nonlinear cyberphysical systems with tracking errors constraint," *International Journal of Fuzzy Systems*, vol. 26, no. 2, pp. 596–612, Mar. 2024, doi: 10.1007/s40815-023-01619-6.
- [21] N. Zhao, H. Sun, H. Zhang, and K. Mathiyalagan, "Resilient event-triggered dynamic output feedback control for networked Takagi–Sugeno fuzzy systems under denial-of-service attacks," *International Journal of Fuzzy Systems*, vol. 26, no. 1, pp. 357–367, Feb. 2024, doi: 10.1007/s40815-023-01592-0.
- [22] M. Revathi and S. K. Devi, "Hybrid architecture for mitigating DDoS and other intrusions in SDN-IoT using MHDBN-W deep learning model," *International Journal of Machine Learning and Cybernetics*, May 2024, doi: 10.1007/s13042-024-02147-x.
- [23] D. M. Rajan and D. D. J. Aravindhar, "Detection and mitigation of DDOS attack in SDN environment using hybrid CNN-LSTM," *Migration Letters*, vol. 20, no. S13, pp. 407–419, Dec. 2023, doi: 10.59670/ml.v20iS13.6472.
- [24] A. Mansoor, M. Anbar, A. Bahashwan, B. Alabsi, and S. Rihan, "Deep learning-based approach for detecting DDoS attack on software-defined networking controller," *Systems*, vol. 11, no. 6, Jun. 2023, doi: 10.3390/systems11060296.
- [25] Z. Yao, Y. Tang, S. Yuan, and Y. Qi, "Filtering for PDT switched T–S fuzzy systems with attacks and disorders: a TCSI engine fault detection," *International Journal of Fuzzy Systems*, vol. 25, no. 4, pp. 1429–1443, Jun. 2023, doi: 10.1007/s40815-022-01443-4.
- [26] L. Zhou, Y. Zhu, Y. Xiang, F. Paolucci, and T. Zong, "A novel feature-based framework enabling multi-type DDoS attacks detection," *World Wide Web*, vol. 26, no. 1, pp. 163–185, Jan. 2023, doi: 10.1007/s11280-022-01040-3.
- [27] P. Krishnan, K. Jain, A. Aldweesh, P. Prabu, and R. Buyya, "OpenStackDP: a scalable network security framework for SDN-based OpenStack cloud infrastructure," *Journal of Cloud Computing*, vol. 12, no. 1, Feb. 2023, doi: 10.1186/s13677-023-00406-w.
- [28] F. Musumeci, A. C. Fidanci, F. Paolucci, F. Cugini, and M. Tornatore, "Machine-learning-enabled DDoS attacks detection in P4 programmable networks," *Journal of Network and Systems Management*, vol. 30, no. 1, Jan. 2022, doi: 10.1007/s10922-021-09633-5.
- [29] S. Swati, S. Roy, J. Singh, and J. Mathew, "Design and analysis of DDoS mitigating network architecture," *International Journal of Information Security*, vol. 22, no. 2, pp. 333–345, Apr. 2023, doi: 10.1007/s10207-022-00635-1.
- [30] J. Ye, X. Cheng, J. Zhu, L. Feng, and L. Song, "A DDoS attack detection method based on SVM in software defined network," *Security and Communication Networks*, vol. 2018, pp. 1–8, 2018, doi: 10.1155/2018/9804061.
- [31] N. Bharot, P. Verma, S. Sharma, and V. Suraparaju, "Distributed denial-of-service attack detection and mitigation using feature selection and intensive care request processing unit," *Arabian Journal for Science and Engineering*, vol. 43, no. 2, pp. 959–967, Feb. 2018, doi: 10.1007/s13369-017-2844-0.
- [32] F. G. and R. J. Anandhi, "Enhancing SDN resilience against DDoS attacks through dynamic virtual controller deployment and attack level detection algorithm," *International Journal of Information Technology*, vol. 16, no. 7, pp. 4701–4712, Oct. 2024, doi: 10.1007/s41870-024-02064-w.

## BIOGRAPHIES OF AUTHORS



**Florance G.**    is a Ph.D. Research Scholar at the Department of Information Science and Engineering, New Horizon College of Engineering, Bangalore, Karnataka, India. She received her B.E. degree in computer science and engineering from Anna University, Chennai, India in 2006, and her M.E. degree in computer science and engineering from Anna University, Chennai, India in 2008. Her current areas of interest include cybersecurity, security issues in software-defined networking, and DDoS attacks. She can be contacted at email: vijiflorance59@gmail.com.



**Dr. R. J. Anandhi**    received her B.E. degree from Bharathiar University, Coimbatore, India, in 1991, her M.Tech. degree from Pondicherry Central University in 1995, and her Ph.D. degree from Dr. M.G.R. University, India, in 2011. She is currently a Professor and Dean at New Horizon College of Engineering, Bangalore, India, with over 28 years of experience in teaching and research. She has published approximately 59 research papers in various international/national journals and conferences. She has also coordinated several research and development projects within the department and the institute. She has guided ten Ph.D. and is currently supervising several more. Her research interests include spatial data mining, cybersecurity, cloud computing, software-defined networking, among others. She actively serves as a reviewer for many reputed journals published by IEEE, Springer, Elsevier, Taylor & Francis, and Wiley. She can be contacted at email: rjanandhi@hotmail.com.