A New DDoS Detection Method in Software Defined Network

afsaneh banitalebi dehkordi (banitalebi97@gmail.com)
Islamic Azad University Khorasgan Branch

MohammadReza Soltanaghaei
Islamic Azad University Khorasgan Branch

farsad zamani boroujeni
Islamic Azad University Khorasgan Branch

Research

Keywords: Distributed Denial of Service attacks, Software Defined Networks, Network Security, Machine Learning

Posted Date: February 21st, 2020

DOI: https://doi.org/10.21203/rs.2.24212/v1

A New DDoS Detection Method in Software Defined Network

Afsaneh. Banitalebi Dehkordi[1], MohammadReza.Soltanaghaie[2], Farsad.Zamani Boroujeni[3]
1- Department of Computer Engineering, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran.
Email: banitalebi97@gmail.com
2- Department of Computer Engineering, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran.
Email: soltan@khuisf.ac.ir (Corresponding author)
3- Department of Computer Engineering, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran.
Email: farsad.zamanii@yahoo.com

ABSTRACT:
Software Defined Networking (SDN) is a new network architecture in which network control is separated from direct traffic and is programmed directly. Any change in network information and its configuration can be easily implemented in software by using the controller. Although SDN networks with their new structure and controller make way for new and innovative applications for network administrators, but the security challenges and attacks of SDN networks have created problems for these networks. One of these malicious attacks is Distributed Denial of Service (DDoS) attacks. The DDoS attack is aimed at removing machine and network resources from its legitimate users. In this paper, we propose a hybrid method for detecting DDoS attacks in SDN Networks. This method is consisting of statistical and machine learning method. Statistical method calculates the new correlation measure among all features and the dynamic thresholds, then extracts a portion of the data is recognized as attack. This portion is then redirected to the machine learning section to increase the DDoS detection accuracy. The experimental results on UNB-ISCX, CTU-13 and ISOT datasets showed that the proposed method outperforms the existing techniques in terms of the accuracy of detecting DDOS attacks in SDN networks.

KEYWORDS: Distributed Denial of Service attacks; Software Defined Networks; Network Security, Machine Learning

1 INTRODUCTION

SDN architecture [1] consists of application plane, control plane, and data plane. Several applications are provided by application plane including security monitoring, and access controls. Fig 1 shows a simple overview of the network architecture, including the SDN controller, the location of the applications running on the controller, and the openflow switches controlled by the controllers through the openflow interface.

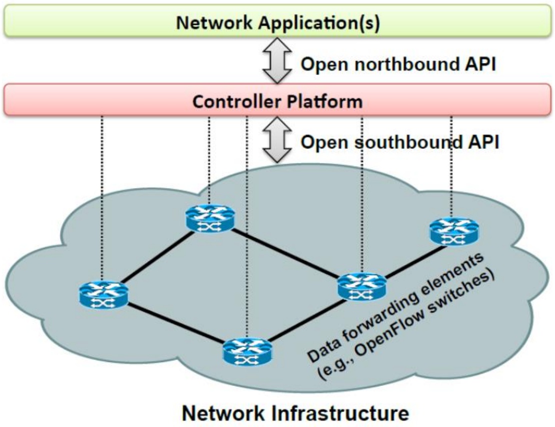

Fig. 1 SDN Architecture

The data plane constitutes underlying network infrastructures known as the infrastructure layer of the SDN. Since SDN creates new networking applications, security in these networks has become a major concern, as it is not an intrinsic feature of SDN architecture yet. Research studies [2] shows that different security attacks, against SDN, can be carried out through different network components of SDN. DDoS attacks are among the most serious threats because they affect network performance, increase the delay, and discard the legitimate packets. For OpenFlow networks, DDoS attacks can be more destructive because there is a constant flow between the controllers and switches. DDoS attacks can start easy and defend hard. For this reason, detection of DDoS attacks in SDN networks is crucial for the future of SDN networks. There are several ways to detect DDoS attacks on SDN networks. Previous research studies [3-7] identify several challenges in detecting DDoS attacks. Failure to investigate all types of DDoS attack, difficulty in selecting the appropriate time intervals for monitoring the traffic in periodic methods, low accuracy of detection, to name a few. The short comes and delays in detecting DDoS attacks may lead to losing resources like bandwidth and CPU, deactivation of the controller and switches and undesirable increase in response time. Furthermore, the importance of maintaining the network security impose high cost of adding hardware to enhance network security, necessitate the major changes in network topology and modifying the security policies on the owner organization. In this paper, we propose a method for detecting DDoS attacks in SDN Networks. The implementation of this method consists of statistical and machine learning methods.

Statistical method calculates the new correlation measure among all features and the dynamic thresholds, then extracts a portion of the data for which the TPR reaches to 100% and is recognized as attack. This portion is then redirected to the classification section to reduce the FPR rate.

Machine learning section extracts 16 features for the hosts of the same flow and records data samples for incoming packets. The samples are fed into the classification section as the training inputs to create models using various classification algorithms such as BayesNet, J48, RandomTree, Logistic regression, Reptree and Naive Bayes algorithms.

In this technique, periodic monitoring and scheduled traffic screening increases the efficiency of the controller in terms of the workload. Another benefit of this idea is needless to add custom hardware to detect attacks. This technique increases in the accuracy of DDoS detection and provides independence from the network topology. This paper is divided into the following sections. We present some methods on DDoS detection in Section 2. Section 3 proposes a new method to detect DDoS attack based on correlation measure and classification algorithms. Section 4 presents the Datasets. Section 5 includes performance evaluations of our proposed detection methods on UNB-ISCX, CTU13 and ISOT datasets. Section 6 compares the methods presented in this study against some existing methods. Finally, Section 7 presents a brief conclusion and future works.

2 RELATED WORK

In the last decade, various studied were conducted to enhance the detection of DDoS attacks in SDN networks.

Dhawan in [8] introduced a method by using the flow graph to estimate the actual network operations. This technique dynamically learns new network behaviors and uses alarm when it detects an attack. In [9] a hybrid diagnostic model using a multidimensional Gaussian method and Expectation Maximization algorithm is proposed for detecting normal and abnormal traffic. In this method, the distance between the parameters was calculated and the detection was performed based on the comparison of these distances with a specific threshold. Various features are used in [10] to infer whether an attack is occurred or not. Since there is more than one factor in judging DDoS attacks, the major issue is to determine the parameters with significant relevance. For example, the destination IP address is considered as one of the relevant parameters in attack detection. Therefore, the attacks can be identified by calculating entropy measure on destination IP address. In [11] an attack detection mechanism presented for responding as quickly as possible to a DDoS attack and reducing the workload of controllers and switches.

This DDoS attack detection mechanism uses a neural network as a classification model for packet classification.

In this study a combination of correlation measure between traffic flows and classification algorithms was presented.

3 THE PROPOSED METHOD

In the present study, a new hybrid method was used for detecting DDoS attacks. The flowchart in Fig. 2 indicates the steps of the presented method. This technique is based on the flows received by the switches and controller. The controller computes correlation among all extracted features and generates a normal level during the analysis period. It also generated a test level for observed traffic using the same correlation measure. If the difference between normal level and observed traffic is higher than dynamic threshold, an alarm will be generated indicating that an attack has occurred and an alarm rate increased one.

$$P = 1 - \frac{1}{n}\sum_{i=1}^{n} \frac{\left| X_{(i,t)} - Y_{(i,t)} \right|}{\left\| \mu_{X_{(i,t)}} - \sigma_{X_{(i,t)}} \right| - X_{(i,t)} \Big| + \Big\| \mu_{Y_{(i,t)}} - \sigma_{Y_{(i,t)}} \right| - Y_{(i,t)} \Big|} \tag{1}$$

$$\mu_{(x_{i,t})} = \sum_{i=1}^{n} X_i \tag{2}$$

$$\mu_{(Y_{i,t})} = \sum_{i=1}^{n} Y_i \tag{3}$$

$$\sigma_{X(i,t)} = \sqrt{\left| \mu_{x^2_{(i,t)}} - (\mu_{x(i,t)})^2 \right|} \tag{4}$$

$$\sigma_{Y(i,t)} = \sqrt{\left| \mu_{Y^2_{(i,t)}} - (\mu_{Y(i,t)})^2 \right|} \tag{5}$$

$$\left| p_{normal-traffic} - p_{observed-traffic} \right| > T_1 \tag{6}$$

Now, assume F1, F2, F3, F4 and F5 are five network traffic objects. In Table 1 the correlation values among different them are calculated.

Table 1 Correlation Values between two flows

Flow pair	P
(F1 ,F2)	O.9876
(F2,F3)	0.5764
(F3,F1)	0.5532
(F3,F5)	0.9961
(F5,F2)	0.5790

3.1 Threshold Calculation

Oshimavet al. [12] introduced dynamic threshold and examined the detection of DDoS attacks with DARPA2000 [13]. DARPA2000 datasets are recognized by experts based on the DDoS attacking software leading that these attacks have the simplicity of structure and type in spite of the complexity of the real data. In this study, this threshold was evaluated for DDoS attacks by testing on datasets collected from actual SDN networks. In order to calculate the dynamic threshold, a computational method based on time sequence was used. The main purpose of using this threshold is fast detection of DDoS attacks in small time windows. The dynamic threshold is calculated as follows:

$$T_1 = \bar{H}_{(i,t-1)} + \alpha.\sigma_{H_{(i,t-1)}} \tag{7}$$

In this equation, α represents a constant value representing a coefficient determined based on experiments. It is not dependent on the time period and the value of previous entropy. In equation (7), the mean values of the entropies $\bar{H}_{(i,t)}$ and standard deviation $\sigma_{H_{(i,t)}}$, calculated at time t, are obtained using the following equations:

$$H_{(i,t)} = -\log \frac{X_{(i,t)}}{\sum_{i=1}^{n} X_{(i,t)}} + \tau_{(i,t)} \tag{8}$$

$$\tau_{(i,t)} = \left| \log \frac{x_{(i,t+1)}}{x_{(i,t)}} \right|, \mathrm{x}_{(i,t)} \geq x_{(i,t+1)}$$

$$\tau_{(i,t)} = \left| \log \frac{x_{(i,t)}}{x_{(i,t+1)}} \right|, \mathrm{x}_{(i,t)} < x_{(i,t+1)}$$

$$\bar{H}_{(i,t)} = \frac{1}{t}\sum_{i=1}^{t} H_{(i,t)} \tag{9}$$

$$\sigma_{H_{(i,t)}} = \frac{1}{t}\sum_{i=1}^{t} (H_{(i,t)} - \bar{H}_{(i,t)})^2 \tag{10}$$

In threshold formula, α is an experimental parameter and its value has a high impact on the accuracy of attack detection. Since selecting the best value for α is a subjective task and depends on various parameters, to calculate the best α for each time period, it is advised to consider an interaction between the different factors. One of these factor is related to the ability of detecting all attacks. Also, it should not make the number of time periods to be very different and must require less computational burden and produce low false alarm rates. For this reason, we consider the α value for which all attacks are correctly detected, i.e. the TPR value is 100, as the best α In this case, although all attacks are correctly detected, some normal flows may also be mistakenly recognized as attack which results in undesirable increase in the FPR parameter or false alarms. By considering the best α value for each time period, by comparing the optimal values for each time period, the best time period is considered as the period for which the

FPR value is lower than that of other time periods. By determining the best time period and best α value, that part of the flow for the attack is detected, is selected and forwarded to the classification step to increase the accuracy of the attack detection. Since this step eliminates a portion of the normal flow that is correctly detected, it balances the number of normal flow and attack flow, before delivering to the classification step. In the classification step, the classification algorithms provide further differentiation between the actual attacks and false alarms achieving higher accuracy in detecting attacks. The controller sends this message to all applications that have requested this type of message. Fig. 2 shows the flowchart of the proposed method.

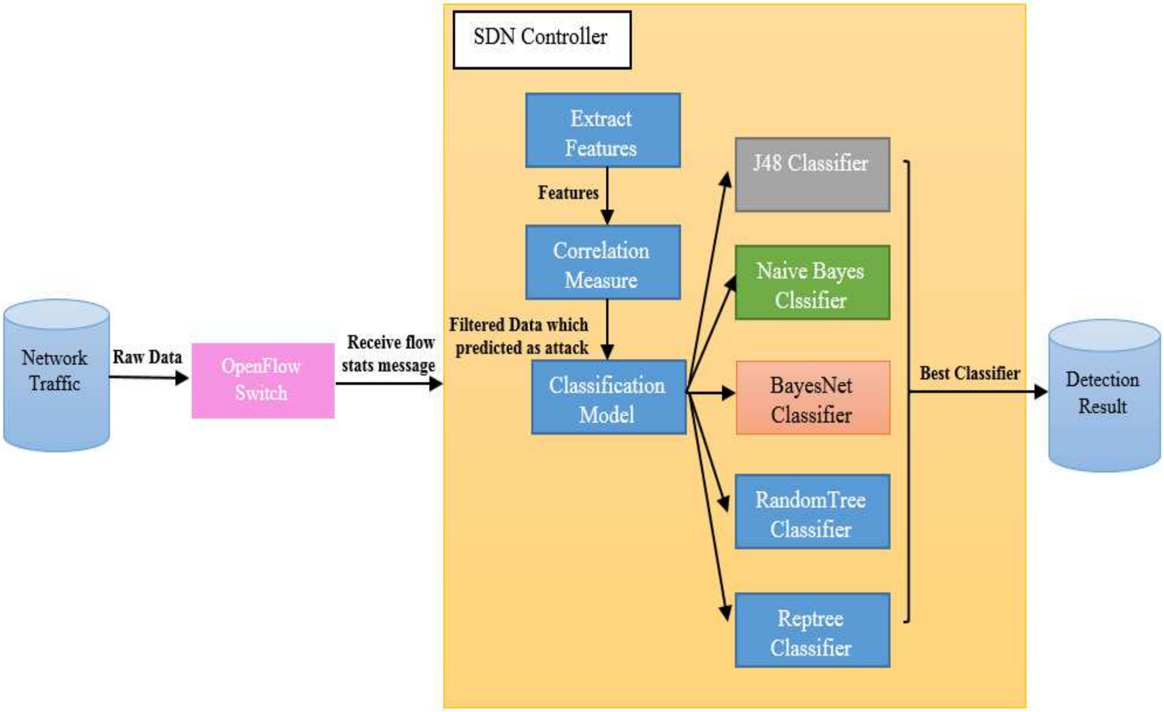

Fig.2 The method presented in this study

The method presented in Fig. 2 consist of several applications. These applications work together to detect DDoS attacks running in the Floodlight controller. The each section is introduced in the following text.

3.2 Feature Extraction

An important challenge of the classification tasks is finding effective features that improve the accuracy of the results. In this section, the most relevant features are selected based on the data and input flow received from the previous stage. Each flow is considered as an edge in the graph and each host, i.e. one of the two ends of a flow, represents a node in the graph. To extract these features, each IP is first considered as a node, then all the connections between those two nodes and other nodes are used to obtain the features. Finally, a weighted directed graph is constructed based on the existing flows. A set of 16 features are extracted for training the classifiers. These features and their explanations are presented in Table 2.

Table 2 The features extracted

	Feature	Explanations
Host A	SenderSrc	The ratio of the number of one-way connections which were the host of the transmitter to the total connections of the desired node
	ReceiverSrc	The ratio of the number of connections which were the host of the recipient to the total connections of the desired node
	EntropyByte PerPacketSentSrc	Calculating the entropy of the flows which were the desired host of the transmitter
	EntropyReceiveSrc	Calculating the entropy of the flows which were the desired host of the recipient
	CountSentSrc	The number of the flows which were the desired host of the transmitter.
	CountReceiveSrc	The number of the flows which were the desired host of the recipient.
Host B	SenderDst	The ratio of the number of one-way connections which were the host of the transmitter to the total connections of the desired node
	ReceiverDst	The ratio of the number of connections which were the host of the recipient to the total connections of the desired node
	EntropyByte PerPacketSentDst	Calculating the entropy of the flows which were the desired host of the transmitter
	EntropyReceiveDst	Calculating the entropy of the flows which were the desired host of the recipient
	CountSentDst	The number of flows which were the desired host of the transmitter
	CountReceiveDst	The number of flows which were the desired host of the recipient
Both A and B	CountPacket	The number of packets in the relevant flow
	SumByte	The total number of bytes for which there is a specified flow
	Packet In	This feature is the first packet which transmits a flow for any host starting the flow whether A or B and is raised in SDN networks.
	Flow Request Rate	The number pf packets go to SDN controller per second
	Duration	length (number of seconds) of the connection

In this article, the data samples are divided into normal and attack classes. After extracting the features, the training samples are given as inputs to the classification algorithms including BayesNet, J48, RandomTree, Logistic regression, Reptree, and Naive Bayes classifiers [14] to construct classification models. In this section, extracted features are provided as inputs to the classification algorithms for detecting attacks. By comparing the results, the best classification algorithm that improve the accuracy of attack detection is selected.

The importance of this method is that we can compute correlation value between any two flows with low number of parameters.

4 THE DATASETS

To evaluate the performance of the proposed method, well-known datasets, namely UNB- ISCX[1] [15] and CTU-13[2] [16], were selected and used in the experiments. In addition to these datasets, the ISOT[3] [17] dataset is also used for normal traffic. The first dataset, UNB-ISCX, were prepared by Canadian Institute of Cyber security. The UNB-ISCX dataset consists of several sections. In this article, the two section of this dataset, namely, ISCX-SlowDos-2016 and ISCX-IDS 2012 are used to detect DDoS attacks. ISCX-SlowDos-2016 dataset contains DDoS attacks generated by several tools. ISCX-IDS-2012 dataset contains various types of attacks such as HTTP GET DDoS. HTTP GET DDoS attack is generated by an IRC botnet, and a brute force SSH attack. Each scenario contains a pcap file including both attack flows and normal flows. The CTU-13 dataset is collected from the Czech University and aims to create a real traffic for the botnet combined with normal and background traffic. This dataset was captured from 13 different samples of different botnet scenarios. In this study, the scenarios 10 and 11 are used to detect DDoS attacks. The traffic used in the CTU-10 scenario is of the type UDP DDoS and in the CTU-11 scenario is ICMP DDoS. The ISOT dataset was created by the Information Technology Research Center at the University of Victoria. Normal traffic was combined from two different sources one from the Ericsson Research Center and the other from the Berkeley National Laboratory. Table 3 provides statistics of ISCX-SlowDDoS2016 datasets.

Table 3 Attack Statistics in ISCX-SlowDDoS2016 Dataset

	Tools		Destination IP		Duration
1	Slowbody2	To	75.127.97.72	After	00.53 minutes
2	Slowread	To	75.127.97.72	After	01:58 minutes
3	Ddossim	To	75.127.97.72	After	02.22 minutes
4	Goldeneye	To	75.127.97.72	After	02:50 170 minutes
5	Slowheaders	To	74.63.40.21	After	02:57 177 minutes
6	Rudy	To	75.127.97.72	After	03:08 188 minutes
7	Ddossim	To	97.74.144.108	After	03:28 208 minutes
8	Rudy	To	208.113.162.153	After	03:29 209 minutes
9	Hulk	To	69.84.133.138	After	04:38 278 minutes
10	Slowheaders	To	67.220.214.50	After	06:00 360 minutes
11	Goldeneye	To	97.74.144.108	After	07:06 426 minutes
12	Slowbody2	To	69.192.24.88	After	08:13 493 minutes
13	Slowbody2	To	97.74.144.108	After	09:03 543 minutes
14	Slowbody2	To	203.73.24.75	After	09:09 549 minutes
15	Rudy	To	97.74.144.108	After	09.20 560 minutes
16	Slowread	To	74.55.1.4	After	11.02 662 minutes
17	Slowheaders	To	97.74.104.201	After	11:27 687 minutes
18	Hulk	To	74.55.1.4	After	13:33 813 minutes
19	Hulk	To	69.192.24.88	After	13:47 827 minutes
20	Slowloris	To	97.74.144.108	After	15:20 920 minutes
21	Slowheaders	To	97.74.144.108	After	15:47 947 minutes
22	Slowloris	To	75.127.97.72	After	16:33 993 minutes
23	Slowheaders	To	75.127.97.72	After	17:13 1033 minutes

[1] http://www.unb.ca/cic/datasets/ids-2017.html

[2] https://www.stratosphereips.org/datasets-ctu13/

[3] ttps://www.uvic.ca/engineering/ece/isot/datasets/

24	Goldeneye	To	69.192.24.88	After	19:23 1163 minutes
25	Hulk	To	75.127.97.72	After	19:25 1165 minutes
26	Rudy	To	74.55.1.4	After	20:59 1259 minutes

5 EVALUATION

In this study, the evaluation results of the proposed method were presented separately for detecting DDoS attacks. The K-Fold cross-validation method [18] was used for training and validation of the classification model. The number of folds was chosen to be 10. The performance of the proposed solution was measured by Accuracy (ACC), Precision (PR), F-Measure (F1), True Positive Rate (TPR) and False Positive Rate (FPR) metrics[19] , which are calculated in Table 4. Let P be the number of actual positive (attack) examples and N be the number of actual negative (normal) one. The TPR (True Positive Rate) measure represents the ratio of the attacks that were correctly recognized as attack and the False Positive Rate (FPR) is the ratio of normal cases that are wrongly classified as attack. The Alarm-Rate (AR) is the ratio of examples classified as attack with respect to the total number of classified samples.

Table 4 Parameter Formula

Parameter	Formula
Accuracy	$\frac{TP+TN}{TP+FN+TN+FP}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F-measure	$\frac{2\times Precision\times Recall}{Precision+Recall}$
TPR	$\frac{TP}{FN+TP}$
FPR	$\frac{FP}{FP+TN}$
Alarm-Rate	$\frac{(TP+FP)}{(TP+TN+FP+FN)}$

5.1 Implementation Environment and Tools

The experiments were conducted on an ASUS laptop with an AMD (Bristl Ridge), FX-9830P CPU 2.8GHz processor and 12GB of RAM. The operating system was Linux Ubuntu 14.04 LTS run on Window 8.1 host machine. We chose Floodlight [20] as the network controller. It uses Mininet2.2.1 [21] for network simulation [22].

5.2 Results of the experiments

The results are described in two sections. In the first section, the correlation value for each objective traffics and dynamic threshold are calculated for each time period and each specific α . If the difference between this value and normal level is higher than the threshold the attack is detected and a value is added to the alarm rate parameter, that calculates the amount of attack alerts. The best α value was calculated in this experiment and is highlighted in Tables 5 to 8 for each time period. By specifying the best time period and best value of parameter α , which are outlined in Table 9 for each dataset, the part of the flow that is identified as attack is selected for the best time period and the best α value.

Table 5 The evaluation results of Dynamic threshold for DDoS attack detection in the ISCX-SlowDos-2016 dataset

Time-period	α	AR (%)	TPR (%)	FPR (%)	ACC (%)	PR (%)	F1 (%)
	0	35.48	94.20	28.87	73.46	26.85	41.80
10	1	36.15	96.47	29.46	73.12	26.62	41.73
	2	50.00	100	42.42	63.15	26.31	41.66
	0	54.89	99.26	52.21	50.13	9.22	16.88
20	1	59.37	100	51.85	56.25	26.31	41.66
	2	20.06	100	13.88	87.11	35.74	52.66
	0	11.57	59.87	7.84	89.84	37.08	45.80
50	1	54.84	100	52.29	50.50	9.74	17.75
	2	87.24	100	86.52	18.09	6.12	11.54
	0	46.45	100	43.49	58.78	11.28	20.27
100	1	42.18	100	32.72	71.87	33.33	50.00
	2	52.63	100	45.45	60.52	25.00	40.00
	0	66.66	100	58.82	52.38	28.57	44.44
200	1	53.85	100	51.16	51.65	10.22	18.55
	2	87.16	100	86.47	17.93	5.85	11.06

Table 6 The results of Dynamic threshold for DDoS attack detection in the ISCX-IDS-2012 dataset

Time-period	α	AR (%)	TPR (%)	FPR (%)	ACC (%)	PR (%)	.F1 (%)
	0	11.45	63.03	5.88	92.42	38.12	47.51
10	1	22.30	89.41	14.86	85.56	40.00	55.27
	2	20.03	100	13.95	87.03	35.24	52.12
	0	11.5	61.53	7.73	90.11	37.47	46.58
20	1	18.50	78.65	13.94	85.54	29.97	43.41
	2	60.52	100	54.54	52.63	21.73	35.71
	0	11.49	60.85	7.74	90.03	37.36	46.31
50	1	18.57	79.84	13.92	85.64	30.34	43.97
	2	11.47	60.49	7.75	90.00	37.17	46.05
	0	18.57	79.93	13.95	85.62	30.16	43.79
100	1	57.14	100	47.05	61.90	33.33	50.00
	2	18.64	80.86	10.83	88.72	30.04	43.81
	0	54.80	100	52.30	50.43	9.56	17.45
200	2	87.27	100	86.57	17.96	6.00	11.33
	2	20.01	100	13.98	86.99	35.02	51.88

Table 7 The results of Dynamic threshold for DDoS attack detection in the CTU-10 dataset

Time period	α	AR (%)	TPR (%)	FPR (%)	ACC (%)	PR (%)	F1 (%)
10	0	24.44	88.59	17.27	83.31	36.46	51.66
	1	38.21	92.98	32.09	70.43	24.48	38.75
	2	83.14	97.36	81.55	26.39	11.78	21.02
20	0	93.25	98.55	92.65	16.56	10.69	19.29
	1	18.37	92.30	9.94%	90.28	51.42	66.05
	2	21.26	93.10	13.11	87.52	44.62	60.33
50	0	87.16	100	86.47	17.93	5.85	11.06
	1	20.21	83.53	17.29	82.75	21.68	34.43
	2	20.54	88.60	16.88	83.39	22.00	35.25
100	0	21.40	89.85	13.70	86.6	42.46	57.67
	1	73.43	100	69.09	40.62	19.14	32.14
	2	92.37	98.82	91.65	17.37	10.67	19.26
200	0	54.89	99.26	52.21	50.13	9.22%	16.88
	1	34.12	94.87	27.19	75.06	28.46	43.78
	2	95.07	98.27	94.71	14.76	10.53	19.03

Table 8 The results of Dynamic threshold for DDoS attack detection in the CTU-11 dataset

Time period	α	AR(%)	TPR (%)	FPR (%)	ACC (%)	PR (%)	F1 (%)
10	0	71.39	97.43	68.42	38.32	13.97	24.43
	1	54.68	100	47.27	59.37	25.71	40.90
	2	47.22	100	44.37	57.89	10.82	19.53
20	0	19.99	100	10.84	89.74	34.65	51.46
	1	62.50	100	56.09	52.08	23.33	37.88
	2	68.75	100	63.41	45.83	21.21	35.00
50	0	87.27	100	86.53	18.23	6.31	11.87
	1	55.03	100	52.61	50.07	9.28	17.00
	2	20.03	100	13.97	87.01	35.20	52.07
100	0	53.12	100	44.44	62.50	29.41	45.45
	1	86.35	100	85.62	18.75	5.92	11.17
	2	65.62	100	59.25	50.00	23.80	38.46
200	0	40.16	100	36.67	65.34	13.71	24.12
	1	66.66	100	58.82	52.38	28.57	44.44
	2	83.13	100	82.69	19.94	3.83	7.38

Table 9 The value of the best α and the best time period for attack detection in different datasets

Dataset name	Best Time Period	Optimal α Value
ISCX-SlowDos-2016	20	**2**
ISCX-IDS-2012	10	**2**
CTU-10	20	**2**
CTU-11	10	**0**

The results indicated that dynamic thresholding results in obtaining high TPR and high FPR. Since high FPR is undesirable, classification algorithm techniques were used to identify false positive cases and improve the performance of the attack detection. This part is forwarded to the classification step to increase the accuracy of attack detection. Some part of the dataset being detected as attack by the correlation-based section with dynamic threshold were selected as the training set for the classification task for more investigation. However, the rest of the dataset was filtered out and not used as the input for classification task. Various classification algorithms such as BayesNet, J48, Logistic regression, RandomTree, and Reptree are used in this paper to model and accurately detect attacks. Most of the parameters of the classification algorithms are set to default.

Table 10 Classification technique results for different datasets in different algorithms

	Algorithms	TPR (%)	FPR (%)	ACC (%)	PR (%)	F1 (%)
ISCX-SlowDos-2016	BayesNet	95.68	1.09	98.74	82.36	88.52
	J48	98.60	0.35	99.12	99.64	99.11
	Logistic regression	87.31	1.85	97.84	92.39	93.57
	Naïve Bayes	93.69	1.73	96.93	95.68	94.67
	RandomTree	97.17	1.95	97.85	93.07	95.08
	RepTree	98.47	5.50	98.26	99.67	99.07
ISCX-IDS-2012	J48	99.64%	0.10%	99.83%	99.66%	99.65%
	BayesNet	96.14%	0.11%	98.01%	99.88%	97.97%
	Logistic regression	96.69%	4.00%	96.14%	86.47%	91.29%
	Naive Bayes	90.76%	2.58%	95.84%	91.56%	91.16%
	RandomTree	99.68%	0.10%	99.84%	99.66%	99.67%
	RepTree	98.10%	0.34%	98.87%	99.64%	98.86%
CTU-10	J48	99.99%	9.20%	99.64%	99.63%	99.80%
	BayesNet	96.24%	0.03%	98.10%	99.96%	98.06%
	Logistic regression	97.41%	0.62%	98.39%	99.38%	97.89%
	Naive Bayes	93.69	1.73	96.93	95.68	94.67
	RandomTree	98.59	0.09	99.87	97.53	98.06
	RepTree	94.51	1.13	97.64	97.01	95.74
CTU-11	J48	99.70	13.49	98.56	98.73	99.22
	BayesNet	95.65	4.55	95.53	94.36	95.00
	Logistic regression	95.84	3.26	96.35	95.62	95.73
	Naive Bayes	36.72	3.19	36.79	99.98	53.71
	RandomTree	92.56	4.67	94.80	81.88	86.90
	RepTree	99.65	0.12	99.82	99.60	99.62

The results of Table 10 indicated that in the classification section, the best algorithm for detecting DDoS attacks in ISCX-SlowDos-2016 dataset was the J48 algorithm with accuracy of 99.12% and FPR value of 0.35%. The evaluation results of the ISCX-IDS-2012 dataset indicated that the RandomTree algorithm with an accuracy of 99.84% and a FPR value of 0.10% was the best algorithm to detect DDoS attacks. For the CTU-10 dataset, the RandomTree algorithm with an accuracy of 99.87% and FPR value of 0.09% was the best algorithm to detect DDOS attacks. The results of classification section for CTU-11 dataset indicated that the Reptree algorithm with an accuracy of 99.82% and FPR value of 0.12% was the best algorithm to detect DDOS attacks

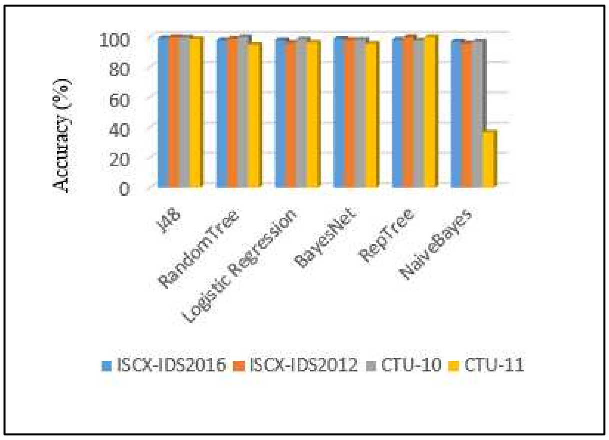

Fig .3 Comparing the accuracy with different algorithms

Table 11 Comparing the accuracy of results with different algorithms and datasets

Datasets	Algorithm	ACC (%)	FPR (%)
ISCX-SlowDos-2016	J48	99.12	0.35
ISCX-IDS-2012	RepTree	99.84%	0.1%
CTU-10	RandomTree	99.87	0.09
CTU-11	RepTree	99.82	0.12%

The results based on Fig.3 and Table 11 indicated that tree algorithms resulted in better results based on the desired dataset.

6 Comparative Performance Experiments

This section compares the method proposed in this study with some existing methods. It should be noted that all these studies aimed at detecting DDoS attacks in UNB ISCX and CTU-13 datasets. The comparative results are summarized in Fig.4 and Fig.5.

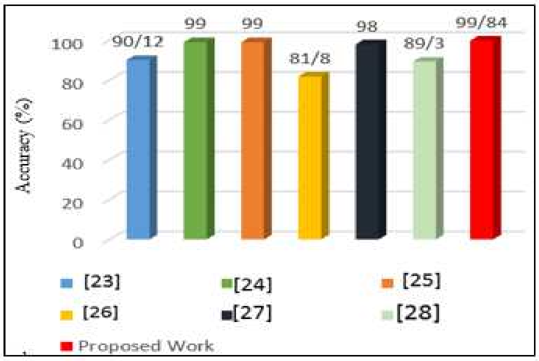

Fig .4 Comparing the accuracy of the proposed method to other studies for the UNB-ISCX dataset

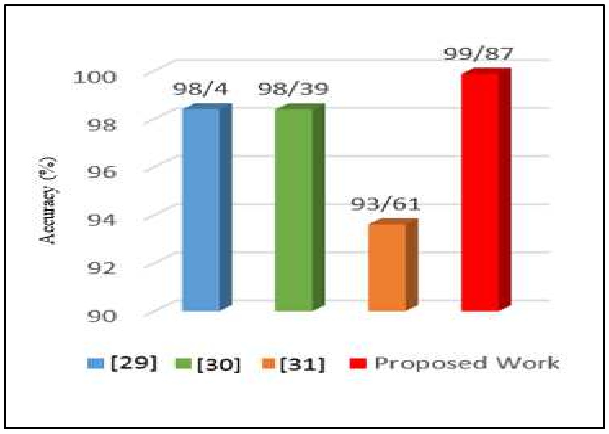

Fig .5 Comparing the accuracy of the proposed method to other studies for the CTU-13 dataset

The main contribution of the current study is to combine statistical methods and machine learning to improve the detection DDoS Attacks in SDN networks. Previous methods did not consider using the strengths of both techniques motivating us to propose an efficient method based on statistical filtering of SDN traffic and supervised learning to achieve higher detection performance. In the statistical step, correlation measure is utilized to recognize the majority of attacks. It is easily developed and implemented in SDN network environments, which requires low CPU load and is easily implemented by the controller. The advantage of this study is also the use of periodic DDoS attack detection technique in using SDN networks over other methods of attack detection in SDN networks. Considering periods that are neither too short nor too long has a great impact on detecting attacks. Because selecting short periods causes losing computational resources such as network bandwidth and CPU cycles. On the other hand, considering long periods increases response time and late detections which results in harmful damages to controllers, switches and network security. Therefore, the choice of detection method by considering selected periods and using dynamic thresholds, which is independent from time periods, can increase the speed of attack detection. In addition, it can preserve resources, protect the controller and switches from harmful damages caused by attacks. Also, eliminating a portion of the normal flows in the correlation method results in balancing normal and attack flows. It acts as a pre-processing step for the machine learning stage. Also, there is also no need for hardware infrastructure to enhance network security, which is another strength of the proposed method. Extraction of features that are independent of the speed and type of attack during machine learning has made the proposed method able to detect both High-rate and Low-rate DDoS attacks. Fig. 5 and 6 illustrate the comparative performance of the proposed method against traditional methods when dealing with real datasets collected from actual SDN networks. The results show that the proposed method outperforms its existing counterpart methods in terms of accuracy and efficiency.

7 Conclusion

Today, SDN have gained considerable popularity among corporate networks due to the flexibility in network management services and reduced operating costs. However, the issue of security and preventing attacks such as DDoS attack on these networks is inevitable. To improve the security of SDN networks, this study introduced a new method for detecting DDoS attacks using a combination of statistical and supervised learning techniques. The proposed method was evaluated and analyzed and its results were investigated in separate sections. The evaluations indicated that the Correlation-based sections with dynamic threshold do not produce appropriate results according to experiments on different datasets. However, better results were obtained for the dynamic thresholding at the cost of high FPR. In order to solve this problem, different classification algorithms were used and more accurate results were obtained. Finally, the significance of the proposed method was determined by comparing the accuracy of the proposed method with previous studies. Results indicated that the accuracy of the proposed method is higher than other similar methods.

Abbrreviations

DDoS: Distributed Denial Of Service;SDN: Software Defined Network; TP: True Positive; TN:True Negative;

FP:False Positive; FN:False Negative; TPR: True Positive Rate; FPR:False Positive Rate; ACC:Accuracy

Acknowledgements

Not applicable

Authors'contribution

All authors have contributed to this research work. All authors have read and approved the final manuscript

Authors' information

Not applicable

Funding

Not applicable

Availability of data and materials

The datasets generated and/or analaysed during the current study are available in the [Canadian Institute for Cybersecurity Institute, Stratosphere Lab and University of Victoria] repository,
[http://www.unb.ca/cic/datasets/ids-2017.html]
[https://www.stratosphereips.org/datasets-ctu13]
[https://www.uvic.ca/engineering/ece/isot/contact/index.php]

Competing interests

The authors declare that they have no competing interests.

REFERENCES

1. Anithaashri, A., G. Ravichandran, and R. Baskaran, *Security enhancement for software defined network using game theoretical approach.* Computer Networks, 2019.
2. Todorova, M.S. and S.T. Todorova, *DDoS Attack Detection in SDN-based VANET Architectures*. 2016, AALBORG.
3. Behal, S., K. Kumar, and M. Sachdevac, *D-FACE: An anomaly based distributed approach for early detection of DDoS attacks and flash events.* Journal of Network and Computer Applications, 2018: p. 49-63.
4. Cui, J., et al., *DDoS detection and defense mechanism based on cognitive-inspired computing in SDN.* Future Generation Computer Systems, 2019.
5. Yadav, A., et al., *SDN Control Plan Security in Cloud Computing Against DDoS Attack.* IJARIIE, 2016. **2**(3): p. 426-430.
6. Priyadarshini, R., *A deep learning based intelligent framework to mitigate DDoS attack in fog environment.* Journal of King Saud University –Computer and Information Sciences, 2019.
7. Sagar Sahoo, K., et al., *An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics.* Future Generation Computer Systems, 2018.
8. Dhawan, M., et al., *SPHINX:Detecting Security Attacks in Software Defined Networks*, in *Network and Distributed System Security Symposium*. 2015.
9. Cepheli, O., S. Buyukcorak, and G. Karabulut *Hybrid Intrusion Detection System for DDOS Attacks.* journal of electrical and computing engineering, 2016.
10. YAN, Q., Q. GONG, and F. DENG, *Detection of DDOS Attacks Against Wireless SDN Controllers Based on the Fuzzy Synthetic Evaluation Decision-making Model.* Ad Hoc & Sensor Wireless Networks, 2016. **33**: p. 275-299.
11. Cui, Y., et al., *SD-Anti-DDOS:Fast and Efficient DDOS Defense in software -Defined Networks.* Journal of Network and Computer Applications, 2016. **68**: p. 65-79.
12. Oshima, S., T. Nakashima, and T. Sueyosh, *DDOS Detection Technique using Statistical Analysis to generate Quick Response time*, in *Intrnational Conference on Broadband,Wireless Computing,Communication and Applications*. 2010.
13. *LINCOLN LABORATORY MASSACHUSETTS INSTITUTE OF TECHNOLOGY*. Available from: https://www.ll.mit.edu/r-d/datasets.
14. *Buczak, A. and E. Guven, A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys and Tutorials, 2016. 18(2): p. 1153-1176*
15. Hadian Jazi, H., H. Gonzalez, and N. Stakhanova, *Detecting HTTP-based Application Layer DoS attacks on Web Servers in the presence of sampling.* computer Networks, 2016.
16. Yavanoglu, O. and M. Aydos, *A Review on Cyber Security Datasets for Machine Learning Algorithms*, in *IEEE International Conference on Big Data*. 2017.
17. Bhamare, D., et al., *Feasibility of Supervised Machine Learning for Cloud Security.* IEEE, 2016.
18. Cross-validation (statistics). Available: https://en.wikipedia.org/wiki/Cross-validation_(statistics).
19. *Beyond Accuracy: Precision and Recall. Available: https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c.*
20. Asadollahi, S. and B. Goswami, *Experimenting with scalability of floodlight controller in software defined networks*, in *International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT)*. 2017.
21. *Mininet ,An Instant Virtual Network on your Laptop (or other PC)*. Available from: http://mininet.org/.
22. Akhunzada, A., et al., *Secure and Dependable software defined networks.* journal of network and computer, 2015. **61**: p. 199-221
23. Tan, Z., A. Jamdagni, and X. He, *Detection of Denial-of-Service Attacks Based on Computer Vision Techniques*, in *IEEE TRANSACTIONS ON COMPUTERS*. 2015.
24. YASSIN, W., et al., *ANOMALY-BASED INTRUSION DETECTION THROUGH KMEANS CLUSTERING AND NAIVES BAYES CLASSIFICATION*, in *4th International Conference on Computing and Informatics, ICOCI*. 2013.
25. Fallahi, N., A. Sami, and M. Tajbakhsh, *Automated Flow-based Rule Generation for Network Intrusion Detection Systems*, in *24th Iranian Conference on Electrical Engineering (ICEE)*. 2016.
26. Catania, C. and C. Garcia Garino, *Towards Reducing Human Effort in Network Intrusion Detection*, in *The 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. 2013: Berlin, Germany.
27. Saied, A., R. Overill, and T. Radzik, *Detection of known and unknown DDOS attacks using Artifitial Neural Networks.* Neurocomputing, 2015.
28. Wang, B., et al., *DDoS Attack Protection in the Era of Cloud Computing and Software-Defined Networking.* Computer Networks, 2015.
29. Kalaivani, p. and M.S. Vijaya, *Mining Based Detection of botnet traffic in Network Flow*, in *IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS)*. 2016.
30. Bansal, A. and S. Mahapatra, *A Comparative Analysis of Machine Learning Techniques for Botnet Detection.* Jaipur, 2017.
31. Chen, R., et al., *An Effective Conversation-Based Botnet Detection Method.* Mathematical Problems in Engineering, 2017.

Figure number: Figure 2

Short title of figure: SDN Architecture
Detailed legend: Fig 1 shows a simple overview of the network architecture, including the SDN controller, the location of the applications running on the controller, and the openflow switches controlled by the controllers through the openflow interface.

Figure number: Figure 2

Short title of figure: The method presented in this study
Detailed legend: Fig. 2 shows the flowchart of the proposed method.

Figure number: Figure 3

Short title of figure: Comparing the accuracy with different algorithms
Detailed legend: The results based on Fig.3 and Table 11 indicated that tree algorithms resulted in better results based on the desired dataset

Figure number: Figure 4

Short title of figure: Comparing the accuracy of the proposed method to other studies for the UNB-ISCX dataset
Detailed legend: The comparative results are summarized in Fig.4 and Fig.5.

Figure number: Figure 5

Short title of figure: Comparing the accuracy of the proposed method to other studies for the CTU-13 dataset
Detailed legend: The comparative results are summarized in Fig.4 and Fig.5.

Figures

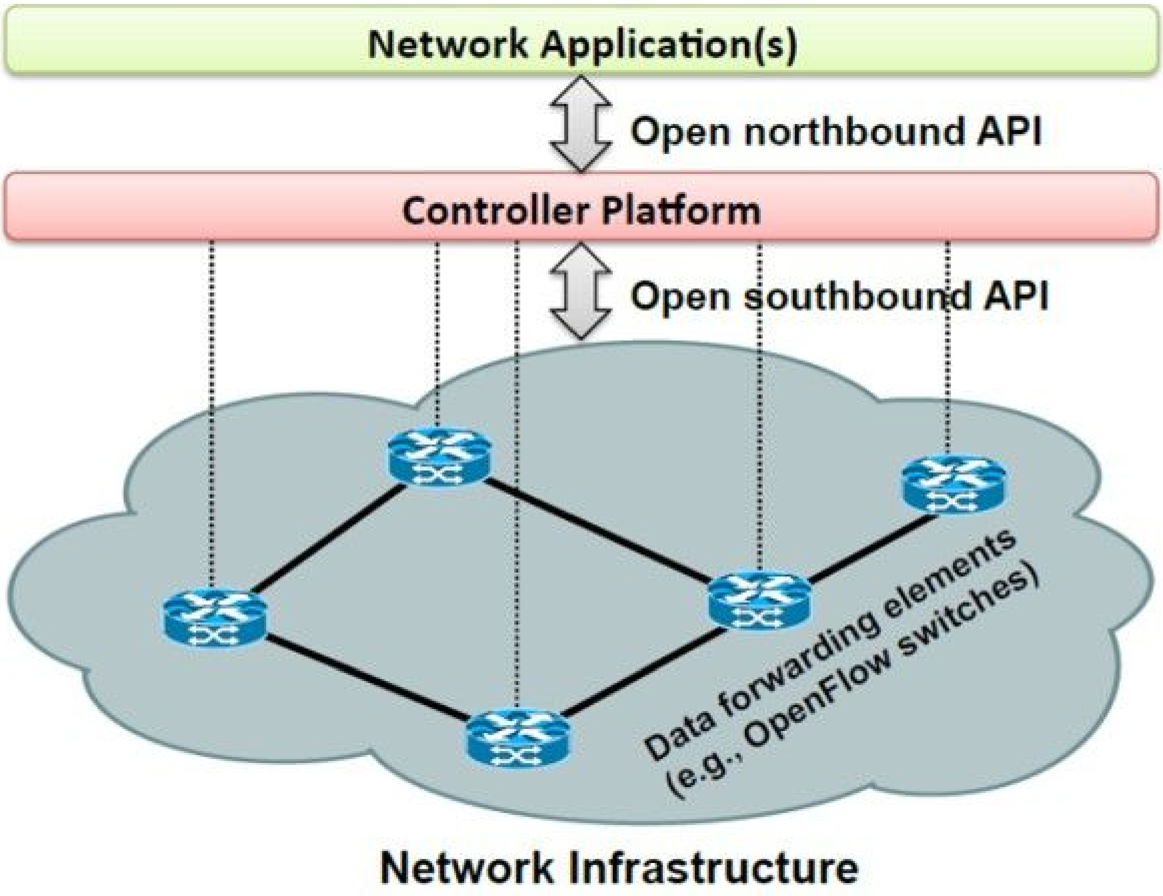

Figure 1

SDN Architecture Detailed legend: Fig 1 shows a simple overview of the network architecture, including the SDN controller, the location of the applications running on the controller, and the openflow switches controlled by the controllers through the openflow interface.

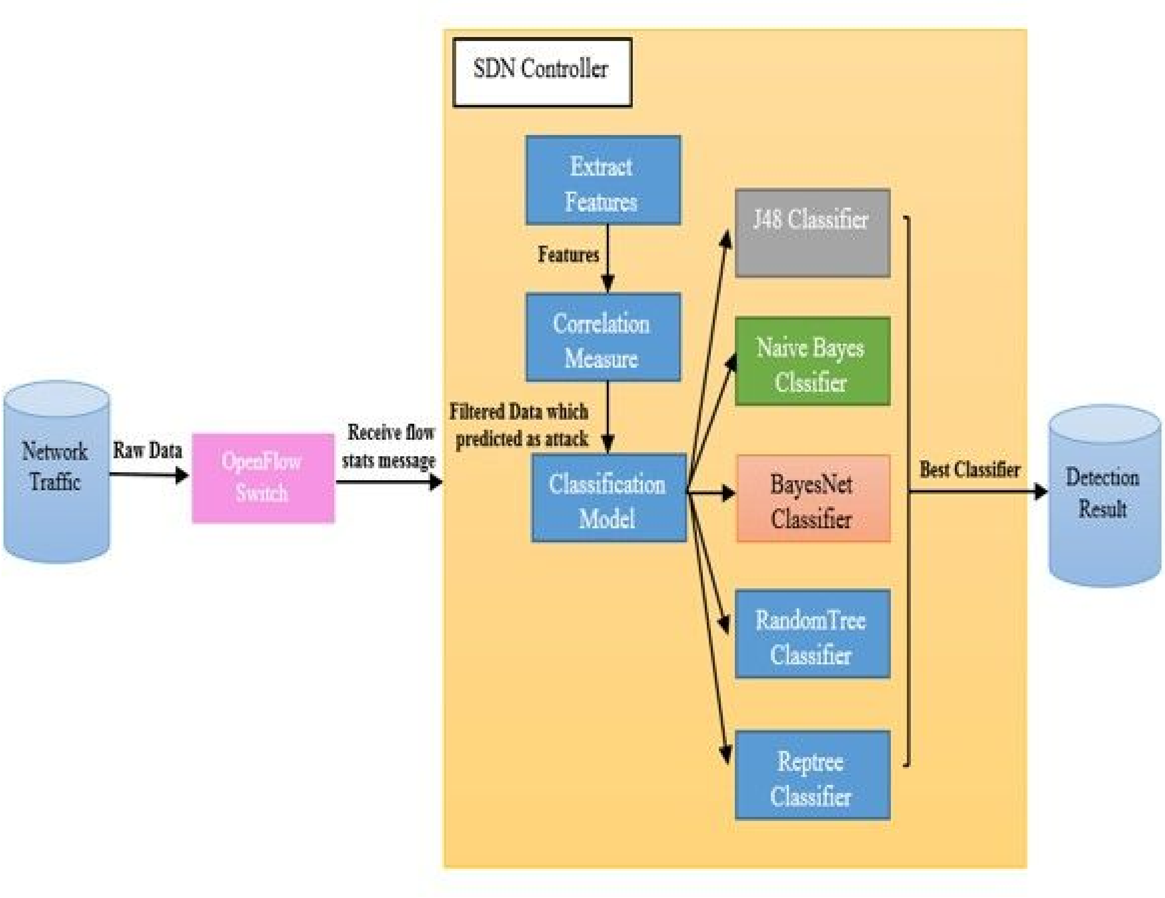

Figure 2

Short title of figure: The method presented in this study Detailed legend: Fig. 2 shows the flowchart of the proposed method.

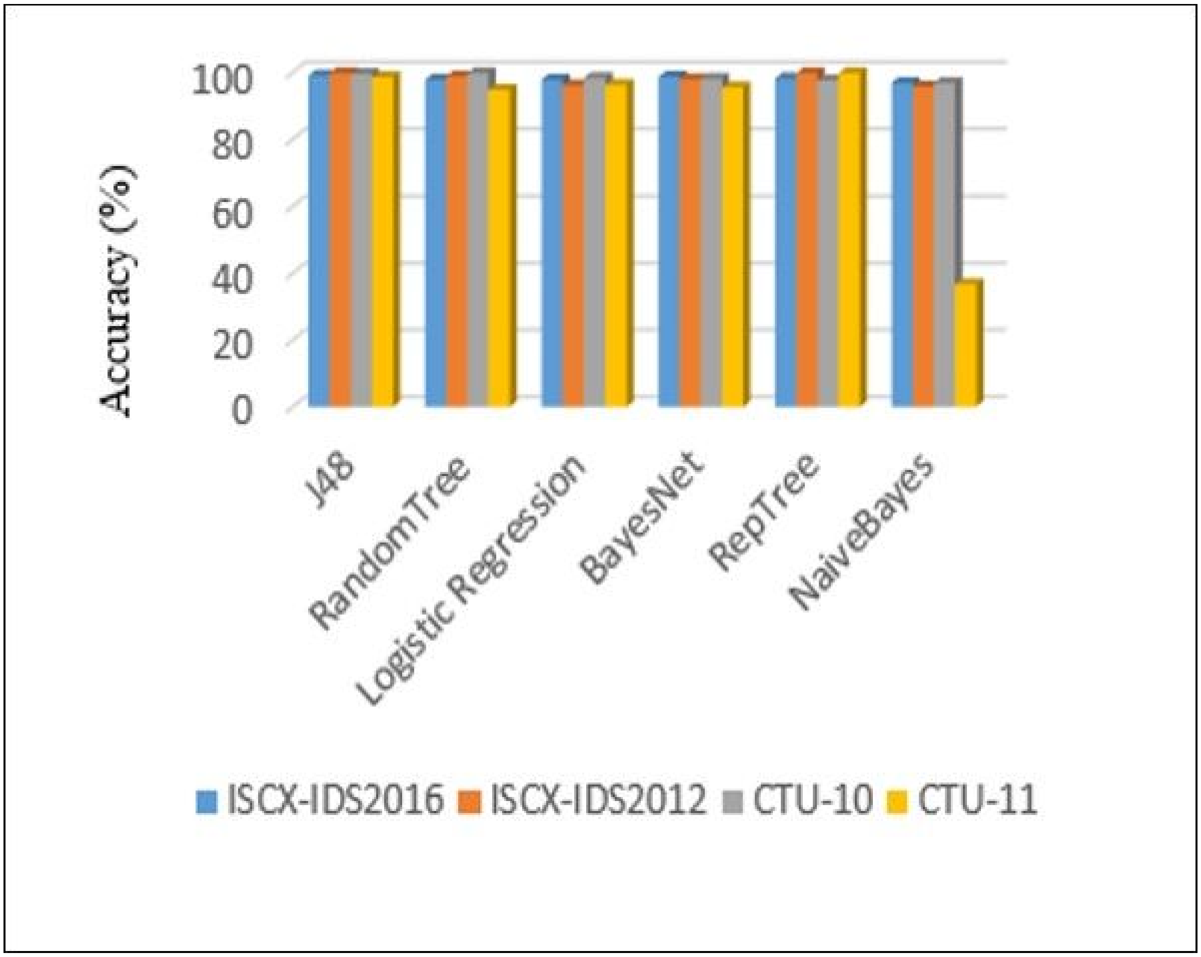

Figure 3

Short title of figure: Comparing the accuracy with different algorithms Detailed legend: The results based on Fig.3 and Table 11 indicated that tree algorithms resulted in better results based on the desired dataset

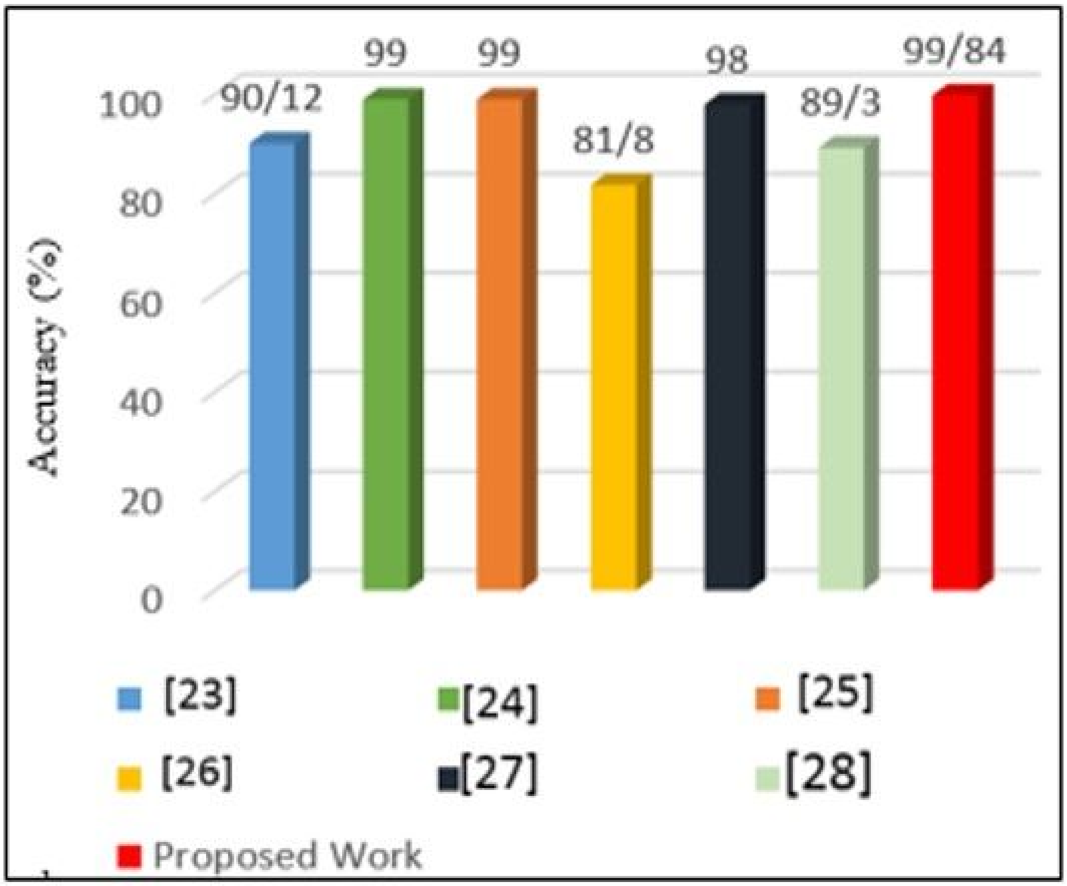

Figure 4

Short title of figure: Comparing the accuracy of the proposed method to other studies for the UNB-ISCX dataset Detailed legend: The comparative results are summarized in Fig.4 and Fig.5.

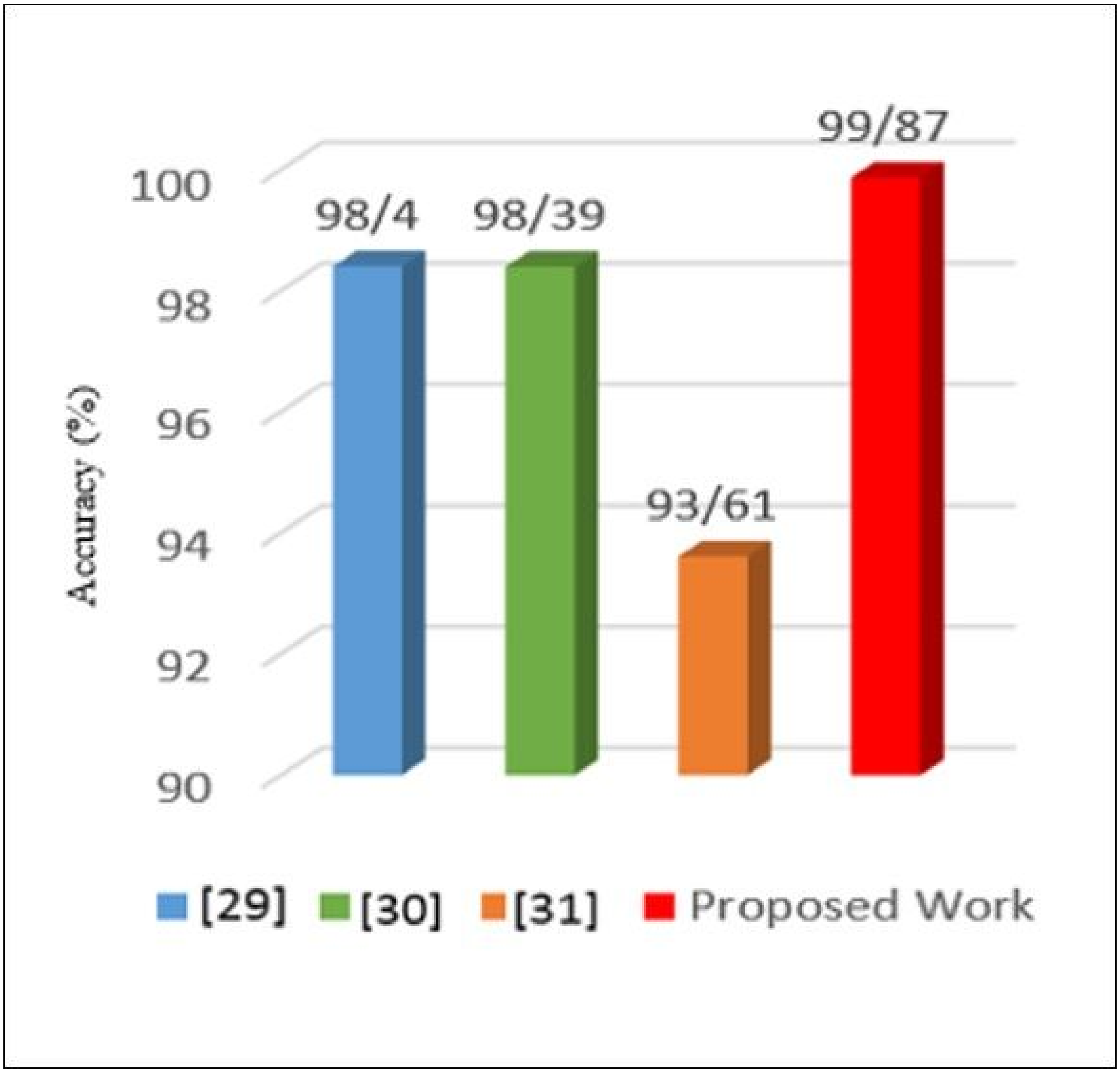

Figure 5

Short title of figure: Comparing the accuracy of the proposed method to other studies for the CTU-13 dataset Detailed legend: The comparative results are summarized in Fig.4 and Fig.5.