SDN Enabled DDoS Attack Detection and Mitigation for 5G Networks

Bhulok Aryal, Robert Abbas, and Iain B. Collings
Macquarie University, Sydney, Australia
Email: bhulok.aryal@students.mq.edu.au; {robert.abbas; iain.collings}@mq.edu.au

Abstract —This paper proposes a hybrid technique for distributed denial-of-service (DDoS) attack detection that combines statistical analysis and machine learning, with software defined networking (SDN) security. Data sets are analyzed in an iterative approach and compared to a dynamic threshold. Sixteen features are extracted, and machine learning is used to examine correlation measures between the features. A dynamically configured SDN is employed with software defined security (SDS), to provide a robust policy framework to protect the availability and integrity, and to maintain privacy of all the networks with quick response remediation. Machine learning is further employed to increase the precision of detection. This increases the accuracy from 87/88% to 99.86%, with reduced false positive ratio (FPR). The results obtained based on experimental data-sets outperformed existing techniques.

Index Terms—DDoS, Software Defined Networking (SDN), 5G Security, Internet of Things (IoT) security, Machine Learning

I. Introduction

Ever-evolving security threats and new deployment scenarios of critical infrastructure means that security of 5G networks has gained more importance than ever. The security of software defined networking (SDN) implementations is particularly critical [1]. Distributed denial-of-service (DDoS) attacks are one of the most prominent threats as they disrupt network performance and can even discard valid packets thereby increasing delay. Challenges include overcoming failures in investigating varieties of attacks, complication on choosing the suitable time interval for observing the traffic in periodic method, and low accuracy of detection. Inconsistency and delay in detecting DDoS attacks can lead to adverse increase in the response time. Likewise, the necessity of retaining the network security requires high cost of enhancing hardware to improve the security over network.

In this paper, we proposed a model to detect DDoS attacks in 5G Networks with SDN. A novel hybrid technology is implemented with calculation of a dynamic threshold, using five different machine learning classifiers i.e., RepTree, Naive Bayes, BayesNet, RandomTree and J48 algorithms with SDN embedded within the system with different sets of rules. Different features are extracted having same flow, and the data are recorded to be processed further in the detection process. A correlation measure between the extracted features and the dynamic threshold is calculated and a segment of data with sensitivity 100% is extracted and recognized as an attack. Our implementation of SDN has resulted in a dynamic configuration of the system with SDS. Because of regular examining and scheduled traffic screening, it enhances the effectiveness of the controller regarding the capacity to handle workloads. This hybrid technique does not require adding custom hardware for attack detection. Increased accuracy and independence from network topology are another benefit of our hybrid approach.

Manuscript received December 22, 2020; revised June 15, 2021.
Corresponding author email: bhulok.aryal@students.mq.edu.au
doi:10.12720/jcm.16.7.267-275

II. Related Works

Paper [2] discussed a collaborative defense scheme for 5G MEC against DDoS by leveraging SDN and Network Function Virtualization (NFV). It proposed an online algorithm with guarantee over their performance of the system. Unlike [2], [3] proposed a novel technique which enhances the potentiality of prevailing Intrusion Detection Systems (IDS), to precisely ascertain attacking nodes on the network irrespective to several traffic encapsulations. [4] showed that upon fully involvement of 5G, the dependency of cyber-physical systems (CPSs) on the cellular network will be more promising, making the system more vulnerable to the intruders. A secure network framework was proposed where they implemented the deep learning and data from the real network to prepare for the early detection of DDoS attack orchestrated by the bots. [5] simulated a 5G-like environment in OMNeT++ discrete event simulator that allows diverse testing of 5G typologies along with the performance measurement of SDN in relation to detection and mitigation mechanisms. Output of the system was analyzed to nominalize the SYN flood attacks and even claimed their output concluded SDN as the chief security component in 5G networks. Paper [6] also discussed the DDoS threats for SDN on 5G but unlike [5], instead of simulating in virtual environment, entropy of traffic status generated within the network environment is calculated for acquiring the dynamic threshold so that it could be pre-determined whether the network environment is subjected to DDoS attacks or not. The proposed solution uses the adaptive threshold of traffic

and entropy threshold to assess the entire 5G SDN environment. Papers [2]-[6] justify their proposed solution upon the theoretical basis of modelling the system using simulator, empirical formulas etc. but, [7] proposes the autonomous security system to bespeak the systematic protection of the network contrary to DDoS attacks by elucidating the definite countermeasures apprehended by the autonomous system rather than a human.

The paper [8] approaches with an idea to amalgamate the IDS with SDN for detection of anomalies of mobile networks. Although the proposed architecture tries to reduce malicious traffic generated by the users and restricts its flow towards gateway, it neither really define the detection mechanism utilized by Detection-as-a-Service (DaaS) nodes implemented on their system nor provide any specifics on which type of DDoS attack it tried to discover and prevent. Also, the study does not perceive testing and validating their system's efficiency. The researchers of [9] explored the implementation of security framework for Internet of Radio Light (IoRL) system which is also SDN based system. Their aim was to detect and prevent various types of TCP SYN attacks. On their system, the number of unsuccessful connection request attempted by a host is counted by TCP connection state information. If the number of unsuccessful attempts during the defined time frame exceeds the initially adjusted threshold, source IP address of those requests will be banned for certain time. They have tested and evaluated their proposed system's efficiency and has obtained a substantial outcome. OMNET++/INET extension was introduced in [10] which allows some of the performance and security measurements of SDN controller. They simulated a basic DDoS attack scenario comprising SDN controller, switch, four client hosts with three UPD application running servers. The simulation was done on entropy basis; SDN controller utilizes the drop action for malicious host as per their flow table to mitigate the DDoS attack. [11] and [12] proposes a scheduling mechanism for SDN controller to mitigate the attacks targeting SDN. The scheduling method they have applied is 'MultiSlot' which incorporates time slicing for allocation procedure. This multiqueue scheduling approach aids in distinguishing the attacked switch and normal OpenFlow switch, and hence serves the legitimate requests first which prevents controller to get overloaded by malicious flow requests. [13] describes that the most important player in the network is data and getting to know it more closely and precisely is half the work done. Studying data in a network and analyzing the pattern and volume of data leads to the emergence of a solid Intrusion Detection System (IDS), that keeps the network healthy and a safe place to share confidential information. [14] initiated the flow graph method for estimating the genuine network operations. This approach dynamically acquires a knowledge of new behaviors on the network and triggers alarm upon detection of attack. [15] can be considered as a hybrid-diagnostic model comprising of Expectation maximization algorithm and Gaussian multidimensional algorithms. These combination of algorithms helps in differentiating the normal and abnormal behavior on the network. Here, parameter's distance was compared with the preset threshold. [16] implemented different feature to detect an attack. Because of the multiple factors, there aroused a problem in selecting the relevant parameter prior to detecting the DDoS attack. [17] presented an approach to react the DDoS attack promptly. Also, this process reduces the workload of switches and SDN controllers.

A key step of our paper is to assess the vulnerability of systems to DDoS attacks by using the simulated and standard real-time data sets in accordance with calculated dynamic threshold and different machine learning algorithms while setting the system configuration dynamically with SDN and detect the attack with more accurate precision factor than any other previous studies had done before.

III. The Proposed System

The key novel idea of this paper is to calculate a measure of correlation between the traffic flows and machine learning with different classification models. An important aspect is that dynamic configuration of the system is made with SDN which helps to detect the attacks with less response time and higher precision. A hybrid method for the DDoS detection is presented in step wise procedure in Fig. 1. Based upon the flows received by the switches and controller; the controller computes the correlation between all the features extracted and generates a normal level during its computational period using (1). Also, normal experimental level of perceived traffic is generated by using the same correlation measure. If difference among normal level traffic and the perceived experimental level traffic exceeds the dynamic threshold value, then an alarm will be generated signifying that attack has occurred.

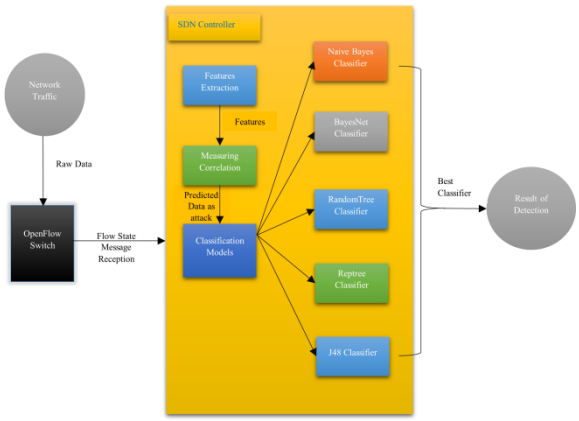

Fig. 1. Flow diagram of proposed approach

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

where,

$$\mu_{X_{(i,t)}} = \sum_{i=1}^{n} X_i \quad , \mu_{Y_{(i,t)}} = \sum_{i=1}^{n} Y_i$$

$$\sigma_{X_{(i,t)}} = \sqrt{|\mu_{X^2_{(i,t)}} - (\mu_{X_{(i,t)}})^2} \quad ,$$

$$\sigma_{Y_{(i,t)}} = \sqrt{|\mu_{Y^2_{(i,t)}} - (\mu_{Y_{(i,t)}})^2}$$

and, $|\rho_{nor\ traffic} - \rho_{perceived\ traffic}| > T_1$

Let us consider five flows of network traffic and the correlation between the flows are tabulated below in Table I.

TABLE I: VALUE OF CORRELATION BETWEEN FLOWS

Paris of Flow	*Correlation value*
(F1, F2)	0.9786
(F2, F3)	0.5674
(F3, F1)	0.5353
(F3, F5)	0.9916
(F5, F2)	0.5709

A. Calculation of Threshold Value

[31] deliberated the dynamic threshold alongside scrutinized the DDoS Attacks detection with DARPA2000 datasets. These datasets are much appreciated based on attacking software primal to that these attacks have simple structure and type despite the complexity of real time data. For our paper, a threshold was calculated for the attacks by experimenting on the datasets from the simulated SDN networks. For dynamic threshold, time sequence-based method for computation was used for quick detection of DDoS attacks over a small time and is calculated by using (2).

$$T_1 = H'_{(i,t-1)} + \alpha.\ \sigma_{H_{(i,t-1)}} \tag{2}$$

In (2), α is a constant that resembles the experimental based coefficient. In order to calculate the average value of entropy H, and standard deviation σ over a period t, the equations (3), (4), and (5) are used.

$$\overline{H}_{(i,t)} = \frac{1}{t}\sum_{i=1}^{t} H_{(i,t)} \tag{3}$$

and,

$$\sigma_{H_{(i,t)}} = \frac{1}{t}\sum_{i=1}^{t} (H_{(i,t)} - \bar{H}_{(i,t)})^2 \tag{4}$$

Also,

$$H_{(i,t)} = -\log \frac{X_{(i,t)}}{\sum_{i=1}^{n} X_{(i,t)}} + \tau_{(i,t)} \tag{5}$$

where,

$$\tau_{(i,t)} = \left| \log \frac{x_{(i,t+1)}}{x_{(i,t)}} \right|, \mathrm{x}_{(i,t)} \geq x_{(i,t+1)} \tag{6}$$

$$\tau_{(i,t)} = \left| \log \frac{x_{(i,t)}}{x_{(i,t+1)}} \right|, \mathrm{x}_{(i,t)} < x_{(i,t+1)} \tag{7}$$

While calculating the dynamic threshold, α being an experimental parameter has great influence in accuracy of the detection of the attack. However, best value adoption for α is a subjective task depending upon different parameters, interaction of various factors is incorporated for the best selection of αAmongst all one of the factors is associated with ˙ the ability to detect the attacks. Besides, there should not be numerous time slots and should accompany the less burden for the computational factor concerning low false alarm rates. Because of this, α is considered in our paper with True Positive Ratio (TPR) value perfect i.e., TPR=100. Even though the selection of α is prioritized to perfect TPR, some normal flow might also be acknowledged as the attack which undesirably increases the FPR. While considering the precise α for each optimal time, best time period is observed in which the FPR value tends to be less than other time periods. After optimizing the best value for α the attack flow is detected and forwarded to ML classification steps in order to even enhance the precision of detection. However, it annihilates the segment of normal traffic flow which might be correctly detected, it stabilizes the normal flow and attack flow before conveying to the ML classifiers. The ML classification algorithms further deliver higher precision in accuracy.

B. Extraction of Features

TABLE II: EXTRACTED FEATURES BASED ON IDENTICAL DATA FLOW

	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 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 flows which were the desired host of the transmitter
	CountReceiveSrc	The number of 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 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 Duration	The number of packets to go to SDN controller per second length (number of seconds) of the connection

The prime defiance in classification algorithm implementation was extraction of the features that helps

to increase the detection accuracy of the project. The most apposite features are selected based on data and flow of input traffic received from the simulation steps. For the initiation of detection method, features are extracted based upon the flow of data and input obtained from the OpenFlow switch. Individual flow represents an edge while individual host represents the nodes on graph as shown in Fig. 2. To extract these relevant features, every one of the IP is regarded as node and all contact with those two nodes and even other nodes are exploited to acquire the features.16 features are extracted by the machine learning section for the host with identical flow as in Table II and the data samples for the incoming packets are recorded.

The data obtained from the switch are subdivided into two class: ordinary and attack. After the features are extracted, Data sample of ordinary class which are used for training are fed as input to the different classifiers as shown in Fig. 1 for developing classification models. Even the extracted features from attack data class are given as inputs to classifiers such as RandomTree, J48, Naive Bayes, BayesNet and Reptree to detect the attacks. The pre-eminent classification model that excels the detection accuracy will then be selected comparing the results obtained from two procedures. The significance of this approach is correlation value between the flows can be achieved even upon consideration of few parameters.

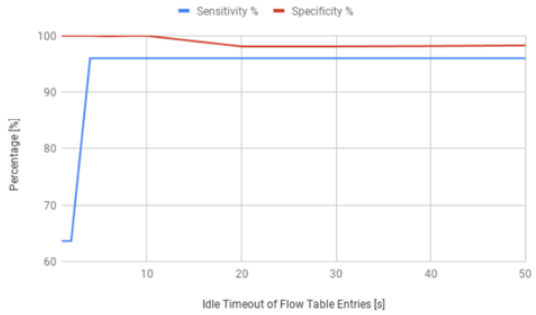

Fig. 2. SDN application performance with regards to sensitivity and ideal timeout of flow entries

IV. Data-Sets

Widely used data-sets are considered for this paper, namely: CTU-13 [18] and UNB-ISCX [19] is used for the experimental purpose. On top of these two data-sets, ISOT [20] is used for the standard traffic. The data-set CTU-13 encompasses sample data from thirteen distinctive botnets with different scenarios. For this paper, only data from scenario 10 and 11 is used for DDoS attack detection because other scenario from the datasets were from different botnets scenario rather than DDoS like from peak to peak (p2p), Internet Relay Chat (which blocks the normal conversation by increasing server load), Spam traffic (emails with undeniable links) etc. Traffics that were used on scenario 10 was UDP DDoS whilst the traffic used on scenario 11 was ICMP DDoS. Also, UNB-ISCX comprises of different sections but, for this paper only two sections are implemented namely: ISCX-IDS 2016 and SCX-SlowDoS 2016 for DDoS attack detection. Both models consist of different DDoS attacks engendered with various tools. In ISOT, there has been a combination of normal traffics from two different source namely: Berkeley National Laboratory and Ericsson Research Center.

V. Evaluation Criteria

K-Fold cross-validation technique is used for training and testing purpose. The performance of the proposed approach was evaluated upon accuracy, precision, false positive rate, true positive rate, and F-Measure (correlation between the traffic flows) parameters. Ture Positive (TP) is the consequence of the model accurately predicting the positive class. True Negative (TN) is the consequence of the model accurately predicting the negative class. False Positive (FP) is the consequence of the model which predicts the positive class not in accordance with the standards. False Negative (FN) is the consequence of the model which predicts the negative class not in accordance with the standards. The formula to calculate the parameters is tabulated below in Table III.

Table III: Formulas to Calculate Parameters [21]

Parameters	Formulas
Accuracy	$\frac{TP + N}{TP + P + TN + N}$
Precision	$\frac{TP}{TP + P}$
Recall	$\frac{TP}{TP + N}$
F-Measure	$\frac{2 * Precision * Recall}{Precision + ecall}$
TPR	$\frac{TP}{TP + N}$
FPR	$\frac{FP}{TN + P}$
Rate of Alarm	$\frac{TP + P}{TP + P + N + N}$

A. Tools and Environment of Implementation

The network controller used is Floodlight. The Floodlight Controller is elaborated to a specific degree by open community of developers, most of them are from Big Switch Networks, which incorporates the e OpenFlow protocol to organize traffic flows in a SDN environment. This controller run effectively on the Linux Ubuntu 14.04 LTS platform. But, for this paper the network controller was carried out in virtual environment of Windows 10 host machine. In this paper, Floodlight controller is equipped with Mininet which is used to create different hosts and switches.

VI. Experimental Results, Findings and Contribution

The outcome of the study is carried out into three sub sections. In the first section, the simulation of the 5g Network prototype is done with DDoS attack where the entropy value of the normal traffic and the attack traffic was analyzed. Wireshark was used for the data capture of normal traffic and attack traffic on the host that has been attacked. Instantly when the attack packets are sent from host h1, because of the destination IP being identical along with each transmitted packet, the entropy values eventually started decreasing and reached to zero.

Eventually, when the attack traffic and background traffic are running simultaneously the value of the entropy expands from zero and again there is randomness in the network. When the attack is launched, because of the decreased value in entropy on controller, the attacked host stops receiving the packets as a result neither traffic is seen in Wireshark and tcpdump. However, background traffic and ICMP traffic is still received.

The performance of the model while varying the flow entry time out is illustrated in Fig. 2 where the DDoS defense application performance in regard to specificity decreases from 100 % at flow entry timeout values of 10 seconds and lower, to 98% at timeout values of 20 seconds and higher.

TABLE IV: EXPERIMENTAL RESULT OF DYNAMIC THRESHOLD IN SLOWDDOS2016

Time Period	Dynamic Threshold (α)	AR (%)	TPR (%)	FPR (%)	ACR (%)	Precision (%)	F1 (%)
	0	35.47	94.19	28.78	73.64	25.89	41.79
10	1	36.11	95.89	29.64	72.13	25.53	40.99
	2	49.98	100	42.24	63.16	25.13	41.23
	0	53.89	99.02	52.23	50.15	9.12	15.99
20	1	58.99	100	51.88	55.96	26.13	41.44
	2	20.04	100	13.86	87.12	35.77	52.46
	0	11.49	59.78	7.48	89.48	37.07	44.98
50	1	54.23	100	52.92	50.02	9.72	17.12
	2	87.22	100	85.25	18.05	6.16	11.45
	0	46.44	100	43.49	57.89	11.12	20.12
100	1	42.16	100	31.76	70.99	33.33	50.34
	2	51.56	100	44.45	60.25	26	39.88
	0	66.01	100	58.28	52.83	28.76	44.42
200	1	53.58	100	50.12	51.56	10.24	18.35
	2	86.95	100	86.34	17.39	5.67	12.03

TABLE V: EXPERIMENTAL RESULT OF DYNAMIC THRESHOLD FOR ISCX-IDS-2012

Time Period	Dynamic Threshold (α)	AR (%)	TPR (%)	FPR (%)	ACC (%)	PR (%)	F1 (%)
	0	10.99	62.99	5.78	91.98	38.22	47.16
10	1	22.13	89.14	14.88	84.99	39.89	54.98
	2	20.12	100	13.67	86.99	35.34	52.19
	0	11.49	61.24	7.37	90.13	37.27	46.34
20	1	18..49	78.43	13.23	85.35	29.36	43.14
	2	60.25	100	54.33	52.22	21.35	35.67
	0	11.56	60.43	7.26	89.99	37.12	46.89
50	1	18.67	79.48	12.94	85.87	30.99	43.56
	2	11.34	59.89	7.57	89.89	37.27	47.01
	0	18.75	78.56	13.67	84.56	30.13	43.23
100	1	57.23	100	46.34	61.34	33.33	50.34
	2	18.38	79.89	9.83	88.45	29.06	42.13
	0	53.68	100	52.37	49.56	9.67	17.33
200	1	86.35	100	85.87	17.56	5.99	11.23
	2	19.99	100	14.23	85.13	36.22	51.78

TABLE VI: EXPERIMENTAL RESULT OF DYNAMIC THRESHOLD FOR CTU-10

Time Period	Dynamic Threshold (α)	AR (%)	TPR (%)	FPR (%)	ACC (%)	PR (%)	F1 (%)
	0	23.34	87.56	16.89	83.44	36.75	51.76
10	1	38.41	91.99	31.56	70.46	23.56	38.56
	2	82.89	97.89	82.33	26.56	11.67	21.89
	0	94.23	98.56	92.78	16.34	10.45	19.45
20	1	18.35	92.56	9.56	90.34	52.34	67.08
	2	21.48	94.56	13.89	87.56	44.78	60.87
	0	88.12	100	85.78	17.98	5.89	11.89
50	1	20.78	84.34	17.57	82.57	21.69	34.77
	2	20.57	88.78	16.89	84.56	21.99	35.76
	0	21.56	90.65	13.7	86.65	42.29	57.89
100	1	73.89	100	69.78	40.48	19.37	33.54
	2	93.22	98,29	92.34	17.45	10.78	19.56
	0	55.78	98.36	52.31	50.65	9.56	16.56
200	1	34.22	94.78	27.56	76.98	29.34	44.32
	2	95.89	98.45	94.23	14.78	10.45	19.46

Eventually with the simulation, correlation value for both the dynamic threshold and the traffics are deliberated for each time and each particular α. When the observation is seen in such a way that the difference between those values and normal level exceeds the threshold then the detection of the attack is done, increasing the value of the alarm rate. After that, on calculating the number of attacks alerts the finest α was considered in the project and is emphasized in the tables below from Table IV to VII for each instance. Upon identifying the best instance and best α as indicated on these tables for each of the three data-sets, the flow segment which is recognized as an attack is taken for those best instances of period and α.

TABLE VII: EXPERIMENTAL RESULT OF DYNAMIC THRESHOLD FOR CTU-11

Time Period	Dynamic Threshold (α)	AR (%)	TPR (%)	FPR (%)	ACC (%)	PR (%)	F1 (%)
	0	23.34	96.57	67.89	38.78	14.67	25.89
10	1	54.89	100	47.56	60.98	25.45	39.99
	2	47.89	100	44.23	57.89	10.87	19.45
	0	19.89	100	10.84	89.77	34.66	51.55
20	1	63.45	100	56.92	52.09	23.35	37.78
	2	68.89	100	63.45	45.56	21.22	35.78
	0	87.53	100	86.45	18.45	6.33	11.89
50	1	55.09	100	52.13	49.89	9.34	16.99
	2	20.01	100	13.78	87.01	35.08	52.18
	0	52.34	100	44.34	62.56	29.14	44.89
100	1	85.89	100	86.09	18.23	5.95	11.13
	2	65.73	100	59.34	49.99	24.8	38.45
	0	40.45	100	36.78	64.34	12.99	24.44
200	1	66.66	100	58.87	53.12	27.85	44.34
	2	84.12	100	82.68	19.84	3.85	7.83

TABLE VIII: EXPERIMENTAL RESULT OF DATA-SETS WHEN PASSED TO ML CLASSIFIERS

	Classifiers	TPR (%)	FPR (%)	ACR (%)	Precision (%)	F1 (%)
ISCX-SlowDos-2016	BayesNet	95.78	1.07	98.79	82.46	88.64
	J48	98.56	0.45	99.34	99.56	99.15
	Naïve Bayes	93.57	1.67	96.39	95.78	94.56
	RandomTree	97.28	1.98	97.58	93.07	95.12
	RepTree	98.36	5.7	98.12	99.23	99.01
ISCX-IDS	BayesNet	96.27	0.13	98.12	99.87	97.96
	J48	99.6	0.11	99.72	99.56	99.56
	Naïve Bayes	90.73	2.68	95.23	92.34	91.23
	RandomTree	99.72	0.11	99.86	99.87	99.58
	RepTree	98.25	0.32	98.35	99.56	98.58
CTU-10	BayesNet	96.12	0.04	97.89	99.86	98
	J48	99.99	8.58	99.56	99.12	99.75
	Naïve Bayes	92.56	1.65	96.21	95.34	94.87
	RandomTree	98.23	0.07	99.78	96.23	98.06
	RepTree	94.31	1.23	97.27	97.03	95.47
CTU-11	BayesNet	94.56	4.52	95.43	94.16	94.87
	J48	99.75	12.27	98.65	98.21	99.1
	Naïve Bayes	35.35	3.14	36.45	99.98	54.56
	RandomTree	92.67	4.17	94.29	80.89	85.87
	RepTree	99.56	0.12	99.37	99.45	99.53

When the statistical method only is implemented, the experimental results we attained signified that the dynamic threshold of any network consequently obtained higher values of the True positive ratio and False positive ratio. However, the false positive ratio is unacceptable, ML algorithms are then implied for recognizing the FPR and hence approach of ML classifiers are employed to enhance the precision of attack detection. Since some segments of the data-sets are recognized as attack because of the use of correlation-based section with dynamic threshold; those are again sent to the classifiers as the training set for more exploration. Nevertheless, other portion of datasets was filtered out. The classifiers that were mentioned in earlier section of this report are used for modelling and detecting attacks more precisely. Almost every parameter in the classifier were set as default value. The experimental results that are obtained by applying ML algorithms for all the data-set is presented in Table VIII.

It is observed that the J48 classifier is recognized as the pre-eminent classifier for ISCX-SlowDos with 99.34 % accuracy and 0.45 % false positive ratio. Similarly, RandomTree Classifier is considered as the best suited classifier with 99.86% accuracy and 0.11% FPR and 99.78% accuracy and 0.07% FPR for ISCX-IDS and CTU-10 respectively whereas accuracy with 99.37% and FPR of 0.12% lead the RepTree classifier as the better algorithm to detect the attacks. Table IX gives the accuracy comparison result for the data-sets used in the experiment.

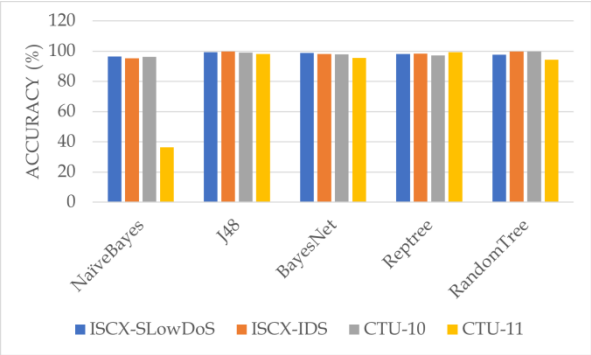

Fig. 3. Accuracy comparison Chart for different ML classifiers

TABLE IX: COMPARISON OF ACCURACY WITH EACH DATA-SET USED

Datasets	Classifiers	ACR (%)	FPR (%)
ISCX-SlowDos	J48	99.34	0.45
ISCX-IDS	RandomTree	99.86	0.11
CTU-10	RandomTree	99.78	0.07
CTU-11	RepTree	99.37	0.12

The experimental results obtained from the project as represented in Fig. 3 and Table IX disclosed that tree classifiers gave the best result over the applied data-set.

In SDN environment, it is easy to monitor and analyze the network traffic because of the self-regulating rules set in the controller. Because of the DDoS attack we can observe the unbalanced load with network fluctuation in the network. We used the IP trace back algorithm and analyzed the load on the server with the help of SDN and bypassed the attacking IP by applying the different filtering rules we had set on SDN controller as remediation measure.

A. Findings

In this paper, a DDoS attack was tested in simulated environment with SDN controller. Upon iterating the experiment by changing the parameters, a performance sensitive analysis was done in terms of entropy, flow entries, number of benign and malicious nodes, normal traffic packet rate and attack packet rate alongside with dynamic threshold for detection.

1) Number of Benign and Malicious Nodes: The result obtained from hybrid method signified that our detection is independent of the malicious nodes and does not affect the performance of the experiment, which suggests that scalability is possible on controller application regarding number of attack hosts, specified that the switches with OpenFlow enabled has enough memory for the flow table. However, the performance is altered when there change in the benign nodes and by means of expansion which increase the traffic of the network despite of malicious node remaining same amount. The experiment's performance degrades undesirably low when the rate of attack is nominal.

2) Entropy: Entropy is defined as the measure of randomness, uncertainty, or distortion of the system. In our project, we came to conclude that when the incoming traffic is stopped, there is attack being occurred. However, the SDN controller permits traffic reception based upon the entropy values. Whenever the attack is being occurred, there will be the decrease in the entropy values of controller which stops receiving the packets and neither tcpdump nor Wire-shark shows the incoming traffic.

3) Flow Entry Timeout: Attack detection has an inevitable performance parameter as flow-entry timeout when selected as low timeout can make the most of the attackers vulnerable using enormous number of attacking nodes with attack transmission time being more extensive than the flow entry timeout session. From the experiment it was also concluded that because of high timeout value the system blocks the malicious node longer than usual time which sometimes undesirably block the benign nodes for longer time assuming as a malicious node.

B. Contribution

The main results of this paper are highlighted and summarized in Fig. 4 and Fig. 5, respectively. All the experiments used UNB-ISCX and CTU-13 data sets.

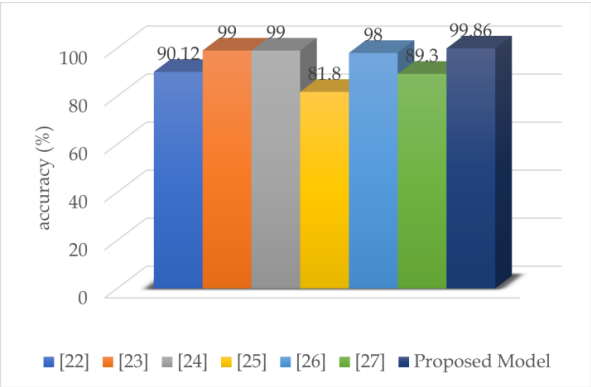

Fig. 4. Accuracy comparison of proposed model to other studies for UNB-ISCX data-set

The main fundamental new contribution of this paper is to mix statistical techniques and machine learning to increase detection accuracy against DDoS attacks over 5G network by passing the simulated and real-time data generated against different botnet scenarios into different machine learning algorithms mentioned in previous section which drastically increases the accuracy from 89% to 99.86% keeping the FPR as much possible low (< 0.5) for different classifier algorithm. Moreover, a

dynamically configured SDN is employed with SDS, which provides a strong policy framework protecting the availability, integrity, and privacy of the network centrally and close to real time scenario.

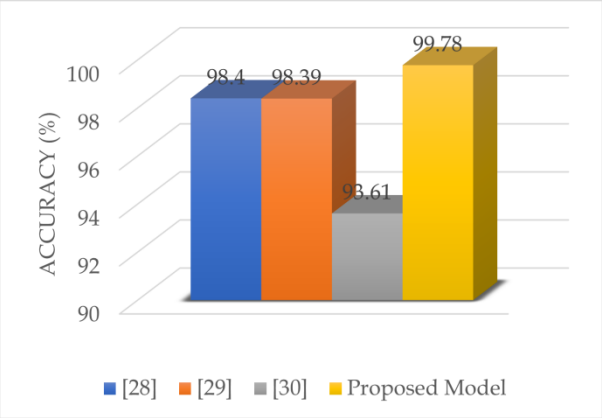

Fig. 5. Accuracy comparison of proposed model to other studies for CTU-13 data-set

Our model achieves maximum productivity with minimum effort or expense build on statistical filtering of the SDN traffic and supervised learning to achieve higher degree of performance. The statistical method makes effective use of a correlation measure for detecting the attacks which requires low CPU and can be implemented easily by the controller that are easily developed an applied on most of the SDN network environments. One of the benefits of this study is the usage of periodic DDoS attack detection technique using SDN networks over the methods of detection in mobile networks.

Computational resource loss like CPU cycles and network bandwidth can be seen when short time stamps are considered and if longer time periods are taken into consideration there will be a decrease in detection rate with bigger response time resulting the damage of switch, controller and even network security. The selection of detection method by using the best dynamic threshold independent from the time period and the best time period can expand the attack detection speed. Moreover, resource is preserved and the network elements like controller, switches are also protected from harmful damage brought about by the attacks. Besides, stability between the normal and attack flows is created upon eliminating the small portion of normal flows in correlation method which acts as a preliminary processing step for classification algorithms. Another supremacy of the proposed method is that it does not require hardware infrastructure to enhance the security of network. Feature extraction process which is totally unconventional approach regardless to speed and attack type during the machine learning have boosted the proposed approach to be able to detect both low-rate and high-rate DDoS attacks. The performance comparison of proposed model against traditional methods is illustrated in Fig. 3 and Fig. 4 when dealing with real data-sets collected from actual SDN networks. The results exhibit that the proposed approach perform better than the prevailing conventional methods regarding accuracy, preciseness, and efficiency.

VII. Conclusion

This paper has proposed a hybrid method indulging statistical and machine learning techniques. A correlation-based result with dynamic threshold did not offer suitable results on its own, according to experiments on various data-sets with high FPR. To resolve this, different machine learning algorithms were used, and the result obtained indicated that our model outperformed others in terms of accuracy and preciseness.

Conflict of Interest

The authors declare no conflict of interest.

Author Contribution

Bhulok Aryal conceived of the presented idea developed the theory and performed computations. Robert Abbas verified the analytical methods alongside encouraging Bhulok Aryal to investigate on SDS and 5G Network Security issues and supervised the findings of this work. Iain Collings contributed with proof reading and contributed to final abstract and conclusion. All authors discussed the results and contributed to the final manuscript.

References

[1] F. Bensalah, N. Elkamoun, and Y. Baddi, "SDNStat-Sec: A statistical defense mechanism against DDoS attacks in SDN-based VANET," in *Advances on Smart and Soft Computing*, Springer, pp. 527–540.

[2] H. Li and L. Wang, "Online orchestration of cooperative defense against DDoS attacks for 5G MEC," in *Proc. IEEE Wireless Communications and Networking Conference*, 2018.

[3] A. S. Mamolar, Z. Pervez, J. M. A. Calero, and A. M. Khattak, "Towards the transversal detection of DDoS network attacks in 5G multi-tenant overlay networks," *Computers & Security*, vol. 79, pp. 132–147, 2018.

[4] B. Hussain, Q. Du, B. Sun, and Z. Han, "Deep learning-based DDoS-Attack detection for cyber-physical system over 5G network," *IEEE Transactions on Industrial Informatics*, 2020.

[5] M. K. Forland, K. Kralevska, M. Garau, and D. Gligoroski, "Preventing DDoS with SDN in 5G," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2019.

[6] G. C. Hong, C. N. Lee, and M. F. Lee, "Dynamic threshold for DDoS mitigation in SDN environment," in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2019.

[7] A. S. Mamolar, P. Salvá-García, E. Chirivella-Perez, Z. Pervez, J. M. A. Calero, and Q. Wang, "Autonomic protection of multi-tenant 5G mobile networks against UDP flooding DDoS attacks," *Journal of Network and Computer Applications*, vol. 145, p. 102416, 2019.

[8] M. Monshizadeh, V. Khatri, and R. Kantola, "Detection as a service: an SDN application," in *Proc. 19th International Conference on Advanced Communication Technology*, 2017.

[9] K. Cabaj, M. Gregorczyk, W. Mazurczyk, P. Nowakowski, and P. Żórawski, "Sdn-based mitigation of scanning attacks for the 5g internet of radio light system," in *Proc. 13th International Conference on Availability, Reliability and Security*, 2018.

[10] M. Tiloca, A. Stagkopoulou, and G. Dini, "Performance and security evaluation of SDN networks in OMNeT++/INET," arXiv preprint arXiv:1609.04554, 2016.

[11] J. Lam and R. Abbas, "Machine learning based anomaly detection for 5G networks," arXiv preprint arXiv:2003.03474, 2020.

[12] Q. Yan, Q. Gong, and F. R. Yu, "Effective software-defined networking controller scheduling method to mitigate DDoS attacks," *Electronics Letters*, vol. 53, p. 469–471, 2017.

[13] S. S. Dhaliwal, A. A. Nahid, and R. Abbas, "Effective intrusion detection system using XGBoost," *Information*, vol. 9, p. 149, 2018.

[14] H. Polat, O. Polat, and A. Cetin, "Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models," *Sustainability*, vol. 12, p. 1035, 2020.

[15] S. Smys, A. Basar, and H. Wang, "Hybrid intrusion detection system for Internet of Things (IoT)," *Journal of ISMAC*, vol. 2, pp. 190–199, 2020.

[16] C. S. Rajpoot, A. K. Bairwa, and V. K. Sharma, "Mitigating the impact of DDoS attack on upsurge network performance in MANET," in *Proc. International Conference on Communication and Computational Technologies*, 2020.

[17] Y. Cui, L. Yan, S. Li, H. Xing, W. Pan, J. Zhu, and X. Zheng, "SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks," *Journal of Network and Computer Applications*, vol. 68, pp. 65–79, 2016.

[18] O. Yavanoglu and M. Aydos, "A review on cyber security datasets for machine learning algorithms," in *Proc. IEEE International Conference on Big Data*, 2017.

[19] H. H. Jazi, H. Gonzalez, N. Stakhanova, and A. A. Ghorbani, "Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling," *Computer Networks*, vol. 121, pp. 25–36, 2017.

[20] D. Bhamare, T. Salman, M. Samaka, A. Erbad, and R. Jain, "Feasibility of supervised machine learning for cloud security," in *Proc. International Conference on Information Science and Security*, 2016.

[21] W. Koehrson, "Beyond accuracy: Precision and recall," *Towards Data Science*, 2018.

[22] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, and J. Hu, "Detection of denial-of-service attacks based on computer vision techniques," 2018.

[23] W. Yassin, N. I. Udzir, Z. Muda, M. N. Sulaiman, and others, "Anomaly-based intrusion detection through k-means clustering and naives bayes classification," 2017.

[24] N. Fallahi, A. Sami, and M. Tajbakhsh, "Automated flow-based rule generation for network intrusion detection systems," in *Proc. 24th Iranian Conference on Electrical Engineering*, 2016.

[25] C. Catania and C. G. Garino, "Towards reducing human effort in network intrusion detection," in *Proc. IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems*, 2019.

[26] A. Saied, R. E. Overill, and T. Radzik, "Detection of known and unknown DDoS attacks using artificial neural networks," *Neurocomputing*, vol. 172, pp. 385–393, 2016.

[27] B. Wang, Y. Zheng, W. Lou, and Y. T. Hou, "DDoS attack protection in the era of cloud computing and software-defined networking," *Computer Networks*, vol. 81, pp. 308–319, 2018.

[28] P. Kalaivani and M. Vijaya, "Mining based detection of botnet traffic in network flow," *International Journal of Computer Science and Information Technology & Security*, 2016.

[29] A. Bansal and S. Mahapatra, "A comparative analysis of machine learning techniques for botnet detection," in *Proc. 10th International Conference on Security of Information and Networks*, 2017.

[30] R. Chen, W. Niu, X. Zhang, Z. Zhuo, and F. Lv, "An effective conversation-based botnet detection method," *Mathematical Problems in Engineering*, vol. 2017, 2017.

[31] S. Oshima, T. Nakashima, and T. Sueyosh, "Detection technique using statistical analysis to generate quick response time," in *Proc. International Conference on Broadband, Wireless Computing, Communication and Applications*, 2017.

Bhulok Aryal received the Bachelor of Engineering in Electronics and Communication from Tribhuvan University (TU), Nepal in 2017 and Master of Engineering in Networking and Telecommunications from Macquarie University (MQ), Australia, in 2021. His research interest includes Software Defined Network Security, 5G/6G Network, Intrusion Detection System, Internet of Things, Wireless Networks, and cybersecurity.

Robert Abbas received the master's degree (Hons.) in Wireless Engineering from the University of Odessa, in 1981, and the M.S. and Ph.D. degrees in Mobile data Networks Technische Universität Dresden Germany 1989. From 1984 until 1988, he was PhD student at TU Dresden,

Germany. From 2007 until 2012, he was an Associate Professor in the Faculty of Engineering and Information Technology, University of Tishreen. From 2016, he has been Senior Lecturer at the School of Engineering, Macquarie University, Sydney, Australia. His research interests include Network softwarisation, Mobile Networks Analytics & Intelligence, Systems Security & Automation, AI/ML Defined Network Security for 6G, 5G, IoT, SDN, MEC, V2X, and NGFWs

Iain Collings is the Deputy Dean Research for the School of Engineering at Macquarie University, Sydney, Australia. Previously he worked for 9 years at the CSIRO, Australia, 6 years at the University of Sydney, and 3 years at the University of Melbourne. He is a Fellow of the IEEE and has published over 330 research papers. He was awarded the Engineers Australia IREE Neville Thiele Award 2009 for outstanding achievements in engineering, and the IEEE CommSoc Stephen O. Rice Award in 2011. He served as an Editor for IEEE Transactions on Wireless Communications, and the MDPI journal Sensors, and a as Guest Editor for the EURASIP Journal on Advanced Signal Processing. He has Co-Chaired numerous Technical Program Committees of major international conferences.