DDoS Attacks Detection Using Dynamic Entropy InSoftware-Defined Network Practical Environment

Dinh Thi Thai Mai, Nguyen Tien Dat, Pham Minh Bao, Can Quang Truong, Nguyen Thanh Tung

Faculty of Electronics and Telecommunications, VNU University of Engineering and Technology, Hanoi, Vietnam

Abstract

Software-Defined Network (SDN) is an innovative network architecture with the goal of providing the flexibility and simplicity in network operation and management through a centralized controller. These features help SDN to easily adapt to the expansion of network requirements, but it is also a weakness when it comes to security. With centralized architecture, SDN is vulnerable to cyber-attacks, especially Distributed Denial of Service (DDoS) attack. DDoS is a popular attack type which consumes all network resources and causes congestion in the entire network. In this research, we will introduce a DDoS detection model based on the statistical method with a dynamic threshold value that changes over time. Along with the simulation result, we build a practical SDN model to apply our method, the results show that our method can detect DDoS attacks rapidly with high accuracy.

Keywords

SDN, DDoS attacks, network security, machine learning, statistical analysis method, entropy, dynamic entropy.

1. Introduction

Nowadays, with the rapid development of network technology, the explosion of network devices and the requirements for controlling network services are becoming increasingly important, and the requirements for high security and control are higher than ever. With a traditional network, managing millions of devices can be a daunting challenge. From that, Software Defined Network (SDN), the novel network architecture, is introduced with the goals to overcome the barriers and disadvantages of traditional networks with centralized control, high scalability and flexibility. SDN network architecture consists of 3 separate layers: The application layer, the control layer, and the infrastructure layer [1].

The application layer includes the applications and functions required for the network. This layer connects to the control layer through the north API interface [2], which allows the administrator at the application layer to program control functions for the network. The SDN controller in the control layer plays a vital role in managing all network operations. The controller will handle the packets and give actions to the packet- forwarding devices at the lower layer. The infrastructure layer includes hardware devices for the network that perform packet forwarding functions and communicates with the controller via OpenFlow protocol [3]. The separation of the control plane and data plane, as shown in Figure 1, bring lots of advantages to SDN. A centralized controller can simplify network management as well as bring programmable and scalability to the SDN

DOI: 10.5121/ijcnc.2023.15307

system, which helps SDN to get over the limitations of traditional networks. From there, the administrator can have a global view of the whole network and manage the network resources in realtime. [4]

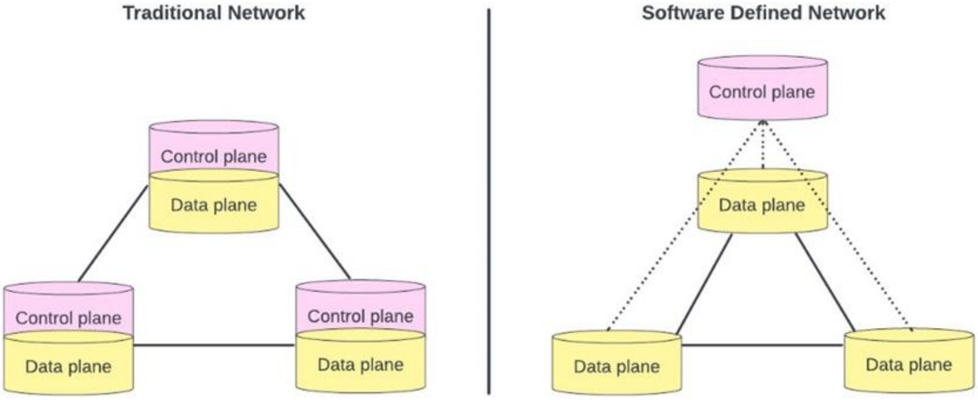


Figure 1. Traditional vs SDN network architecture

However, it is also a weakness and challenge for the administrator if the controller is attacked. SDN is affected by many serious and dreadful attacks and the most common of which is DDoS[5]. In DDoS, at a time, a lot of packets with fake source addresses are sent to the controller. This causes an overload of bandwidth, resources exhaustion [6], then, leads to the controller not being able to handle the user services. Therefore, the early detection and prevention of DDoS attacks is an urgent and essential requirement for the network to protect and prevent network security risks.

The DDoS detection mechanism can be divided into 2 can be divided into two categories: intrinsic and extrinsic solutions [7]. While the former relied on the SDN modules and their elements, the latter related to the traffic flow properties and network characteristics. In this study, we mainly focus on the extrinsic solutions as it helps the system to accurately detect the attacks, compared to intrinsic solutions, by adopting a highly precise method [8]. These methods can be categorized as the machine learning methods and statistics methods. DDoS attack detection based on these methods has been thoroughly studied so far including papers [9-12]. In the machine learning method, a machine learning algorithm is trained and applied to a detection model to detect abnormal events [13]. On the other hand, in the statistical method, a statistical model system will be operated to examine the traffic flows of the network. A traffic sample of the traffic parameter will be extracted and compared with the statistics model. Based on that result, the model can determine if the traffic is malicious. Entropy is the most frequently used technique to detect DDoS attacks in statistics methods [14]. It can measure the randomness of a network parameter and then compare it with a threshold to examine the traffic status. In this research, we discovered that there are still a lot of limitations when calculating a fixed entropy value, as in papers [15-16], it will lower the accuracy, constrain flexibility and be inappropriate for various service providers. We therefore suggest a technique altering the mentioned dynamic thresholds that change over time in response to the entropy value variability of each window. We can predict the state of the system based on this dynamic threshold value and compare it to the present entropy value. Moreover, most of the studies related to detecting DDoSattacks in SDN are only implemented in the simulation environment. So, the research’s main contribution is the execution of the SDN practical model to apply our proposed method and then compared it with the simulation result.

The rest of this article is laid out as follows: In section2, we present the related works. Section 3 gives a general overview of the Entropy-based approach for DDoS attack detection, and in section 4, we will evaluate the system's performance in both environments (simulation and real network) and provide implementation results. Section 5 concludes with our conclusions and the direction of the remaining work.

2. Related Works

Many methods have been proposed to detect DDoS in SDN [17]. In the same topic, the authors of [18-19] developed a technique for identifying DDoS attacks based on fixed entropy values. After calculating, the entropy value will be compared with a pre-set threshold value. If it is below the threshold, then the possibility of an attack is indicated. Otherwise, the entropy value will be set to the current entropy calculation to avoid further inaccurate analysis. This enables the detection algorithm to adjust in adapt response to the characteristics of traffic flow. While the paper [18] utilizes a window size of 50 packets, the authors of [19] evaluated 5 window sizes: 20, 40, 60, 80, 100 and concluded that the window size of 60 has the best cost-benefit ratio.

The authors in [20] used a different entropy method that combined information entropy and log energy entropy into fusion entropy, which greatly reduced in attack event and help detect attack rapidly. According to experimental result, the fusion entropy decreases by 91.25% when an attack occurs. However, this method still depends on a static threshold which is not optimized in term of detection accuracy.

In [21], another technique for identifying DDoS attacks is suggested. This article suggested a technique that based on the extraction of six characteristic variables from traffic in the flow table acquired from the switch. They then also used the SVM algorithm to classify data which helps accurately detect DDoS attacks. According to experimental data, the method's average accuracy rate was 95.24% even though only a small amount of flow data was collected. The authors in [22] enhanced this algorithm with an advanced SVM technique. They customized a reaction mechanism which informs DDoS attack by taking the security requirements of the application into account. The method then still extracts 5 characteristics value to train the model. The detection rate of this enhanced technique is approximately 97%.

In [23], the authors proposed a new method to detect DDoS attack in an SDN using Principal Component Analysis (PCA). In the paper, they suggested a unique real-time DDoS detection system for the SDN environment to examine the network status on traffic data. To minimize the calculating workload, the network is split into various parts. The residual vector value is then estimated in real time. The DDoS will be detected if this value falls below a threshold over a specific time period. In [24], using another improved PCA technique, the authors provided this solution with weighted principal components to counter the DDoS attack. The updated PCA technique is used separately for each subnet after the entire network has been divided into various subnets. This method helped successfully detect the DDoS attacks targeted at the controller or switch at the rate of 95.24%.

Another method of detecting DDoS attacks is proposed in [25]. This article proposes a mechanism that uses a hybrid model combined Entropy with SVM. When a new packet comes in, its feature will be extracted and compared with entropy threshold value. If the packet is identified as abnormal, it will be examined again with SVM detection model that extract 4 features to train. However, their test results are not impressive with an accuracy of 76.12%. The accuracy is not high mainly because of the features using to train the SVM model is not optimized.

In the paper [26], H. Lotfalizadeh *et al* suggested the use of real-time entropy to differentiate between normal and attack traffic. Each flow statistic is only applied to the associated time window. In other words, statistics of flows are retrieved for each time window without any data from earlier time windows. Consequently, the threshold will change over time and help the system detect the new flow of attack more accurately. The suggested approach was tested on 3-time windows of 10, 30 and 60 seconds in which the 10 seconds window provides the best result. Table 1 gave a brief comparison between all the related works and also analyzed their strengths and weaknesses.

Table 1. Summary of the related work

Author	Description	Methods	Strengths	Weaknesses
Tamer Omar et al. [18]	Detection of DDoS in SDN Environment Using Entropy-based Detection	Entropy	- Lightweight, simple	- Use a static threshold - Only evaluate in simulation environment
R. Neres Carvalho et al. [19]	Entropy-Based DoS Attack Identification in SDN	Entropy	- Lightweight, simple - Examine on various window sizes of packet	- Use a static threshold - Only evaluate in simulation environment
Fan et al. [20]	Detection of DDoS Attacks in Software Defined Networking Using Entropy	Fusion Entropy	- Lightweight, simple - Rapid detection due to fusion entropy	- Use a static threshold - Only evaluate in simulation environment
Ye J. et al. [21]	A DDoS attack detection method based on SVM in software defined network	SVM	- Lightweight - Small dataset - High detection rate	- High response time - Only evaluate in simulation environment
M Myint Oo et al. [22]	Advanced Support Vector Machine-based detection for Distributed Denial of Service attack on Software Defined Networking	Advanced SVM	- High detection rate - Examine various types of DDoS attacks	- Complex model - Only evaluate in simulation environment
D. Wu et al. [23]	A Novel Distributed Denial-of-Service Attack Detection Scheme for Software Defined Networking Environments	PCA	- Lightweight, simple	- Lacks the comparison with other methods - Only evaluate in simulation environment
S. Salaria et al. [24]	Implementation and Analysis of an Improved PCA technique for DDoS Detection	Improved PCA	- High detection rates	- Lacks the comparison with other methods - Only evaluate in simulation environment
KM Aung et al. [25]	Anomaly Detection in	Hybrid method	- Combining two simple but	- Training features is not optimized

	SDN's Control Plane using Combining Entropy with SVM		efficiency method	which leads to low accuracy - Only evaluate in simulation environment
H. Lotfalizadeh et al. [26]	Investigating Real-Time Entropy Features of DDoS Attack Based on Categorized Partial-Flows	Entropy	- Use real-time entropy to increase detection rate - Examine on various time window sizes	- High response time - Only evaluate in simulation environment - Lacks the comparison with other methods

3. ENTROPY-BASED DETECTION APPROACH

3.1. Entropy

We will outline the definition and formula used to determine the entropy value in this section. Entropy is a value to calculate the randomness of an event in a period. In this research, we evaluate the arriving target IP, which is gathered by the central controller, to calculate the entropy.

Consider a collection $W$with $n$items ($n \leq N$) that represent a window of NIP addresses and represent the number of distinct destination IP addresses in the incoming packet headers:

$$W = \{x_1, x_2, x_3, \dots, x_n\} \quad (1)$$

Then, the entropy value is determined using to the following formula:

$$H = -\sum_{i=1}^{N} p_i log_{(p_i)} \quad (2)$$

The probability of an IP address in W is:

$$p_i = x_i / N \quad (3)$$

Where x_i represents the number of IP addresses x in W while N is the size of the W (the total IP address). N stands for the window's size.

In (2), if H decreases and approaches zero, it means that there is an anomalous event is occurring throughout the system. Whereas, in normal event, packets are sent to different destinations with almost the same speed, no destinations receive a disproportionately large number of packets compared to other destinations. As a result, H will be in an optimal average approximated state.

In [27], a static test threshold is chosen based on the execution of many attacks in order to detect a DDoS attack.

$$ConfidenceInterval = \overline{X} \pm Z.\frac{\sigma}{\sqrt{N}} \quad (4)$$

In (4), $\overline{X}$ stands for the sample mean while the remaining is called the margin of error: Z is a confidence coefficient, σ is the sample standard deviation and N is the sample size. The chosen confidence level is 95% (Z =1.9599).

Firstly, we will find the difference $\triangle = H_{n_{min}} - H_{a_{max}}$ in which $H_{n_{min}}$ is calculated as normal average traffic minus the reliability interval and $H_{a_{max}}$is equivalent to the average entropy value in attack event plus a confidence interval. Finally, the static threshold is determined as $H_{n_{max}} - \Delta$. This static threshold is fixed and any entropy value below it will be regarded as an ongoing attack [27].

However, this static threshold value based on previous attack data. As a result, it limits the ability to adjust the threshold for identifying new attacks. In this study, the threshold that we utilized will not be fixed but it will fluctuate over time based on the changing of entropy value in incoming traffic. Once the entropy values have been calculated, it will be stored in the window. Based in these parameters, we will calculate the average entropy value $\overline{H_t}$and the standard deviation σ_t for each window.

$$\overline{H_t} = \frac{1}{t}\sum_{i=1}^{t} H_i \tag{5}$$

$$\sigma_t = \frac{1}{t}\sum_{i=1}^{t}\left(H_i - \overline{H_t}\right)^2 \tag{6}$$

In (5) and (6), H_i is the entropy value over period $t$which denotes the number of windows calculated using the previously described in (2). Depending on the parameters determined above, we consider a dynamic threshold value $T_{dynamic}$with the formula defined as follows:

$$T_{dynamic} = \overline{H_t} + C_d.\sigma_t \tag{7}$$

In (7), $\overline{H_t}$ and σ_t denote the average entropy value and standard deviation at the time of$t$, respectively. The normal distribution indicates that 95% of entropy values will fall within the range $\overline{H_t} \pm 2\sigma H_t$. These values, which are smaller than $\overline{H_t} - 2\sigma H_t$, will not significantly affectthe result. Then, we can choose $C_d$for this system based on this fact. In [28], C_d is a constant value and equal to -2 according to experiment.

3.2. Ddos Detection with Entropy

In this research, we use the entropy value of destination IP address field in the incoming packet to detect DDoS attack. Based on this field, the entropy can be calculated and shows the probability of an IP address appearance. From there, we can find the abnormal traffic in the network. The process of DDoS detection is presented in Figure 2. Firstly, the incoming packet is collected. Secondly, we compare the arriving destination IP address with the existed destination IP address. If it is matched, the window packet count is added up so as have to enough 50 packets to calculate the entropy. The packet window is set at 50 to limit the new connections in the network as well as the number of devices connect to the controller. It also speeds up the computing process to reduce response time. If the calculated entropy value is below the threshold, the DDoS attack is identified and vice versa.

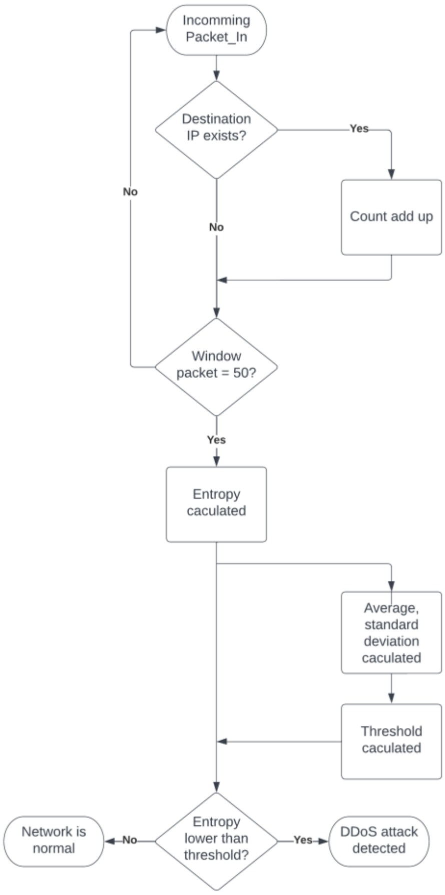


Figure 2. DDoS detection process with entropy

4. IMPLEMENTATION AND RESULT

4.1. Virtual Environmental Simulation

Our simulation was carried out on a Lenovo computer with an Intel® Core™ i5 - 9300H processor operating at 1.2 GHz and 8 GB of DDR4 RAM 2666 MHz, along with Ubuntu 20.04 as operating system. We chose Mininet [29] as a network emulator with a POX controller for simulation purposes. POX [30]is a python based SDN controller which is the improved version of NOX. Compared with other controllers, the usage of POX controllers in detection algorithm is more straightforward and effective. With Mininet, we could create an attack on a virtual server and examine the outcomes of our DDoS attack detection model. Then, we apply our proposed method to detect the DDoS in this model.

In this study, we simulated a DDoS attack with 64 hosts and 9 Open vSwitch, with 1 core Open vSwitch and 8 access switches connect to 8 hosts, as illustrated in Figure3.

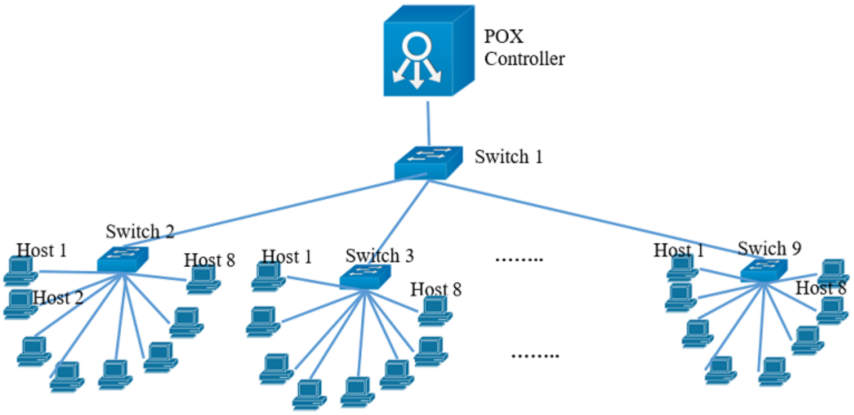


Figure 3. System simulation model

In order to make the host communicate with each other through POX, we use the $l3_learning$module in Pox. This module offers layer 3 learning capabilities by storing a list of IP address information between nodes. $l3\_learning$ will analyse and extract the IP address from each new packet that comes in. This information will be compared with the list and if there is no similar path, the module will start ARP protocol to start the request. In addition, we edited integrated algorithms that make it possible for the POX controller to calculate entropy values and parameters needed to detect attacks when there is an unusual change in incoming traffic.

Scapy [31] handled packet initialization and transmission in the system. Scapy is used to generate UDP packets and spoof their source IP addresses to simulate attack and normal traffic in the simulation system. The hosts in the model are given IP addresses that increase gradually, starting from 10.0.0.1.

1) *Phase 1: The system is in normal state:* In normal state, we use a host to initiate traffic and distribute packets to the whole system. The packet is sent every 0.1 second with a destination port of 80 and a source port of 2. 500 packets which equivalent to 10 windows will be delivered in all during a single run.
   We use formula (2) and (3) to determine the current entropy in a window of 50 packets. Formulas (5) and (6) are used, respectively, to calculate Average Entropy and Standard Deviation. The dynamic entropy threshold is then calculated using the above value and formula (7).
   For instance, the immediate entropy value is almost 0 with 50 identical destination IP addresses. In contrast, when there are 50 separate IP addresses in a window, this figure peaks at about 1.5.
   In normal event, packets are transmitted to a wide range of network destination addresses. Therefore, the randomness will increase as well as the entropy value at that time. As the immediate entropy value exceeds the dynamic entropy threshold value, the system can conclude that the system is in normal state.
2) *Phase 2: The system is in a State of Attack:* There are 2 attack scenarios that we perform in Phase 2 which related to different attack intensities on the system, 50% and 75%, respectively. The rate of an attack is determined by:

$$R_{attackrate} = 1 - \frac{I_a}{I_n+I_a}.100\% \qquad (8)$$

   In (8), I_a and I_n are the period of time where attack traffic and normal traffic occur, sequentially. In the system, normal traffic is randomly forwarded to all hosts, whereas attack traffic is only intended for one host.

First, we launch a 50% attack rate on a host on 10 times. It will allow the controller to detect any attack with packets accounting for 50% of incoming traffic or more. Then the higher-rate tests of 75% were performed on a host to examine a more focused attack. The changes in entropy between the two events can be seen more clearly in these simulations.

4.2. Practical Environmental Simulation

In the preceding session, we detect the DDoS attack in the simulation environment. In this section, the model will be put into practice on the Aruba Switch 2930F in which OpenFlow protocol is enabled.

We build a practical topology with 1 controller, 2 switches and 8 hosts as shown in Figure 4. By using the IP address and listening to the port of the controller interface, we can acquire the flow status of the switch and calculate entropy. The normal and attack script is implemented as same as in the simulation environment. Host 10.10.0.6 produce samples of normal traffic and forward them to the whole network. The attack scenario will be implemented using Scapy. Host 10.10.0.3 is the attacker, from there we use to flood UDP packets to the target, host 10.10.0.7. The network is affected by the DDoS and can not communicate as normal.

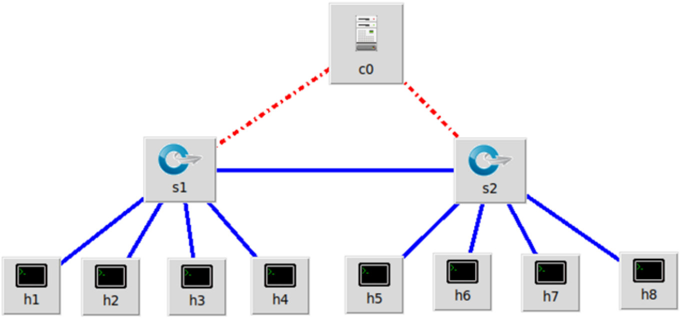


Figure 4. Topology in practical model

4.3. Evaluated Metrics

We gathered 1000 samples to evaluate metrics for the proposed method. Table 2 below shows the overview of the system parameter. There are 7 parameters that determine the performance of methods. True Positive (TP) represent the percentage of attacks event which are successfully detected by the system, while False Positive (FP) is the rate of attack event detected as normal. In contrast, True Negative (TN) stands for the normal event that is successfully detected and False Negative (FN) is the percentage of normal events wrongly detected as an attack. Precision is the ratio of true attack detected flows to all attack-detected flows. The Recall is the ratio of true attack- detected flows to all attack flows. Finally, accuracy is the detection rate of all system.

Table 2. Evaluated metrics for the proposed dynamic entropy.

	TP (%)	**FP (%)**	**TN (%)**	**FN (%)**	**Precision (%)**	**Recall (%)**	**Accuracy (%)**
Simulation	96.39	3.61	98.8	1.2	96.38	97.56	98
Practical	92.16	7.84	94.9	5.1	92.17	90	93.99

The results in Table 2 show that the simulation result is quite higher than the practical results. This happens because our practical topology is quite small compared with the simulation topology. Because there are just 8 hosts in the system, and the window size is 50, the IP address will be duplicated for an average of 4 or 5 times. Therefore, the randomness will decrease and cause the accuracy fall. The fluctuation of entropy, the average and dynamic threshold value in 2 attack scenarios indicate the same result, as shown in Figures5 and 6. In these figures, the green lines represent the entropy value, the black lines represent for the average entropy value and the grey lines are the dynamic threshold value over time. The solid lines illustrate the simulation value while the dashes lines illustrate the practical one.

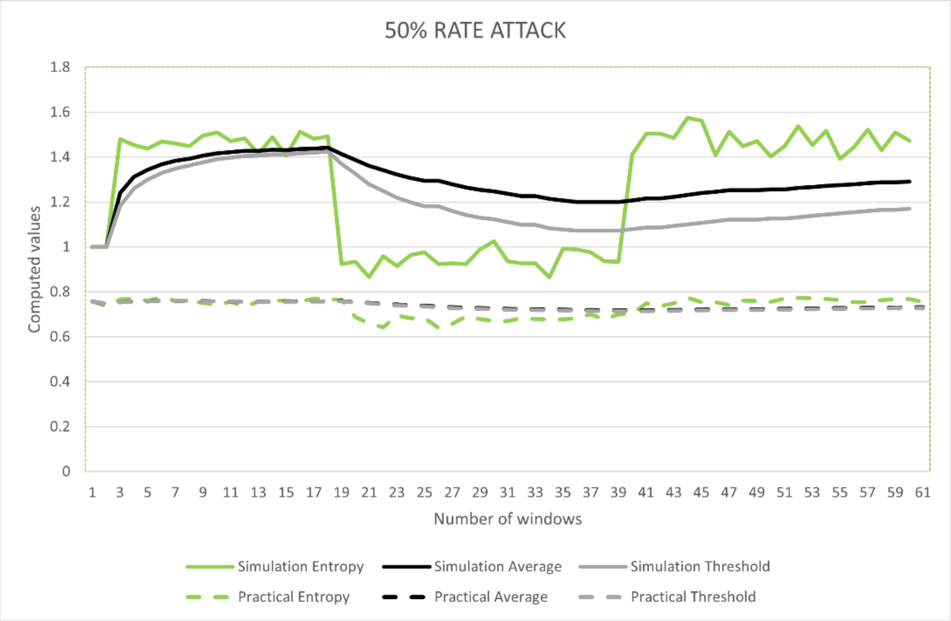


Figure 5. The attack rate of 50% in both models

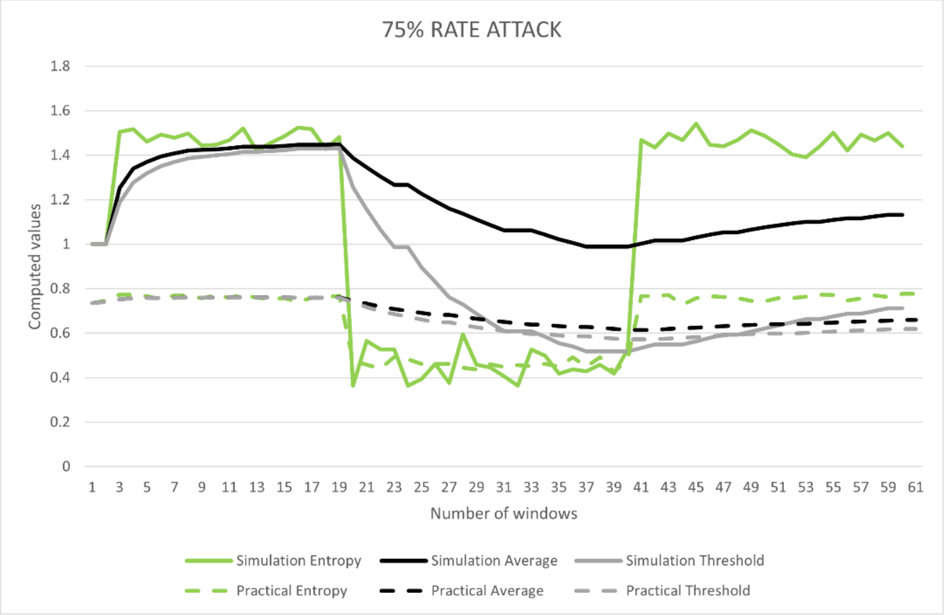


Figure 6. The attack rate of 75% in both models

Each point on the horizontal diagram axis shows the number of packet windows collected and the vertical axis displays the computed value for that period. All the value that illustrated in the figure in the practical model is lower than in the simulation model. As we can see, the rate of 50% and 75% is almost the same shape, however, in the 75% rate, the depth of the attack entropy, as well as the slope of the threshold and average level, will be steeper than the attack at the rate of 50%, in both simulation and practical model. As the speed of attack increases and the number of attack packets generated is fixed, the percentage of attack packets in the window will increase. This will result in a deeper and narrower attack chart.

The rate drop of entropy, as calculated in Table 3, also indicates that the higher the attack rate, the lower the entropy value. In the Table 3, we collect the statistical data between the normal traffic and attack traffic to see the change in entropy value.

Table 3. Statistics among scenarios of the proposed method

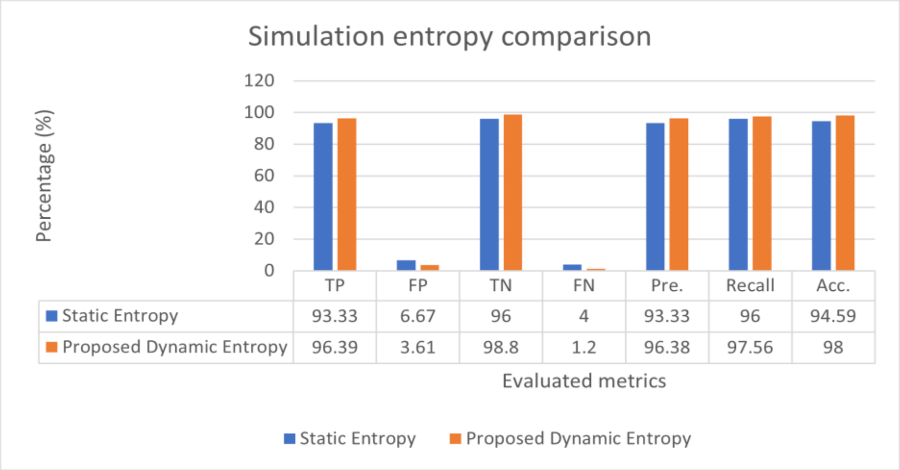
	Normal		50% attack		75% attack	
	Simulation	Practical	Simulation	Practical	Simulation	Practical
Average entropy	1.46897	0.763506	1.29085	0.7316	1.1327544	0.658636
Standard deviation	0.015367	0.020003	0.260507	0.040937	0.494858	0.144209
Margin of error	0.007776	0.010122	0.131828	0.020718	0.24825	0.07298
Max confidence interval	1.476746	0.773628	1.422678	0.752318	1.3810044	0.731616
Min confidence interval	1.46194	0.753384	1.159022	0.710882	0.8845044	0.585656
Min normal traffic – Max attack traffic			0.038516	0.001066	0.0801896	0.021768
Rate drop of entropy			2.64%	0.14%	5.49%	2.89%

To get these values, we take the following steps:

1. Calculate the average entropy and standard deviation based on the collected entropy value
2. Calculate the margin of error due to (4)
3. Find the min and max confidence interval, which equals average entropy minus and plus the margin of error, respectively. The min confidence interval in normal traffic is also the lowest entropy value of normal traffic, whereas the max confidence interval in attack traffic is the highest entropy of attack traffic. Depending on the difference between these two, we can find the rate drop of the entropy.

4.4. Comparison of Our Proposed Method with Other Methods

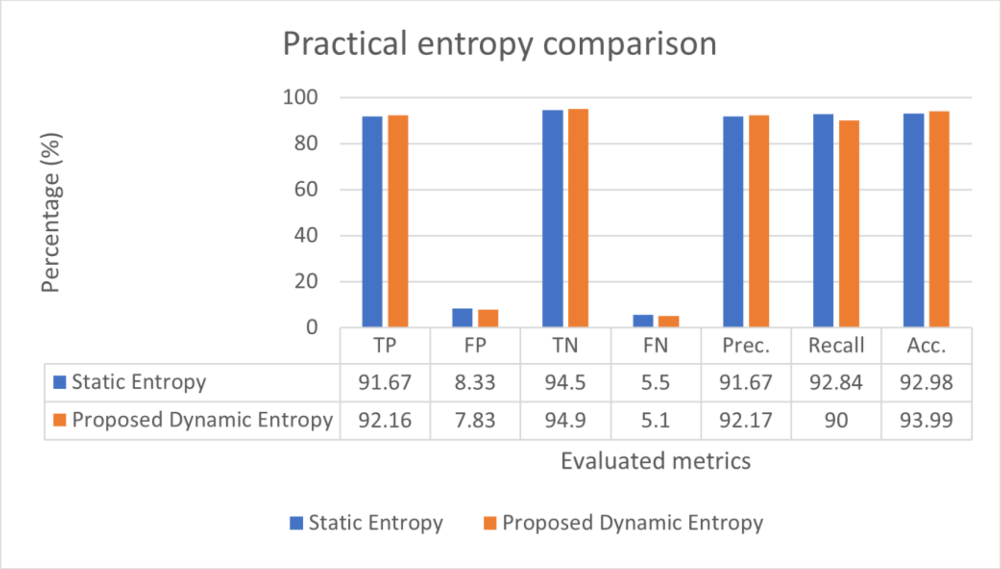
We built another entropy model but used a static threshold to evaluate the performance of our method. Figure 7 shows the comparison of the evaluated metrics above between the proposed dynamic and static entropy in the simulation environment. We can easily see that all the metrics related to positive predicted samples and the correct ratio of dynamic entropy results is higher than the static one. As the probability of an IP address appearance can rise sharply in an attack event, especially for a small window size, it causes a significant fluctuation of entropy value which leads to conflict in identifying a DDoS attack. So, using the dynamic thresholds that change over time can adapt to the entropy value variability of each window. So, the comparison of the entropy value and the threshold will be more precise and increase the accuracy, as shown in the figure below.



	TP	FP	TN	FN	Pre.	Recall	Acc.
■ Static Entropy	93.33	6.67	96	4	93.33	96	94.59
■ Proposed Dynamic Entropy	96.39	3.61	98.8	1.2	96.38	97.56	98

Figure 7. Simulation entropy comparison

The practical result is nearly the same as the simulation with the proposed dynamic entropy results has more positive parameters than the static entropy result, as illustrated in Figure 8. Due to the lack of randomness of an IP address in the window size, the accuracy of both the model decreases and the corresponding metric in each model is pretty close. The recall metric is higher in the static entropy because in the normal event, the random IP address is high and the wrongly detected attack flow ina normal scenario is really low. In addition, the threshold is fixed so that it can not vary as the traffic change. Consequently, the entropy in a normal event is hardly below the pre-set threshold. So, the accurately detected attack flow will have the majority of all predicted attacked flow and lead to the recall metric being higher in this situation.



	TP	FP	TN	FN	Prec.	Recall	Acc.
■ Static Entropy	91.67	8.33	94.5	5.5	91.67	92.84	92.98
■ Proposed Dynamic Entropy	92.16	7.83	94.9	5.1	92.17	90	93.99

Figure 8. Practical entropy comparison

Figure 9 compares the response time between the proposed dynamic method, static threshold and real-time entropy method [26] with the response time of the dynamic method being the fastest. The response time of our simulation and practical model is similar. In our model, we use a window size of 50 packets to evaluate the entropy value and the response time of about 3.5

seconds, which is much faster than the response time of 10 seconds corresponding to the window size used in the paper [26].

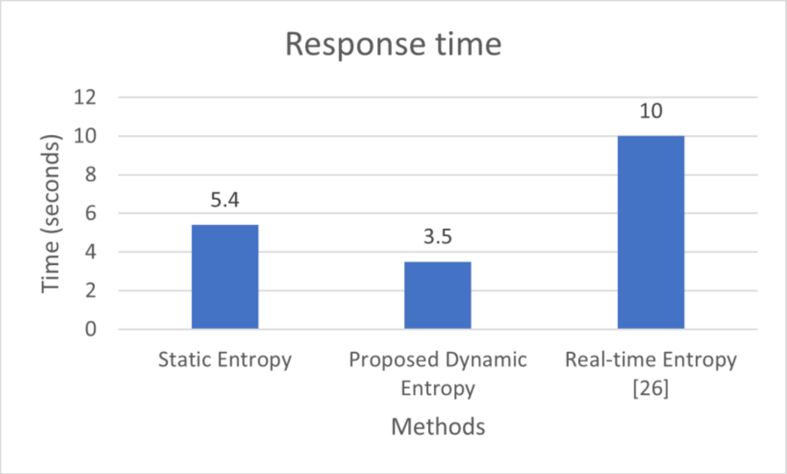


Figure 9. Average response time with different methods

Finally, we compared our method with some of the techniques that we mentioned in the related work. As all of the techniques implemented in simulation, we use our simulation result to compare. The result in Figure 10 shows that the proposed method has the best accuracy.

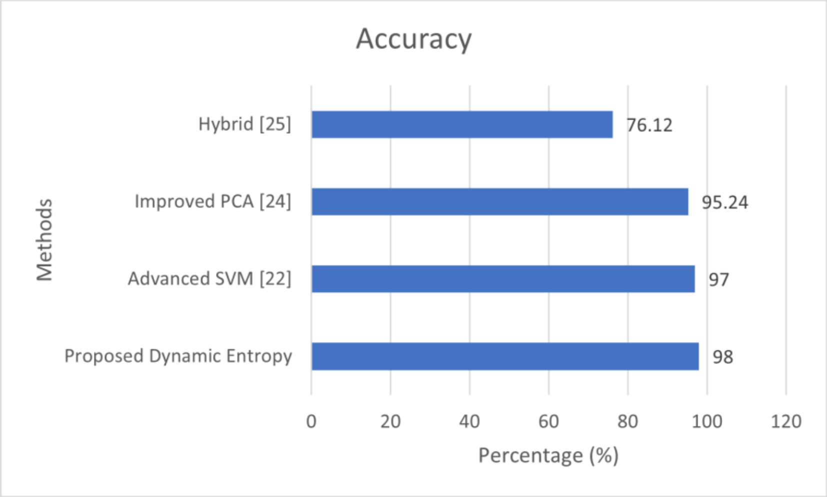


Figure 10. Accuracy with different methods

5. CONCLUSIONS

This study's main content revolves around detecting DDoS attacks using entropy method with the use of dynamic threshold in SDN practical environment. The result shows that dynamic entropy could be used as an accurate method of detecting DDoS attacks and the model can also run on the hardware system with an acceptable result.

However, our practical topology is quite small comparing to the simulation topology, further test case or evaluation of practical model will be performed in the future to improve this method.

This problem related to DDoS attacks in the SDN network is fundamental because of the whole architecture of the SDN network has a center point that controls the entire forwarding plane below it. And there are many directions that this topic could be able to develop in the future. The most possible way is combining machine learning with statistical value to build a Hybrid model to detect DDoS. If the advantages of both methods can be optimized, then a highly detected model with rapid response time can be proposed. Another potential direction is DDoS mitigation as the detection of DDoS attacks must go in tandem with the attack's mitigation. The mitigation process first step is to detect the attack source. It is possible to find the attacker by measuring the traffic rate from each port, and then different solutions can be done to mitigate the whole network.

REFERENCES

[1] D’Arienzo M., Napolitano M.,& Romano S. P. (2022), “Controller Placement Problem Resiliency Evaluation in SDN-based Architectures”, International Journal of Computer Networks & Communications (IJCNC), Vol. 14, No. 5, pp. 101-116.

[2] W. Xia, Y. Wen, C. H. Foh, D. Niyato & H. Xie (2015), "A Survey on Software-Defined Networking," IEEE Communications Surveys & Tutorials, Vol. 17, No. 1, pp. 27-51.

[3] Hamid Farhady, HyunYong Lee & Akihiro Nakao (2015), "Software-Defined Networking: A survey", Computer Networks, Vol. 14, pp. 79-95.

[4] Vanitchasatit M.,Sanguankotchakorn T. (2022), “A Class-based Adaptive QoS Control Scheme Adopting Optimization Technique over WLAN SDN Architecture”, International Journal of Computer Networks & Communications (IJCNC), Vol. 14, No. 3, pp. 55-72.

[5] M. Parashar, A. Poonia& K. Satish (2019), "A Survey of Attacks and their Mitigations in Software Defined Networks," 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-8.

[6] Mahajan, Anmol & Bhandari, Abhinav (2020), "Attacks in Software-Defined Networking: A Review", Proceedings of the International Conference on Innovative Computing & Communications (ICICC).

[7] Kalkan K., Gur, G., &Alagoz F. (2017) “Defense Mechanisms against DDoS Attacks in SDNEnvironment. IEEE Communications Magazine”,Vol. 55, Iss. 9, pp.175–179.

[8] C. Xu, H. Lin, Y. Wu, X. Guo & W. Lin (2019), "An SDNFV-Based DDoSDefense Technology for Smart Cities", IEEE Access, vol. 7, pp. 137856-137874.

[9] L. Yang & H. Zhao (2018), “DDoS Attack Identification and Defense Using SDN Based on Machine Learning Method,” International Symposium on Pervasive Systems, Algorithms and Networks (I-SPAN), pp. 174-178.

[10] H. Kousar, M. M. Mulla, P. Shettar& D. G. Narayan (2021), “Detection of DDoS Attacks in Software Defined Network using Decision Tree”, IEEE International Conference on Communication Systems and Network Technologies (CSNT), pp. 783-788

[11] A. T. Kyaw, M. ZinOo& C. S. Khin (2020), “Machine-Learning Based DDOS Attack Classifier in Software Defined Network”, International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 431-434/

[12] Kalkan K., Gur, G., &Alagoz F. (2018), “JESS: Joint Entropy-Based DDoSDefense Scheme in SDN,” in IEEE Journal on Selected Areas in Communications, Vol. 36, No. 10, pp. 2358-2372.

[13] Lubna Fayez Eliyan& Roberto Di Pietro (2021), “DoS and DDoS attacks in Software Defined Networks: A survey of existing solutions and research challenges”, Future Generation Computer Systems, Vol. 122, pp. 149-171.

[14] Aladaileh, Mohammad Adnan, Mohammed Anbar, Ahmed J. Hintaw, Iznan H. Hasbullah, Abdullah Ahmed Bahashwan, TaiefAlaa Al-Amiedy, &Dyala R. Ibrahim (2023), "Effectiveness of an Entropy-Based Approach for Detecting Low- and High-Rate DDoS Attacks against the SDN Controller: Experimental Analysis", Applied Sciences 13, no. 2: 775.

[15] Mao, J.; Deng, W.& Shen, F. (2018), “DDoS Flooding Attack Detection Based on Joint-Entropy with Multiple Traffic Features”, IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp. 237–243.

[16] Jiang, Y.; Zhang, X.; Zhou, Q.& Cheng, Z. (2016),“An Entropy-Based DDoSDefense Mechanism in Software Defined Networks”, Int. Conf. Commun. Netw.,Vol.1, pp.169–178.

[17] Lubna Fayez Eliyan& Roberto Di Pietro (2021), "DoS and DDoS attacks in Software Defined Networks: A survey of existing solutions and research challenges”, Future Generation Computer Systems, Vol. 122, pp. 149-171.

[18] Tamer Omar, Anthony Ho & Brian Urbina (2019), “Detection of DDoS in SDN Environment Using Entropy-based Detection”, IEEE International Symposium on Technologies for Homeland Security (HST), pp. 1-4.

[19] R. NeresCarvalho, J. LuizBordim& E. AdilioPelinsonAlchieri (2019), "Entropy-Based DoS Attack Identification in SDN," IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 627-634.

[20] Fan, C.; Kaliyamurthy, N.M.; Chen, S.; Jiang, H.; Zhou, Y. & Campbell, C. (2022) "Detection of DDoS Attacks in Software Defined Networking Using Entropy", Applied Science, Vol. 12, Iss. 1, Arc. 370.

[21] Ye J, Cheng X, Zhu J, Feng L & Song L (2018), “A DDoS attack detection method based on SVM in software defined network”, Security and Communication Networks, Vol. 2018, Hindawi.

[22] MyoMyintOo, SinchaiKamolphiwong, ThossapornKamolphiwong&SangsureeVasupongayya (2019), “Advanced Support Vector Machine-(ASVM-) based detection for Distributed Denial of Service (DDoS) attack on Software Defined Networking (SDN)", Journal of Computer Networks and Communications, Vol. 2019, Hindawi.

[23] D. Wu, J. Li, S. K. Das, J. Wu, Y. Ji & Z. Li (2018), “A Novel Distributed Denial-of-Service Attack Detection Scheme for Software Defined Networking Environments,” 2018 IEEE International Conference on Communications (ICC), pp. 1-6.

[24] S. Salaria, S. Arora, N. Goyal, P. Goyal & S. Sharma (2020), “Implementation and Analysis of an Improved PCA technique for DDoS Detection,” 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), pp. 280-285.

[25] KhaingMarlarAung& Nay Min Htaik (2020), "Anomaly Detection in SDN’s Control Plane using Combining Entropy with SVM", International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 122-126.

[26] H. Lotfalizadeh& D. S. Kim (2020), "Investigating Real-Time Entropy Features of DDoS Attack Based on Categorized Partial-Flows," 14th International Conference on Ubiquitous Information Management and Communication (IMCOM), pp. 1-6.

[27] T. Nakashima, T. Sueyoshi & S. Oshima (2010), “Early DoS/DDoS Detection Method using Short-term Statistics”, in International Conference on Complex, Intelligent and Software Intensive Systems, pp. 168-173.

[28] Guo-Chih Hong, Chung-Nan Lee & Ming-Feng Lee, 2019, “Dynamic Threshold for DDoS Mitigation in SDN Environment”, AsiaPacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1-7.

[29] Mininet. [Online]. Available at: http://mininet.org/. [Accessed Dec., 2022].

[30] GitHub, POX Controller. [Online]. Available at: https://noxrepo.github.io/pox-doc/html/. [Accessed Dec. 2022].

[31] Scapy. [Online]. Available at: https://scapy.net/. [Accessed Dec. 2022].

AUTHORS

Dinh Thi Thai Mai graduated from Post and Telecommunication Institute of Technology, Vietnam, in 2006. She received her M.Sc. degree from University of Paris Sud 11, France, in 2008, and her PhD. degree from VNU University of Engineering and Technology, Hanoi, Vietnam, in 2017. She is Deputy Head of Department of Telecommunications Systems, Faculty of Electronics and Telecommunications, VNU University of Engineering and Technology, Hanoi, Vietnam. Currently, her research interests include 5G/6G mobile networks, wireless communications, localization techniques and Security in SDN.

Nguyen Tien Dat is a senior working toward a B.S Degree at VNU University of Engineering and Technology. He is a member of the Telecommunication Systems Laboratory at VNU University of Engineering and Technology. His research interests include Software Defined Networks, Cyber Security and Network Monitoring.

Pham Minh Bao is a senior working toward a B.S Degree at VNU University of Engineering and Technology. He is a member of Telecommunication Systems Laboratory at VNU University of Engineering and Technology. His research interests include Software Defined Networks, Cyber Security and Cloud Computing.

Can Quang Truong is a senior working toward a B.S Degree at VNU University of Engineering and Technology. He is a member of Telecommunication Systems Laboratory at VNU University of Engineering and Technology. His research interests include Software Defined Networks, Cyber Security and Network Automation.

Nguyen Thanh Tung is a senior working toward a B.S Degree at VNU University of Engineering and Technology. He is a member of Telecommunication Systems Laboratory at VNU University of Engineering and Technology. His research interests include Software Defined Networks, Cyber Security and System Designing.