applied sciences

MDPI

Article

Detection of DDoS Attacks in Software Defined Networking Using Entropy

Cong Fan [1,2,*], Nitheesh Murugan Kaliyamurthy [2], Shi Chen [1], He Jiang [1], Yiwen Zhou [1] and Carlene Campbell [2]

[1] School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China; chenshi@whut.edu.cn (S.C.); Hejiang2019@whut.edu.cn (H.J.); 293423@whut.edu.cn (Y.Z.)
[2] Wales Institute of Science and Art, University of Wales Trinity Saint David, Swansea SA1 8PH, UK; n.kaliyamurthy@uwtsd.ac.uk (N.M.K.); carlene.campbell@uwtsd.ac.uk (C.C.)
* Correspondence: 293410@whut.edu.cn

Featured Application: This study proposes a detection method of Distributed Denial of Service attacks in Software Defined Networking, which uses the property of entropy to measure the occurrence of attack behavior in the network. The significance of this study is to quickly and effectively detect Distributed Denial of Service attacks in the Software Defined Networking and protect the SDN controller against security threats.

Abstract: Software Defined Networking (SDN) is one of the most commonly used network architectures in recent years. With the substantial increase in the number of Internet users, network security threats appear more frequently, which brings more concerns to SDN. Distributed denial of Service (DDoS) attacks are one of the most dangerous and frequent attacks in software defined networks. The traditional attack detection method using entropy has some defects such as slow attack detection and poor detection effect. In order to solve this problem, this paper proposed a method of fusion entropy, which detects attacks by measuring the randomness of network events. This method has the advantages of fast attack detection speed and obvious decrease in entropy value. The complementarity of information entropy and log energy entropy is effectively utilized. The experimental results show that the entropy value of the attack scenarios 91.25% lower than normal scenarios, which has greater advantages and significance compared with other attack detection methods.

Keywords: software defined networking; entropy; distributed denial of service attacks

Citation: Fan, C.; Kaliyamurthy, N.M.; Chen, S.; Jiang, H.; Zhou, Y.; Campbell, C. Detection of DDoS Attacks in Software Defined Networking Using Entropy. *Appl. Sci.* **2022**, *12*, 370. https://doi.org/10.3390/app12010370

Academic Editor: Christos Bouras

Received: 30 November 2021
Accepted: 28 December 2021
Published: 31 December 2021

1. Introduction

SDN breaks the shackles of traditional network complexity and coupling and makes it possible for network architecture to satisfy flexibility, reliability and security at the same time. It separates the control plane from the data plane and separates the control function of the network from the data forwarding function [1]. The control plane is only responsible for routing decisions, while the data plane realizes these decisions by forwarding packets and other behaviors. The separation of the two planes can improve the abstraction and programming ability of the network and makes the network structure less tedious and redundant.

Centralized control and implementation of network programming solve the interdependence problem in three planes which are control plane, data plane and application plane. The control plane communicates with the application plane through northbound API. The southbound API is mainly the OpenFlow protocol, through which the control plane is able to communicate with the data plane.

Essentially, SDN provides a brain (controller) for the underlying network, making each network plane device centrally programmable [2] and giving the administrator absolute control over the use of software to manage network functions through a centralized control point.

Although SDN architecture has many advantages compared with traditional network, it is often subjected to network threats and attacks. Network threats to SDN are mainly reflected in security device licensing and global view acquisition [3]. Because packets are passed according to flow rules, physical security devices do not have the right to decide, and attackers can bypass security devices before deployment. The controller is the core of the entire network and can obtain various network status information. The attacker can use the controller to directly grasp the global view of the network to launch serious attacks. SDN has a clear plane structure, and the attack objects at different planes are different. At the control plane, because the controller manages the entire network, an attacker damaging the controller can cause serious damage. Controllers are the main targets of attacks in recent years.

In emerging SDN deployment scenarios such as data centers, the centralized control plane is the main cornerstone, which makes SDN more scalable [4]. The controller has the right of absolute control over the whole SDN. They communicate with switches through commands and perform network operations through packet switching and routing SDN applications. It acts as a core brain, controlling the forwarding operation of the data plane and managing the traffic behavior of the entire network. The controller has the unique property of providing a global view of the network, making it the highest priority target in the network [5]. The global view of the network includes flow rules for network devices and various statistics.

The main security challenges on the control plane are as follows:

- Application threats: Applications implemented on the control plane may seriously threaten the security of the control plane.
- Scalability Threats: Due to lack of scalability, the control plane gradually becomes saturated and cannot handle more flows [6].
- DDoS attack: Controllers are vulnerable to DDoS attacks [7]. When receiving a packet, the switch matches the packet with the flow table entry. If the packet fails to match, the switch encapsulates the packet header into a packet_in message and sends the packet to the controller. When an attacker sends a large number of packets to the network, the switch forwards these packets to the controller. A large number of DDoS spoofing packets exhaust controller resources and make the controller unable to work properly, posing serious threats to the controller and the entire network.
- Other threats: Inconsistent and conflicting controller configurations may result in failure to receive network information [8]. Malicious applications can easily be developed to apply to controllers. There is no effective trust management mechanism between controller and application.

Among the confirmed security vulnerabilities, DDoS is still one of the most important security problems at the control plane [9]. In view of the importance of controllers in software-defined networks, DDoS attacks on controllers are very dangerous and protecting controllers from attacks is also a major concern of researchers.

DDoS attacks send a large amount of traffic to the network and consume network resources and cause network congestion. Many DDoS attacks are launched from distributed hosts [10]. DDoS attacks have two stages. First, an attacker creates a distributed attack network of thousands of targeted computers, known as zombies, robots or attack hosts. The attack host then sends massive traffic to the victim either on the attacker's command or automatically [11]. To build an attack network, attackers seek out computers that are less secure, such as those that have not been properly patched.

A DDoS attack is an aggressive and destructive network attack that causes the system to stop working by depleting system resources. It can destroy the user's available network services, thus seriously threatening the network. When malicious data packets are sent by attackers on the network, normal traffic is processed or even cannot be processed due to the consumption of network resources. As a result, the network and servers become jammed and normal services are interrupted. Attackers who apply DDoS often target SDN mainly because of its unique characteristics.

In order to solve the impact of DDoS attacks on the SDN controller, this paper proposes to use fusion entropy for attack detection. Through research, it is found that information entropy has the characteristics of rapid entropy reduction in attack scenarios, but it does not have the function of rapid detection. Although the log energy entropy reduces the entropy value in the attack scenario less than the information entropy, it can detect the attack at the beginning of the attack. Based on this, this paper proposes the fusion entropy, which combines the two entropies by weighting. The fusion entropy can effectively utilize the complementarity between the two entropies, which can quickly detect attacks and significantly reduce the entropy value. The remainder of the paper is organized as follows: Section 2 introduces related work of this study. Section 3 gives a detailed introduction to the proposed method. Section 4 is about the attack detection simulation. Section 5 is the experimental results and discussion. Section 6 presents the conclusion.

2. Related Works

The security threats facing SDN have received extensive attention. The most common and well-known attack in SDN is the DDoS attacks. So far, many DDoS detection methods have been proposed.

2.1. Software Defined Networking

SDN is a network architecture in which the control plane manages the forwarding state of the data plane [12]. The control plane is decoupled from the data plane so that network devices only need to implement the forwarding function, which is the biggest advantage of SDN, as shown in Figure 1. The forward decision is based on a stream that combines packet field values and operation instructions. Streams abstract the behavior of different types of network devices, such as switches, routers, and firewalls.

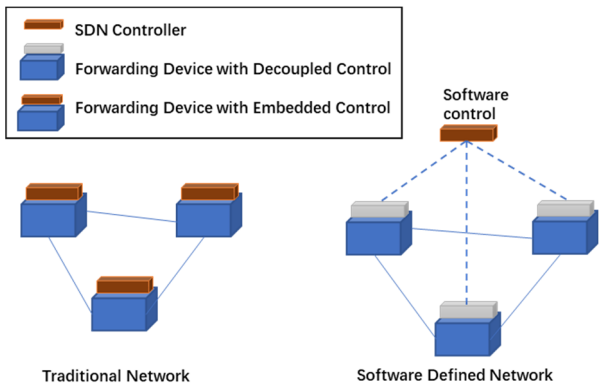

Figure 1. Decoupled control in SDN vs. traditional network.

In the existing traditional network, this abstraction is not implemented. Because network devices in traditional networks are interrelated and dependent on each other [13]. Programming for streams provides a lot of flexibility. The key to achieving flexibility is the introduction of functional separation between the implementation of switching hardware and the forwarding of legitimate traffic. Because of these advantages, Cisco, Juniper and other major Internet companies are investing in this technology [14]. Enterprise engineers or network administrators can dynamically change the centralized traffic console on demand. This behavior does not affect the underlying network devices such as switches. This rapid response to changes in requirements is critical and urgently needed for the application of real scenarios.

SDN has the advantages of abstraction and flexibility, various enterprises and companies have stepped on the road to use this technology. Although SDN brings a lot of advantages, when a network structure system is attacked, the consequences are serious and how to protect SDN from attack need serious consideration.

2.2. Analysis of Threats over SDN Network

From 2020 to 2025, some enablers such as cloud service providers will greatly promote the use of SDN, which is expected to increase by 19% [15]. However, SDN architecture still has some security holes. These security vulnerabilities are related to some of their own characteristics [16].

SDN separates the control plane from the data plane through OpenFlow protocol. Frequent communication between switches and controllers makes SDN vulnerable to attacks, which greatly reduces the network performance. Switches in traditional networks not only complete forwarding tasks but also need to formulate flow rules, while SDN formulates flow rules through controllers. This means that if one of the controllers and switches is attacked, the other will be affected. SDN is attacked mainly for the following four reasons:

- The lack of security protection mechanism makes the controller in a potential threat;
- Complex interactions between various applications increase the burden of flow rules;
- Lack of authentication mechanism and application authorization;
- There is not enough secure encryption for the flow rules, making it easy for the flow rules to be tampered with after publication.

Although SDN brings the benefits of programmability, centralization and flexibility, it also brings new security issues. In ref. [17], the author analyzes the security problems of the three planes in SDN architecture and puts forward corresponding security solutions. In order to deal with the security threats of SDN, relevant organizations have developed corresponding security challenges and solutions. It is necessary to analyze the security threats of this architecture from different functional planes.

In SDN architecture, the security of the control plane (namely OpenFlow controller) directly affects the data forwarding plane. As the decision center, the controller has a high probability to become the target of attackers. D. Melkov et al. [18] analyzed the advantages and disadvantages of SDN in detail and classify security threats into SDN's unique challenges and the same challenges as classical networks. The security vulnerability of controller is studied and discussed from different angles, because it is the most vulnerable component in SDN architecture. In a multi-controller architecture, the threat to distributed multi-controllers is also a security problem. The coexistence and cooperation of multiple controllers will lead to configuration conflict. In addition, malicious access can occur to applications running on the controller, which requires a different security policy for each application. J.H. Cox et al. [19] paid attention to how SDN is further applied in practice and discuss structural security and attack detection measures. More consideration has been given to the security of the controller. A DDoS attack on a controller prevents legitimate traffic from using controller resources and functions by exhausting computing or memory resources. An attacker launches an attack on SDN within a short period of time, and the attack traffic is mixed with normal traffic. Under normal circumstances, it is difficult to distinguish the two types.

2.3. Distributed Denial of Service in SDN

A DDoS attack is one of the biggest security threats to SDN network in recent ten years. It can not only prevent legitimate users from accessing and using network resources, but also destroy the entire network [20]. Therefore, it is vital to protect the SDN network from DDoS attacks.

Attackers combine multiple hosts to form zombie host groups that meet their attack requirements. These zombie hosts send a large number of useless data packets to the target, making it consume a large amount of resources such as CPU and bandwidth to process these packets. Once the target host receives far more data packets than the load, it will not work properly and cannot process legitimate data packets. DDoS attacks are easy to implement and favored by attackers.

When the controller is attacked, the switch receives the attack packets and matches them with flow entries one by one. The attack packet is not valid and of course cannot

match the flow table in the flow table entry. In this case, the switch encapsulates the packet as packet-in message and sends it to the controller. Then the controller makes decisions on the direction of the data packet. Attackers send a large number of attack packets so that packet-in messages continuously enter the controller and occupy a large number of controller resources [21]. As a result, the controller cannot process legitimate traffic data, and the controller cannot work properly or is faulty.

The above content shows that the security problems under SDN cannot be underestimated. Although it subverts the traditional network and brings many amazing benefits, the security problems hidden behind it pose a great threat to its own development and application. It is very important to further discover the security vulnerabilities of SDN architecture and find appropriate solutions to promote the further development of SDN.

DDoS attacks are of various types, such as TCP flood [22], UDP flood [23], and ICMP flood. An attacker sends a large amount of garbage traffic to the target network by operating the attack source, which sharply reduces the available bandwidth and prevents the target host from communicating with the outside world. TCP flood and UDP flood attacks use massive TCP and UDP packets to attack victims. ICMP flood is applied by displaying ICMP request packets to disturb normal traffic destined for the target host. In a TCP flood attack, the attacker sends a large number of forged IP address packets to the destination host. As the IP addresses of the packets are forged, the destination host cannot receive the response from the sender. The difference between UDP flooding attacks and TCP flooding attacks is that UDP is connectionless, which is commonly used in voice and video applications [24]. An attacker initiates UDP flooding attacks by generating excessive UDP packets to random ports of the destination host and prevents the attack target from responding to legitimate users.

When a DDoS attack occurs on a controller, the impact on the SDN is more serious. Therefore, effective detection methods are of great significance to attack detection and response. So far, there have been many methods to detect DDoS attacks against SDN controllers. Each technology approach has a different design and definition in terms of architecture, timing, and parameters.

Packet-based detection and flow-based detection are two main DDoS attack detection methods in SDN environment. The flow-based DDoS attack detection method checks flow tables on the switch, while the packet-based detection method should check all packets on the network. In the flow-based method, the pre-set trigger mechanism can first judge whether there is an attack in the network, and then start the attack detection algorithm if there is an attack. The package-based approach needs to examine every packet in the network, regardless of whether the attack time has occurred. In comparison, the flow-based detection method consumes less system resources and is more efficient.

J. Boite et al. [25] proposed a new method based on switching processing capability. This method can quickly respond to DDoS attacks. However, the author could have added more details about the complex flows in the switch. The scheme in [26] defines the predicted value of the number of switch requests according to Taylor series, and can filter out the requests that lead to a sharp decrease in entropy. P. Kumar et al. [27] proposed a scheme for single TCP SYN flood, which can improve the processing delay. In ref. [28], the author uses Paspberry Pis as OpenvSwitches and proves that virtual hardware resources can be used as a solution for the Internet of Things. In this paper, entropy is used as a DDoS detection scheme, but the disadvantage is that the threshold cannot be dynamically updated. In ref. [29], the author uses multi-scale principal component analysis to denoise signals, but in this paper, data packets are used for different application objects, so denoising technology can be considered to reduce interference in data packets in subsequent work. In ref. [30], the author proposed the adaptive wavelet transform method for signal decomposition to improve the classification accuracy. In SDN, there are also many kinds of interference, this method can be considered to improve the accuracy of attack detection in the future. In ref. [31], the author proposes to use multivariate empirical wavelet transform to decode motor imagery tasks. Similar to the preprocessing method mentioned above, we will also

preprocess the interference existing in the attack detection process in the future. Similarly, in [32], the author combines dimension reduction technology with neural network to improve system performance. The entropy value used in this paper can be used as feature vector for reduction and neural network, which will be discussed in the future. The authors in [33] propose a statistical feature to describe the traffic characteristic condition entropy of DDoS attacks, and then use support vector machine classifier to identify attacks. This method can distinguish attack traffic from normal traffic. Sabah et al. [34] proposed a distributed artificial neural network method based on anomaly detection and signature detection. This hybrid model can improve the detection performance of DDoS attacks. In ref. [35], entropy is used to detect whether the traffic is abnormal firstly. After extracting the attack characteristics, BiLSTM-RNN neural network algorithm is used to train the data set and classify the real-time traffic to achieve DDoS attack detection. I. Sumantra et al. [36] calculates entropy using various attributes of source IP and TCP tokens in the network. This technology can detect TCP SYN flood, Ping flood, and Slow HTTP attacks, making good use of the programmability and flexibility provided by centralized control. T.A. Tang et al. [37] proposed an intrusion detection system based on the Gated Recursive Unit Recursive Neural Network (GRU-RNN), which uses the least functions to achieve higher computational efficiency without affecting the performance of the network. The detection method based on the entropy change of destination IP address [38] can be used to identify the attack in the early stage. Entropy is a statistical concept that represents the randomness of a particular dataset. The higher the entropy is, the stronger the divergence of the data set is; the lower the entropy is, the weaker the divergence of the dataset is.

Many DDoS detection methods based on SDN have been designed. In this paper, a fast DDoS detection method is proposed which can effectively detect attacks based on the significant change of entropy and it utilizes the working characteristics of flow table and the randomness of entropy.

3. Proposed Work

3.1. Fusion Entropy

Entropy is an important part of information theory. Entropy can measure the randomness of packets entering the network, which is the main reason for using entropy in DDoS detection. Entropy increases with the increase of randomness and decreases with the decrease of randomness. Common entropy includes information entropy, mean energy, mean Teager Kaiser energy, shannon wavelet entropy and log energy entropy. Considering the probability of using destination IP address in this paper, for the case of only one variable, this paper mainly uses information entropy and log energy entropy and achieves the purpose of improving the detection effect by using complementarity through fusion.

To calculate the information entropy, the first is to calculate the probability of the destination IP address. The variable x is used to define the destination IP address of the packet, and the probability of x is calculated by using Equation (1). Set the number of packets in the window to n. The probability of each element in the window is defined as p.

$$p_i = \frac{x_i}{\sum_{i=1}^{n} x_i} \tag{1}$$

In Equation (1), $i = \{1,2,3...n\}$, $0 \leq p_i \leq 1$.

Then calculate information entropy, the calculation formula is Equation (2). H is the information entropy of the destination IP address of packets appearing in a specific window.

$$H1 = -\sum_{i=1}^{n} p_i \log p_i \tag{2}$$

Log energy entropy as another kind of entropy, its calculation formula is shown in Equation (3). In Equation (3), n and p_i still represent the number of packets and the probability of destination IP addresses.

$$H2 = -\sum_{i=1}^{n} \log {p_i}^2 \tag{3}$$

Through research and experiments, it is found that the entropy value of information entropy will be significantly reduced when an attack occurs, but the attack cannot be detected quickly, while the log energy entropy can quickly detect the attack, but the entropy value is not as obvious as the information entropy. This paper considers the fusion of the two entropies through weighting, so as to achieve complementary effects. Since p ranges from 0 to 1, according to the mathematical properties of the logarithmic function, the log energy entropy will get a negative value. In order to better integrate with the information entropy, we multiply the log energy entropy by minus one and weighted with information entropy. This change is shown in Equation (3). The selection of weight is based on the change rate of entropy decline of the two kinds of entropy when the attack occurs. The fusion entropy calculation is shown in Equation (4). In Equation (4), 0.75 in weight w_1 is the change rate of entropy corresponding to information entropy, while 0.13 in weight w_2 is the change rate of entropy corresponding to log energy entropy. The subsequent experimental results show that the fusion entropy effectively realizes the complementary advantages of information entropy and log energy entropy, which can not only quickly detect the attack but also have a high decline rate of entropy.

$$\begin{aligned} H3 &= w_1 * H1 + w_2 * H2 \\ w_1 &= \frac{0.75}{0.75 + 0.13} \\ w_2 &= \frac{0.13}{0.75 + 0.13} \end{aligned} \tag{4}$$

When multiple data packets are received on the same host or switch port in a specific window and the number of data packets exceeds the threshold, DDoS attacks are detected. The basic steps handled by this detection method are shown in Figure 2. During an attack, if the computation entropy of a specified window continuously drops below the threshold, the target port on the specified switch is blocked. In the experiment, data packets will continuously flow into the network. For each incoming packet, the entropy is calculated. If the entropy value is higher than or equal to the threshold, the next calculation will be carried out normally. If the entropy value falls below the threshold, the packet is logged. The threshold is determined based on the entropy fluctuation range in normal traffic scenarios. In the later experiments, the threshold will be described in more detail.

3.2. Packet Generation

Packet generation is done by scapy. It is a very powerful tool for package generation, scanning, sniffing, attacking, and forgery. Two parameters set in scapy are packet type and packet generation interval. The packet type includes TCP and UDP packets. UDP packets are used to cheat the source IP addresses of packets. The interval is set to fit the test case.

In this experiment, using scapy to generate network traffic so that normal and attack traffic can be simulated. With scapy, forged IP packets can be generated, in which way the identity of the real attacker is hidden. Run the scapy program on the host to send packets, both normal packets and attack packets (UDP packets and TCP packets) to the host in the designed network topology.

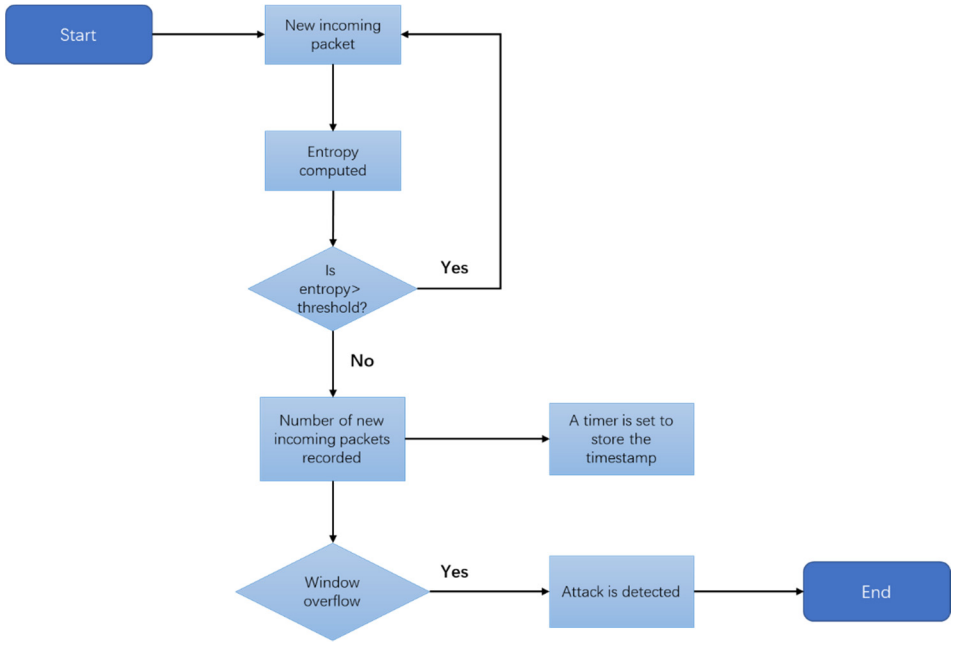

Figure 2. Flowchart of attack detection method.

3.3. Window Size

The window size should be set to less than or equal to the number of hosts to provide accurate calculations. In this paper, the window size is set to 50. The primary reason for choosing 50 is the limited number of incoming new connections to each host in the network. In SDN, once a connection is established, packets will not pass through the controller unless there is a new request.

The second reason is that each controller can only connect to a limited number of switches and hosts. Finally, the third reason for choosing this size is the amount of computation done for each window. A list of 50 values is much faster to compute than a list of 500 values, and attacks within the 50 packets window are detected earlier. Given the controller's limited resources, this window size is ideal for networks with only one controller and a few hundred hosts.

4. DDoS Attacks Detection on Simulated SDN Network

4.1. Experiment Environment Setup

This experiment was carried out on an HP laptop with 8 GB RAM. The experimental operating system adopts Linux Ubuntu 20.04 with 20 GB memory and uses Ubuntu image files to run in Oracle VM VirtualBox software. The version of mininet is 2.3.0 and runs locally on Linux. The network topology shown in Figure 3 is created using mininet. The network is a tree structure with a depth of 2. The switch adopts OpenFlow switch, which can support OpenFlow protocol.

In order to realize the function of the controller, it is necessary to turn on the controller to observe the changes of traffic in the network during simulation. In SDN, when a data flow arrives, the Openflow switch will parse the data packet and match it with the local flow entry. If the match is successful, the data flow will be processed according to the forwarding policy. The flow table counter corresponding to this flow table entry is increased by 1. If the match fails, the packet is encapsulated as a packet_in message and sent to the controller through the Openflow switch. Then, the controller sends the updated flow table information to the Openflow switch. The flow table includes three modules: matching domain, counter and forwarding policy.

In the experiment, the destination IP address of the packet is extracted from the matching and the corresponding number of packets is obtained from the counter. Before the experiment, the window size and threshold size need to be set in advance. If the

window setting is too small, not enough samples will be obtained, which will affect the accuracy of attack detection. If the window setting is too large, the computing load of the controller will increase.

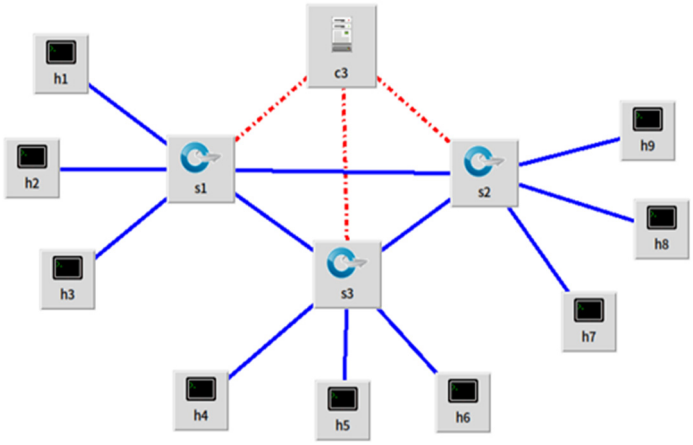

Figure 3. Designed network topology.

4.2. Pox Controller

It is very necessary to choose a suitable controller. At present, there are many controllers available for researchers and developers. This paper selects some representative controllers and compares their basic information, as shown in Table 1. The Pox controller is used in this experiment. It is widely used in experiments with high speed and light weight. It is designed as a platform, so a user-defined controller can be built on it. Pox is an open source controller written in Python. Its advantage is to facilitate the application interface to the controller so that they can run in parallel with the controller.

Table 1. SDN Controllers.

Controller	Language	Openflow	Thread	Release
Ryu	Python	1.0~1.4	Single	03/2015
Pox	Python	1.0	Single	10/2013
Nox	C++	1.0&1.3	Multiple	02/2014
Beacon	Java	1.0	Multiple	09/2013
Floodlight	Java	1.0	Multiple	11/2012
Opendaylight	Java	1.0&1.3	/	03/2015
ONOS	Java	1.0&1.3	/	03/2015

The Pox controller is developed by Stanford and is based on Openflow. Kernel and component are two important parts of Pox. The kernel is the assembly point for all components, and components can interact with each other through the kernel. Pox provides Openflow interface for topology discovery and path selection and can customize components to realize specific functions. Pox controller supports rapid development of controller prototype functions and it can produce superior performance applications, which decreases the burden of developers and improves development efficiency.

In addition to the advantages of programming language, Pox also has the advantages of clear architecture, good performance and strong scalability, so it has attracted the attention of many developers and researchers. Based on the above reasons, the Pox controller was selected for the simulation experiment in this paper.

5. Experimental Results and Discussion

In this experiment, each test contains 250 data packets, divided by window value of 50, 5 results will be obtained, and a total of 16 times of running, and we will get a total of

80 data results. The results are plotted as a discounted graph, where the horizontal axis represents representative packets and the vertical axis represents the value of calculated entropy. Table 2 shows the information entropy values of partial data packets in the normal and attack scenario. The entropy values of the data packets with 10 nodes in the above representative data packets are selected. The entropy value calculated in the normal scenario is that host 1 randomly sends normal traffic to the simulated network topology. To simulate the attack scenario, two hosts 4 and 6 send attack traffic to the target host 56.

Table 2. Information entropy values.

Information Entropy		Incoming Packets
Normal	**Attack**	
1.11439	1.13712	1
1.1174	1.13712	10
0.84288	0.20198	20
1.0235	0.02878	30
1.0235	0.20452	40
1.05419	0.20017	50
1.0235	0.01419	60
1.0235	1.0235	70
0.84288	1.10643	80

Figure 4 shows the comparison of information entropy between the two scenarios. Among the 80 packets, the green line represents the information entropy value in the normal scenario, and the blue line represents the information entropy value in the attack scenario. The green line shows the packet's information entropy fluctuates between 0.8 and 1.4. The minimum entropy value of 0.8 was adopted as the threshold of our experiment, which could ensure the minimum error.

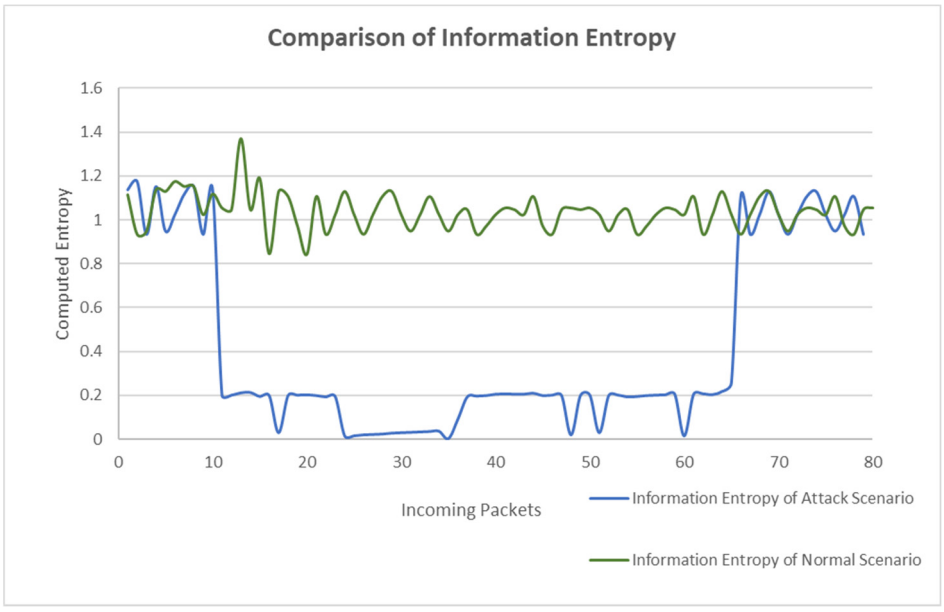

Figure 4. Information entropy variation.

From the 10th packet to the 65th packet, the entropy value drops sharply and remains in the low value region between 0 and 0.21. The lowest point of normal flow entropy is 0.84288, and the highest point of attack flow entropy is 0.20882. The difference between the two is 0.63406, indicating a 75% decrease in information entropy under attack in the normal scenario. According to the experimental results of information entropy, when an attack occurs, the entropy value decreases obviously, but the attack is not detected immediately.

Without changing other experimental conditions, log energy entropy is used to measure the randomness of the network. Table 3 shows the calculated log energy entropy values of partial data packets in the normal and attack scenario. The entropy values of the data packets with 10 nodes in the above representative data packets are selected.

Table 3. Log Energy Entropy values.

Log Energy Entropy		Incoming Packets
Normal	**Attack**	
2.66956	2.67964	1
3.05772	1.23388	10
2.97717	1.0398	20
3.05772	1.0398	30
2.97717	1.04654	40
3.05772	1.04654	50
2.97717	1.09536	60
3.05772	1.0398	70
2.97717	2.96453	80

Figure 5 shows the comparison of log energy entropy under the two scenarios. In normal traffic scenarios, the entropy value of packets ranges from 1.7 to 3.5. We use the minimum entropy value of 1.7 as the threshold to ensure a small error. It can be seen from the figure that in the attack scenario, the value of log energy entropy starts to decline from the third packet, showing an earlier change. The lowest point of entropy of normal flow is 1.87379. The maximum entropy of attack traffic is 1.62796 when very few abnormal data are discarded. The difference between the two is 0.24583, indicating that the log energy entropy decreases by 13% when the attack occurs. According to the experimental results of log energy entropy, when an attack occurs, the attack can be detected immediately, but the change of entropy is small, and even a few extremely high entropy values will appear.

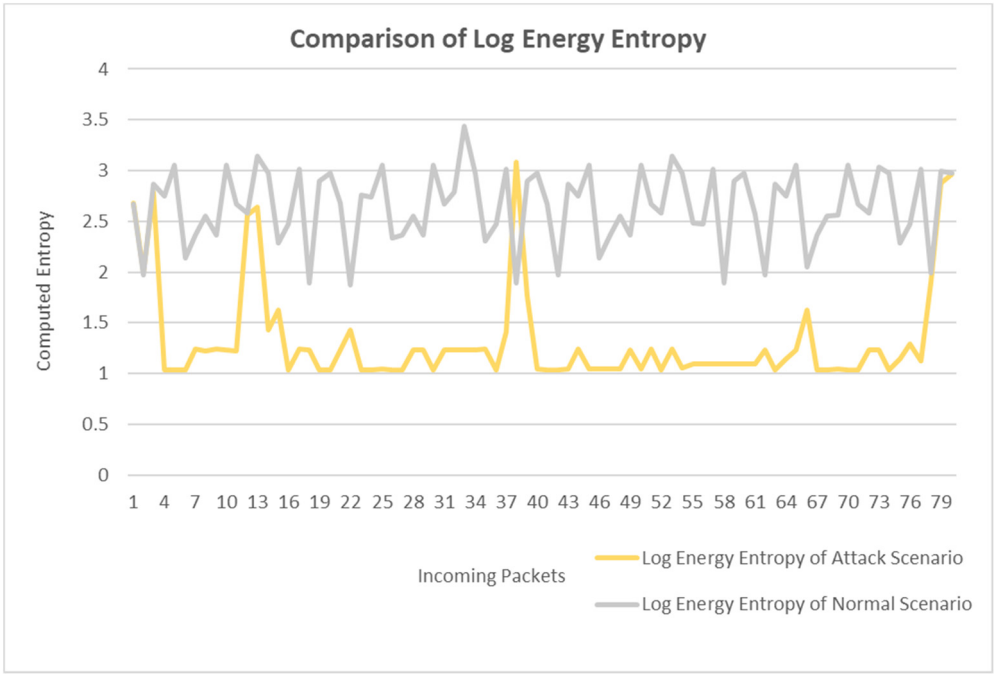

Figure 5. Log energy entropy variation.

Without changing other experimental conditions, fusion entropy is used to measure the randomness of the network. Table 4 shows the calculated fusion entropy values of

partial data packets in the normal and attack scenario. The entropy values of the data packets with 10 nodes in the above representative data packets are selected.

Table 4. Fusion entropy values.

Fusion Entropy		Incoming Packets
Normal	Attack	
3.03872	2.93872	1
2.18832	0.13869	10
2.25056	0.13869	20
2.08832	0.1411	30
2.25056	0.13869	40
2.18832	0.14284	50
2.25056	0.16812	60
2.28832	0.14284	70
2.25056	2.35056	80

Figure 6 shows the comparison of fusion entropy under the two scenarios. In normal traffic scenarios, the entropy value of packets ranges from 2 to 3.3. We use the minimum entropy value of 2 as the threshold to ensure a small error. It can be seen from the figure that in the attack scenario, the value of fusion entropy starts to decline from the third packet, showing an earlier change. The lowest point of entropy of normal flow is 2.01975. The maximum entropy of attack traffic is 0.17669 when very few abnormal data are discarded. The difference between the two is 1.84306, indicating that the fusion entropy decreases by 91.25% when the attack occurs. The experimental results of fusion entropy show that when an attack occurs, the fusion entropy can quickly detect the attack, inheriting the advantages of log energy entropy, while the entropy value decreases significantly, inheriting the advantages of information entropy, and making good use of the complementarity of the two entropies to effectively detect the attack.

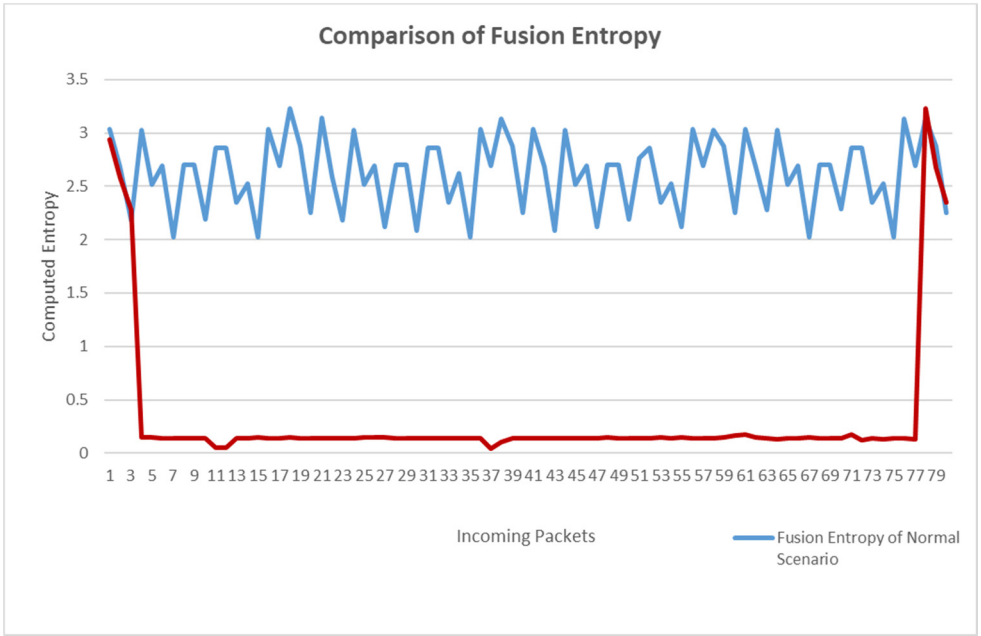

Figure 6. Fusion entropy variation.

Figure 7 lists the comparison between different entropy calculation methods. It can be seen from the figure that the fusion entropy represented by solid line is obviously better than the other two kinds of entropy represented by dotted line and has good effects in rapid detection and entropy value decrease, thus achieving the purpose of attack detection.

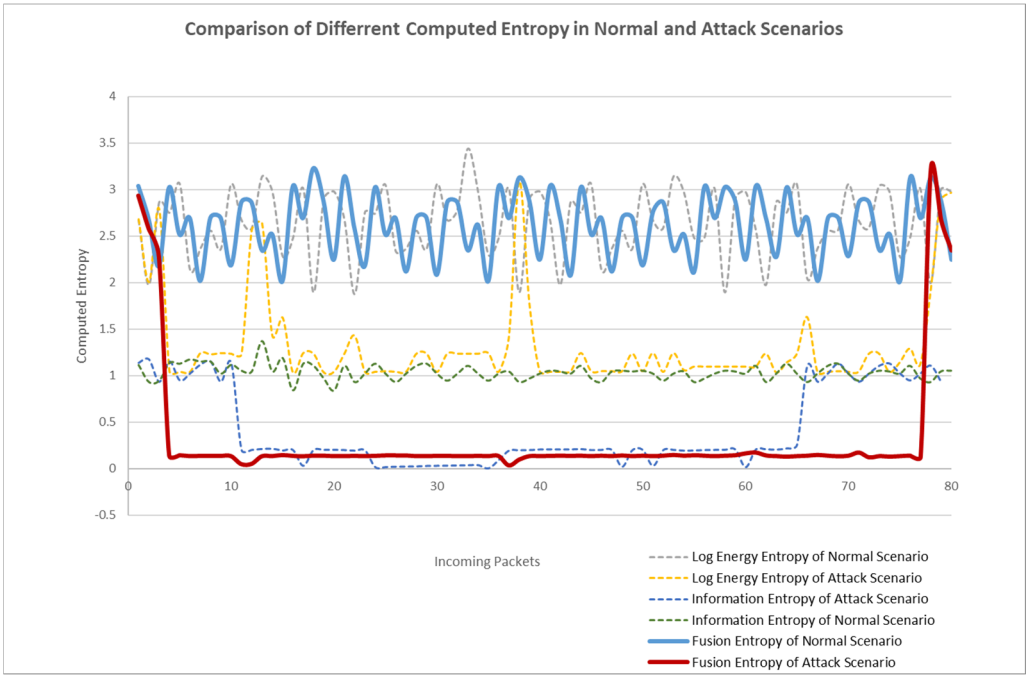

Figure 7. Comparison of different entropy variation.

Table 5 lists the comparison between the proposed method and other attack detection methods. The results show that the fusion entropy detection method used in this paper can achieve a higher entropy decline rate and has the function of rapid detection.

Table 5. Comparison with other attack detection methods.

Method	Decline Rate of Entropy
Average normalized entropy [27]	89.90%
Information entropy + trigger [35]	82.14%
Shannon entropy [38]	74.75%
Proposed method	91.25%

The above experimental results show that the fusion entropy has a good attack detection performance and can effectively make use of the complementarity of information entropy and log energy entropy. It not only has a fast attack detection speed, but also has a high entropy decline rate, which is significantly improved compared with other methods, which indicates the method can well detect DDoS attacks in simulated network scenarios.

6. Conclusions

This paper focuses on protecting the controller as the core of a software-defined network by detecting DDoS attacks. In order to achieve the purpose of attack detection, this paper proposes a fusion entropy method. This method effectively uses the advantages of information entropy and log energy entropy to achieve complementarity. Fusion entropy has the advantages of rapid detection and obvious changes in entropy value, which can be used in attacks. The attack is detected in the fastest time. At the same time, the entropy value when the attack comes is about 91.25% lower than the entropy value in the normal scenario, which can effectively detect the attack. Compared with other attack detection methods, it has great advantages in protecting SDN controller from DDoS attack.

In the future, detecting DDoS attacks in structures that connect or communicate between multiple controllers will be the subject of further research. In addition, since low-rate DDoS attacks are more difficult to detect because the difference between normal traffic and low-rate DDoS attacks is not obvious when they occur on the network, low-rate

DDoS attacks can also be the subject of further research in this paper. At the same time, to reduce the interference in attack, detection is also a key research issue in the future. It can be considered to combine entropy value with dimension reduction technology and machine learning method to achieve better attack detection.

Author Contributions: Conceptualization, C.F., S.C.; methodology, C.F., N.M.K.; software, C.F.; validation, C.F., Y.Z. and H.J.; formal analysis, C.F., Y.Z.; investigation, C.F.; resources, H.J., S.C.; data curation, C.F.; writing—original draft preparation, C.F.; writing—review and editing, Y.Z.; visualization, C.F.; supervision, S.C., N.M.K.; project administration, S.C., C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of Hubei Province, China, under Grant No. 2021CFA001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: For the experimental data and results of this article, you can contact us at 293410@whut.edu.cn.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Al-Adaileh, M.A.; Anbar, M.; Chong, Y.W.; Al-Ani, A. Proposed statistical-based approach for detecting distribute denial of service against the controller of software defined network (SADDCS). In *MATEC Web of Conferences*; EDP Sciences: Les Ulis, France, 2018; Volume 218, p. 02012.
2. Sanjeetha, R.; Srivastava, S.; Pokharna, R.; Shafiq, S.; Kanavalli, A. Mitigation of DDoS attack instigated by compromised switches on SDN controller by analyzing the flow rule request traffic. *Int. J. Eng. Technol. (UAE)* **2018**, *7*, 46–49.
3. Mladenov, B.; Iliev, G. Searching for Optimal Software Defined Network Controller Against DDoS Attacks. In Proceedings of the 2020 International Symposium on Networks, Computers and Communications (ISNCC), Montreal, QC, Canada, 20–22 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–4.
4. Sanjeetha, R.; Benoor, P.; Kanavalli, A. Mitigation of DDoS attacks in Software Defined Networks at application level. In Proceedings of the 2019 PhD Colloquium on Ethically Driven Innovation and Technology for Society (PhD EDITS), Bangalore, India, 18 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–3.
5. Zhao, Y.; Iannone, L.; Riguidel, M. On the performance of SDN controllers: A reality check. In Proceedings of the 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN), San Francisco, CA, USA, 18–21 November 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 79–85.
6. Nanda, S.; Zafari, F.; DeCusatis, C.; Wedaa, E.; Yang, B. Predicting network attack patterns in SDN using machine learning approach. In Proceedings of the 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Palo Alto, CA, USA, 7–10 November 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 167–172.
7. Dong, S.; Sarem, M. DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks. *IEEE Access* **2019**, *8*, 5039–5048. [CrossRef]
8. Gkountis, C.; Taha, M.; Lloret, J.; Kambourakis, G. Lightweight algorithm for protecting SDN controller against DDoS attacks. In Proceedings of the 2017 10th IFIP Wireless and Mobile Networking Conference (WMNC), Valencia, Spain, 25–27 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.
9. Dong, P.; Du, X.; Zhang, H.; Xu, T. A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–6.
10. Ferrag, M.A.; Maglaras, L.; Moschoyiannis, S.; Janicke, H. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *J. Inf. Secur. Appl.* **2020**, *50*, 102419. [CrossRef]
11. Bouras, C.; Kollia, A.; Papazois, A. SDN & NFV in 5G: Advancements and challenges. In Proceedings of the 2017 20th Conference on innovations in clouds, internet and networks (ICIN), Paris, France, 7–9 March 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 107–111.
12. Kreutz, D.; Ramos, F.M.; Verissimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-defined networking: A comprehensive survey. *Proc. IEEE* **2014**, *103*, 14–76. [CrossRef]
13. Gangadhara, S.; Hasyagar, S.N.; Damotharan, U. Deployable SDN architecture for network applications: An investigative survey. In Proceedings of the 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 15–16 March 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 43–49.

14. Hatagundi, M.D.; Kumaraswamy, H.V. A comprehensive survey on different attacks on SDN and approaches to mitigate. In Proceedings of the 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 27–29 March 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 624–627.
15. Yadav, S.K.; Suguna, P.; Velusamy, R.L. Entropy based mitigation of Distributed-Denial-of-Service (DDoS) attack on Control Plane in Software-Defined-Network (SDN). In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–7.
16. Jiménez, M.B.; Fernández, D.; Rivadeneira, J.E.; Bellido, L.; Cárdenas, A. A Survey of the Main Security Issues and Solutions for the SDN Architecture. *IEEE Access* **2021**, *9*, 122016–122038. [CrossRef]
17. Kaur, G.; Gupta, P. Hybrid approach for detecting ddos attacks in software defined networks. In Proceedings of the 2019 Twelfth International Conference on Contemporary Computing (IC3), Noida, India, 8–10 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
18. Liu, Y.; Zhao, B.; Zhao, P.; Fan, P.; Liu, H. A survey: Typical security issues of software-defined networking. *China Commun.* **2019**, *16*, 13–31. [CrossRef]
19. Ahmad, I.; Namal, S.; Ylianttila, M.; Gurtov, A. Security in software defined networks: A survey. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2317–2346. [CrossRef]
20. Melkov, D.; Paulikas, S. Security Benefits and Drawbacks of Software-Defined Networking. In Proceedings of the 2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, 22 April 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–4.
21. Cox, J.H.; Chung, J.; Donovan, S.; Ivey, J.; Clark, R.J.; Riley, G.; Owen, H.L. Advancing software-defined networks: A survey. *IEEE Access* **2017**, *5*, 25487–25526. [CrossRef]
22. Aladaileh, M.A.; Anbar, M.; Hasbullah, I.H.; Chong, Y.W.; Sanjalawe, Y.K. Detection techniques of distributed denial of service attacks on software-defined networking controller—A review. *IEEE Access* **2020**, *8*, 143985–143995. [CrossRef]
23. Deore, S.; Patil, A. Survey denial of service classification and attack with protect mechanism for TCP SYN flooding attacks. *IRJET* **2016**, *3*, 1–4.
24. Rajkumar, M.N. A survey on latest DoS attacks: Classification and defense mechanisms. *Int. J. Innov. Res. Comput. Commun. Eng.* **2013**, *1*, 1847–1860.
25. Boite, J.; Nardin, P.A.; Rebecchi, F.; Bouet, M.; Conan, V. Statesec: Stateful monitoring for DDoS protection in software defined networks. In Proceedings of the 2017 IEEE Conference on Network Softwarization (NetSoft), Bologna, Italy, 3–7 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–9.
26. Huang, X.; Du, X.; Song, B. An effective DDoS defense scheme for SDN. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.
27. Kumar, P.; Tripathi, M.; Nehra, A.; Conti, M.; Lal, C. SAFETY: Early detection and mitigation of TCP SYN flood utilizing entropy in SDN. *IEEE Trans. Netw. Serv. Manag.* **2018**, *15*, 1545–1559. [CrossRef]
28. Sambandam, N.; Hussein, M.; Siddiqi, N.; Lung, C.H. Network security for iot using sdn: Timely ddos detection. In Proceedings of the 2018 IEEE Conference on Dependable and Secure Computing (DSC), Kaohsiung, Taiwan, 10–13 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–2.
29. Sadiq, M.T.; Yu, X.; Yuan, Z.; Aziz, M.Z. Motor imagery BCI classification based on novel two-dimensional modelling in empirical wavelet transform. *Electron. Lett.* **2020**, *56*, 1367–1369. [CrossRef]
30. Sadiq, M.T.; Yu, X.; Yuan, Z.; Fan, Z.; Rehman, A.U.; Li, G.; Xiao, G. Motor imagery EEG signals classification based on mode amplitude and frequency components using empirical wavelet transform. *IEEE Access* **2019**, *7*, 127678–127692. [CrossRef]
31. Sadiq, M.T.; Yu, X.; Yuan, Z.; Zeming, F.; Rehman, A.U.; Ullah, I.; Li, G.; Xiao, G. Motor imagery EEG signals decoding by multivariate empirical wavelet transform-based framework for robust brain–computer interfaces. *IEEE Access* **2019**, *7*, 171431–171451. [CrossRef]
32. Sadiq, M.T.; Yu, X.; Yuan, Z. Exploiting dimensionality reduction and neural network techniques for the development of expert brain–computer interfaces. *Expert Syst. Appl.* **2021**, *164*, 114031. [CrossRef]
33. Liu, Y.; Yin, J.; Cheng, J.; Zhang, B. Detecting DDoS attacks using conditional entropy. In Proceedings of the 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), Taiyuan, China, 22–24 October 2010; IEEE: Piscataway, NJ, USA, 2010; Volume 13, V13-278–V13-282.
34. Ahuja, N.; Singal, G. DDoS attack detection & prevention in SDN using OpenFlow statistics. In Proceedings of the 2019 IEEE 9th International Conference on Advanced Computing (IACC), Tiruchirappalli, India, 13–14 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 147–152.
35. You, X.; Feng, Y.; Sakurai, K. Packet in message based DDoS attack detection in SDN network using OpenFlow. In Proceedings of the 2017 Fifth International Symposium on Computing and Networking (CANDAR), Aomori, Japan, 19–22 November 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 522–528.
36. Alzahrani, S.; Hong, L. Detection of distributed denial of service (DDoS) attacks using artificial intelligence on cloud. In Proceedings of the 2018 IEEE World Congress on Services (SERVICES), San Francisco, CA, USA, 2–7 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 35–36.

37. Sun, W.; Li, Y.; Guan, S. An improved method of DDoS attack detection for controller of SDN. In Proceedings of the 2019 IEEE 2nd International Conference on Computer and Communication Engineering Technology (CCET), Beijing, China, 16–18 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 249–253.
38. Sumantra, I.; Gandhi, S.I. DDoS attack Detection and Mitigation in Software Defined Networks. In Proceedings of the 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 3–4 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.