future internet

MDPI

Article

A HMM-R Approach to Detect L-DDoS Attack Adaptively on SDN Controller

Wentao Wang, Xuan Ke * and Lingxia Wang

College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China; wangwt@mail.scuec.edu.cn (W.W.); 2015110294@mail.scuec.edu.cn (L.W.)

* Correspondence: 2017120290@mail.scuec.edu.cn

Received: 27 July 2018; Accepted: 21 August 2018; Published: 23 August 2018

Abstract: A data center network is vulnerable to suffer from concealed low-rate distributed denial of service (L-DDoS) attacks because its data flow has the characteristics of data flow delay, diversity, and synchronization. Several studies have proposed addressing the detection of L-DDoS attacks, most of them are only detect L-DDoS attacks at a fixed rate. These methods cause low true positive and high false positive in detecting multi-rate L-DDoS attacks. Software defined network (SDN) is a new network architecture that can centrally control the network. We use an SDN controller to collect and analyze data packets entering the data center network and calculate the Renyi entropies base on IP of data packets, and then combine them with the hidden Markov model to get a probability model HMM-R to detect L-DDoS attacks at different rates. Compared with the four common attack detection algorithms (KNN, SVM, SOM, BP), HMM-R is superior to them in terms of the true positive rate, the false positive rate, and the adaptivity.

Keywords: L-DDoS attacks; SDN; data center network; adaptive detection; HMM-R

1. Introduction

With the rise of cloud computing, big data, and virtualization technology, data centers are increasingly playing an important role in providing them with large-scale computing and storage functions [1]. The data center network has the characteristics of low delay, diversity, and synchronization, which leads to the concealment and diversity of the attack mode. As the bridge connecting the large-scale servers in the data center for distributed computing and the core warehouse of data transmission, calculation, and storage, attack detection in the data center network is of great practical significance.

As a new distributed denial of service (DDoS) attacks, low-rate distributed denial of service (L-DDoS) attacks mainly take advantage of some security vulnerabilities of the network protocol or adaptive mechanism to reduce network performance [2]. An intelligent attack-CrossFire is proposed in [3], this attack initiates link attacks at a low rate, and then generates aggregated traffic flow through multiple attack sources to achieve the purpose of blocking services in the data center. It can be known that L-DDoS not only could reduce the service performance of the data center, but also could cause a collapse in the network link.

Traditional L-DDoS detection algorithms mainly have some problems. In terms of collecting network traffic, full network traffic control in traditional networks requires a large number of experimental equipment, which is difficult to achieve in general experiments [2], so most studies are based on the assumption that the full network traffic control can be implemented, but it is not enough to meet the reality, resulting in incomplete and unintelligent of traffic flow collection. In terms of detection period setting in traditional networks, the detection period is a custom period of time, if the time is too short, some attack traffic will be missed; if the time is too long, although the detection

rate is improved, the time consumption is too long. In terms of statistical attributes, most traditional machine learning algorithms use Shannon entropy to measure the attribute, but L-DDoS attack traffic is very similar to normal traffic. Shannon entropy performs well in the detection of the relatively large traffic attacks with obvious attack behavior, it may lead to a high false positive using Shannon entropy to detect L-DDoS attacks [4]. In terms of detection performance, traditional methods use Shannon entropy to pre-process the data, which makes the detection data can only be divided into two categories, when L-DDoS attacks with different rates occur in the network, they cannot be adapted to detect, and may classify some degree of attacks into normal classes, resulting in a high false alarm rate. In addition, if the similarity between linked packets is considered, the accuracy will be improved to a certain extent.

To address above-mentioned problems, we propose an adaptive detection method to detect L-DDoS attacks. The main contributions of our work can be summarized as follows:

1. We combine software defined network (SDN) to achieve complete control of network traffic that improves the detection of single link or single host of traditional data center network and utilize OpenFlow's massage mechanism to increase the flexibility of the detection period.
2. We use the PACKET_IN message of OpenFlow mechanism to set up a trigger detection period.
3. We propose a Renyi entropy to add the entropy difference between the normal traffic and L-DDoS to reduce the false positive in attribute statistics.
4. We propose a probabilistic model of combining Renyi entropy and hidden Markov model (HMM-R) to define a variety of states with double stochastic processes of hidden states and observed states to improve the true positive and reduce the false positive and increase the flexibility.

The rest of the paper is organized as follows. We provide an overview of the related work in Section 2. We introduce the background about L-DDoS, SDN, Renyi entropy, and the hidden Markov model in Section 3. An adaptive HMM-R model of L-DDoS attacks is presented in Section 4. Section 5 describes the implementation and evaluation. Finally, we conclude the paper in Section 6.

2. Related Work

The detection algorithms of L-DDoS attacks in the data center network can be divided into non-machine learning algorithms and machine learning algorithms.

2.1. Non-Machine Learning Algorithms

Non-machine learning algorithms include periodic detection, spectrum analysis, information metrics, correlation coefficient analysis, and so on. Chaovalit et al. [5] propose discrete wavelet (DWT) to analyze abnormal time series data. However, one of the most obvious disadvantages of time-frequency domain transformation is the high complexity of detection time, which is not suitable for low delay data center network. Oshima et al. [6] and Bhuyan et al. [7,8] propose the entropy theory to measure the change of source IP distribution. If the source IP entropy is less than the threshold, it indicates that there may be an abnormal attack. Mousavi and Sthilaire [9] propose an early detection method of information entropy based on destination IP, to prevent the controller from paralyzing and reducing the controller's burden in SDN. This kind of detection is effective with relatively large flow attacks, but the effect is not obvious in a low-rate attack similar to a normal traffic [4], it may make a high false positive rate. Jadhav and Patil [10] propose the maximum entropy model for detection. Xiao et al. [11] propose the correlation analysis between data flows and combined with k-nearest neighbor (KNN) algorithm for detection. In [12–14], there is a linear relationship between the traffic in a normal state. No matter what the traffic obeys, Spearman and partial rank correlation coefficient are used to detect the attack. However, this detection method is not effective for L-DDoS attack with similarity, which will result in a low detection rate. Hoque et al. [15] propose a multivariate analysis method that makes full use of the feature with an attack attribute change at least in an attack state,

and for threshold detection, but it can only detect attacks with high intensity of attack. Zhang et al. [16] use congestion control characteristics to detect the packet loss rate. In all, all these methods assume that the network has complete control ability. However, traditional networks cannot control all routers, they cannot respond to traffic information in time. These detection methods tend to cause a low detection rate.

2.2. Machine Learning Algorithms

In terms of machine learning algorithms. Suresh et al. [17] combine with chi-square evaluation and fuzzy theory for adaptively detecting attacks. Yusof et al. [18] combine KNN and support vector machine (SVM) to detect slow attacks and extended attacks on the application layer caused by DNS services. Priyanka et al. [19] propose SVM to measure the low-rate and high-rate DDoS attacks in the distributed network. Compared with other measurement methods, SVM plays an important role in reducing false positive rate, but it is not suitable for new attacks and large capacity samples. Yan et al. [20] propose a lightweight fuzzy synthetic decision model for detection of wireless network attacks, considering multiple attack factors and using SDN controller centralized control to reduce detection resources. Braga et al. [21] propose an unsupervised neural network algorithm-self-organization map (SOM) algorithm-to detect attacks. Giotis et al. [22] propose a mode to reduce load, using SFLOW traffic collection technology to statistics OpenFlow flow table for detecting local switch and OpenFlow switch traffic attacks. Cui et al. [23] uses an error back propagation (BP) algorithm to detect the attack attributes in the OpenFlow flow table. Even though the high accuracy of the neural network method, it also has the disadvantage of slow convergence of training time.

3. Background

In this section, we mainly introduce L-DDoS attack, data center network topology, Renyi entropy, and HMM-R model.

3.1. L-DDoS Attacks

Low-rate distributed denial of service (L-DDoS) attacks are a new type of DDoS attacks. L-DDoS attacks take advantage of the vulnerability of TCP's congestion avoidance algorithm. The RTO of the TCP flow is used as the pulse period, and the RTT is used as the pulse width to send the pulse flow of the cycle, which makes the target host continuously lose the packet and enter the congestion avoidance state, resulting in a large reduction in the throughput of the target host. The average traffic of L-DDoS attacks is much lower than that of traditional DDoS attacks and it is similar to normal traffic. Compared with DDoS attacks, the efficiency of L-DDoS attacks is greatly improved, and L-DDoS attacks are very effective to evade detection and prevention, which will have greater harm.

3.2. Software Defined Network

Since it is not easy to implement full network traffic control in traditional networks [2], we need a technology that can easily implement full network traffic control. Software defined network (SDN) is an innovative network architecture and stems from Stanford University in the United States [24]. The core idea of SDN is to separate the coupled control plane from the data plane. A centralized controller is used to control the data plane; the controller connects the switch actively or passively, and the control instruction is sent; the network element is responsible for the simple data forwarding according to the received information command. From the perspective of application function, the controller realizes various network management application services by defining various flexible interfaces and developing northbound interfaces [25]. As a centralized control management technology, SDN can be used to collect the real-time traffic of data center network in an all-round way. It lays the foundation for the good performance of detection algorithms, and finally achieves the purpose of flexible detection of concealed L-DDoS attacks.

3.3. Renyi Entropy

In real-time networks, attack traffic does not always exist, but tends to be stable from scratch. Renyi entropy is α order generalized entropy that is proposed in phase space and it is a common generalized entropy [26]. Before the attack begins, the high probability events in the data center network are normal traffic accesses, when attack occurs, high probability decreases. With the increase of attack rate, the probability of L-DDoS attack increases, and the high probability event that eventually transits to the data center network is the L-DDoS attack. In this transition process, there is a certain order of Renyi entropy which makes the difference between normal traffic and attack traffic maximum. The following is the definition of Renyi entropy:

Definition 1. *Renyi entropy is defined as: This is an example of an equation:*

$$H_\alpha(x) = \frac{1}{1-\alpha}\log\left(\sum_{i=1}^{n} p_i^\alpha\right) \tag{1}$$

where p_i is the probability of the random variable x_i, and $\sum_{i=1}^{n} p_i = 1$.

Renyi entropy has several entropy forms that are used frequently, they are maximum entropy, minimum entropy, and Shannon entropy.

- Maximum entropy: when $\alpha = 0$ or $p_{x_1} = p_{x_2} = \cdots = p_{x_n}$, a maximum entropy is obtained. The maximum entropy model is to use the weakest randomness of maximum entropy to obtain the model parameters at the highest time. Then the model parameters are obtained at the highest uncertain time.

$$\max(H_\alpha(x)\ ) = \log(n) \tag{2}$$

- Minimum entropy: when $\alpha \to \infty$, $H_\alpha(x)$ converges to the minimum entropy. When the possible number of all events is b, the probability of all events is shown to be $1/2^{-b}$ in the minimum entropy. Minimum entropy is the smallest of the entropy family and is the most lightweight method to measure unpredictability, it pays an important role in theoretical computer science.

$$H_\infty(\alpha)\ = \min(-\log(p_i)) = \max(\log(p_i)) = -\log(\max(p_i)) \tag{3}$$

- Shannon entropy: when $\alpha \to 1$, $H_\alpha(x)$ converges to Shannon entropy.

$$\lim_{\alpha\to 1} H_x(\alpha) = \lim_{\alpha\to 1}\frac{1}{1-\alpha}\log_2\left(\sum_{i=1}^{n} p_i^\alpha\right) = \lim_{\alpha\to 1}\frac{\sum_{i=1}^{n} p_i^\alpha \ln(p_i)}{\sum_{i=1}^{n} p_i^\alpha \ln(2)}$$

$\sum_{i=1}^{n} p_i = 1$, so:

$$\lim_{\alpha\to 1} H_x(\alpha) = -\sum_{i=1}^{n} p_i * \log_2(p_i) \tag{4}$$

When a random variable obeys the uniform distribution, the entropy of all kinds of Renyi is the same, and the detection effect has few differences. However, in the case of uneven distribution, the entropy of all kinds of Renyi is quite different, which can reflect the unpredictability in the process of detection.

3.4. Hidden Markov Model

Hidden Markov model (HMM) is composed of a quintuple: $\lambda = (X, Y, A, B, \pi)$, where X represents the set of hidden states; Y represents the set of observational states; A is a hidden state probability transfer matrix; B represents the probability transfer matrix of observation state; and π represents the probability in the initial time of each hidden state. The transfer between hidden states is a random process, and the transfer between observed states is also a random process, so the most obvious feature of HMM is a double random process.

HMM is a probabilistic model that predicts the possibility of attack in the form of probability. For L-DDoS attacks with background traffic, HMM can remove noise and fully distinguish attack events in sequential events. Unlike traditional detection methods, which classify states into two categories-attack or no attack-HMM can define the number of states according to different degrees of attack, and finally detect L-DDoS attacks at different rates adaptively.

$$\begin{aligned} &A = \left[a_{ij}\right]_{N\times N}, a_{ij} = P\left(x_{t+1} = s_j \middle| x_t = s_i\right), i \geq 1, j \leq N \\ &B = \left[b_{ij}\right]_{N\times N}, b_{ij} = P\left(y_t = o_j \middle| x_t = s_i\right), 1 \leq i \leq N, 1 \leq j \leq M \\ &\pi = (\pi_1, \pi_2, \cdots, \pi_N), \pi_i = P(x_i = s_i), 1 \leq i \leq N \end{aligned} \tag{5}$$

The Equation (5) showed that at time t, if the state in previous moment is s_i, a_{ij} will be the probability for next moment when the state is s_j; if the state is s_i, b_{ij} will be the probability when the observed value is o_j; π_i represents the probability when $t = 1$ and the state is s_i. A and π determine the hidden Markov chain and generate an unobservable state sequence. B determines how to generate observations from the state and integrate with the state sequence to determine how to generate observation sequences.

4. HMM-R Detection Scheme

This scheme uses the corresponding relationship between the HMM hidden state sequence and the feature observation sequence to regard the states of L-DDoS attacks at different rates as hidden states and regard the data features after preprocessing as the feature observation sequence. As two discrete random variables, hidden state and observed state depict the possibility of attack by probability model. We use Renyi entropy to get data features and make use of the transfer relation between states to make adaptive detection of L-DDoS attacks.

4.1. Overall Architecture

As shown in Figure 1, our data center network topology is fat-tree and the detection scheme architecture consists of four modules: data preprocessing, model initializing, model training, and model detecting. We collected and parsed the PACKET_IN data packets, then gathered Renyi entropies of source IP and destination IP; we use Baum–Welch algorithm for training the observation sequence data and Viterbi algorithm for detecting L-DDoS attacks. After introducing the detection features and performance indices in the next section, the four modules of HMM-R detection scheme for L-DDoS attacks are described in detail.

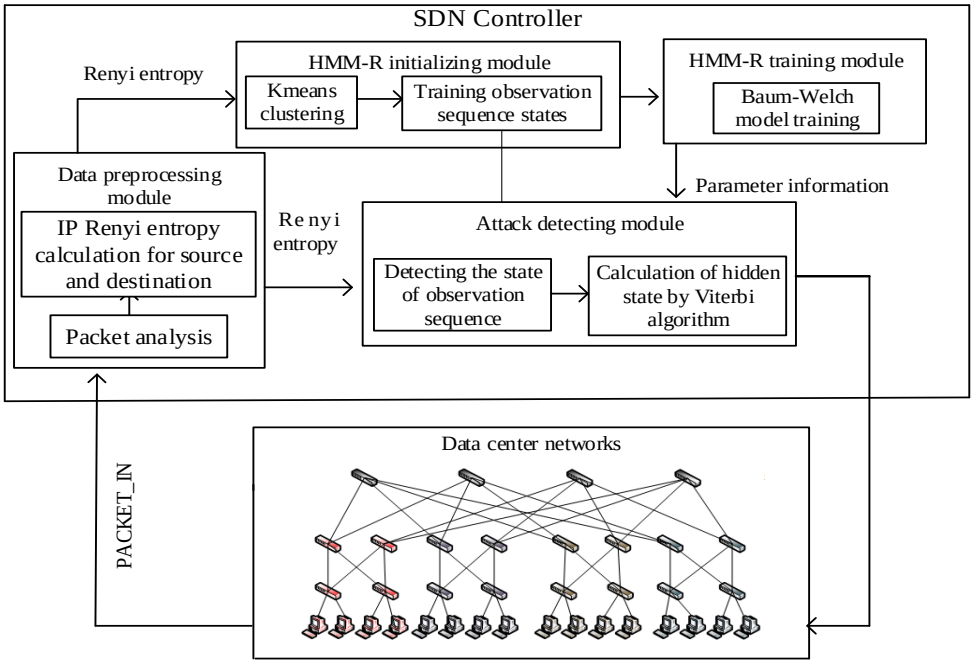

Figure 1. Architecture of HMM-R detection algorithm.

4.2. Detection Features and Performance Indices

4.2.1. Detection Features

The detection features we use are source IP and destination IP of data traffic, and their Renyi entropy is calculated to analyze their distribution changes.

The attack sources of L-DDoS attacks are dispersed, attackers use multiple attack sources to cause bad experience of servers or links. Therefore, when the attack occurs, although the attack rate is very small, the distribution of the source IP will also change, especially when the attack speed is high, the distribution will change dramatically, resulting in the increase of the Renyi entropy of source IP.

In a normal data center network, because of multi service requests, multiple servers cooperate and communicate with each other, thus the transmission of data packets between a server and multiple servers, the distribution of IP is relatively uniform. Most of the attacks are to explore the vulnerable hosts, servers, and links in the network. In order to get a better attack effect, the attacker generally chooses the same target to attack, thus the destination IP distribution is concentrated and its Renyi entropy decreases.

We can distinguish normal traffic from attack traffic by the Renyi entropy changes of source IP and destination IP.

4.2.2. Performance Indices

The performance indices are mainly divided into the true positive rate, false positive rate, and false negative rate. The true positive is the probability that the normal traffic is regarded as the normal and the abnormal flow as the abnormal; it will be convenient for defending and tracing if the flow type is to be detected correctly. The false positive refers to the probability of taking the normal traffic as the abnormal, and it may cause the normal traffic to be cleaned or defended. The false negative is the probability that the abnormal traffic is regarded as the normal, which may cause some anomalies to be missed, and result in the reduction of network performance.

$$RP = \frac{TP + TN}{TN + FN + TP + FP}, FA = \frac{FP}{TN + FP}, DR = \frac{FN}{TP + FN} \quad (6)$$

In Formula (6), RP indicates the true positive rate, FA indicates the false positive rate, and DR indicates the false negative rate, where TP means attack traffic is classified as an attack, TN indicates a normal traffic is classified as a normal, FP means a normal traffic is classified as an abnormal, and FN indicates that an attack traffic is classified as normal.

For a detection model, if its RP is higher and its FA and DR are lower, the performance of the model is better.

4.3. Data Preprocessing

After building a data center network topology environment, if there is no attack flow in the SDN switch flow table, the SDN switch will send the PACKET_IN message to the SDN controller, then the controller will send out the strategy to deal with the situation. The packets we need to collect are PACKET_IN packets, we monitor the PACKET_IN message through the OPENFLOW_PACKET_IN listener of the SDN controller module, if we monitor the PACKET_IN message, the OPENFLOW_PACKET_IN function will analyze the packet. The switch gets the input port from the packet, if the packet is Ethernet type, the source MAC and destination MAC addresses are obtained from the Ethernet data frame header; if the Ethernet data frame is IP type, continue to parse the DATA domain of the packet, and finally get the source IP and the destination IP. The data parsing process can be seen in Figure 2.

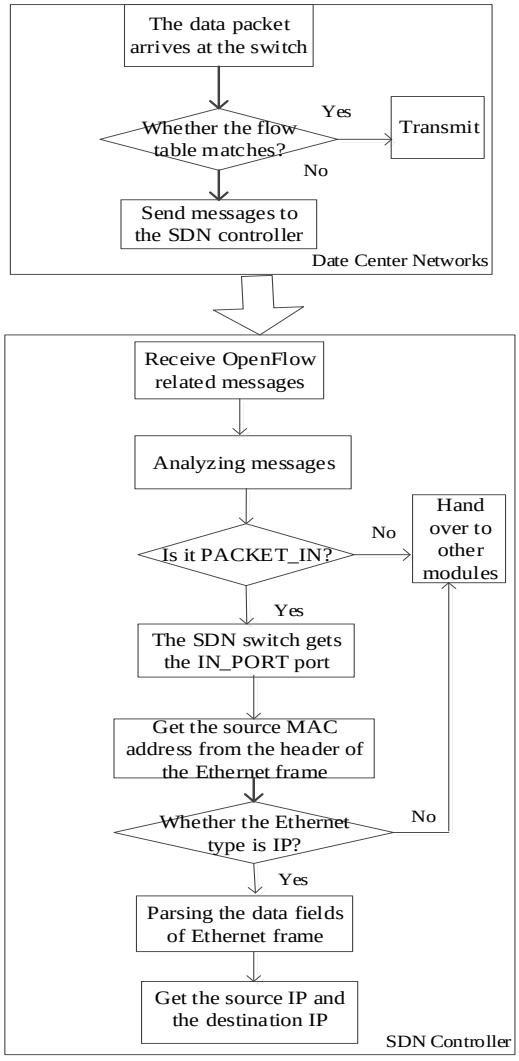

Figure 2. Packet analysis process.

4.4. Model Initializing

Before training HMM-R model, we needed to get the HMM-R hidden state space, the observed state and the initial probability matrix. The range of state space is the primary parameter for HMM-R. However, because of the large scope of Renyi entropy, the state space may be large, then we use k-means clustering to narrow the range of the state space and reduce the complexity of HMM-R. The training data observation sequence is obtained by k-means clustering and the parameters of different parameter models are produced by the observation sequences with different lengths. The setting of the initial hidden state transfer matrix is given randomly, that is to say, $\lambda^{(0)}$ is given randomly. Under different attack rates, the ranges of hidden state and the observation state and the size of the observation matrix will affect the final HMM training performance. Therefore, we choose different parameters under different attack rates and optimize them by the training algorithms.

4.5. Model Training

Model training is to learn HMM-R by Baum–Welch algorithm. The purpose is to seek the parameters of expectation maximization through data iteration and its core idea is EM. The initial parameter $\lambda^{(0)}$ and the observation data sequence are obtained in the initialization module. The training process is as shown in Figure 3. M is the total times of EM training iterations, and the times of initialization training are 0. We judge whether the times of iterations are more than M, if not, the expectation of hidden variables can be solved through the given parameters and observation data and it will get new parameters.

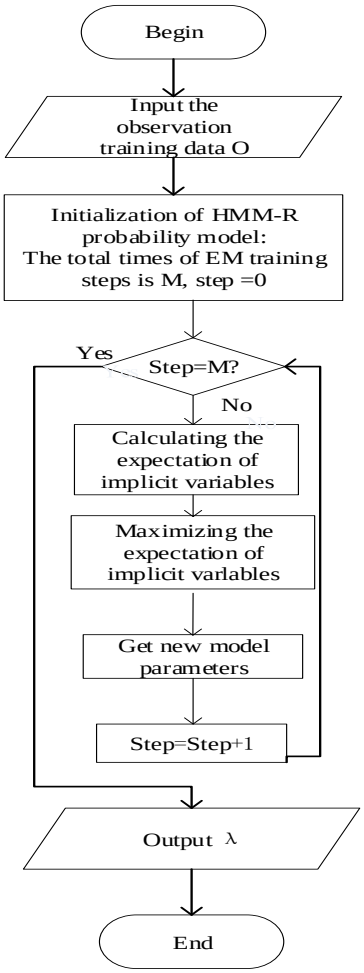

Figure 3. Baum–Welch training process.

4.6. Model Detecting

We collected the PACKET_IN packets under a window with a certain time by SDN, then preprocess these packets, and get the Renyi entropy of source and destination IP. These preprocessed data and k-means clustering data are calculated for Euclidean distance, as shown in Formula (7).

$$\begin{aligned} d(x_i) &= \frac{\sum_{j=1}^{n} \sqrt{\sum_{z=1}^{m} \left(C_{ijz} - x_{ijz}\right)^2}}{n} \\ OF(x) &= \text{argmin}_{1 \le i \le k} d(x_i) \end{aligned} \tag{7}$$

In Formula (7), C_{ijz} represents the z dimension of the j data in class i; m represents the dimension of data and m is 2; n represents the total number of data in each class; k represents the total number of k-means clustering; $OF(x)$ represents the category of a data.

We calculate the Euclidean distance for Renyi entropies of source IP and destination IP, and select the minimum Euclidean distance as the state of the detection data, that is, the category. The Viterbi algorithm is used to solve the problem of HMM-R decoding, the specific process is shown in Figure 4. We needed to obtain the five model parameters of HMM-R: the state space, the observation state space, the state transition matrix, the observation transfer matrix and the initial probability transfer matrix, to detect the prediction of observed data.

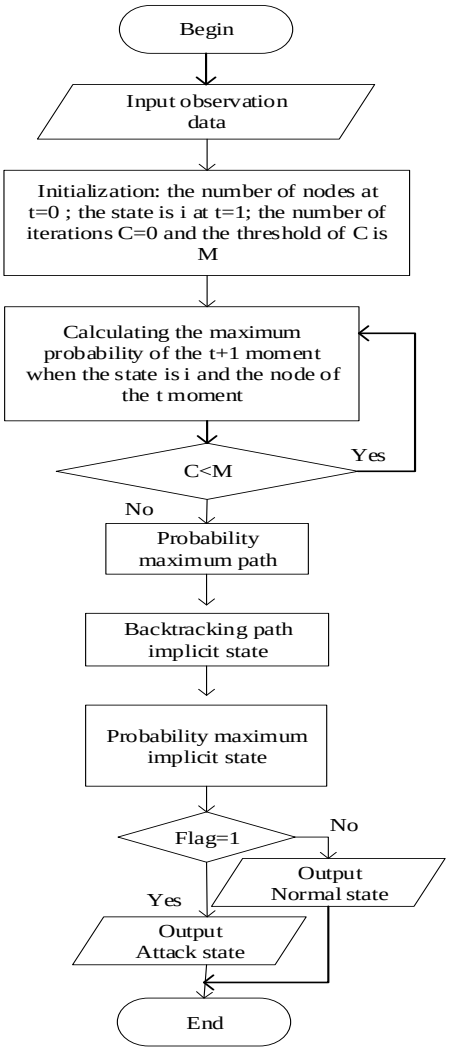

Figure 4. Viterbi algorithm process.

The HMM-R detection algorithm is proposed as Algorithm 1:

Algorithm 1 HMM-R detection algorithm

Input: λ, T, W
Output: S

$P\_C \leftarrow 0, COUNT \leftarrow 0, O \leftarrow \{\ \}, Src\_Ip \leftarrow \{\}, Dst\_Ip \leftarrow \{\}$
FOR $i \leftarrow 1$ to $|X|$
  $\delta_1(i) \leftarrow \pi_i b_i(o_1), \psi_1(i) \leftarrow 0$
IF $P\_C <= W$ THEN
  $Src\_Ip \leftarrow Src\_Ip + Packet\_In\_OpenFlow(event)$
  //Function $Packet\_In\_OpenFlow(event)$ handles $Packet\_In$ messages
  $Dst\_Ip \leftarrow Dst\_Ip + Packet\_In\_OpenFlow(event)$
  $P\_C \leftarrow P\_C + 1$
ELSE
  IF $COUNT <= T$ THEN
    $Src\_Ent, Dst\_Ent \leftarrow Entropy(Src\_Ip, Dst\_Ip)$
    $//Entropy()$ is a function to calculate Renyi entropy
    $COUNT \leftarrow COUNT + 1$
    $O \leftarrow O + (Src\_Ent, Dst\_Ent)$
  ELSE
    $O \leftarrow Compute(\lambda, O)$ $//Compute()$ is a function to calculate Euclidean distance
    $S \leftarrow Viterbi(\lambda, O)$

Table 1 shows the meaning of the variable names in Algorithm 1.

Table 1. Meaning of the variable names in the HMM-R detection algorithm.

Variable Name	Meanings
λ	Parameter of HMM-R model
T	Length of observation sequence
W	Threshold of window size
P_C	Statistics of window size
X	Hidden state space
Src_Ip	Source IP address
Dst_Ip	Destination IP address
Src_Ent	Renyi entropy of source IP
Dst_Ent	Renyi entropy of destination IP
O	Observation sequence

5. Implementation and Evaluation

This experiment uses Mininet [27] to build the data center network topology. It is a virtualized network topology simulator. The topology can be customized through a graphical interface. Each network device has a unique namespace that can be accessed in the form of a namespace. The Mininet simulator has the advantages of both a hardware test bed and an emulator. Thus, it can support any topology structure with flexibility. We use POX [28] as a SDN controller. The POX is a lightweight controller that can automatically add flow table entries through the command line. The POX controller

is a variant of NOX. Its core components are composed of OpenFlow module and Of_01 module. The controller frequently monitors the state information by registering the self-defining traffic collection function module. This experiment uses Trafgen as an attack tool in the Netsniff-ng suite as a high-speed, dynamic, and multi-threading approach [29]. We selected the destination host with the IP address 10.0.1.2 and 10.1.1.3 as attack hosts. One host sends a SYN attack and another launches an ACK attack. The attack period is 10 s for 10 times and the rate of attacks is random.

We collect packets in chronological order, data is obtained by parsing the source IP and destination IP of data packets. Data is divided into the training data and test data. The training data is adopted to train our model and the test data is used to examine our model. We compared HMM-R with KNN, SVM, SOM, and BP in forms of RP, FA, and DR. At the same time, we analyzed the detection performance under different windows and rates; and Renyi entropy is also verified by setting the order α = 1, 3, 5, 8. The size of the sliding window is the number of packets passed per second. We continuously adjusted the proportion of attack traffic. The following proportion of attack traffic is the ratio of pure attack traffic to the normal traffic.

5.1. Model Detection Performance

We use Shannon entropy (α = 1) to statistic attribute features and compare the performance of 4 common machine learning algorithms and HMM-R algorithms with the standard of RP, FA and DR when the L-DDoS attack ratio is 0.01. As can be seen from Table 2, the HMM-R algorithm increases the true positive rate by at least 2% and the false positive rate by at least 5% or even 20% compared to the other 4 machine learning algorithms. It is because the state transfer matrix and the observation probability transfer matrix in the HMM-R algorithm are in the form of probability to relate the potential association between data packets in the data center network. Since the HMM-R algorithm obtains the best model parameters at different orders, different k-means clustering results and observation sequence lengths are set. Table 3 shows the result of the k-mean clustering and the length of the observation sequences in each sequence for multi-rate attack streams at the attack ratio of 0.01. The different clustering results show that the hidden state and the state of the observed sequence are different. The classification of Renyi entropy clustering is more, indicating that it is more fine-grained in the classification. Table 4 shows the performance of HMM-R at different attack ratios. With the increase of L-DDoS attack traffic, the gap between normal traffic and attack traffic is increasing. The probability that the attack traffic is mistaken for normal traffic is getting smaller and smaller. Therefore, the false negative rate gradually decreases and the true positive rate increases. At any attack ratio, the HMM-R detection performance is superior to other algorithms in both the true positive and the false positive because HMM-R through the state transfer matrix to consider the L-DDoS attack state in a variety of rates adaptively.

Table 2. Performance of algorithms at attack ratio of 0.01.

Detection Algorithm	RP	FA	DR
KNN	0.8990	0.2530	0.0
SVM	0.8990	0.2409	0.007
SOM	0.9250	0.0636	0.0834
BP	0.9230	0.1315	0.0
HMM-R	**0.9461**	**0.01**	**0.08**

Table 3. Parameters of HMM-R at attack ratio of 0.01.

Order of Renyi Entropy	K	L
$\alpha = 1$	10	13
$\alpha = 3$	11	13
$\alpha = 5$	10	13
$\alpha = 8$	11	13

Table 4. Performance of HMM-R at different attack ratios (α = 1).

Attack Ratio	RP	FA	DR
0.01	0.9461	0.0181	0.0800
0.02	0.9692	0.0	0.0533
1	0.9711	0.0	0.0500

5.2. Sliding Window

In order to analyze the impact of the statistical window on the detection performance, we have counted the experimental data of the sliding window with 40, 60 and 100 packets under the attack ratio of 0.025. Figure 5 show the comparison of RP, FA, and DR of the detection algorithms under different windows respectively. With the increase of the sliding window, the true positive rate of each algorithm is increasing and the false positive rate is decreasing; the false negative rate of all algorithms is decreasing. It can be seen from the contrast diagram that the performance of the HMM-R algorithm is very good, but only slightly worse on the false negative. This is because as the window increases, a large amount of normal traffic is mixed in the L-DDoS traffic, making the entropy difference between the attack traffic and the normal traffic very small.

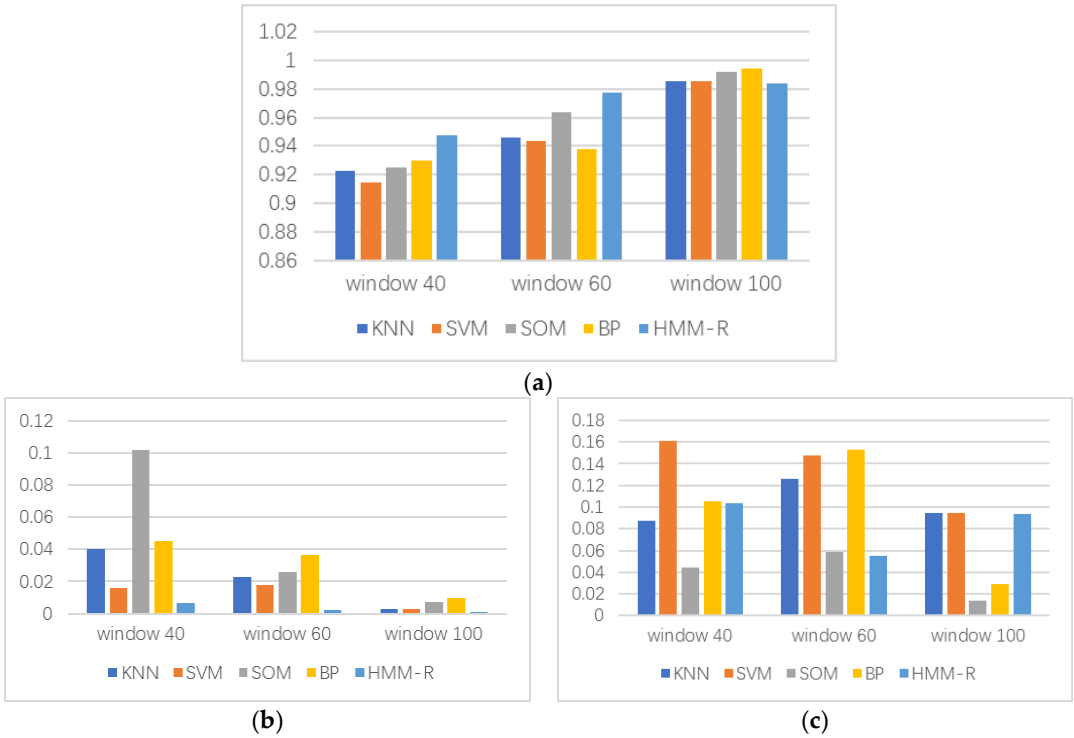

Figure 5. Comparison diagram of performance Indices of algorithms at attack ratio of 0.025. (**a**) Comparison of RP of algorithms; (**b**) comparison of FA of algorithms; (**c**) comparison of DR of algorithms.

Figure 6 show the comparison of RP, FA, and DR of each order of the HMM-R under different windows respectively. It can be seen from the graph that the best performance of HMM-R at the attack ratio of 0.025 is obtained when the sliding window is 60 and the order of statistical attributes is at α = 8. The different sizes of windows and the order of statistical attributes will lead to different performance of HMM-R. Choosing the appropriate detection window and order has a significant impact on detection performance.

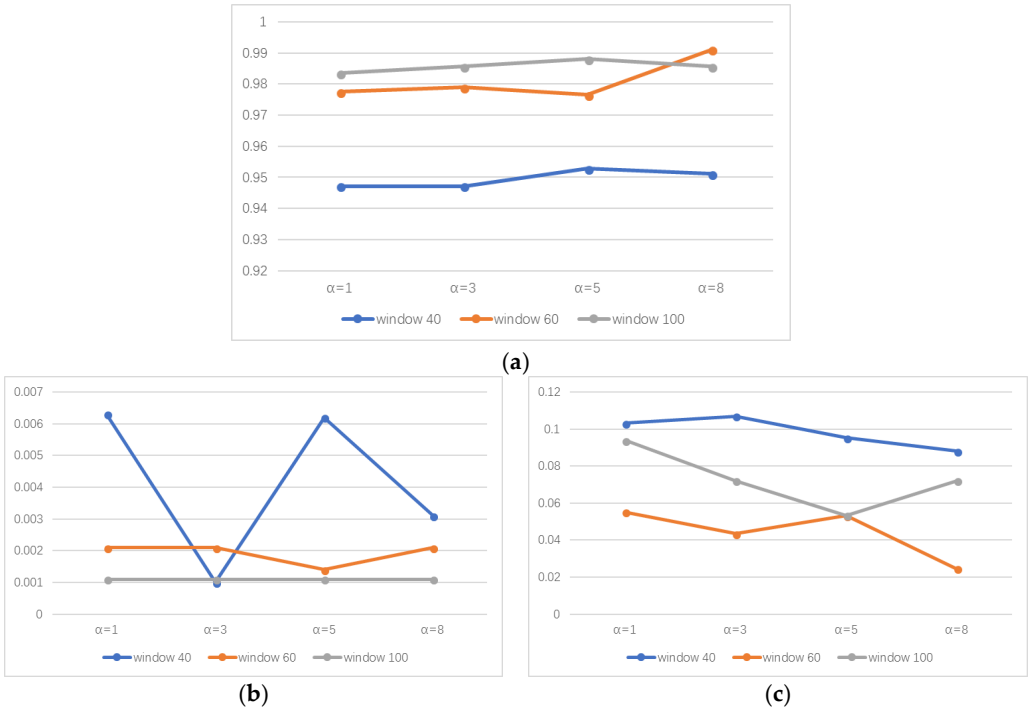

Figure 6. Comparison diagram of performance Indices of each order of HMM-R under different windows. (**a**) Comparison of RP of HMM-R; (**b**) comparison of FA of HMM-R; (**c**) comparison of DR of HMM-R.

5.3. Robustness

We have measured the experimental data with attack ratios of 0.25 and 2.5 in window 60 to analyze the effect of different rates on HMM-R detection performance. Table 5 shows the performance of algorithms. With the increase of L-DDoS attack traffic, the true positive rate of the four machine learning methods gradually increased; the false positive and the false negative gradually decreased. When the attack ratio is low, the HMM-R algorithm is superior to other algorithms in three performance indicators. There are some optimizations when the attack ratio is relatively high. It shows that with the increase of L-DDoS attack traffic, the difference between the entropy of normal and L-DDoS attack traffic gradually increases; that is, it deviates from the normal traffic behavior profile to a greater extent. It also can be seen that HMM-R is better than the other four algorithms at a lower attack ratio (equivalent to a lower rate).

Table 5. Performance of algorithms with attack ratio of 0.25 and 2.5 in window 60.

Detection Algorithm	RP	FA	DR
KNN	0.9833 0.9960	0.0147 0.0025	0.0247 0.0088
SVM	0.9814 0.9960	0.0147 0.0025	0.0247 0.0088
SOM	0.9814 0.9911	0.0147 0.0041	0.0247 0.0237
BP	0.9792 0.9903	0.0181 0.0307	0.0454 0.0153
HMM-R	**0.9847** **0.9950**	**0.0072** **0.0049**	**0.0197** **0.0052**

5.4. Time Performance

We analyze the detection time of different algorithms and HMM-R which is under different windows and different ratios. It showed that HMM-R performs better at detecting time than other algorithms, and the detection time of HMM-R is related to the size of the sliding window and the rate of attack.

When the attack ratio is 0.025, Table 6 shows the detection time of each algorithm when the size of the window is 40, of which the BP and SOM algorithms have the longest detection time, indicating that the neural network algorithm is not suitable for the data center network with low delay flow. The data in Table 7 is the detection time of HMM-R under different windows. It shows that the training time of HMM-R increases with the increase of windows. It can be concluded that the HMM-R detection time is related to the size of the sliding window, so the time complexity of the HMM-R can be reduced by adjusting the size of the window at a certain ratio. Table 8 shows the HMM-R detection time at different attack ratios. It can be seen if the attack ratio increases, the detection time will increase gradually. This is due to the increase of attack ratio, which increases the attack rate and the number of attack packets, thus increasing the number of observed sequence states, that is, the observed state matrix becomes larger and eventually leads to an increase in the time complexity. Therefore, in a certain detection window, HMM-R time complexity is also related to the attack ratio.

Table 6. Detection time of each algorithm in window 40 with attack ratio of 0.025.

Detection Algorithm	Time (s)
KNN	0.0009
SVM	0.0246
SOM	0.5576
BP	119.7
HMM-R	0.2227

Table 7. Detection time of HMM-R in attack ratio of 0.025 with different windows.

Window	Time (s)
40	0.1642
60	0.1700
100	0.2316

Table 8. Detection time for HMM-R in window 40 with different attack ratios.

Attack Ratio	Time (s)
0.025	0.1642
0.25	0.2227
2.5	0.2831

6. Conclusions

HMM-R algorithm for detecting L-DDoS attacks in the data center network is proposed in this paper. We used SDN technology to realize the intelligent control and collection of the traffic, and PACKET_IN message is used to set the detection period to reduce the detection time. Then we made the Renyi entropy as a statistical attribute to reduce the false positive in the attribute. Finally, we used HMM-R to define a variety of states to detect L-DDoS attacks at different rates in the form of probability. We also compared HMM-R with KNN, SVM, SOM, and BP. We gathered that the HMM-R can improve the true positive and reduce the false positive rate at different attack rates. It has a comprehensive detection performance, especially at a low rate, where the performance of the HMM-R algorithm increases significantly. Attack detection, defense, and traceability are three

important parts of the anomaly detection system. This paper is mainly for L-DDoS attack detection in the data center network. We will use OpenFlow flow table information of SDN to defend against attacks to different degrees in the future.

Author Contributions: W.W. and L.W. discussed and confirmed the idea; L.W. carried out the experiment and analyzed the data; X.K. collected the data and made a paper writing.

Funding: This research was funded by "The Fundamental Research Funds for the Central Universities", South-Central University for Nationalities (CZY18014).

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

References

1. Gang, D.; Gong, Z.; Hong, W. Characteristics research on modern data center network. *J. Comput. Res. Dev.* **2014**, *51*, 395–407. Available online: http://crad.ict.ac.cn/EN/Y2014/V51/I2/395 (accessed on 18 July 2018).
2. Wen, K.; Yang, J.H.; Zhang, B. Survey on research and progress of low-rate denial of service attacks. *J. Softw.* **2014**, *533*, 37. [CrossRef]
3. Min, S.K.; Lee, S.B.; Gligor, V.D. The crossfire attack. In Proceedings of the IEEE Symposium on Security & Privacy, Berkeley, CA, USA, 19–22 May 2013; pp. 127–141. [CrossRef]
4. Xiang, Y.; Li, K.; Zhou, W. Low-rate DDoS attacks detection and traceback by using new information metrics. *IEEE Trans. Inf. Forensics Secur.* **2011**, *6*, 426–437. [CrossRef]
5. Chaovalit, P.; Gangopadhyay, A.; Karabatis, G.; Chen, Z. Discrete wavelet transform-based time series analysis and mining. *ACM Comput. Surv.* **2011**, *43*, 6. [CrossRef]
6. Oshima, S.; Nakashima, T.; Sueyoshi, T. Early DoS/DDoS Detection Method using Short-term Statistics. In Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems, Krakow, Poland, 15–18 February 2010; pp. 168–173. [CrossRef]
7. Bhuyan, M.H.; Bhattacharyya, D.K.; Kalita, J.K. An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection. *Pattern Recognit. Lett.* **2015**, *51*, 1–7. [CrossRef]
8. Bhuyan, M.H.; Bhattacharyya, D.K.; Kalita, J.K. Information metrics for low-rate DDoS attack detection: A comparative evaluation. In Proceedings of the International Conference on Contemporary Computing, Noida, India, 7–9 August 2014; pp. 80–84. [CrossRef]
9. Mousavi, S.M.; Sthilaire, M. Early detection of DDoS attacks against SDN controllers. In Proceedings of the International Conference on Computing, NETWORKING and Communications, Garden Grove, CA, USA, 16–19 February 2015; Volume 17, pp. 77–81. [CrossRef]
10. Jadhav, P.N.; Patil, B.M. Low-rate DDoS attack detection using optimal objective entropy method. *Int. J. Comput. Appl.* **2014**, *78*, 33–38. [CrossRef]
11. Xiao, P.; Qu, W.; Qi, H.; Li, Z. Detecting DDoS attacks against data center with correlation analysis. *Comput. Commun.* **2015**, *67*, 66–74. [CrossRef]
12. Ain, A.; Bhuyan, M.H.; Bhattacharyya, D.K.; Kalita, J.K. Rank correlation for low-rate DDoS attack detection: An empirical evaluation. *Int. J. Netw. Secur.* **2016**, *18*, 474–480. [CrossRef]
13. Bhuyan, M.H.; Kalwar, A.; Goswami, A.; Bhattacharyya, D.K.; Kalita, J.K. Low-Rate and High-Rate Distributed DoS Attack Detection Using Partial Rank Correlation. In Proceedings of the IEEE Fifth International Conference on Communication Systems and Network Technologies, Gwalior, India, 4–6 April 2015; pp. 706–710. [CrossRef]
14. Wei, W.; Chen, F.; Xia, Y.; Jin, G. A rank correlation based detection against distributed reflection dos attacks. *IEEE Commun. Lett.* **2013**, *17*, 173–175. [CrossRef]
15. Hoque, N.; Bhattacharyya, D.K.; Kalita, J.K. A novel measure for low-rate and high-rate DDoS attack detection using multivariate data analysis. In Proceedings of the IEEE International Conference on Communication Systems and Networks, Bangalore, India, 5–10 January 2016; pp. 1–2. [CrossRef]
16. Zhang, C.; Cai, Z.; Chen, W.; Luo, X.; Yin, J. Flow level detection and filtering of low-rate DDoS. *Comput. Netw. Int. J. Comput. Telecommun. Netw.* **2012**, *56*, 3417–3431. [CrossRef]
17. Suresh, M.; Anitha, R. Evaluating machine learning algorithms for detecting DDoS attacks. In Proceedings of the Advances in Network Security and Applications: 4th International Conference, CNSA 2011, Chennai, India, 15–17 July 2011; pp. 441–452. [CrossRef]

18. Yusof, A.R.; Udzir, N.I.; Selamat, A. An evaluation on KNN-SVM algorithm for detection and prediction of DDoS attack. *Parasitology* **2011**, *138*, 578–582. [CrossRef]
19. Priyanka, P.S.; Gowrishankar, A.; Priyanka, P.S.; Gowrishankar, A. Detection of Low and High Rate DDoS Attack Using Metrics with SVM in FireCol Distributed Network. Available online: https://www.ijcaonline.org/proceedings/icaccthpa2014/.../19445-6027 (accessed on 18 July 2018).
20. Yan, Q.; Gong, Q.; Deng, F.A. Detection of DDoS attacks against wireless sdn controllers based on the fuzzy synthetic evaluation decision-making model. *Ad Hoc Sens. Wirel. Netw.* **2016**, *33*, 275–299. Available online: http://ahswn-volume-33-number-1-4-2016/ahswn-33-1-4-p-275-299/ (accessed on 18 July 2018).
21. Braga, R.; Mota, E.; Passito, A. Lightweight DDoS flooding attack detection using NOX/OpenFlow. In Proceedings of the IEEE Conference on Local Computer Networks, Denver, CO, USA, 10–14 October 2010; Volume 8, pp. 408–415. [CrossRef]
22. Giotis, K.; Argyropoulos, C.; Androulidakis, G.; Kalogeras, D.; Maglaris, V. Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments. *Comput. Netw.* **2014**, *62*, 122–136. [CrossRef]
23. Cui, Y.; Yan, L.; Li, S.; Xing, H.; Pan, W.; Zhu, J.; Zheng, X. SD-anti-DDoS: Fast and efficient DDoS defense in software-defined networks. *J. Netw. Comput. Appl.* **2016**, *68*, 65–79. [CrossRef]
24. Farhady, H.; Lee, H.Y.; Nakao, A. Software-defined networking: A survey. *Comput. Netw.* **2015**, *81*, 79–95. [CrossRef]
25. Zhang, C.K.; Cui, Y.; Tang, H.Y.; Wu, J.P. State-of-the-art survey on software-defined networking (SDN). *J. Softw.* **2015**, *26*, 62–81. [CrossRef]
26. Terrence, L. Foundations of Probability. In *Advanced Real Analysis. Cornerstones*; Birkhäuser: Boston, MA, USA, 2005; ISBN 978-0-8176-4382-9. [CrossRef]
27. Lantz, B.; Heller, B.; Mckeown, N. A network in a laptop: Rapid prototyping for software-defined networks. In Proceedings of the ACM Workshop on Hot Topics in Networks, HOTNETS 2010, Monterey, CA, USA, 20–21 October 2010; pp. 1–6.
28. POX Controller. Available online: https://github.com/pkpk8/pox (accessed on 18 July 2018).
29. Netsniff-ng Toolkit. Available online: http://www.netsniff-ng.org/ (accessed on 18 July 2018).