symmetry

MDPI

Article

# A DRDoS Detection and Defense Method Based on Deep Forest in the Big Data Environment

**Ruomeng Xu [1], Jieren Cheng [1,2,*], Fengkai Wang [3], Xiangyan Tang [1] and Jinying Xu [4]**

1 School of Information Science and Technology, Hainan University, Haikou 570228, China; 0xbbc@0xbbc.com (R.X.); tangxy36@163.com (X.T.)
2 State Key Laboratory of Marine Resource Utilization in South China Sea, Haikou 570228, China
3 Rossier School, University of Southern California, California, CA 90089, USA; fengkaiw@usc.edu
4 Zhejiang Science and Technology Information Institute, Hangzhou 310006, China; xujy@zjinfo.gov.cn
* Correspondence: cjr@hainu.edu.cn; Tel.: +86-151-0891-0688

Received: 23 November 2018; Accepted: 27 December 2018; Published: 11 January 2019


**Abstract:** Distributed Denial of Service (DDoS) has developed multiple variants, one of which is Distributed Reflective Denial of Service (DRDoS). With the increasing number of Internet of Things (IoT) devices, the threat of DRDoS attack is growing, and the damage of a DRDoS attack is more destructive than other types. The existing DDoS detection methods cannot be generalized in DRDoS early detection, which leads to heavy load or degradation of service when deployed at the final point. In this paper, we propose a DRDoS detection and defense method based on deep forest model (DDDF), and then we integrate differentiated service into defense model to filter out DRDoS attack flow. Firstly, from the statistics perspective on different stages of DRDoS attack flow in the big data environment, we extract a host-based DRDoS threat index (HDTI) from the network flows. Secondly, using the HDTI feature we build a DRDoS detection and defense model based on the deep forest, which consists of 1 extreme gradient boost (XGBoost) forest estimator, 2 random forest estimators, and 2 extra random forest estimators in each layer. Lastly, the differentiated service procedure applies the detection result from DDDF to drop the traffic identified in different stages and different detection points. Theoretical analysis and experiments show that the method we proposed can effectively identify DRDoS attack with higher detection rate and a lower false alarm rate, the defense model also shows distinguishing ability to effectively eliminate the DRDoS attack flows, and dramatically mitigate the damage of a DRDoS attack.

**Keywords:** DRDoS; deep forest; IoT; big data; differentiated service

## 1. Introduction

The service providers, security practitioners are struggling to eliminate numerous information security threats that against modern organizations in the era of big data. With the rapid development of network-based systems and lower marginal cost of learning skills about cyber-attacks, it is totally foreseeable that the number, the frequency and the magnitude of the attacks, will grow faster than ever. On 28 February 2018, GitHub's code hosting website was hit with the largest-ever DDoS attack that peaked at recorded 1.35 Tbps, and the most sustained DDoS attack lasted 297 h in the first quarter of this year, according to the report from Kaspersky lab [1]. One out of many DRDoS methods is that the attackers send request packets to many open domain name servers (DNS) or network time protocol (NTP) servers with source IP set as intended victim's IP, then those servers respond to the intended victim, which amplifies the effect of the attack. According to Cloudflare [2], attacker with a single 87 Mbps source server achieved 400 Gbps attack flow via DRDoS attack method. The attack flow is nearly 5000 times amplified with only a single source. By injecting a huge number of worthless

access requests, and the internal server or devices reflecting more packets, the DRDoS attack brings the danger of congestion effectively, and the high accessibility of conducting such attacks is achieved due to the easy-use tools and relatively low cost.

In 26 September 2016, and 21 October 2016, the network in the United States was also attacked by DRDoS, the attackers unitized tens of millions of webcams and digital video recorders (DVR), sending packets to the Internet service provider (ISP), causing over 1200 websites, including Twitter, Amazon, Reddit, and Netflix to be inaccessible for millions of Internet users. Moreover, the DRDoS attacks witnessed this year have the trend to be more dedicated and sophisticated with higher diversities, as a consequence, a swift, smart and solid cyber-attack detection mechanism is needed for security control for the increasingly vulnerable network.

The rest of the paper is organized as follows. Related work is discussed in Section 2. In Section 3, we analyze the feature of general type DRDoS attack as the base theorem for the method we proposed in Section 4, where we will introduce the algorithm of the DRDoS detection method and characterized each part of it. We will then introduce the deep forest model for classification and apply differentiated service in the defense method. And we define differentiated service as a procedure that implements a simple and scalable mechanism for treating network traffic.

Here follow our running experiments. In Section 5, we simulated the DNS DRDoS attack and then collected the network flow for evaluating the proposed method with different parameters to verify that both detection and defense methods could effectively distinguish normal network flow and DRDoS attack flow. We then conducted a real-world experiment on Memcached based DRDoS attack to test the ability when generalizing other types of DRDoS attack. Lastly, in Section 6, we conclude the advantage of the method we proposed, and discuss some further work for the enhancement of this method.

## 2. Related Works

In 1999, CERT published the first report to warn the Internet about the threat of DDoS attacks, with concrete preventive actions to mitigate the threat in their articles [3]. After a few months, the first massive DDoS attack was witnessed, and many years after that, the style of the attacks were not changed as much [4]. Since the first DDoS attack, researchers had begun to disassemble, study and analyze some DDoS attack tools, measuring their impact on the Internet [5,6]. After the pressure test and studies, a number of defensive approaches came to the world. Gradually, those efforts brought up a set of efficient, effective and reliable anti-DDoS commercial products provided as independent appliances and cloud-based services. Jieren C. et al., proposed a DDoS attack detection method using IP address feature interaction in 2009 [7]. Then a DDoS attack detection using three-state partition based on flow interaction is proposed [8]. Given previous work, a DDoS detection method based on multi-feature fusion is presented [8]. And a better method was presented based on IP flow interaction [9]. An adaptive DDoS attack detection method based on multiple-kernel learning was also proposed [10]. The multiple-kernel learning was further improved by Xinzhong Z et al., [11]. Moreover, a hyper parameter selection method was proposed by Siqi W et al., [12] for self-adaptive data shifting. And a change-point DDoS attack detection method based on half interaction anomaly degree was presented [13]. Recently, an abnormal network flow based DDoS detection method was presented [14], which showed a better performance among other existing methods. Also, Jieren C. proposed an DDoS detection method for socially aware networking [15] and a method using flow correlation degree [16]. Ruizhi Z. presented a DDoS attack security situation assessment model [17] that formed the basic evaluation solution to the attacks.

Security detection is an important tool that can strengthen the security of information and communication in networks [18–22]. The purpose of security detection is detecting the attacks with high efficiency, high reliability, and low cost, by implementing corresponding detection mechanisms. Currently, the security detection can be classified as host-based security detection and fusion-based security detection. For the host-based detection, using local information collected from the node or its

neighbors, every network node can monitor itself, and therefore the decisions are provided separately within the system; as for the fusion-based detection, the method focused on the global information, making the decisions in variable ways [23–25]. With processor centered collecting all of the data from every node, the decision can be made from fused information.

For wireless differentiated service, S. Rajeev, S. N. Sivanandam, P. Pradeep et al., had proposed a new authentication protocol "Distributed Substring Authentication Protocol (DSAP)" with a database of user authentication information at the Wireless Service Provider [26]. David Black from EMC and Paul Jones from SISCO described the interaction between differentiated services network quality-of-service (QoS) functionality and the efficiency of real-time network communication, with communication based on the real-time transport protocol (RTP) [27]. In their model, the diffserv was based on network nodes applying different forwarding treatments to packets with those IP headers were marked with different Diffserv Codepoints (DSCPs). By Vishal V. Mahale et al., a co-operative cross layer mechanism for mitigation of DDoS attack was introduced [28]. In their consideration, to enhance the overall reliability against DDoS attacks, a combination of Device-Driver Packet Filter and Remote Firewall were a solution as a cross-layer approach in their model. Łukasz Apiecionek et al., brought an overview of the quality of service method as a DDoS protection tool, proposing QoS features method attached to a DDoS attack model for development purposes [29].

However, these years, due to the proliferation of new attack strategies such as Slowloris attacks and DRDoS attacks, the interest in defending against complicated DDoS attacks has been increased. Georgios et al., developed a fair solution for DNS amplification attacks, which is a one-to-one mapping of DNS requests and responses and could achieve spoof detection/prevention [30]. Marios et al., they focused on the DNSSEC-powered amplification attack and they proposed some measures such as source validation, disable open recursion, detection of DNS amplification and DNS Response Rate Limiting for improving detection [31]. Jing Li et al., they proposed a verifiable chebyshev maps-based chaotic encryption schemes with outsourcing computations in the cloud/fog scenarios [32]. Jin Li, Xiaofeng Chen and Sherman S. M. Chow proposed multi-authority fine-grained access control with accountability approach for cloud computing [33]. A privacy-preserving naive bayesian classifier, secure against the substitution-then-comparison attack, was presented by Chong-zhi Gao, Qiong Cheng, and Pei He [34]; we could adapt the privacy-preserving technique into our method in the future. The technique presented in L-EncDB also uses privacy-preserving technique [35]. Tong Li, Jin Li, and Zheli Liu proposed a differentially private naive bayesian learning over multiple data sources [36]. We could apply this technique into our method because there are very likely multiply data sources in the big data environment. There is a GMM and CNN hybrid method proposed by Zheli Liu, Zhendong Wu, and Tong Li for short utterance speaker recognition [37]; the method can be adapted in the future model in our method. To detect malware in Android, Jin Li, Lichao Sun, and Qiben Yan presented a machine learning-based significant permission identification [38], some ideas insides their work could be adapted in the future model. Ya Li, Guangrun Wang, and Lin Nie proposed a distance metric optimization for convolutional neural network when doing the age invariant face recognition [39]; the distance metric optimization could be applied in our method. We could further use the cloud-aided techniques, as presented by Jian Shen, Ziyuan Gui, and Sai Ji [40].

In order to provide an open environment for researchers to study different types of DDoS attacks, Christian Rossow started an AmpPot Project back in 2014 [41]. The project offered a honeypot that acts as an amplifier. Rossow et al., used the data collected by those AmpPot deployments across the world to measure, analyze, and display new patterns about the victims under DRDoS attacks, and the effort had been confirmed by other independent researches [42,43]. For Internet service providers (ISP), Michael Aupetit, Yury Zhauniarovich, and Giorgos Vasiliadis et al., systematically presented a visualization tool that helps technicians with an ISP to understand the amount of resources wasted due to a DRDoS attack [44,45]. The tool they created could also simulate the efficiency and the implication of various mitigation strategies that are available to ISPs against such attacks [46].

And another finding about DRDoS attacks was that, those attacks were firstly perpetrated against Internet gamers, as a form of cheating or for financial gain [47,48].

As a succession consequence, it is clear that there is a large fraction of victims are end-users, who cannot afford the commercial anti-DRDoS services at present. Traditional methods about detecting DRDoS attacks based on information metrics. Boosted by the growing of network-based services and devices on networks, the features of big data questioned existing solutions on internet security. With the development of data-mining and machine-learning, scholars and researchers were able to analyze, detect and defense DDoS attacks by extracting features throughout network flows, enabling low margin cost assembling models with enormous data in the big data environment. However, these approaches can't evaluate or repel DRDoS attacks effectively, since the potential cost and overall time consuming is high.

According to the real DRDoS attack record on the Internet, there was a DNS DRDoS targeted Spamhaus used 30,956 open DNS servers, and the attack flow reached 300 Gbps. For comparison, the largest NTP DRDoS attack flow was 400 Gbps with 4529 NTP servers. Liu, Li and Wei et al., proposed the SF-DRDoS attack, which used the peer-to-peer network such as Kad and BT-DHT. The SF-DRDoS attack could further push the amplification as large as 2400 [49], making it one of the most destructive attacks in the net. For comparison, we conducted a more destructive DRDoS attack which utilizes Memcached service, which could achieves 50,000 times amplification. We'll detail this real-world experiment in Section 5.

## 3. Analysis of General Type DRDoS Attack

The purpose of security detection is detecting the attacks with high efficiency, high reliability and low cost, by implementing accessible, available and applicable detection mechanisms. Currently, the security detection can be classified as host-based security detection and fusion-based security detection. For the host-based detection, using local information collected from the node or its neighbors, every network node can monitor itself, and therefore the decisions can be provided separately within the system; as for the fusion-based detection, the method is focused on the global information, to make the decisions in variable ways.

Based on the theory and our consideration, this paper is going to focus on host-based detection, as every node could respond to the suspicious network flow independently and efficiently. The features our detection implementation needs are from request packets and response packets passing through each node.

The UDP amplification attack is one out of many DRDoS attack methods, it begins with a server controlled by an attacker allowing IP address spoofing. Attackers would send request packets to many open DNS or NTP servers with source IP set as the intended victim's IP. And for DNS DRDoS attack, attackers could set a very long text DNS record on that server, when the request packet reached the DNS server, DNS server will send response packets with extra-long records, which leads to the UDP amplification. For the NTP DRDoS attack, the attacker would use a command named MONLIST. When an NTP server received the request packet with this command, this server will respond up to 600 IP addresses recently accessed to. If there is such an NTP server on the attack path, the responding packet of that server will be around 206 times bigger than the request packet. Figure 1 shows a possible means of conducting a DRDoS attack.

Because the UDP protocol lacks authentication and is stateless, the attacker could easily spoof the source IP address in the request packets. With the high variety of devices in the network nowadays, attackers can launch a valid DRDoS attack using not only numerous DNS/NTP servers but also enormous Internet of Things (IoT) devices. In fact, the most destructive DRDoS attacks around these years are tending to use billions of IoT devices instead of zombie computers [50]. Especially, compared with regular terminals and computers, the IoT devices other than the form of terminals, such as civilian cameras and vending machines, are regarded as less maintained, updated, or patched, leading them vulnerable as the projector of the DRDoS attacks nowadays and in the

future. Due to the specifications and technique features, this paper will discuss DRDoS attacks on DNS protocols as the main point, whereas the methodology can also be put into other scenarios with other protocols.

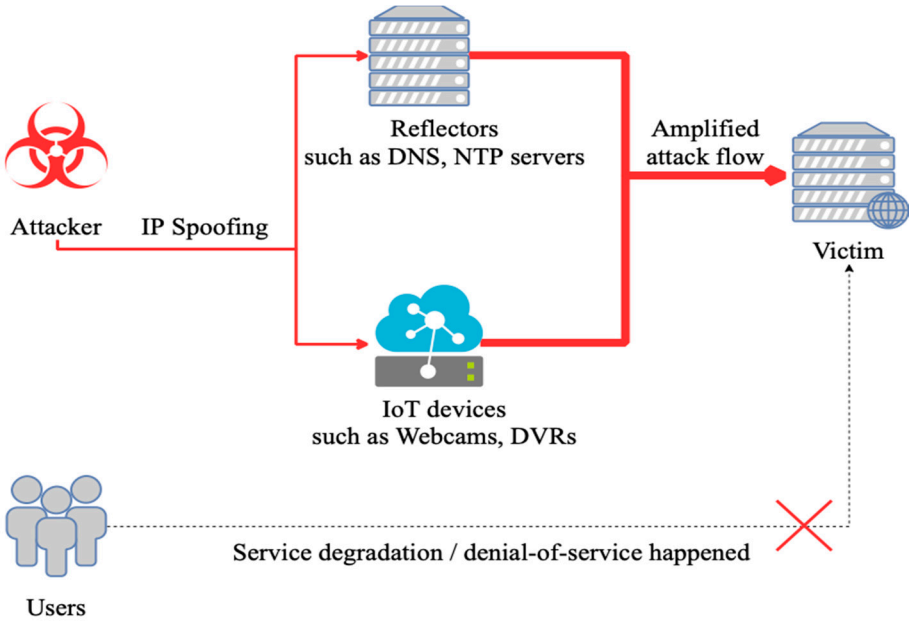


**Figure 1.** Distributed Denial of Service (DDoS) attack. DNS: domain name servers; NTP: network time protocol; DVR: digital video recorders; IoT: Internet of Things.

### *3.1. IP Layer*

At this layer, an IP packet consists of a source IP and a destination IP. One of the most commonly used means for attackers to hide themselves is IP spoofing. With this trick, attackers could fabricate any source IP in the packet, which makes it harder to trace down the real attackers. Meanwhile, attackers could control many zombie computers, webcams or DVRs to initiate a DRDoS attack. Both means would result in the changes of the distribution of the source IP, especially the former scheme.

When an attacker initiates a DRDoS attack, the router near to the attacker would be able to detect massive unique source IPs, and merely a few destination IP addresses, which shows a one-to-many relationship. Also, the many-to-one relationship can be witnessed near the victim's side. These could be used as a feature value in our proposed method, as detailed description as follows.

### *3.2. Transport Layer*

There are normally TCP packet and UDP packet in the transport layer, and in this paper, we mainly focus on UDP protocol since DNS and NTP are both designed using UDP. Other types of DRDoS attack could be easily adapted to the same method.

During the DRDoS attack, the number of request packets sent per second is unusually large near the attacker side. This leads to high bandwidth usage and the wide distribution of the source port being occupied. Correspondingly, on the way from the reflectors to the victim, the high bandwidth can still be observed, whereas the wide distribution of the destination port can be seen.

Suppose there is a successful attack and the attacker is launching a typical DRDoS attack with IP Spoofing; by the theory, we can assume a huge stream of service request packers showing a one-to-many relationship witnessed in the nodes near the attacker's side. After the attacking stream got amplified and pointed towards the victim, a blast to the number of response packets showing a many-to-one relationship can be detected, and the length of these packets could be very huge, as the bandwidth they took is directly related to the effectiveness of an attack.

## 4. The Proposed Method

To construct a valid detection and defense methods meets the needs of the big data environment, we will face two main issues. The first issue is the availability. Since the fast growing devices are attached to the network, the distribution and variation of devices are no longer dominated with regular computers such as terminals, servers or routers; instead, the nodes in a network are generally being occupied by IoT devices, such as civilian camera, vending machines, even advertisement show boards. Nowadays, those devices plugged into the net are becoming the biggest part of the nodes in a network, and these devices can be used by attackers easily with rather low cost, compared with regular methodology using zombie machines. For the method can be fit into this situation, and can be deployed on every kind of node, we should pick the features as basic as possible. The second issue we are facing is time complexity. With the high velocity, volume of data flowing in the network waiting to be processed, the detection algorithm must be efficiently enough to handle the large amount of data.

### *4.1. Extract Features at Different Layers*

Given a network flow $A$ with $n$ sample packets to be detected, we define each packet as the standard packet at the 3rd layer of the OSI reference model, i.e., they are IP packets. And normally, underneath the IP layer, the payload should be either TCP or UDP packet. And step further, here comes the application layer, for example, DNS, or NTP, as shown in Figure 2.

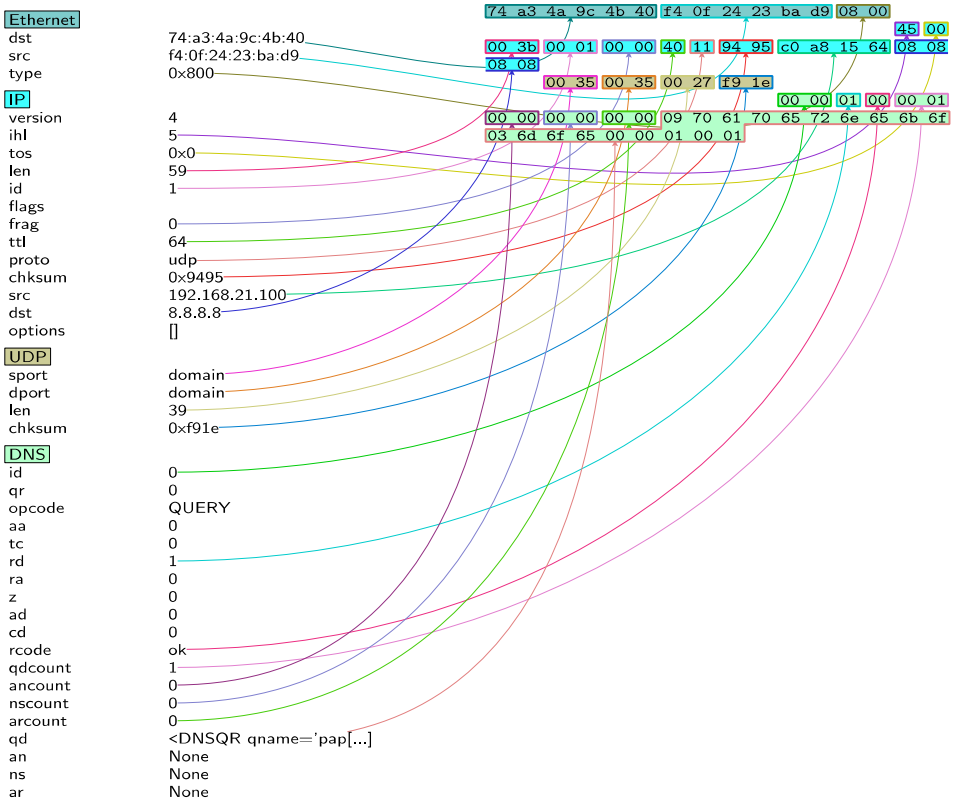


**Figure 2.** Domain name servers (DNS) request packet.

We define each IP packet as $\text{IP}_i = (S_i, D_i, T_i, P_{si}, P_{di})$, where we define $S_i$ as the source IP, $D_i$ as the destination IP, $P_{si}$ as the source port, $P_{di}$ as the destination port, and $T_i$ as the payload, i.e., application layer packet, of $IP_i$. We will take $\Delta t$ as the parameter for sampling time. And we define vulnerable service (for example, DNS, NTP, UPnP, BT-DHT et al.,) for DRDoS as *VSD*.

In each sample interval, we merge all the source IP and destination IP into one single set $M$. And we suggested extracting features for the $k$-th IP in the set $M$ by $\left(C_{qk}, V_{qk}, P_{qk}, C_{rk}, V_{rk}, P_{rk}\right)$.

$C_{qk}$ represents the amount of request packets of *VSD*s by the $k$-th source IP through the node in a fixed period of time; $V_{qk}$ is the volume per unit time of these *VSD* request packets with the $k$-th source IP, and $P_{qk}$ represents the number of unique port numbers of these packets going through from $k$-th source IP; $C_{rk}$ represents the amount of response packets to the destination IP through current node in a fixed period of time; $V_{rk}$ is the volume per unit time of these response packets with the $k$-th destination IP; also, $P_{rk}$ represents the number of unique port numbers of those packets designated to each destination IP.

These features are going to be calculated for each IP in each sample interval, as follows:

1. When attacker initiates the DRDoS attack, for some $S_i$, there would be a large number of request packets and response packets from the reflectors. Thus we count the number of request packets and response packets for each source IP and destination IP respectively. And for request packets to the *VSD*, we use a dictionary $W_q$ with its key denotes the source IP, and $W_q[S_i]$ denote the corresponding number of request packets declared from the $S_i$. Whereas, for the response packets from the *VSD*, we use $W_r$ with its key denotes the destination IP, and $W_r[D_i]$ denote the corresponding number of response packets that send to the $D_i$.

$$\forall_{(S_i,D_i,T_i,P_{si},P_{di})} \begin{cases} W_q[S_i] = W_q[S_i] + 1 & \text{if } T_i \text{ is a request packetto } VSD \\ W_r[D_i] = W_r[D_i] + 1 & \text{if } T_i \text{ is a responsepacketfrom } VSD \end{cases} \tag{1}$$

At the end of each sample interval, we calculate the amount of request and response packets respectively for each IP.

$$\forall_{M_k \in M} \begin{cases} C_{qk} = W_q[V_k], & \text{if } M_k \text{ exists in } W_q \\ C_{qk} = 0, & \text{otherwise} \end{cases} \tag{2}$$

$$\forall_{M_k \in M} \begin{cases} C_{rk} = W_r[V_k], & \text{if } M_k \text{ exists in } W_r \\ C_{rk} = 0, & \text{otherwise} \end{cases} \tag{3}$$

2. For request and response packets, we calculate the volume per unit time of these packets with the same source IP and destination IP separately. We define the length of each packet as $L_i$, and for request packets, we use a dictionary $Q_q$ with source IP as its key, the corresponding total length from that source IP as its value. Meanwhile, for the response packet, a dictionary $Q_r$ is defined with destination IP as its key.

$$\forall_{(S_i,D_i,T_i,P_{si},P_{di})} \begin{cases} Q_q[S_i] = Q_q[S_i] + L_i & \text{if } T_i \text{ is a request packetto } VSD \\ Q_r[D_i] = Q_r[D_i] + L_i & \text{if } T_i \text{ is a responsepacketfrom } VSD \end{cases} \tag{4}$$

Then we could calculate the volume per unit time for each IP in $M$ as Equations (5) and (6).

$$\forall_{M_k \in M} \begin{cases} V_{qk} = \frac{Q_q[M_k]}{\Delta t}, & \text{if } M_k \text{ exists in } Q_q \\ V_{qk} = 0, & \text{otherwise} \end{cases} \tag{5}$$

$$\forall_{M_k \in M} \begin{cases} V_{rk} = \frac{Q_r[M_k]}{\Delta t}, & \text{if } M_k \text{ exists in } Q_r \\ V_{rk} = 0, & \text{otherwise} \end{cases} \tag{6}$$

An abnormally gigantic value of $V_{qk}$ shows that there's possibly a DRDoS attack, because some *VSD* requires a larger size of request packets to gain more amplification for response flow from reflectors, thus we extract this basic feature from the request packets to *VSD*. And for $V_{rk}$, it is obviously that this is the key point of the DRDoS attack. If $V_{rk}$ is an abnormally large value, it indicates that this $M_k$ is under DRDoS attack.

3. Because each IP packet occupies one source or one destination port of a machine at a time, we are also taking the amount of ports into consideration. Likely, we use a dictionary $J_q$ with source IP as its key, the corresponding value $J_q[S_k]$ is a set which represents the unique source port from $S_k$. Meanwhile, for the response packet, a dictionary $J_r$ is defined similarly.

$$\forall_{(S_i,D_i,T_i,P_{si},P_{di})} \begin{cases} J_q[S_i] = J_q[S_i] \cup \{P_{si}\} & \text{if } T_i \text{ is a request packet to } VSD \\ J_r[D_i] = J_r[D_i] \cup \{P_{di}\} & \text{if } T_i \text{ is a response packet from } VSD \end{cases} \tag{7}$$

Then we could calculate $P_{qk}$ and $P_{rk}$ as Equations (8) and (9).

$$\forall_{M_k \in M} \begin{cases} P_{qk} = \|J_q[M_k]\|, & \text{if } M_k \text{ exists in } J_q \\ P_{qk} = 0, & \text{otherwise} \end{cases} \tag{8}$$

$$\forall_{M_k \in M} \begin{cases} P_{rk} = \|J_r[M_k]\|, & \text{if } M_k \text{ exists in } J_r \\ P_{rk} = 0, & \text{otherwise} \end{cases} \tag{9}$$

We use $P_{qk}$ and $P_{rk}$ as another two basic features in the HDTI, because when attacker initiate the DRDoS attack, and to make the DRDoS attack effective and valid, the attacker would send request packets to *VSD* as much as possible, which leads to that there are many request packets been sent the same time, and each packet requires a unique source port number, thus the $P_{qk}$ would be an abnormally large number. And based on the principles of TCP/IP, a response packet's destination port number is the same source port number of the corresponding request packet, which suggests that $P_{rk}$ would be an abnormally large number as well if $M_k$ is under DRDoS attack.

### *4.2. Analysis of the Feature Value*

We characterized each part of the proposed six-tuple feature value with real-world observation to explain why it is effective for both detection and defense.

As for a destructive DRDoS attack, when the attacker launched the initial packets to the reflectors, an obvious growth in $C_q$ can be witnessed; also, as for the request of the amount for reflectors, in a fixed sampling time, the amount of occupied ports on victim's source IP can be expanded fast. As for the volume per unit time $V_q$, the growth of the amount may not explode rapidly, but the anomaly can be still visible than normal dataflow.

For the path after reflectors to the victim, a blast of $C_r$ and $B_r$ can be seen. Because of the request packets, the response packets' destination ports of the intended victim would show a uniformed distribution. Alongside with the attack path, the attack flow should be accumulated, which leads to a rapid growth in each component. Based on the phenomena during different stages of a DRDoS attack, the validity on the feature components can be discussed into three situations.

1. Attack Source. A relatively abnormal growth among $C_q$, $V_q$ and $P_q$ can be observed. By applying the features to the deep forest's classifier, we would be able to detect the upstream of the attack flow. With the result from the classifier, we can drop those upstream packets before they can reach to the reflectors using differentiated service, in case of reducing the number of the abnormal packets to reflectors.
2. Intended Victim. An abnormally enormous value among $C_r$, $V_r$ and $P_r$ can be observed. Moreover, the closer to the intended victim, the larger these components extracted in the nodes are, as the attack flow clustering from reflectors to the intended victim. As an answer to this situation, the detection mechanism using random forest deployed on the intended victim's side could alert and activate defense moves by eliminating the downstream of the attack flow towards the intended victim.

3. Internal Nodes in the Internet. The nodes in the internet can obtain both upstream from attack flow and send downstream attack flow, which means that both streams can be observed and extracted. We are calling the flow with these features mentioned as mixed upstream and downstream (MUD). When attack flow lies in the MUD, we can still recognize the threats by classifying this with normal flow with random forest, and initiate differentiated service to drop the attack packets, so that the attack flow could be reduced, and the network load could be relieved.

Given the consideration and assumptions above, we can classify the feature proposed into 4 classes illustrated in Figure 3. We defined 0 as a relatively low value, and 1 stands for a relatively large value in the corresponding position in the 6-tuple feature $\left(C_{qk}, V_{qk}, P_{qk}, C_{rk}, V_{rk}, P_{rk}\right)$.

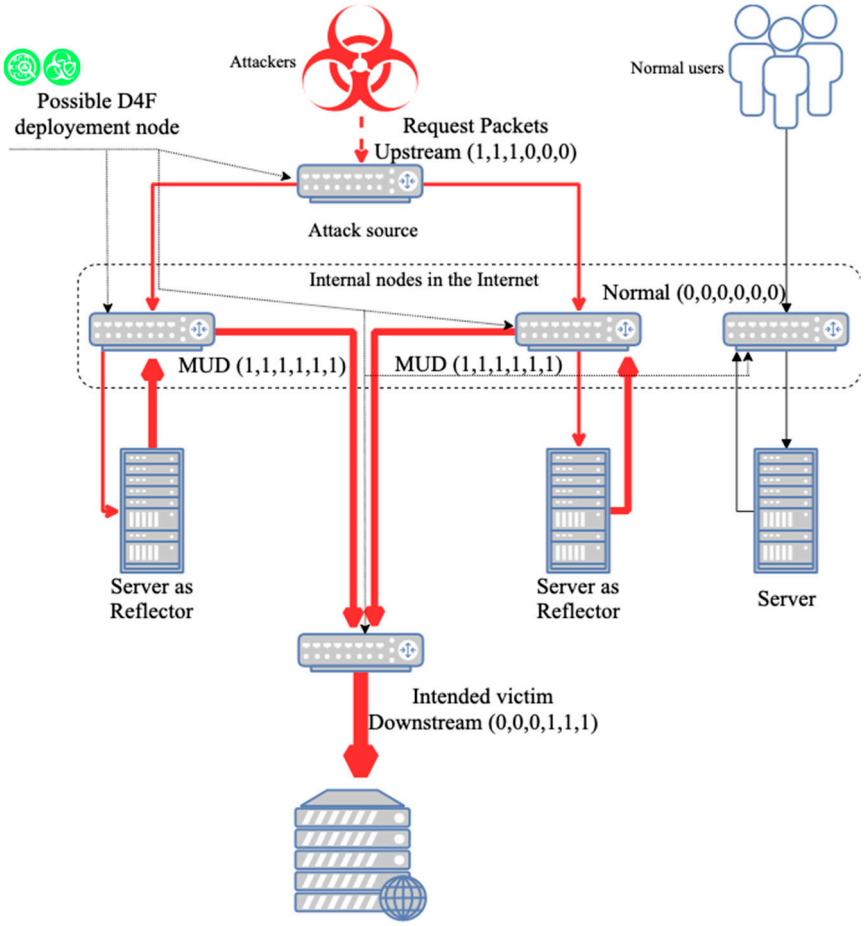


**Figure 3.** Description of 4 classes features.

With the definitions above, the status for any node in a network where under potential threat of a DRDoS attack could be revealed by our proposed detection method, and an efficient defense method could be deployed upon any node in the Internet.

### *4.3. Deep Forest Based DRDoS Detection and Defense Method*

With features gathered from the network flow based on our proposed method above, a valid deep forest model can be trained by HDTI in order to determine if a certain IP was under a DRDoS attack. If the model classified the IP is under threat, i.e., downstream, upstream or MUD type, the differentiated service procedure will be introduced and activated to achieve the elimination of DRDoS attack flow in early, middle and post stages.

Detection Model. To implement this, firstly, we gathered the 6-tuple feature, HDTI, by online sampling from normal network flow and DRDoS simulation. The normal network flow contains the

packets of *VSD*. And because there is no public available dataset of DRDoS attack, we simulated DRDoS attack. Then we take 30 s of normal network and 30 s of DRDoS attack to form a 60-s training set for the deep forest modeling. The model of our deep forest contains 5 estimators, including an XGBoost classifier, 2 random forest classifiers and 2 completely-random tree forest classifiers, as shown in Figure 4.

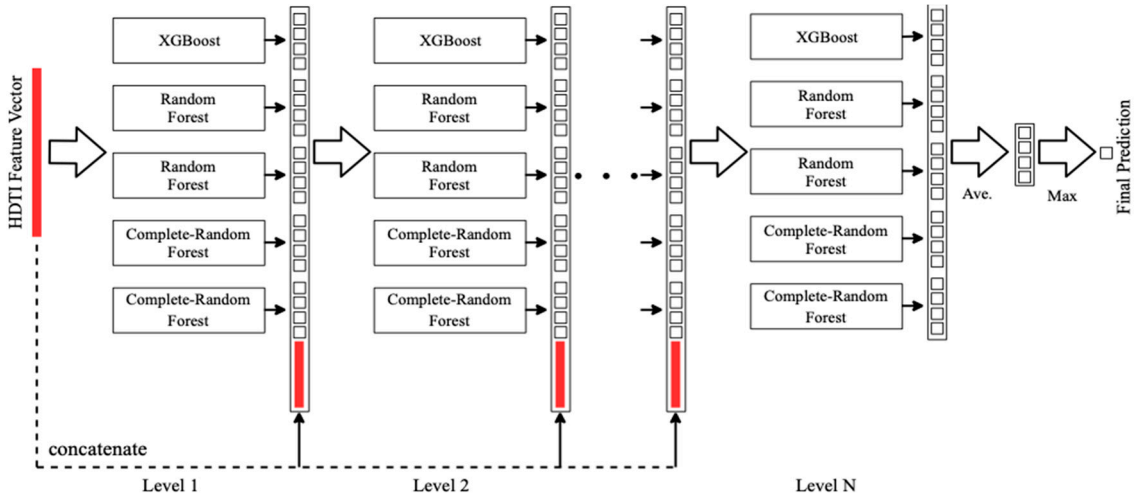


**Figure 4.** Deep DRDoS detection model based on host-based DRDoS threat index (HDTI).

The XGBoost classifier could be described as follows. The basic component of a boosted tree is regression tree, or classification and regression trees (CART). A CART will assign attributes to each leaf, and there is a real value score associated with that leaf. However, we can't make effective prediction only using CART, thus a stronger model named tree ensemble was proposed, and the tree ensemble model could be written as Equation (10).

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F} \tag{10}$$

where the $f_k$ belongs to the function space $\mathcal{F}$, and $\mathcal{F}$ is the set of all regression trees. And we can write the object function as Equation (11).

$$Obj(\Theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{11}$$

As for the additive training of the XGBoost tree, we will choose a function $f_t()$ to minimize the value of object function $Obj$.

$$Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + constant \tag{12}$$

And the $\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$, where $T$ is the number of leaves, and $w_j$ is the weight of $j$-th leaf. Then we could regroup the objective by each leaf. The result is the sum of $T$ independent quadratic functions.

$$Obj^{(t)} \simeq \sum_{j=1}^{T} \left[ \left( \sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left( \sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T \tag{13}$$

Assume the structure $q$ of the tree is fixed, then we could solve the best $w_j^* = -\frac{G_j}{H_j + \lambda}$, and the corresponding maximum gain of the objective, $Obj = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T$.

And XGBoost tree defines the gain as Equation (14). Essentially, it's the score of left child plus the score of right child then minus the score if we do not split, and finally, minus the complexity cost by introducing additional leaf. Now, we could do a left to right linear scan on the sorted instance, and we can obtain the best split along the feature.

$$Gain = \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} - \gamma \tag{14}$$

As for the random forest, firstly, it also utilizes CART as the weak learner. Secondly, it optimized the basic decision tree. It randomly selects a sub part of the features on the node, the number of randomly selected feature obeys $n_{sub} < n$. Then it decides the best split given the $n_{sub}$ features.

The input of a random forest is $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$, and $T$ represents the iteration rounds of weak classifier. The output of a random forest is a strong classifier $f(x)$. For any $t = 1, 2, \cdots, T$, the algorithm of random forest will

1. Sample the train set m times, obtaining the sampling set consists of m sample.
2. Train the $t$-th model of decision tree $G_t(x)$ with randomly selecting $n_{sub}$ features.
3. The most voted class of T weak learners will be selected as the final prediction.

As for the extra trees, it's a variant of random forest, there're only 2 minor differs between them. Firstly, the random forest utilizes bootstrap for sampling the train set, whereas the extra trees use the original input as the train set. Secondly, after randomly selected $n_{sub}$ features, the random forest will decide the best split based on information gain, Gini, or mean square error. However, the extra trees is way more radical, it randomly select a value for splitting the features. Although the randomly selected value will cause the increment of the tree, the ability of generalization is enhanced in extra trees.

The input feature value HDTI is a 6-dimensional feature value. And when feed into the first layer, each estimator outputs the initial classification result, which is a 4-dimensional vector. The results in first layer would produce a 20-dimensional feature value, then we concatenate the 20-dimensional feature value and the 6-dimensional input feature value to form a 26-dimensional augmented feature value. And the 26-dimensional augmented feature value will be used as the input feature value for the second layer, similar training procedure will be processed until there is no significant performance gain. Therefore the number of the layer could be automatically chosen, which enhances the adaptivity of the model, making it applicable to different scales of data and deploy at any node in the network in big data environment.

Defense Model. Within the trained deep forest model, we could identify the type for each IP in the network flow which needs detection, and make corresponding process against different attack packets by differentiated service. The basic idea can be describe as this: If an IP address was identified as normal, then we let all corresponding packets go through. If the IP address was identified as upstream, then we will filter abnormal vulnerable service request packets with source IP declared from that one, which achieves early stage DRDoS attack elimination. When the IP address was classified as downstream, we could filter all related abnormal vulnerable service response packets sending to that IP for post stage DRDoS attack elimination. Whereas the IP address was identified as MUD, then we filter both abnormal request and response packets of the corresponding vulnerable service for that IP.

In the actual setting up of the defense method, to make the differentiated service wise enough, in other words, demolishing attacks where letting normal network flow, a set of applicable thresholds $H$ should be applied. We define both *VSD* request and response packet that exceeded the thresholds $H$ as abnormal packets. And if an IP was classified as MUD, then we will tag it as both upstream and downstream. The differentiated service will drop an abnormal packet when the following condition meets.

1. If the source IP of an abnormal *VSD* request packet was identified as upstream, the differentiated service drops it.

2. If an abnormal *VSD* response packet with its destination IP identified as downstream, it would be also getting filtered.

To get a set of applicable thresholds $H$, we could learn from the normal and legitimated corresponding *VSD* request and response packets separately with the statistics method applied. And in real-world, experts could change their experience into empirically rules for identifying whether a packet is abnormal of not.

In this paper, the packet length is used as one of the rules in the threshold set $H$. We learnt from the dataset and calculated the observed max and min length of legitimated *VSD* request and response packets respectively. We use $G_q, L_q$ as the max and min length of a legitimated *VSD* request packets, and correspondingly, $G_r, L_r$ denote the max and min length of a legitimated *VSD* response packets. We calculate the upper bound of the request and response packet length as $U_q = G_q + \left(G_q - L_q\right)$ and $U_r = G_r + (G_r - L_r)$. Then we could define the rules for *VSD* request and response packets as below.

1. Whether the *VSD* request packet length $L_i$ exceeds $U_q$ or not.

$$H_1 : (T_i \text{ is a } VSD \text{ request packet}) \wedge (L_i > U_q) \tag{15}$$

And for the demonstration of empirically rules, we also add one rule for *VSD* response packet. This rule is used for avoiding the fluctuation of the normal network flow.

2. We use a dictionary $Z$ to store the total filtered length of destination IP $M_k$. If the *VSD* response packet length $L_i$ exceeds $U_r$, then check whether the total transferred length $Z[M_k]$ of the corresponding IP exceeds $5U_r$.

$$H_2 : (T_i \text{ is a } VSD \text{ response packet}) \wedge (L_i > U_r) \wedge (Z[M_k] > 5U_r) \tag{16}$$

With the rule set $H$ defined, we could transform the rules in $H$ into the conjunctive normal form (CNF). A CNF is the conjecture of many disjunctive expressions. We define the following atomic propositions.

$$\begin{gathered} P_1 : T_i \text{ is a } VSD \text{ request packet} \\ P_2 : T_i \text{ is a } VSD \text{ response packet} \\ P_3 : L_i > U_q \\ P_4 : L_i > U_r \\ P_5 : Z[M_k] > 5U_r \end{gathered} \tag{17}$$

Thus the final CNF of the defined rule set $H$ could be represented as Equation (18)

$$A : ((P_1 \wedge P_3) \vee (P_2 \wedge P_4)) \wedge (P_1 \vee P_5) \tag{18}$$

Therefore the formalization of the defense method can be described as Equation (19).

$$\forall_{VSD\text{packets}},\ ((S_i \in \text{Upstream } IPs) \vee (D_i \in \text{Downstream } IPs)) \wedge A \rightarrow \text{drop} \tag{19}$$

The procedure of the deep forest based detection and defense method is shown in pseudo code in Algorithm 1.

**Algorithm 1.** Deep Forest based DRDoS Detection and Defense

**Input:** Training network flow $A$, network flow $V$ to be detected, rule set $H$
Extract HDTi features from $A$ with Equations (2), (4) and (6)
Training deep DRDoS detection and defense forest model with extracted HDTI features
$A \leftarrow$ CNF of $H$
$M \leftarrow \{\}$
**for** each sampling **do**
  **for** each VSD packet $T_i$ **do**
    **if** $T_i$ is a request packet **then**
      **if** $D_i \in$ Upstream IP Set **then**
        **if** proposition $A$ is true for $T_i$ **then**
          drop this packet
        **end if**
      **end if**
      $M \leftarrow M \cup \{S_i\}$
      **end if**
      **if** $T_i$ is a response packet **then**
        **if** $D_i \in$ Downstream IP Set **then**
          **if** proposition $A$ is true for $T_i$ **then**
            drop this packet
            $Z[D_i] = Z[D_i] + L_i$
          **end if**
        **end if**
        $M \leftarrow M \cup \{D_i\}$
      **end if**
    **end for**
    **for** each $M_k \in M$ **do**
      calculate HDTI feature $(C_q, V_q, P_q, C_r, V_r, P_r)$ for $M_k$
      identify the type of $M_k$ using the deep DRDoS detection and defense forest model
      **if** the type of $M_k$ is normal **then**
        do nothing
      **else**
        **if** the type of $M_k$ is Upstream **then**
          add $M_k$ to Upstream IP Set
        **else**
          **if** the type of $M_k$ is Downstream **then**
            add $M_k$ to Downstream IP Set
          **else**
            add $M_k$ to both Upstream and Downstream IP Set
          **end if**
        **end if**
      **end if**
    **end for**
  **end for**
  **return**

### *4.4. Dataset and Assessment Criterions*

Our experiment was based on the WRCCDC 2018 [51], which contains a small company with more than 50 users, 7 to 10 servers, and common Internet services such as a web server, a mail server, and an e-commerce site. We trained our model with different $\Delta t = [0.01, 0.1, 0.5, 1.0]$ seconds, and the proposed method successfully distinguished normal flow and DRDoS flow as shown in Figure 3. Besides the basic DRDoS detection, the differentiated service effectively decrease the AF in the victim side.

To assess the proposed algorithm, we define serval criterions, we firstly define $TN$ as the number of correctly identified DRDoS network flow samples, and $FN$, correspondingly, the number of samples which is normal, but incorrectly flagged DRDoS network flow ones. We let $TP$ denote the amount of

correctly identified normal users, and *FP* denotes the amount of samples which mistakenly flagged as normal network flow, but should be actually DDoS network flow samples.

1. Detection Rate, *DR*. This value denotes the probability of the classifier identifies actual DDoS attack flow. *DR* is calculated as the number of true negative samples divides the sum of both true negative and false negative samples.

$$DR = \frac{TN}{TN + FN} \tag{20}$$

2. Missing Rate, *MR*. This value represents the probability of the classifier fails to identify actual DDoS attack flow. *MR* is calculated as the number of false negative samples divides the sum of both true negative and false negative samples.

$$MR = \frac{FN}{TN \ + \ FN} \tag{21}$$

3. False Alarm Rate, *FAR*. False alarm rate suggests that the probability of normal users are mistakenly flagged as attackers by the classifier. Correspondingly, it calculates as the number of false positive samples divides the sum of both true positive and false positive samples.

$$FAR = \frac{FP}{TP + FP} \tag{22}$$

## 5. Experiment

Inside the nearly 1TB size WRCCDC 2018 dataset, the typical overall, including all unrelated packets, bandwidth is around 166 Mbps. We will mainly conduct following experiment to validate the method we proposed with DNS DRDoS.

1. Directly inject response packets into the network to validate the differentiated service in the victim side. The bandwidth of our injected response packets is configured as $b = (100, 200, 500, 1000)$ Mbps.
2. Inject attacker's request packets to validate the detection method and differentiate service near the attacker side. If there is any DRDoS request packet passed through our detect method, corresponding response packet will be sent to the victim side. And the bandwidth of our injected request packets is configured as $b = (1, 10, 20, 50, 100)$ Mbps.
3. Inject both request packets and response packets into the network to validate the proposed method when deployed at any node in the network. The bandwidth of our injected response packets is configured as $b_r = (100, 200, 500, 1000)$ Mbps, and for the request packets, $b_s = (1, 10, 20, 50, 100)$ Mbps.

And we take another 30 s of normal network flow and 30 s of DRDoS attack flow, where the first 10 s contains only request packets, i.e., upstream, the following 10 s is consisted of response packets, i.e., downstream, and the final 10 s is mixed with both upstream and downstream. Then we compare the method we proposed to support vector machine (SVM), k nearest neighbor (kNN) and pure random forest approaches with the same dataset and experiment settings.

### *5.1. Experiment Result*

The result of our experiment consists of two parts, the first part is DRDoS attack detection between these methods, and the second part demonstrates the DRDoS attack elimination rate of our proposed method. We will discuss the experimental results at the end of each part.

DRDoS Attack Detection Comparison. We conduct various experiments and the representative results are gathered and illustrated in Figures 5–7 for comparison. The rest of the results are shown in Tables 1–6.

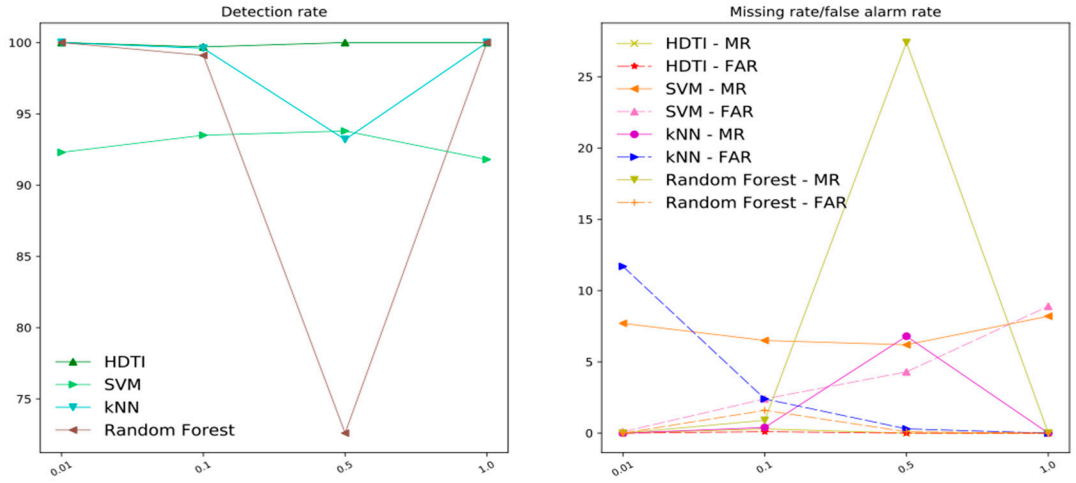


**Figure 5.** Experiment result of $b_r = 1, b_q = 100$. SVM: support vector machine; kNN: k nearest neighbor.

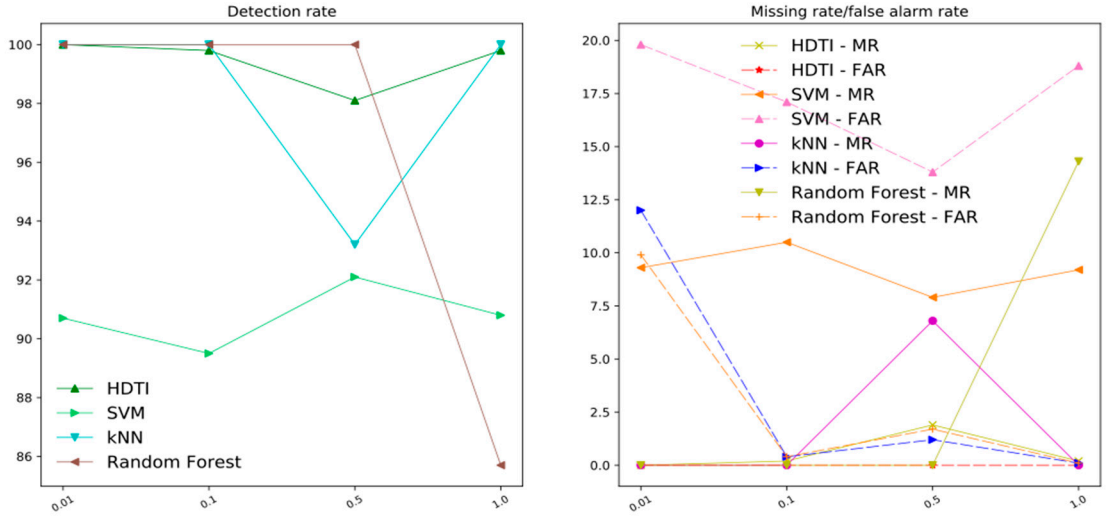


**Figure 6.** Experiment result of $b_r = 20, b_q = 500$.

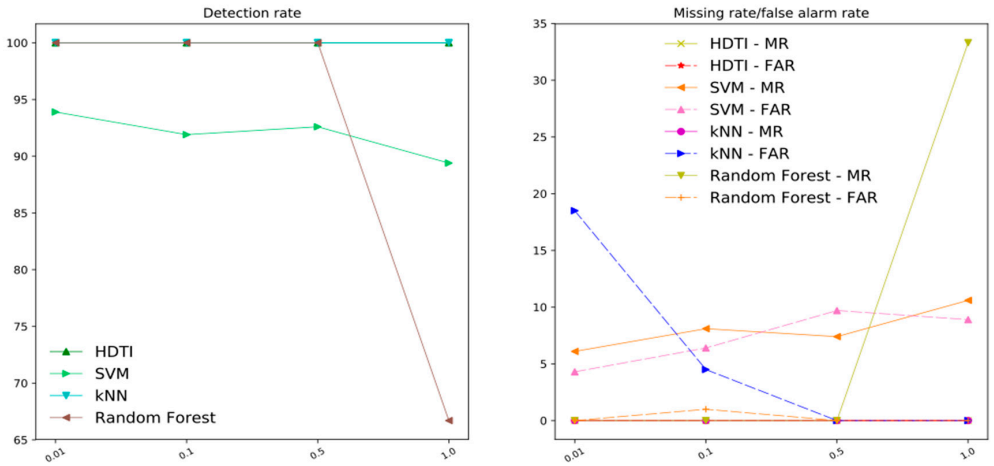


**Figure 7.** Experiment result of $b_r = 100, b_q = 100$.

**Table 1.** Crossed Comparison with DRDoS Detection Methods of $b_r = 1, b_q = 1000$.

| Method | Detection Rate/Missing Rate/False Alarm Rate | | | |
|---|---|---|---|---|
| | $\Delta t = 0.01$ | $\Delta t = 0.1$ | $\Delta t = 0.5$ | $\Delta t = 1.0$ |
| HDTI | 100.0/0.0/0.0 | 99.2/0.8/0.0 | 97.3/2.7/0.0 | 99.5/0.5/0.0 |
| SVM | 94.4/5.6/10.4 | 92.3/7.6/12.6 | 93.7/6.3/15.6 | 94.3/5.6/10.3 |
| Random Forest | 100.0/0.0/9.2 | 100.0/0.0/3.2 | 23.1/76.9/2.4 | 27.3/72.7/0.4 |
| kNN | 99.9/0.1/10.5 | 100.0/0.1/2.7 | 75.0/25.0/3.0 | 100.0/0.1/0.5 |

**Table 2.** Crossed Comparison with DRDoS Detection Methods of $b_r = 10, b_q = 1000$.

| Method | Detection Rate/Missing Rate/False Alarm Rate | | | |
|---|---|---|---|---|
| | $\Delta t = 0.01$ | $\Delta t = 0.1$ | $\Delta t = 0.5$ | $\Delta t = 1.0$ |
| HDTI | 100.0/0.0/0.0 | 100.0/0.0/0.0 | 99.8/0.2/0.0 | 100.0/0.0/0.0 |
| SVM | 93.4/6.5/10.4 | 86.9/13.1/19.4 | 88.4/11.6/23.1 | 91.4/9.5/17.1 |
| Random Forest | 100.0/0.0/12.0 | 100.0/0.0/0.8 | 100.0/0.0/3.0 | 100.0/0.0/0.7 |
| kNN | 100.0/0.0/15.0 | 100.0/0.0/1.2 | 100.0/0.0/0.6 | 100.0/0.0/0.3 |

**Table 3.** Crossed Comparison with DRDoS Detection Methods of $b_r = 10, b_q = 500$.

| Method | Detection Rate/Missing Rate/False Alarm Rate | | | |
|---|---|---|---|---|
| | $\Delta t = 0.01$ | $\Delta t = 0.1$ | $\Delta t = 0.5$ | $\Delta t = 1.0$ |
| HDTI | 100.0/0.0/0.0 | 99.4/0.5/0.0 | 98.8/0.1/0.0 | 100.0/0.0/0.0 |
| SVM | 93.8/6.1/15.2 | 91.2/8.8/14.7 | 89.1/10.8/10.2 | 90.5/9.5/7.2 |
| Random Forest | 100.0/0.0/13.2 | 75.0/25.0/0.2 | 100.0/0.0/1.7 | 100.0/0.0/0.2 |
| kNN | 100.0/0.0/15.3 | 100.0/0.0/0.5 | 100.0/0.0/2.0 | 100.0/0.0/0.1 |

**Table 4.** Crossed Comparison with DRDoS Detection Methods of $b_r = 20, b_q = 200$.

| Method | Detection Rate/Missing Rate/False Alarm Rate | | | |
|---|---|---|---|---|
| | $\Delta t = 0.01$ | $\Delta t = 0.1$ | $\Delta t = 0.5$ | $\Delta t = 1.0$ |
| HDTI | 100.0/0.0/0.0 | 100.0/0.0/0.0 | 99.3/0.7/0.0 | 99.8/0.2/0.0 |
| SVM | 91.2/8.7/20.3 | 93.7/6.2/12.7 | 93.0/7.0/17.9 | 92.6/7.4/16.4 |
| Random Forest | 100.0/0.0/9.1 | 100.0/0.0/0.0 | 100.0/0.0/0.5 | 100.0/0.0/0.3 |
| kNN | 100.0/0.0/13.0 | 100.0/0.0/0.0 | 100.0/0.0/0.7 | 100.0/0.0/0.4 |

**Table 5.** Crossed Comparison with DRDoS Detection Methods of $b_r = 50, b_q = 200$.

| Method | Detection Rate/Missing Rate/False Alarm Rate | | | |
|---|---|---|---|---|
| | $\Delta t = 0.01$ | $\Delta t = 0.1$ | $\Delta t = 0.5$ | $\Delta t = 1.0$ |
| HDTI | 100.0/0.0/0.0 | 100.0/0.0/0.1 | 99.4/0.6/0.0 | 100.0/0.0/0.0 |
| SVM | 90.5/9.5/18.2 | 92.5/7.5/14.2 | 88.1/11.9/10.7 | 89.4/10.6/9.7 |
| Random Forest | 100.0/0.0/1.8 | 23.5/76.4/0.0 | 100.0/0.0/1.1 | 100.0/0.0/0.1 |
| kNN | 100.0/0.0/1.6 | 100.0/0.0/0.2 | 100.0/0.0/0.6 | 100.0/0.0/0.3 |

**Table 6.** Crossed Comparison with DRDoS Detection Methods of $b_r = 50, b_q = 100$.

| Method | Detection Rate/Missing Rate/False Alarm Rate | | | |
|---|---|---|---|---|
| | $\Delta t = 0.01$ | $\Delta t = 0.1$ | $\Delta t = 0.5$ | $\Delta t = 1.0$ |
| HDTI | 100.0/0.0/0.0 | 100.0/0.0/0.0 | 100.0/0.0/0.0 | 100.0/0.0/0.0 |
| SVM | 95.9/4.1/3.1 | 94.1/3.9/1.5 | 92.5/7.5/4.2 | 93.1/6.9/5.1 |
| Random Forest | 100.0/0.0/0.0 | 14.3/85.7/0.0 | 100.0/0.0/0.0 | 100.0/0.0/0.0 |
| kNN | 100.0/0.0/1.8 | 100.0/0.0/0.8 | 100.0/0.0/0.1 | 100.0/0.0/0.0 |

Among these experiment results, we could see that when given a short $\Delta t$, all the methods shows a great detection rate. However, the missing rate of SVM approach is relatively high compared to HDTI and kNN means. And for HDTI and kNN, when given a fixed $\Delta t$, the larger the upstream and downstream are, the better the detection rate is. But the false alarm rate of kNN are also getting

larger and larger. When take this into real-world consideration, a higher false alarm rate is more likely to cause the service interruption for legitimated users. Although the detection rate of kNN approach reached 100 almost in all experiments, and it could indeed identify the DRDoS attack flow, the high false alarm rate would also cause service degradation or interruption. The HDTI approach we proposed shows relatively higher detection rate and lower false alarm rate, which could detect near all DRDoS attack flow without causing service interruption for legitimated users.

As the information shown above, we finished cross comparison with traditional SVM and kNN classification algorithms to test the efficiency and validity of the three methods under differentiated conditions. It is obvious that the detection method we proposed can be adopted to different situations better than SVM and kNN classifier.

We noticed the relatively high value among false alarm rate and missing rate by SVM method, whereas the low accuracy on detecting real threats, this implied the existed disadvantages of the SVM algorithms: the SVM classifier is very sensitive to the missing values and kernel function, which may lead to errors in the actual use. When we are using the database from the real world, the SVM classifier would malfunctioning sometimes due to the missing of a value, which is quite common in big data environment. Besides, the traditional SVM didn't provide neither multi-classifier nor solution to the non-liner problems, this also contributed to the poor performance of the SVM classifier in our experiment.

Although upcoming theories and implementations based on rough set provided the combinative SVM for multi-classifying, the time cost of training such classifiers goes up. Besides, the vector processed in an SVM depends on the size of samples, the memory and time a SVM take towards a large training set could be explode. In short, the traditional SVM classifiers may not meet the needs of host-based DRDoS attacks in big data environment.

The variety, velocity and volume in terms of the big data are all shown in the DRDoS attack. There are multiple vulnerable services that can be used to initiate a DRDoS attack, including but not limited to common services like DNS, NTP, BT-DHT and UPnP. We defined this kind of services as *VSD*. The detection model we proposed could easily adapt to different *VSD* for different network. As for the velocity, Attacker sends malicious *VSD* request at a very high speed to reflectors like DNS servers and IoT devices, and the accumulated response packets from distributed reflectors leads to even higher speed near the victim. Whereas the accumulated response packets also generate a large volume in the network. Given these features in the big data environment into consideration, the HDTI feature we designed and the detection method we proposed show higher detection rate, lower missing rate and false alarm rate compared to other detection methods.

DRDoS Attack Defense Comparison. The experiment results of our DRDoS attack defense method could be evaluated by the attack elimination rate, we conduct the experiment with $\Delta t = 0.1$, and the results are illustrated in Figures 8 and 9, where the $b_r = 1000$ and $b_q = 10$ were set.

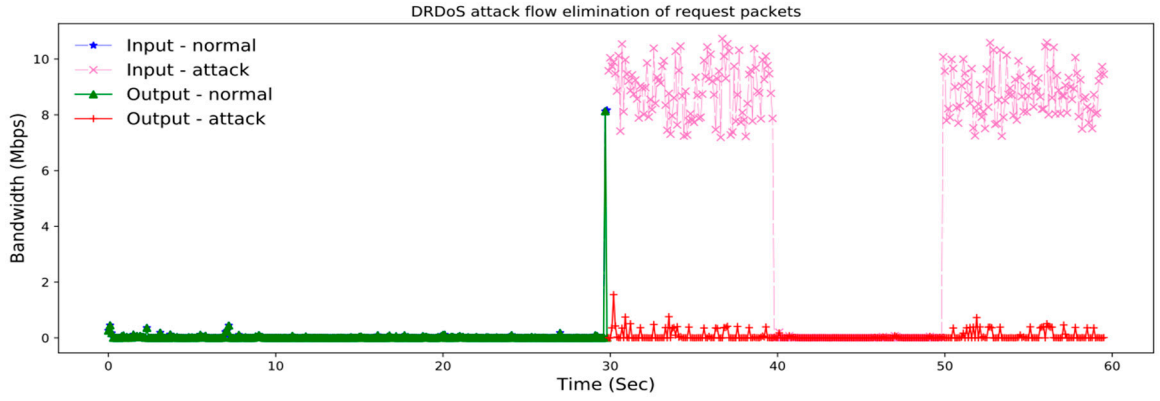


**Figure 8.** The proposed defense method when eliminate the upstream and MUD.

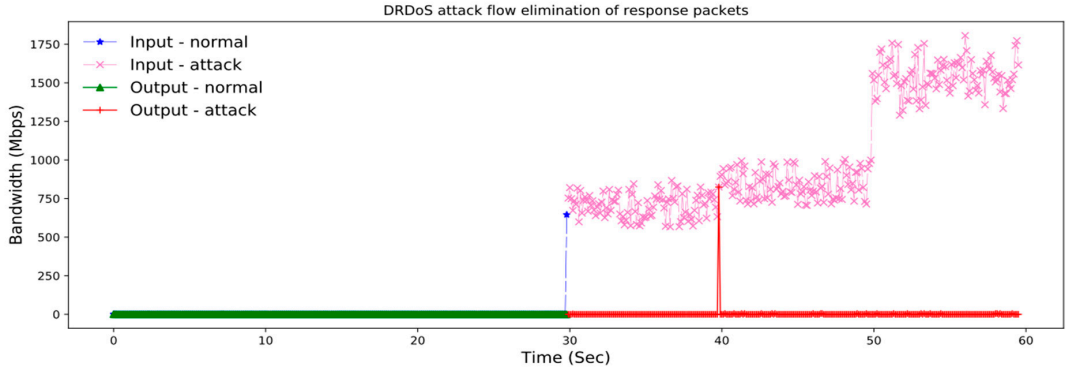


**Figure 9.** The proposed defense method when eliminate the downstream and MUD.

We could see that after applied the defense method we proposed, the DRDoS attack flow showed a gigantic drop, which suggests that the method is valid and could effectively relieve the network load for the intended victim, avoiding service degradation or denial of service for normal users as much as possible. And we illustrated the *AER* for $\Delta t = (0.1, 0.5, 1.0), b_r = 1000$ and $b_q = 10$ in Figure 10.

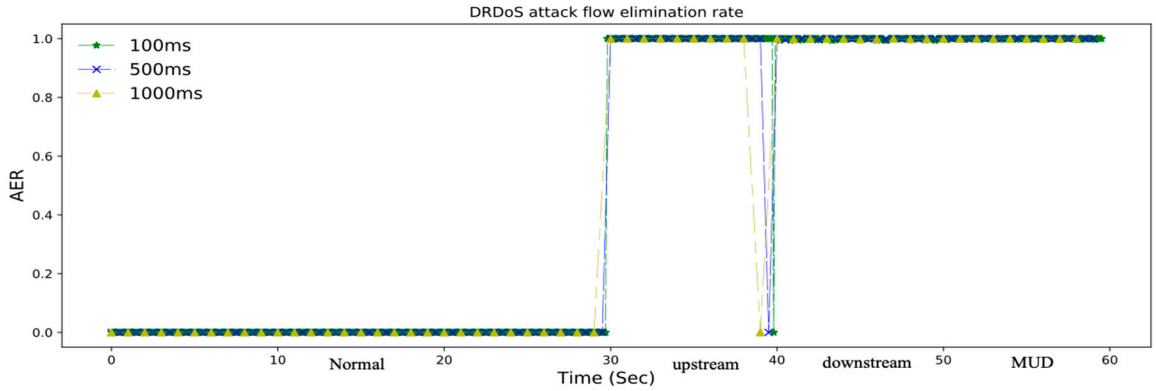


**Figure 10.** The attack flow elimination rate with different sample time.

As we can see in the Figure 10, after applied the method we proposed, the DRDoS attack flow could be reduced a lot in early, middle and post stages, which suggests that this method could be deployed at somewhere near the attack source, any internal node in the Internet, such as the router or switch in ISP, or deployed at the victim side.

### *5.2. Real-World Experiment with Memcached Service*

To demonstrate the method we proposed is effective and could adapt to other $VSD$, we conducted a real-world experiment with Memcached service. In this experiment, we observed the DRDoS attack reached about 50,000 amplification. And after we deployed the detection and defense models, the attack flow that reached the intended victim shows a gigantic drop, which suggested that the strong robustness and the flexibility of the method we proposed. To conduct this real-world experiment,

1. We scan over literally the whole IPv4 address space for finding servers with TCP port 11211 opens, which is the default port of memcached service. This takes more than 15 hours on our machine. And we discovered more than 2 million servers opens TCP port 11211.
2. Then we probe over the 2 million servers to filter out the server that actually runs the memcached service and responses to UDP requests. And we observed 1466 servers that could be used for initiating the memcached DRDoS attack, as shown in Figure 11.
3. In stage 3 we upload the payload to vulnerable servers, the payload is set to about 1 MB. And we actually could upload even larger payload to these servers, for example, 2 MB payload, so that the amplification would get almost doubled again.

4. In this attack phase, we send UDP requests to these vulnerable servers for retrieving the payload with source IP spoofed as the intended victim. The length of request UDP packet payload is 20 bytes, thus the amplification is $52,432.5 = \frac{1,048,650}{20}$ for each request. And to avoid real DRDoS attack, we only send these fake requests to 20 vulnerable servers. Even through, this results in 104 Mbps peak and 90 Mbps average Memcached DRDoS attack flow.
5. Then, at around 30th second, we deploy the method we proposed, we observed a huge drop of attack flow in the intended victim, as shown in Figure 12.

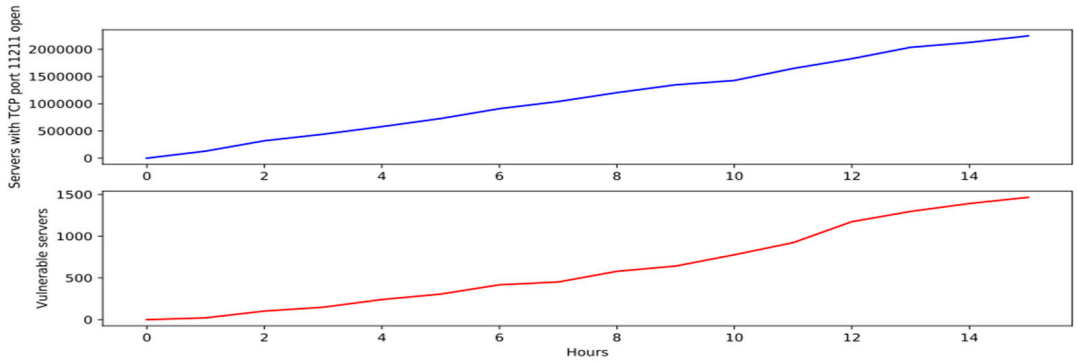


**Figure 11.** Scan and Probe over the whole IPv4 space. At the time we scanned, there're 2,247,438 servers with TCP port 11211 open, and 1466 servers among them actually run memcached service and responses to UDP requests.

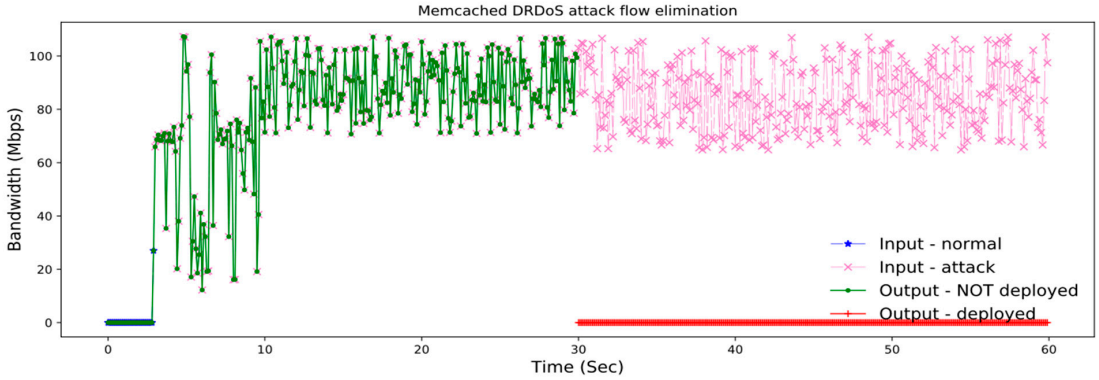


**Figure 12.** Performance of proposed method when generalized to memcached DRDoS.

As the result indicates, the proposed method could well generalize to other types of DRDoS with great performance on both detection and defense, which also proofs the robustness of this method.

## 6. Conclusions and Future Direction

In this paper, we developed a novel method to detect and defense DRDoS attacks with increasing IoT devices in the big data environment. To the part of detecting threats, we analyzed the statistical features of the DRDoS request and response packets through nodes in the network, uniqueness ports, packet lengths, and IP information lied in them. By our new definition upon these processed and combined features, the host-based DRDoS threat index, HDTI, is proposed. The evaluation and experiments had been confirmed the validity, efficiency, and accuracy of our proposed detection method. With the high accessibilities applying to the nodes inside the modern network environment by the proposed method, we are confident to put the detection model and take the HDTI as the main factor in host-based detection scheme and the trigger of the defense method upcoming in this paper.

As for the DRDoS defense method we proposed, it builds the on top of a deep forest model, which consists of 1 XGBoost estimator, 2 random forest estimators and 2 extra trees estimators in

each layer. Together with the differentiated service procedure, a highly effective defense method with low cost and false alarm rate or missing rate can be expected to be put into an actual environment for the test. The following experiments also proofed that the whole mechanism can stay on the expected accuracy dealing with data flow in high velocity and volume when processing in complicated situations with higher noise or distortion. More importantly, as our method is focusing on the most harmful and representative attack among different DDoS attacks, picking universal features which don't need to be sophisticating preprocessed, therefore our detection and defense method can be used against other kinds of DRDoS attacks on different layers in OSI.

Our upcoming tasks lie in the classification in the host-detection method, and the introduction of fusion-based detection and defense method. A universal evaluation of network status can be made by fusion-based methodology, with gathered information from the hosts, the macroscopic threat evaluating system by fusion of the cluster can be achieved afterward, which could optimize our proposed method in a new conventional mean.

**Author Contributions:** Conceptualization, R.X.; Formal analysis, R.X. and F.W.; Investigation, R.X. and F.W.; Methodology, F.W.; Project administration, J.C. and X.T.; Resources, J.C. and J.X.; Software, R.X.; Visualization, F.W.; Writing—original draft, R.X. and F.W.; Writing—review & editing, R.X., J.C. and F.W.

**Funding:** This work was supported by the National Natural Science Foundation of China [61762033, 61702539]; The National Natural Science Foundation of Hainan (617048, 2018CXTD333); Hainan University Doctor Start Fund Project (kyqd1328); Hainan University Youth Fund Project (qnjj1444). This work is partially supported by Social Development Project of Public Welfare Technology Application of Zhejiang Province (LGF18F020019).

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

## References

1. Vukovic, O.; Dan, G. Security of fully distributed power system state estimation: Detection and mitigation of date integrity attacks. *IEEE J. Sel. Areas Commun.* **2014**, *32*, 1500–1508. [CrossRef]
2. Cloudflare. Available online: https://blog.cloudflare.com/the-daily-ddos-ten-days-of-massive-attacks/ (accessed on 7 May 2018).
3. CERT Coordination Center. *Results of the Distributed-Systems Intruder Tools Workshop*; Software Engineering Institute: Pittsburgh, PA, USA, 1999.
4. Garber, L. Denial-of-service attacks rip the Internet. *Computer* **2000**, *33*, 12–17. [CrossRef]
5. Kargl, F.; Maier, J.; Weber, M. Protecting web servers from distributed denial of service attacks. In Proceedings of the 10th international conference on World Wide Web, Hong Kong, China, 1–5 May 2005; ACM: New York, NY, USA, 2001; pp. 514–524.
6. Weiler, N. Honeypots for distributed denial-of-service attacks. In Proceedings of the Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Pittsburgh, PA, USA, 12 June 2002.
7. Cheng, J.; Yin, J.; Liu, Y. DDoS Attack Detection Using IP Address Feature Interaction. In Proceedings of the International Conference on Intelligent NETWORKING and Collaborative Systems, Barcelona, Spain, 4–6 November 2009; IEEE Computer Society: Washington, DC, USA, 2009; pp. 113–118.
8. Cheng, J.; Zhang, B.; Yin, J. DDoS Attack Detection Using Three-State Partition Based on Flow Interaction. *Commun. Comput. Inf. Sci.* **2009**, *29*, 176–184.
9. Cheng, J.; Yin, J.; Liu, Y. Detecting Distributed Denial of Service Attack Based on Multi-feature Fusion. In Proceedings of the Security Technology, Proceedings of the International Conference, Jeju Island, Korea, 10–12 December 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 132–139.
10. Cheng, J.; Tang, X.; Zhu, X.; Yin, J. Distributed denial of service attack detection based on IP Flow Interaction. In Proceedings of the International Conference on E-Business and E-Government (ICEE), Shanghai, China, 6–8 May 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 1–4.
11. Zhu, X.; Li, X.; Liu, M.; Zhu, E.; Liu, L.; Cai, Z.; Yin, J.; Gao, W. Localized Incomplete Multiple Kernel k-means. *IJCAI* **2018**, 3271–3277. [CrossRef]
12. Wang, S.; Liu, Q.; Zhu, E.; Porikli, F.; Yin, J. Hyperparameter selection of one-class support vector machine by self-adaptive data shifting. *Pattern Recognit.* **2018**, *74*, 198–211. [CrossRef]

13. Cheng, J.; Chen, Z.; Tang, X. Adaptive DDoS attack detection method based on multiple-kernel learning. *Secur. Commun. Netw.* **2018**, *2018*, 5198685. [CrossRef]
14. Cheng, J.; Xu, R.; Tang, X. An Abnormal Network Flow Feature Sequence Prediction Approach for DDoS Attacks Detection in Big Data Environment. *Comput. Mater. Contin.* **2018**, *55*, 95–119.
15. Cheng, J.; Zhou, J.; Liu, Q.; Tang, X.; Guo, Y. A DDoS Detection Method for Socially Aware Networking Based on Forecasting Fusion Feature Sequence. *Comput. J.* **2018**, *61*, 959–970. [CrossRef]
16. Cheng, J.; Li, M.; Tang, X.; Sheng, V.S.; Liu, Y.; Guo, W. Flow Correlation Degree Optimization Driven Random Forest for Detecting DDoS Attacks in Cloud Computing. *Secur. Commun. Netw.* **2018**, *2018*, 6459326. [CrossRef]
17. Zhang, R.; Cheng, J.; Tang, X.; Liu, Q.; He, X. DDoS Attack Security Situation Assessment Model Using Fusion Feature Based on Fuzzy C-Means Clustering Algorithm. In Proceedings of the International Conference on Cloud Computing and Security (ICCCS), Haikou, China, 8–10 June 2018; pp. 654–669.
18. Manikopoulos, C.; Papavassiliou, S. Network intrusion and fault detection: A statistical anomaly approach. *IEEE Commun. Mag.* **2002**, *40*, 76–82. [CrossRef]
19. Liao, H.; Lin, C.R.; Lin, Y. Intrusion detection system: A comprehensive review. *J. Netw. Comput. Appl.* **2013**, *36*, 16–24. [CrossRef]
20. Garcia-Teodoro, P.; Diaz-Verdejo, J.; Macia-Fernandez, G. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Comput. Secur.* **2009**, *28*, 18–28. [CrossRef]
21. Li, J.; Yan, Q.; Chang, V. Internet of Things: Security and privacy in a connected world. *Future Gener. Comp. Syst.* **2018**, *78*, 931–932. [CrossRef]
22. Sun, W.; Cai, Z.; Li, Y.; Liu, F.; Fang, S.; Wang, G. Security and Privacy in the Medical Internet of Things: A Review. *Secur. Commun. Netw.* **2018**, *2018*, 5978636. [CrossRef]
23. Alsmadi, I.M.; Karabatis, G.; Aleroud, A. *Information Fusion for Cyber-Security Analytics*; Springer: Basel, Switzerland, 2017.
24. Aleroud, A.; Karabatis, G. Contextual information fusion for intrusion detection: A survey and taxonomy. *Knowl. Inf. Syst.* **2017**, *52*, 563–619. [CrossRef]
25. AlEroud, A.; Karabatis, G. Beyond data: Contextual information fusion for cyber security analytics. In Proceedings of the 31st ACM Symposium on Applied Computing, Pisa, Italy, 4–8 April 2016.
26. Rajeev, S.; Sivanandam, S.N.; Pradeep, P. *Architecture for Authentication in Wireless Differentiated Services Using Distributed Substring Authentication Protocol (DSAP)*; Assumption University: Bangkok, Thailand, 2015.
27. Black, D.; Jones, P. Differentiated Services (DiffServ) and Real-time Communication. 2015. Available online: https://buildbot.tools.ietf.org/html/rfc7657 (accessed on 22 November 2018).
28. Mahale, V.; Pareek, P.; Uttarwar, U. Alleviation of DDoS attack using advance technique. In Proceedings of the International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 21–23 February 2017; IEEE: Piscataway, NJ, USA, 2017; Volume 1, pp. 172–176.
29. Apiecionek, L.; Czerniak, M.; Dobrosielski, T. Quality of services method as a DDoS protection tool. In Proceedings of the Intelligent Systems'2014, Proceedings of the 7th IEEE International Conference Intelligent Systems IS'2014, Warsaw, Poland, 24–26 September 2014; Springer: Berlin/Heidelberg, Germany, 2015.
30. Kambourakis, G.; Moschos, T.; Geneiatakis, D.; Gritzalis, S. A fair solution to DNS amplification attacks. In Proceedings of the Second International Workshop on Digital Forensics and Incident Analysis (WDFIA 2007), Samos, Greece, 27–28 August 2007.
31. Anagnostopoulos, M.; Kambourakis, G.; Kopanos, P.; Louloudakis, G.; Gritzalis, S. DNS amplification attack revisited. *Comput. Secur.* **2013**, *39*, 475–485. [CrossRef]
32. Kramer, L.; Krupp, J.; Makita, D. Amppot: Monitoring and defending against amplification ddos attacks. In Proceedings of the International Workshop on Recent Advances in Intrusion Detection, Kyoto, Japan, 2–4 November 2015; Volume 1, pp. 615–636.
33. Jing, L.; Licheng, W.; Lihua, W. *Verifiable Chebyshev Maps-Based Chaotic Encryption Schemes with Outsourcing Computations in the Cloud/Fog Scenarios. Concurrency and Computation: Practice and Experience*; Wiley Online Library: Hoboken, NJ, USA, 2018. [CrossRef]
34. Li, J.; Chen, X.; Chow, S.S.; Huang, Q.; Wong, D.S.; Liu, Z. Multi-authority fine-grained access control with accountability and its application in cloud. *J. Netw. Comput. Appl.* **2018**, *112*, 89–96. [CrossRef]
35. Gao, C.Z.; Cheng, Q.; He, P.; Susilo, W.; Li, J. Privacy-Preserving Naive Bayes Classifiers Secure against the Substitution-then-Comparison Attack. *Inf. Sci.* **2018**, *444*, 72–88. [CrossRef]

36. Li, J.; Liu, Z.; Chen, X.; Xhafa, F.; Tan, X.; Wong, D.S. L-EncDB: A Lightweight Framework for Privacy-Preserving Data Queries in Cloud Computing. *Knowl.-Based Syst.* **2015**, *79*, 18–26. [CrossRef]
37. Li, T.; Li, J.; Liu, Z.; Li, P.; Jia, C. Differentially Private Naive Bayes Learning over Multiple Data Sources. *Inf. Sci.* **2018**, *444*, 89–104. [CrossRef]
38. Liu, Z.; Wu, Z.; Li, T.; Li, J.; Shen, C. GMM and CNN Hybrid Method for Short Utterance Speaker Recognition. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3224–3252. [CrossRef]
39. Li, J.; Sun, L.; Yan, Q.; Li, Z.; Srisa-an, W.; Ye, H. Significant permission identification for machine learning based android malware detection. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3216–3225. [CrossRef]
40. Li, Y.; Wang, G.; Nie, L.; Wang, Q.; Tan, W. Distance Metric Optimization Driven Convolutional Neural Network for Age Invariant Face Recognition. *Pattern Recognit.* **2018**, *75*, 51–62. [CrossRef]
41. Shen, J.; Gui, Z.; Ji, S.; Shen, J.; Tan, H.; Tang, Y. Cloud-aided Lightweight Certificateless Authentication Protocol with Anonymity for Wireless Body Area Networks. *J. Netw. Comput. Appl.* **2018**, *106*, 117–123. [CrossRef]
42. Rossow, C. *Amplification Hell: Revisiting Network Protocols for DDoS Abuse*; NDSS: New York, NY, USA, 2014.
43. Ryba, F.J.; Orlinski, M.; Wählisch, M.; Rossow, C.; Schmidt, T.C. Amplification and DRDoS Attack Defense—A Survey and New Perspectives. *arXiv*, 2015; arXiv:1505.07892.
44. Czyz, J.; Kallitsis, M.; Gharaibeh, M. Taming the 800 pound gorilla: The rise and decline of NTP DDoS attacks. In Proceedings of the Conference on Internet Measurement, Vancouver, BC, Canada, 5–7 November 2014; ACM: New York, NY, USA, 2014; Volume 1, pp. 435–448.
45. Karami, M.; McCoy, D. Understanding the Emerging Threat of DDoS-as-a-Service. In Proceedings of the 6th USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET'13), Washington, DC, USA, 12 August 2013.
46. Durumeric, Z.; Bailey, M.; Halderman, A. An Internet-Wide View of Internet-Wide Scanning. In Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, 20–22 August 2014; Volume 1, pp. 65–78.
47. Jeff Yan, J.; Choi, H.J. Security issues in online games. *Electron. Libr.* **2002**, *20*, 125–133. [CrossRef]
48. Paulson, A.; Weber, E. Cyberextortion: An overview of distributed denial of service attacks against online gaming companies. *Issues Inf. Syst.* **2006**, *7*, 52–56.
49. Bingshuang, L.; Jun, L.; Tao, W. SF-DRDoS: The store-and-flood distributed reflective denial of service attack. *Comput. Commun.* **2015**, *69*, 107–115.
50. Kolias, C.; Kambourakis, G.; Stavrou, A.; Voas, J. DDoS in the IoT: Mirai and other botnets. *Computer* **2017**, *50*, 80–84. [CrossRef]
51. WRCCDC 2018. Available online: https://archive.wrccdc.org/pcaps/2018/ (accessed on 11 June 2018).