

Article

# Implementation of a Clustering-Based LDDoS Detection Method

Tariq Hussain <sup>1,\*</sup>, Muhammad Irfan Saeed <sup>2</sup>, Irfan Ullah Khan <sup>3,†</sup>, Nida Aslam <sup>4,†</sup>  
and Sumayh S. Aljameel <sup>3</sup>

<sup>1</sup> School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China

<sup>2</sup> Software College, Northeastern University, Shenyang 110819, China

<sup>3</sup> Department of Computer Science, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia

<sup>4</sup> Saudi Aramco Cybersecurity Chair, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia

\* Correspondence: uom.tariq@gmail.com

† These authors contributed equally to this work and are first co-authors.

**Abstract:** With the rapid advancement and transformation of technology, information and communication technologies (ICT), in particular, have attracted everyone's attention. The attackers took advantage of this and can cause serious problems, such as malware attack, ransomware, SQL injection attack, etc. One of the dominant attacks, known as distributed denial-of-service (DDoS), has been observed as the main reason for information hacking. In this paper, we have proposed a secure technique, called the low-rate distributed denial-of-service (LDDoS) technique, to measure attack penetration and secure communication flow. A two-step clustering method was adopted, where the network traffic was controlled by using the characteristics of TCP traffic with discrete sense; then, the suspicious cluster with the abnormal analysis was detected. This method has proven to be reliable and efficient for LDDoS attacks detection, based on the NS-2 simulator, compared to the exponentially weighted moving average (EWMA) technique, which has comparatively very high false-positive rates. Analyzing abnormal test pieces helps us reduce the false positives. The proposed methodology was implemented using Python for scripting and NS-2 simulator for topology, two public trademark datasets, i.e., Web of Information for Development (WIDE) and Lawrence Berkley National Laboratory (LBNL), were selected for experiments. The experiments were analyzed, and the results evaluated using Wireshark. The proposed LDDoS approach achieved good results, compared to the previous techniques.

**Keywords:** low-rate distributed DoS (LDDoS) attacks; attacks detection; two-step clustering; outliers detection



**Citation:** Hussain, T.; Saeed, M.I.; Khan, I.U.; Aslam, N.; Aljameel, S.S. Implementation of a Clustering-Based LDDoS Detection Method. *Electronics* **2022**, *11*, 2804. <https://doi.org/10.3390/electronics11182804>

Academic Editors: Houbing Song, Jehad Ali and Juan-Carlos Cano

Received: 18 June 2022

Accepted: 25 August 2022

Published: 6 September 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In recent few years, the advent of the Internet has introduced mankind to the most fascinating technologies and services. The majority of the world's population is now connected via the Internet [1]. According to the June 2019 Statistical Report on Internet Users, there were a total of 4.33 billion active Internet users, which was 56% of the total world population [2]. The report also mentioned that China was the largest consumer of the internet, with 829 million of active internet users. With the development of science and technology, computer networks [3] spread into people's lives, bringing people's lives together more than ever with a wide range of spectra affecting each section and improving everyday life [4,5]. It has many capabilities and applications for distance education, office automation [6], e-commerce, digital currencies [7], online payments, social media and communication [8]. At the same time, it has many flaws in its network architecture that

make it vulnerable for people and their credentials [9,10]. Since the initial design of the network was poor, as the security factor was not properly considered, there are many difficulties in finding security loopholes, which [11,12] can easily be maliciously exploited and infringe upon the private and property security of others or affect the normal provision of network services [13,14].

With the growth of network applications, the internet security factor becomes more and more serious [15]. However, where there is interest, there is contention, and the Internet is not an exception [16,17]. The internet designers paid more attention to the efficiency of message delivery and less attention to security [18,19]. So, as time has passed, more and more networks have been exposed to security problems, and the forms of network attacks have become diverse, from the initial worm virus to Trojans and phishing, as well as from system vulnerability to the use of web scripts, network security faces more and more challenges. According to the survey [20], on average, a successful cyber-attack takes place every 20 s worldwide. The nature of network security is becoming more and more stringent. Attack and defense have become a new means of “war” [21].

In early 2015, China incorporated cybersecurity into national security as a complete strategy [22]. Numerous types of attacks are classified as social engineering penetration attacks, password cracking, intrusion attacks, cross-site scripting (XSS) attacks, SQL injection attacks, cross-site request forgery (CSRF) attacks, arp-spoofing attacks, phishing attacks, DoS attacks, Trojan horse attacks, etc. [23]. When it comes to multi-network attacks, DoS attacks are considered to be the most dangerous for the Internet [24,25].

Cybersecurity is becoming more and more crucial with the growth of the Internet and its services. The Internet and its services touch nearly 4.5 billion active users, representing 58% of the entire population of the world. So, as internet usage is increasing [26,27], cybercrime rates are also increasing, damaging or infringing on the personal information of billions of users.

The researchers’ attention on this topic has increased, due to the rise of cyber threats and people’s total dependency on the internet, making it more and more important to detect and mitigate cyber threats. Denial-of-service (DoS) attacks first became dangerous [28] with the invention of their distributed category, known as distributed denial-of-service, and they became too severe with the advent of low-rate denial-of-service attacks. Low-rate denial-of-service attacks are more severe, as they behave similar to legitimate traffic while passing through the network, making them very difficult to detect through the available detection mechanisms, meaning they easily fool the available techniques and can disrupt the network by consuming its resources and capacity [29,30].

The proposed research includes the mechanism to detect LDDoS attacks, since they are dangerous and increasing so fast. More specifically, we focus on the early detection of new cyber threats, called LDDoS. To detect LDDoS attacks, we proposed a clustering-based mechanism. In the proposed mechanism, we performed experiments using two-step clustering techniques to detect the LDDoS after analyzing the anomalous properties of TCP traffic caused by LDDoS attacks.

The pre-cluster step uses a sequential clustering approach. It scans the data sets one after the other and uses the distance criterion to decide whether the current data set is to be merged with the previously formed clusters or whether a new cluster needs to be created. For the distance, we used the Mahalanobis distance. The method is implemented by constructing a modified cluster feature (CF) tree. Cluster formation is the second step that groups the sub-clusters resulting from the pre-clustering step into the desired number of clusters. Since the number of sub-clusters is much less than the number of original records, traditional clustering methods can be used effectively. The agglomerative hierarchical clustering technique was adopted for the two-step method because it works well with the auto-cluster.

The two-step cluster analysis method analyzes the TCP traffic on a large timescale, which can result in a high false-positive rate. Therefore, another method based on the analysis of TCP traffic is proposed to reduce the false alarm rate. From a small timescale

perspective, TCP traffic fell off quickly and then recovered under LDDoS attacks [31]. The range of the TCP traffic is much larger than the normal network in short attack duration, so we split each detection unit (DU) in the suspected cluster into many test pieces (TP), with each TP having  $k$  samples. The TP must be greater than the attack period  $T$ , so that we can get a full TCP traffic changing process.

In addition, we chose Kali Linux for the implementation because of its extensive services, in terms of cyber attacking and defense tools, and its command-based operations. Its command-based operations made it difficult to apply the proposed techniques in the Kali Linux environment, specifically for Windows users.

The structure of the paper is as follows. Section 2 discusses the cyber-attacks and their related work. Section 3 proposes the mechanism, while Section 4 is based on the evaluation of the algorithm. Section 5 contains the conclusion of our proposed work.

## 2. Related Works

This section discusses recent developments and research related to cyber threats, specifically DDoS attacks.

The history of DoS began in the 1980s and early 1990s; however, due to the low usage of the Internet, it was not highlighted as what we have today. The alternative idea to this arose when the use of the Internet began to rapidly evolve and connect to the source of the medium [1].

An attack called SYN FLOOD by DoS crippled all internet facilities in New York City, in which a group of ISPs, called Panix, went offline for a week, while Internet usage and web services using that terminology in the New York Times and Internet Chess Club were deactivated [2]. Later, researchers started working on solving this problem and created an effective method such kinds of attacks can be stopped before impersonation and eavesdropping. After two months of the incident, they released a solution to the DoS attack as a product. The idea of this commercial product was to detect the incoming SYN packets from DoS and prevent them from being triggered and executed in a system. The solution was good but failed on the live webcam, and many products went offline [3,4].

The authors of [5,6] proposed a SYN-based technique where the DoS attacks were detected and recognized. For this idea, they proposed a novel scheme called intrusion detection for DoS attacks. It is clear from the algorithm that the proposed mechanism uses the flooding approach, in which the spoofed packets are triggered to detect the DoS attack on the packets. From executing the booby trap mechanism, the authors had to use an alternate path to forward the data. The authors of [7–9] proposed a technique for TCP/ICMP packets, in order to detect the DoS attacks on the packets during transmission. A dedicated path is used to transfer data from one client to another. A ping method [10] was introduced to detect the DoS attacks where the victim was TCP/ICMP packets.

Authors in [11] proposed a report using IRC-based DDoS attacks to execute. To do this, they introduced a robust technique to be applied to the devices from both the sender and receiver sides. The attackers used the legitimate commands sent to the client and server, mimicking each other in nature. The attacks were difficult to detect, due to their friendly nature [11]. They reported that IRC must be the best edition of the product, as well as to utilize all its resources and detection approaches to detect the attacks of DDoS. The authors stated that these attacks can be friendly, which is difficult to detect. The solution is to use the ping command, which uses a unique ID. This ID is kept open only for the registered users and those for whom there was no ID, in order to trigger the attacks [13].

The authors in [14–22] reported on DDoS attacks, introducing its new types, i.e., Teardrop and Bonk. These were the types of DDoS attacks that can conquer any security tier. The key factor was that the main course needs to interface in order to run abnormally.

The authors of [23–32] reported on wormholes in DDoS attacks, which are observed as a disaster for any kind of wireless and wired media. They had reported that this malicious activity can damage any kind of software and slow down the speed of any hardware and software by stealthily installing it while running in another program. In today's world,

DoS attacks are being pronounced as DDoS attacks because distributed devices or botnets are used and controlled to disrupt the victim's target network.

Distributed denial-of-service attacks are the advanced version of denial-of-service attacks; they use the same paradigm to disturb the network of the victims but in a distributed manner. The date 22 July 1999 is ominous in the history of computing. On that day, a computer at the University of Minnesota was suddenly attacked by a network of 114 other computers, which were infected with a malicious script called Trin00 [1]. This marks the start of the distributed version of the denial-of-service attacks. DDoS is considered to be one of the most dangerous attacks to date, due to its magnitude and power to completely abandon the services of the victims for a better range of time.

The distributed version of DoS attacks still poses the major threat to network security and its infrastructure, as well as the entire Internet world, but it gained momentum after the invention of new low pulsing denial-of-service attacks, recently known as low-rate DoS attacks, which are known to attract more attentive attacks. These attacks use very little bandwidth to transmit attacks to the victim's server, and they can easily trick the built-in detection mechanism for distributed DoS attacks because they move slowly and constantly, similar to legitimate traffic.

DoS attack technology has continued to evolve, and the destructiveness is also increasing. Especially in recent years, a kind of LDDoS has emerged that is more threatening than traditional DDoS attacks. Although the traditional DDoS attack is very destructive, it has a common characteristic, due to its attack principle, since the attacker, through a kind of pressure (sledgehammer), has to send many attack packets to the target, forcing the attacker to maintain an attack flow with high frequency and speed. This characteristic causes all types of traditional DoS attacks to have an anomalous statistical characteristic, compared to normal network traffic, making them relatively easy to detect. Therefore, many DoS detection methods take these abnormal statistical characteristics as a feature to identify DoS attacks. Once an attack is detected, the packet filtering mechanism is activated to discard all packets transmitted by the data stream with attack characteristics. Low-rate DDoS attacks are quite different from traditional DDoS attacks, as their traffic is similar to legitimate traffic.

A low-rate DDoS attacker exploits the vulnerability of the TCP's congestion control mechanism by periodically sending burst attack packets repeatedly over short periods of time (pulsing attack) or continuously launching attack packets at a constant low rate (constant attack). As these attacks reduce the average number of attack packets to avoid being detected by existing detection schemes, it is difficult to distinguish such attacks from legitimate traffic with a large measurable distance gap and low false-negative rate. The biggest feature is that it does not need to maintain the high-speed attack flow and exhausts all available resources on the victim's side. Instead, it uses the security and vulnerability in the common adaptive mechanism (such as the congestion control mechanism of TCP) in the network protocol or application services to periodically burst in a specific short time interval, in order to send many attack packets and reduce the service performance to the attacked end. The LDDoS attack only sends data in a specific time interval and does not send any data in other periods of time of the same cycle. The intermittent attack feature makes the average rate of attack flow relatively low, which is not different from the data flow of legitimate user's data flow, and no longer exhibits the above abnormal statistical characteristics.

It is difficult to use the existing methods to prevent this. It can be assumed that an LDDoS attack is an improved form of a traditional DoS attack. Compared with traditional DDoS attacks, it has a more targeted approach, so the attack efficiency has been greatly improved, and it evades detection and prevention. The emergence of LDDoS attacks brings new challenges to the research of attack prevention. The research on LDDoS attacks is still in its infancy, but the related research work has mainly appeared in recent years, showing that it has received enough attention [10]. In 2003, Aleksandar of Rice University first proposed a low-rate denial-of-service attack on TCP protocol at Sigcom, the highest rating conference on computer networks. This attack mainly targets the loopholes in the

TCP congestion control mechanism. On ICNP (in 2004) and Infocom (in 2005), Giurgiu proposed the ROQ attack [21]. It also aimed at the congestion control in the TCP protocol and loopholes in the router queue management mechanism, causing the performance of certain routers to degrade. This type of attack has also appeared in the network layer. Therefore, it is an urgent problem in the field of network security to propose the detection and prevention methods for this kind of attack.

### 3. Proposed Detection Method

The proposed two-step clustering analysis algorithm and sub-algorithms that take place in the pre-cluster and cluster phases are discussed in detail in this section. The abnormal packet analysis is also discussed in detail here, which is critical for reducing the FP rate in LDDoS detection.

#### 3.1. General Idea of the Proposed Algorithm

The two-step clustering mechanism has been proposed to detect LDDoS attacks. For data acquisition, the NS-2 was implemented for the network topology and network traffic collected from the core router installed in it. Two datasets, called the Lawrence Berkley National Laboratory (LBNL) [11] and WIDE project [12], were used for the project. Both are analyzed with a Wireshark traffic analyzer. The TCP traffic was observed from the data characteristics and calculated for the two steps of the clustering mechanism. For the LDDoS attack detection, the analysis of the two-step clustering mechanisms has been adapted for clustering the traffic of the network. From there, the suspected clusters were detected and removed using TCP data. The results showed that the proposed detection for LDDoS worked well.

##### 3.1.1. Abnormal Characteristics of TCP Traffic Caused by LDDoS Attacks

Abnormal characteristics of the TCP traffic caused by the LDDoS attack are shown in Figure 1, with a duration between 0 and 600 s. From Figure 1, it can be seen in Figure 1 that the TCP traffic under an LDDoS attack is somewhat more discrete than the normal environment of the TCP traffic with additional attacks, such as DDoS. The variance is shown in Equations (1)–(3), each representing the different aspects of the terminology. The mean deviation is calculated in Equation (2), using the coefficient of variation with the standardized probability measure and frequency distribution. The ratio of the standard deviation is provided in Equation (3), where the  $N$  denotes the overall number and  $x$  represents the sample value and total samples.

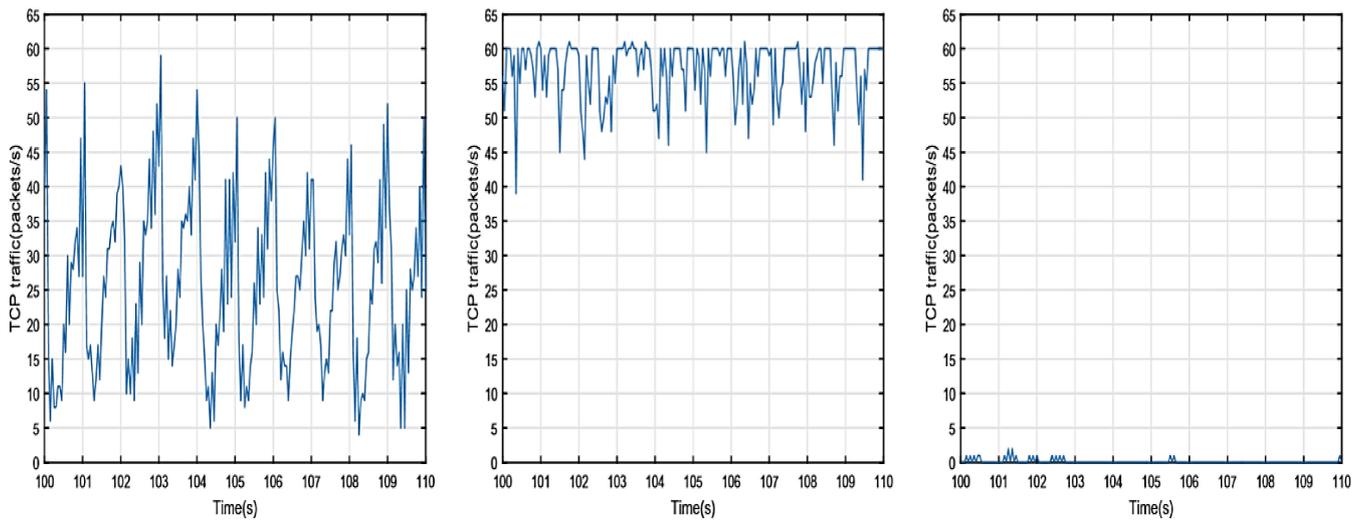
$$V = \frac{1}{N} \sum_{i=1}^n (x_i - m) \quad (1)$$

$$MD = \frac{1}{N} \sum_{i=1}^n |x_i - m| \quad (2)$$

$$CV = \frac{\sqrt{V}}{m} \quad (3)$$

From this viewpoint, in a LDDoS attack, the TCP blockage control instrument would be set off when the assailant sends high-rate blasts to the person in question. Then, the organization would draw certain lines on the TCP traffic at that point, and the TCP traffic begins to recuperate when the attack stops.

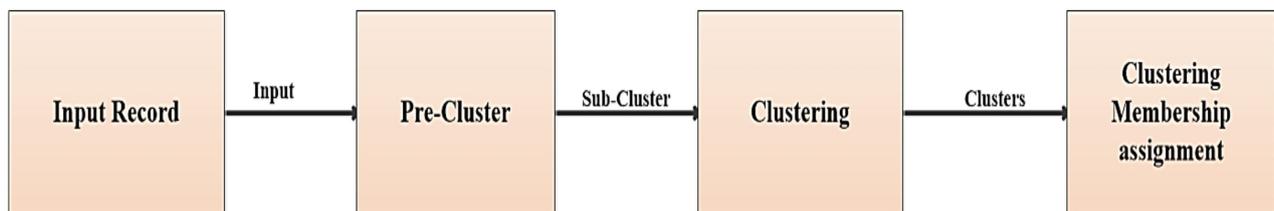
TCP traffic has to go through a mechanism called “steady decline and then steady increase”, while the range of the TCP traffics is much larger, as compared to the normal flow in a network with a limited duration DDoS attack. From Figure 1, TCP traffic was measured with network environments of different scenarios, with a time duration of 100–110 s. The flow of TCP traffic and samples are included in Figure 1.



**Figure 1.** (a) TCP traffics under LDDoS attacks; (b) TCP traffic under no attack; (c) TCP traffic under DDoS, general TCP traffic with different environments in a small-time scale [1].

### 3.1.2. Two-Step Cluster Analysis Algorithm

The proposed research work presented in this article follows the way of two-step clustering to detect the LDDoS attacks after analyzing and evaluating the normal and abnormal characteristics, as well as the traffic of TCP caused by them. The two-step cluster analysis process is given in Figure 2.



**Figure 2.** Flowchart of two-step cluster mechanism.

### 3.2. Algorithm Design

The two-step clustering algorithm is designed based on the pre-clustering and clustering phases. Here, we describe its design, in terms of its phases.

#### 3.2.1. Cluster Feature (CF)

A balanced iterative reducing scheme and clustering using hierarchies (BRICH) is used to minimize the memory requirements in a large dataset by aggregating the information present in a dense area as CF entries.

$$CF_1 + CF_2 = (N_1 + N_2, LS_1 + LS_2, SS_1 + SS_2) \tag{4}$$

where SS means slow-start, LS is the linear sum of N, and SS is the square sum of N.

#### Insertion Algorithm

These algorithmic steps are used to insert the CF entry, whether single data or sub-clusters, into the cluster feature tree. There are few steps to follow:

- *Step 1*—Start.
- *Step 2*—Used for the identification purpose of the appropriate leaf. It starts from the root node and goes down to the child nodes, until it reaches the null.
- *Step 3*—The leaf modification step. It reaches the leaf nodes and, from there, starts the working process, until all the leaf nodes are traversed with CF.

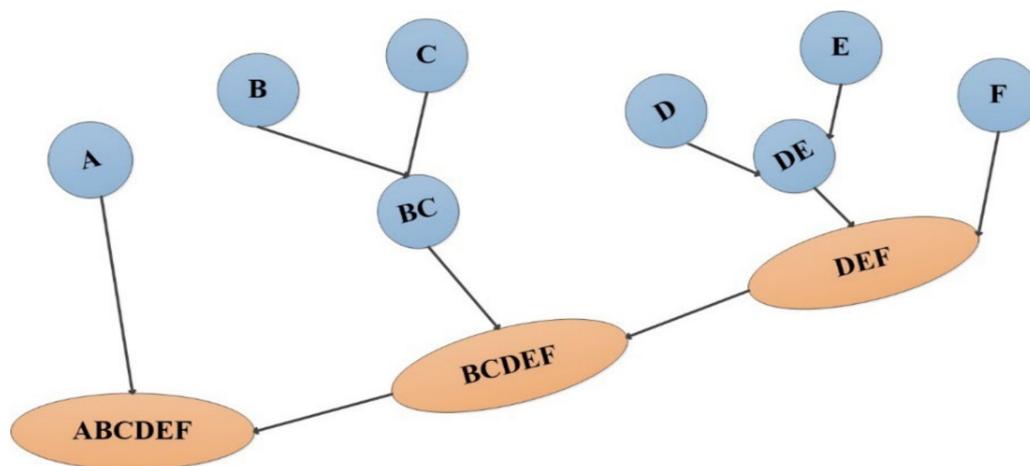
- *Step 4*—For the modification of the leaf node. It modifies the leaf node by traversing it and counts all the possible hops, which can be used by CF to find the suitable node.
- *Step 5*—End.

### 3.2.2. Agglomerate Clustering Algorithm

The below steps create the agglomerate cluster. The steps taken by this algorithm are:

- *Step 1*—Start.
- *Step 2*—In the underlying advance, we work out the nearness of individual places and consider every one of the six data of interest as individual groups.
- *Step 3*—In step two, similar clusters are merged and formed as a single cluster. Let us consider that B C and D E are similar clusters that are merged in step two. Now, we are left with four clusters, which are A, BC, DE, and F.
- *Step 4*—We again ascertain the vicinity of new bunches and union the comparative groups to shape new bunches, i.e., A, BC, and DEF.
- *Step 5*—Calculate the closeness of the new groups. The bunches DEF and BC are comparable and consolidated to frame another group. We are presently left with two bunches, i.e., A and BCDEF.
- *Step 6*—Finally, every one of the bunches is combined and structured into a solitary group.
- *Step 7*—End.

The Figure 3 algorithms present CF entry insertion and agglomerative clustering. Clustering is directly related to TCP traffic. Therefore, when TCP traffic increases, the number of new clusters increases and vice versa. As a result, low-rate denial-of-service attack detection can plot a lot of wrong alarms, which means it has a lot of capability to trigger an increased false-positive rate. To overcome this problem, we have proposed abnormal test packets that analyze the suspected clusters and reduce the false positive rate to get a better result.



**Figure 3.** Agglomerative clustering algorithm.

### 3.2.3. Mahalanobis Distance

The distance metric is a key issue in many machine learning algorithms. For instance, K-means, and the K-nearest neighbor (KNN) classifier should provide a reasonable distance metric, through which adjoining information focuses can be recognized. The Mahalanobis distance is an action between two data of interest in the space characterized by applicable elements, since it represents inconsistent differences, as well as relationships between elements, that adequately assesses the distance by allotting various loads or significant variables to the highlights of the data of interest. Euclidean distance helps prepare brain organization for one-shot learning, but it also has several weaknesses. It handles the contrast between various properties of the example (i.e., each pointer or variable), which sometimes does not meet the actual requirements. Based on the literature, it was observed

that Mahalanobis distance was broadly applied to face recognition, in general, and produced significant results, compared with Euclidean distance. The Mahalanobis distance was proposed by the Indian analyst Mahalanobis to address the covariance distance of the information. It is an effective method for ascertaining the similitude of two obscure example sets. Unlike Euclidean distance, it considers the association between different qualities (for instance, a message about a level will achieve a message weight because the two are connected) and is scale-autonomous (scale invariant), which is free from the estimation scale.

The distance estimation is the first step (pre-requisite stage) in the proposed two-stage research work. So, we use Mahalanobis to observe the distance between two sets, expecting  $a$  and  $b$ . The mathematical expression is:

$$dm(a, b) = \sqrt{\frac{\sum_{i=1}^a (a_i - b_i)^2}{(\delta_{ia})^2 (\delta_{ib})^2}} \quad (5)$$

where  $(\delta_{ib})$  is the standard deviation for aspect  $i$  in bunch  $b$ , and  $(\delta_{ia})$  is the standard deviation for aspect  $i$  in group  $a$ .

Typically, there are two issues with Euclidean distance. The justification behind that will be that can be made sense of with a model. Adding the subsequent guide adds no data to the issue. The Euclidean distance anyway has no chance of realizing those two focuses are indistinguishable [13] and will count similar information two times. This would place the top load on the focuses being referred to. The issue is, to some degree, decreased when there is a fractional relationship, but it is something special to be kept away from overall. Therefore, we chose to utilize the Mahalanobis distance condition much more pertinently in observing anomalies and relationship points for bunching in TCP traffic.

#### 3.2.4. Abnormal Test Piece

The two-step clustering analysis strategy would group input datasets into a few clusters. The clusters with the highest value of information entries would be considered to suffer from the LDDoS attack. The two-step group investigation technique breaks up the TCP traffic on an enormous time scale, which can result in a high misleading positive rate. Therefore, another technique is proposed to parse TCP traffic and reduce spurious benefits. From the viewpoint of modest scale, TCP traffic quickly fell and subsequently recuperated under LDDoS attack. The scope of the TCP traffic is much larger than the typical organization short-term attack, and we split each detection unit in the comparable bunch into many test pieces (TP), and each TP has  $k$  examples. The TP should be greater than the attack time frame  $T$ , so that we can get an overall changing course of the TCP traffic. The means of LDDoS attack identification in a DU are shown below.

- Step 1—Start.
- Step 2—Find the  $S_{max}$  with maximum sample and  $S_{min}$  with minimum sample by the given expression  $TP_i$  ( $i = 1, 2, \dots, k$ ), let the  $range(i) = S_{max} - S_{min}$ .
- Step 3—If the  $range(i) > \Omega_1$ , the  $TP_i$  is defined as the abnormal TP (ATP).
- Step 4—Let  $TPR = \frac{count(ATP)}{k}$ , if  $TPR > \Omega_2$ , DU might suffer from LDDoS attacks.
- Step 5—End.

The edge esteem  $\Omega_1$  is acquired from the preparation information consisting of numerous TPs from ordinary organizations traffic. Working out the reach in each TP, the  $\Omega_1$  is determined as Equation (5),  $mean(range)$  is the typical worth of reaches,  $Std(range)$  is the standard deviation of reaches, and  $z$  is the consistent related with identification exactness. The train information is isolated into numerous DUs, ascertaining the TPR in each DU, and the limit esteem  $\Omega_2$  is determined as (6),  $mean(TPR)$  depends on the normal worth of TPRs,  $Std(TPR)$  depends on the standard deviation of TPRs, and the worth of  $z$  is equivalent to (5).

$$\Omega_1 = Mean(range) + z \times Std(range) \quad (6)$$

$$\Omega_2 = \text{Mean}(T P R) + z \times \text{Std}(T P R) \quad (7)$$

Figure 4 shows the proper flow chart of the proposed mechanism to detect LDDoS attacks through two-step clustering.

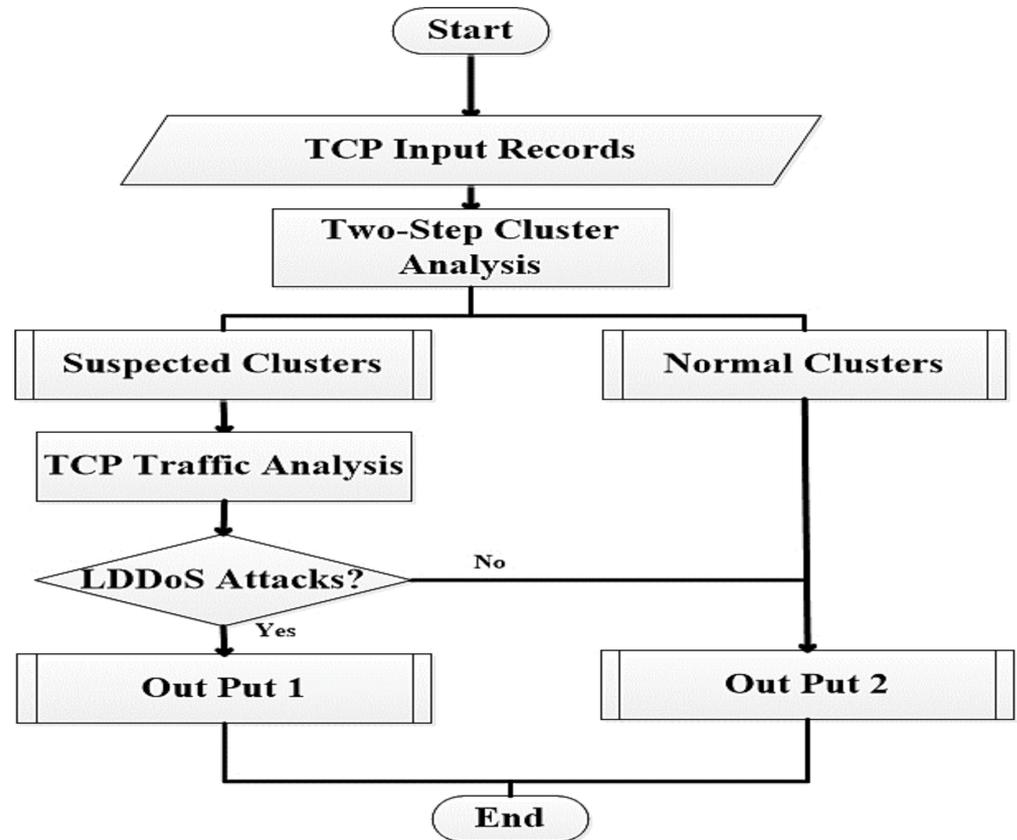


Figure 4. Flow process diagram of LDDoS detection.

### 3.3. Algorithm Implementation

To test the proposed mechanism for detecting outliers in a TCP traffic flow, which is usually caused by the weakness of the TCP congestion control mechanism, we deployed a test case scenario in NS-2 and used a trademark dataset, such as the WIDE and LBNL (United States Department of Energy, Washington, DC, USA) datasets. For implementation, we used the Python programming language, with pycharm as an IDE, in Kali Linux. We used scapy library of Python and Wireshark to detect and analyze data traffic of TCP flow.

#### NS-2 Experiment

The organization's geography is planned in NS-2, as shown in Figure 5. There are three switches in the network geography, and the connection (somewhere in the range of routers 2 and 3) is the bottleneck interface; the data transmission is 10 Mbps, and the delay time of the network is 30 ms. Aside from this, the transmission speed of each connection is 100 Mbps, and the delay time of the network is 15 ms. There are 25 legitimate TCP joins in the geography, and ten of them are set to produce foundation traffic. The attack streams are occasionally sent off by the connection associated with router 1, which focuses on the bottleneck interface.

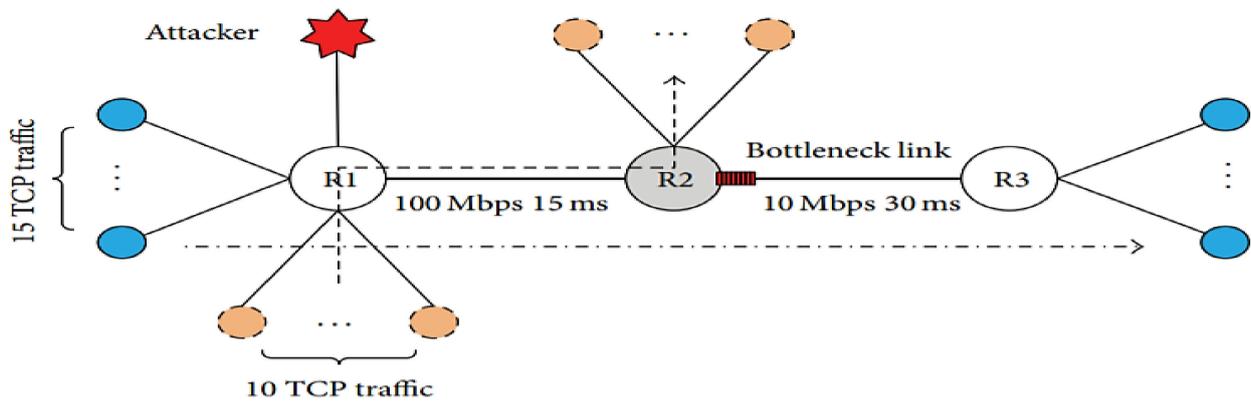


Figure 5. Network topology in NS-2.

As shown in Figure 6, the three routers were set up with multiple additional nodes in the network. Each node has been labeled with a different color for better understanding. From the left side, it shows TCP traffic near the attacker in red color, while the traffic is shown in blue color from both the leftmost and rightmost sides. The orange nodes appear, denoting the 10-value TCP traffic connected to router R1. Internet speed is shown from router R1 to router R<sub>2</sub>, with megabits per second in the range of 100 Mbps, within the 15 milliseconds time limit. Similarly, the orange color of the additional traffic is also shown with router R<sub>2</sub>. Now, from router R<sub>2</sub> to router R<sub>3</sub>, the speed is shown (10 Mbps). Another term is also highlighted here, which is the bottleneck line—this is where the network gets congested and speed is minimized, due to the bottleneck connection. Finally, TCP traffic is shown at router R<sub>3</sub>, which is also 15 traffic connections of router R<sub>1</sub>.

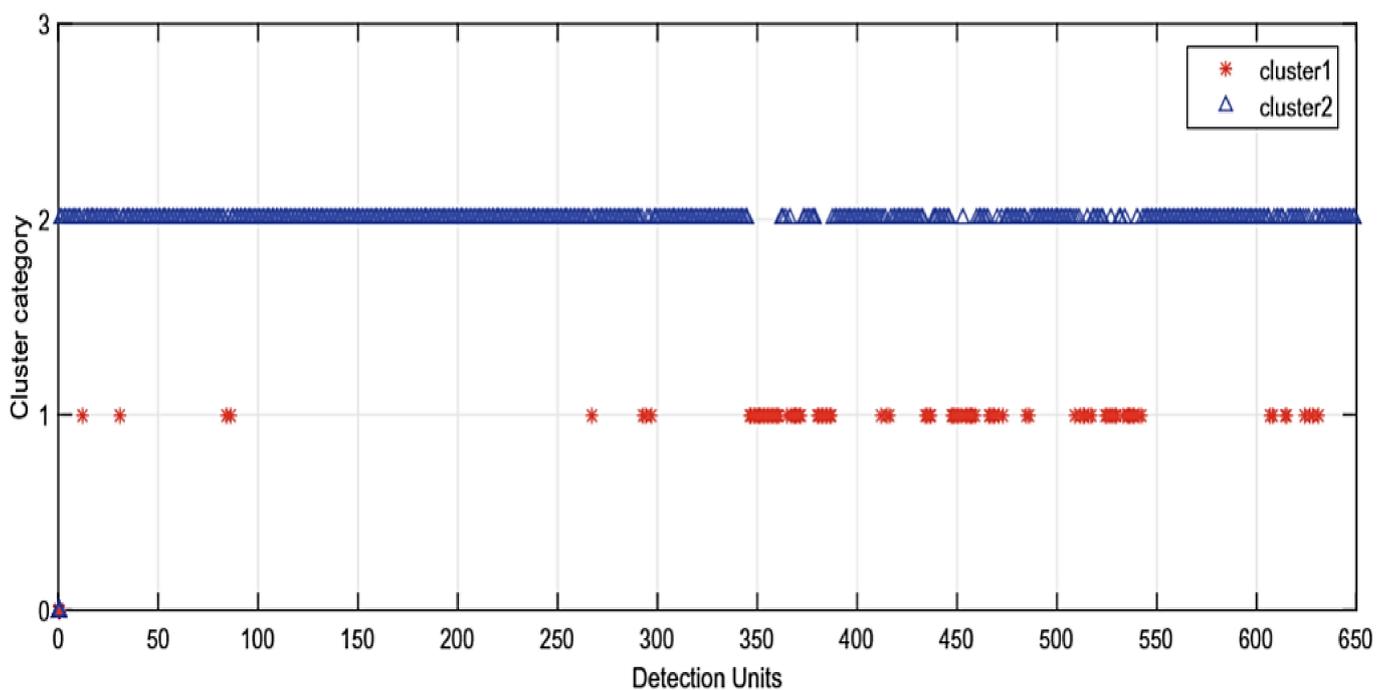


Figure 6. Cluster results of LBNL dataset.

Table 1 is presented with different values, in which the total range of the experiments from 0.2 to 0.4, and the time of the attacks, burst length of the attack, and burst rate of the attack are highlighted, starting from experiment 1, in which the attack period T was 0.2 s up to 0.4 s. Now, in experiment 1, the burst length of the attack was 0.3 s, and the burst rate

of the attack speed is shown as 20 Mbps. Similarly, in experiment 2, the attack time T was 3 s. Attack burst length was 0 s to 0.7 s. Finally, the burst rate of the attack was 20 Mbps. At last, with experiment 3, the attack time was 3 s, and the attack burst length was 0.4 s. At last, the burst rate of the attack started from 5 Mbps, leading up to 20 Mbps.

**Table 1.** Attack parameters in NS-2.

Test of Experiments	Time of Attack (T)	Burst Length of Attack (L)	Burst Rate of Attack (R)
1	0.2–0.6 s	0.5 s	20 Mbps
2	4 s	0–0.8 s	20 Mbps
3	4 s	0.3 s	5–20 Mbps

Table 2 shows the results in a bunch of clusters [1]. All values are shown with left and right and cluster-1 and cluster-2. It has been observed that the attack time was much smaller in length than the burst attack time and length.

**Table 2.** The cluster results.

Experiments	Attack Time	Length	Attack Burst Rate	Cluster 1	Cluster 2
1	0.2–0.6 s 0.5–0.7 s	0.5 s	20 Mbps	DU <sub>1</sub> –DU <sub>11</sub> , DU <sub>13</sub> –DU <sub>16</sub> DU <sub>2</sub> –DU <sub>5</sub> , DU <sub>13</sub> –DU <sub>16</sub> DU <sub>1</sub> –DU <sub>5</sub> , DU <sub>12</sub> –DU <sub>16</sub>	DU <sub>12</sub> DU <sub>1</sub> , DU <sub>6</sub> –DU <sub>12</sub> DU <sub>6</sub> –DU <sub>11</sub>
2	4 s	1 s 0.3 s 0.3–0.8 s	20 Mbps	DU <sub>2</sub> –DU <sub>16</sub> DU <sub>1</sub> –DU <sub>5</sub> , DU <sub>12</sub> –DU <sub>16</sub> DU <sub>2</sub> –DU <sub>5</sub> , DU <sub>12</sub> –DU <sub>16</sub>	DU <sub>1</sub> DU <sub>6</sub> –DU <sub>11</sub> DU <sub>1</sub> , DU <sub>6</sub> –DU <sub>11</sub>
3	4 s	0.3 s	5–10 Mbps 15–30 Mbps	DU <sub>2</sub> –DU <sub>5</sub> , DU <sub>12</sub> –DU <sub>16</sub> DU <sub>1</sub> –DU <sub>5</sub> , DU <sub>12</sub> –DU <sub>16</sub>	DU <sub>1</sub> , DU <sub>6</sub> –DU <sub>11</sub> DU <sub>6</sub> –DU <sub>11</sub>

In experiment 2, an ordinary organization is when L is equivalent to nothing. As shown in Table 2, input records are bunched into two gatherings; the TCP traffic from LDDoS attacks can be effectively grouped from ordinary organization traffic by a two-venture bunch examination technique. Nonetheless, ordinary detection units (DUs), such as DU<sub>1</sub> and DU<sub>12</sub>, are wrongly credited to the gone-after DUs. In the DU<sub>1</sub>, the source generating end of TCP attempts to establish associations with the objective end. The TCP traffic is in a course of development; it has extra unsound for DU<sub>1</sub> than other DUs—so does the DU<sub>12</sub>, which is currently recuperating the TCP traffic from attack.

It is a typical event in that numerous TCP associations begin to interface up with the objective end consistently. For decreasing misleading up-sides, it is important to identify the DUs once again, which are thought to be affected by LDDoS attacks.

The bits of traffic of TCP investigation technique are suggested, in order to distinguish the LDDoS attacks in modest scope. The train information was acquired from the ordinary organizational climate because of the NS-2 stage and continued for 800 s. The test TP was fixed as 2 s, from which, we can obtain the limit  $\Omega_1 = 15.40$ ,  $\Omega_2 = 0.46$ , and steady worth  $z = 2.58$  by the train information. We have tested the DUs of the group (bunch 2) by the TP examination technique. Table 3 displays the outcomes, in light of the bits of traffic of the TCP investigation strategy. It can be witnessed from Table 3 [1] that the DU<sub>1</sub> is taken out from the thought group, and the other DUs that are affected by LDDoS attacks remain. Therefore, here, we do explore different avenues, in terms of a different times and burst rates; instead, we go after rates, so we have changes in location units concerning APR.

**Table 3.** Result on the base of pieces of TCP traffic analysis.

Experiments	Time	Length	Rate	APR < $\Omega_2$	APR > $\Omega_2$
1	0.2 s	0.5 s	20 Mbps	—	DU <sub>12</sub>
	0.3 s			—	DU <sub>12</sub>
	0.4 s			DU <sub>1</sub> , DU <sub>7</sub>	DU <sub>6</sub> , DU <sub>8</sub> –DU <sub>12</sub>
	0.5–0.6 s			DU <sub>1</sub> , DU <sub>12</sub>	DU <sub>6</sub> –DU <sub>11</sub>
	0.7–6 s			—	DU <sub>6</sub> –DU <sub>11</sub>
2	4 s	0 s	20 Mbps	DU <sub>1</sub>	—
		0.2 s		—	DU <sub>6</sub> –DU <sub>11</sub>
		0.3 s		DU <sub>1</sub>	DU <sub>6</sub> –DU <sub>11</sub>
		0.8 s		DU <sub>1</sub>	DU <sub>6</sub> –DU <sub>11</sub>
3	4 s	0.4 s	5–10 Mbps	DU <sub>1</sub>	DU <sub>6</sub> –DU <sub>11</sub>
			15–30 Mbps	—	DU <sub>6</sub> –DU <sub>11</sub>

#### 4. Algorithm Evaluation

This section focuses on the discussion related to the performance of the proposed algorithm. We evaluated the proposed algorithm with two public datasets WIDE and LBNL. Ns-2 is the second version of the network simulator series, which is very effective and the trademark software for analyzing data traffic that is specially designed for defending against cyber security and vulnerabilities. Both the WIDE and LBNL datasets are publicly available [15]. The WIDE data traffic repository is maintained by the MAWI working group of the WIDE project. The LBNL dataset is from the Lawrence Berkley National Laboratory. Wireshark is a packet analysis used to analyze packets coming through the TCP congestion control mechanism and after efficient mechanism to detect outliers in the form of low-rate denial-of-service attacks.

##### 4.1. Experiments on Public Dataset LBNL

We examined the dataset LBNL that comes from the Lawrence Berkeley National Laboratory. Figure 6, as shown in the LBNL dataset, exhibits that the procedure has a low, deceptive rate of positives. Data from the test of LBNL continued north of 18 h, where no LDDoS attacks happened. We have set the assessing time  $st = 0.1$  s, ID unit  $DU = 100$  s, and  $TP = 2$  s. So, we have 649 DUs through and through, and each DU has 50 TP's. Figures 7 and 8 show the outcomes of the assessment, considering the two-adventure bundle examination strategy. There are 94 DUs (bundle 1) that are believed to be affected by LDDoS attacks. Then, we separate the traffic of TCP pieces of the 94 DUs.

The train information is removed from another LBNL dataset that is in an ordinary organization setting, and the train information continues for 600 s. We achieved the boundaries through the train information, with  $z = 1.8489$ ,  $\Omega_1 = 105.95$ , and  $\Omega_2 = 0.88$ . The outcomes, given the TCP traffic technique breakdown bits, are shown in Figure 8. At last, we obtained 16 misleading up-sides after the two-venture bunch examination strategy and strange pieces investigation technique; the false positive rate was 1.8489%.

The TCP traffic of the DUs that are distinguished to be gone after is displayed in Figure 8. We accepted the succeeding 2 DUs as specific illustrations. In a modest scope, the traffic of TCP alternates quickly, as well as how much vacillation of TCP traffic is similar to the traffic that is gone after by LDDoS attacks. At last, we get 12 bogus up-sides after the two-step cluster technique and unusual pieces examination strategy; the misleading positive rate was 1.8489%.

From the dataset of LBNL, with the two-steps clustering mechanism's analysis for detection of LDDoS attacks, the attacks had a lower false positive than the AEWMA (adoptive exponentially weighted moving average) method, which shows the proposed mechanism is more prominent than the traditional AEWMA algorithm analysis, 2.7734%; the outcomes are revealed in Table 4.

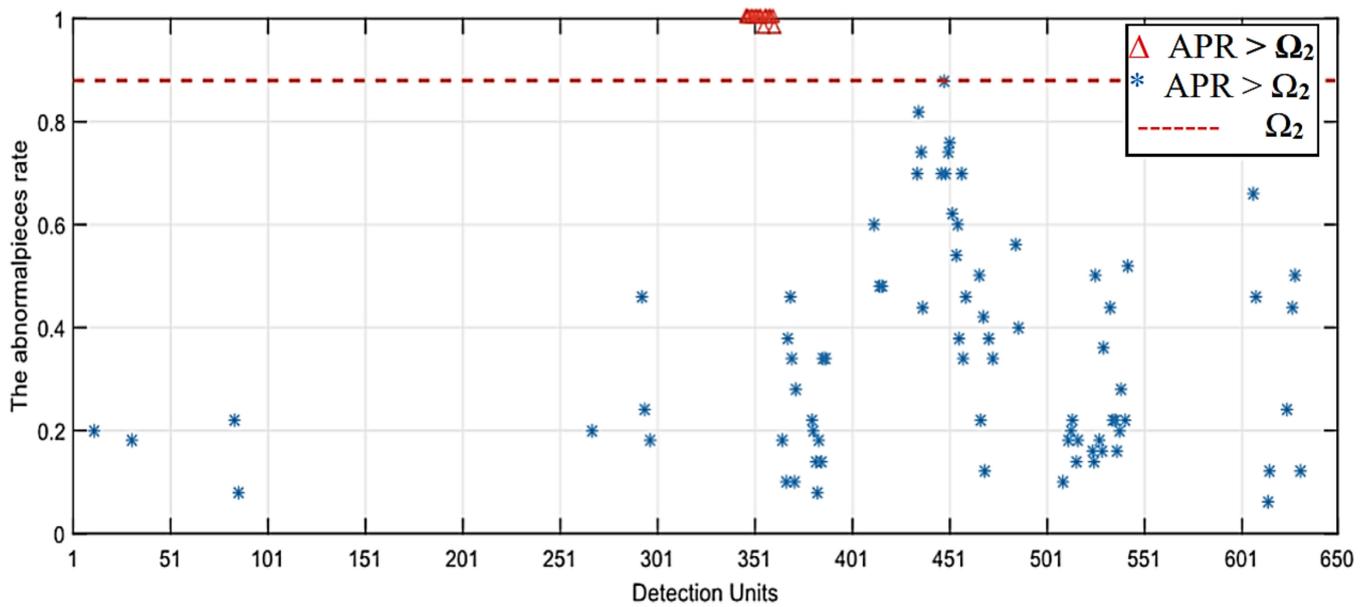


Figure 7. Results based on the abnormal pieces analysis method.

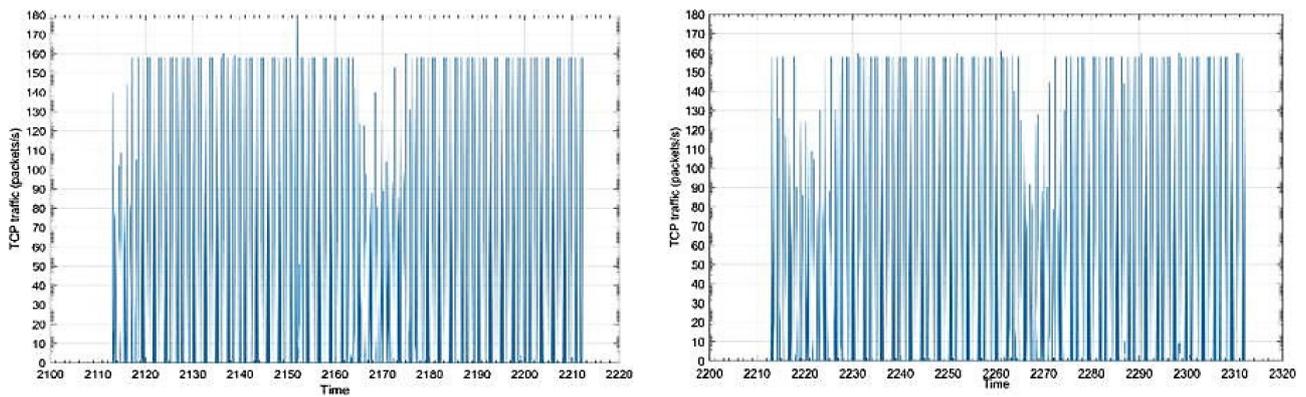


Figure 8. The TCP traffic of the false positive detection units.

Table 4. The comparison of the proposed techniques.

Methods of Detection	Detection Units	The FP	FP Rate
Two-step’s clustering Analysis	649	12	1.8489%
The AEWMA Analysis	649	18	2.7734%

4.2. Experiments on Public Dataset WIDE

With the WIDE dataset, we tested the dataset that accumulated 10 weeks of association traffic in 2018. Monday’s network traffic was recycled as examination data and occurred for 150 min. We set the looking at time  $t = 0.1$  s, recognizable proof unit  $DU = 100$  s, and  $TP = 2$  s; then, we obtained 90 DUs through and through, and every DU had 50 TP’s. Figure 9 displays the outcomes of the examination, considering the two-adventure pack assessment system. There are 14 DUs (bundle 1) that are believed to be affected by LDDoS attacks. Then, at that point, we explored the traffic of TCP pieces of the 14 DUs. For Tuesday’s association, the traffic was utilized to train the data, in which there were no LDDoS attacks that happened. We obtained the limits by the data to train as:  $z = 2.58$ ,  $\Omega_1 = 4540.29$ , and  $\Omega_2 = 0.38234$ . The outcome took the examining pieces of traffic of TCP technique into account; they are revealed in Figure 10. Lastly, we obtained five misdirecting up-sides, and the counterfeit positive rate was 5.56%.

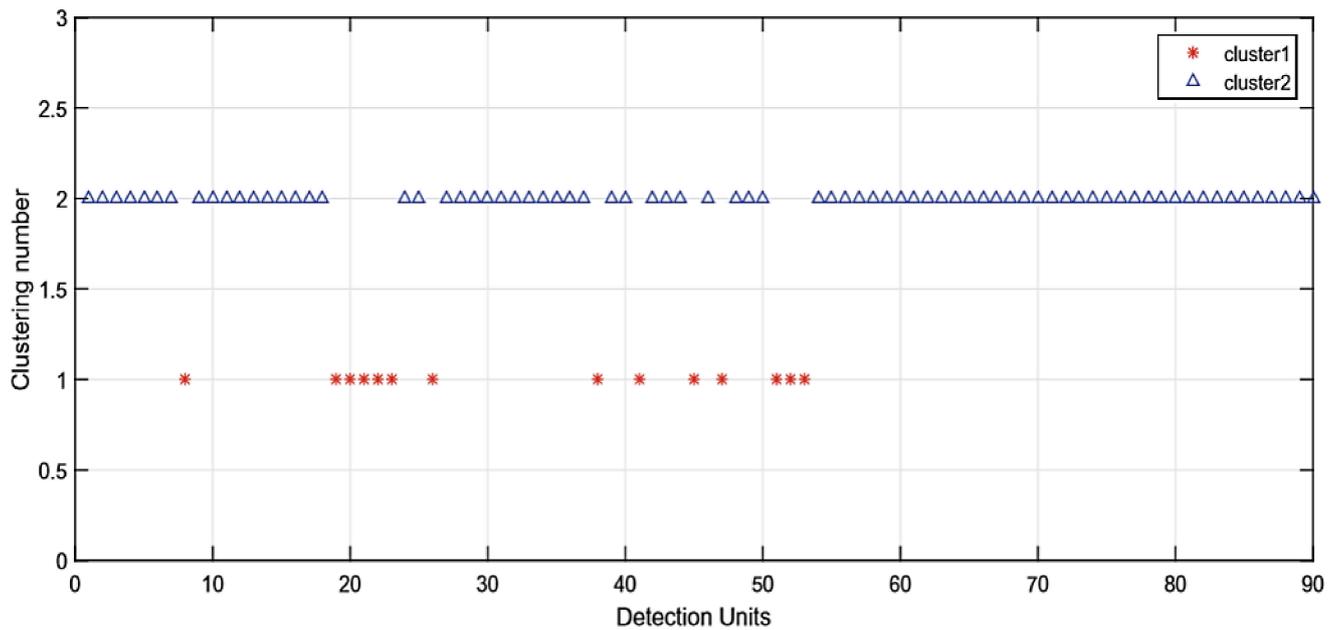


Figure 9. Cluster results of the WIDE dataset.

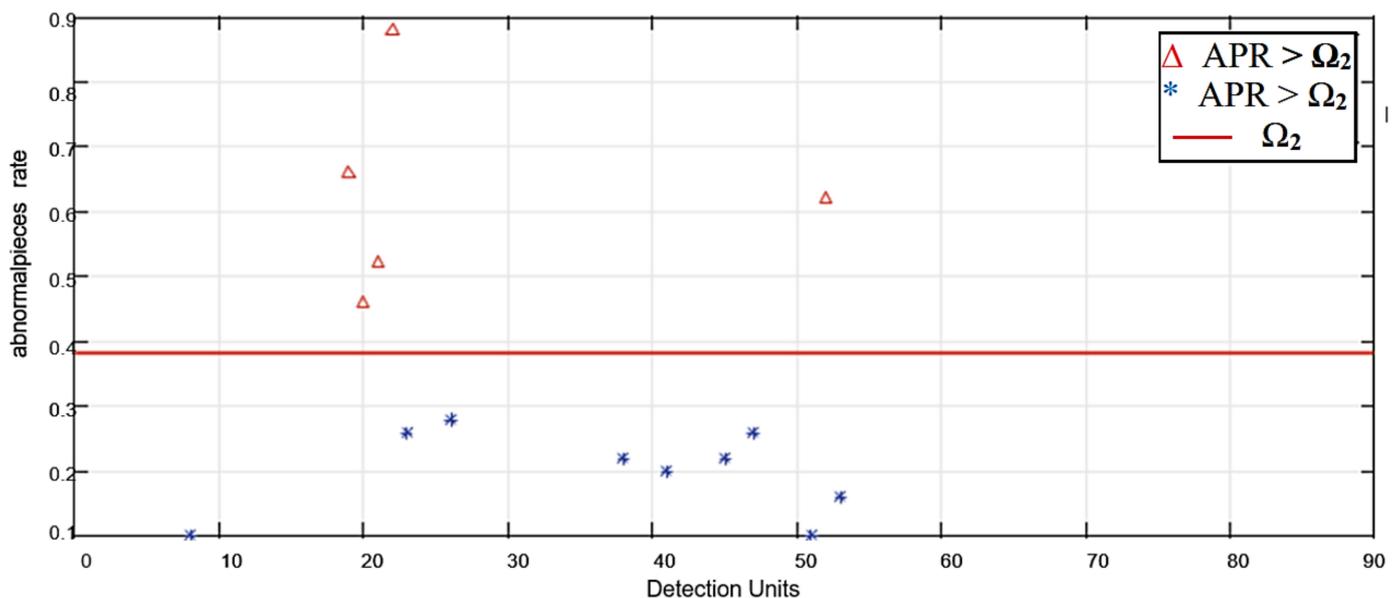


Figure 10. Result based on abnormal piece analysis.

The traffic of TCP, with false positive units 21 and 18, is revealed in Figure 11. The 21st DU contains the traffic of TCP of 2000 to 2100 s, and the 22nd DU contains traffic of TCP of 2100 to 2200 s. Since the traffic of TCP in the 21st DU proceeds to change and TCP traffic is discrete, it was parceled into questionable bundles (bunch 1). The TCP traffic in the 21st DU varied rapidly; the instability of the TCP traffic was similar to the traffic that was pursued by LDDoS attacks. The 22nd DU resembles 21st DU, they both are recognized as affected by LDDoS attacks. The TCP traffic in 21st DU proceeds to differ, and the traffic of TCP is separate; it was isolated into questionable gatherings (bundle 1). The traffic of TCP in 21st DU alterations quickly; the difference in the traffic of TCP was similar to the traffic that was pursued by LDDoS attacks. The 22nd DU resembles the 21st DU; they are both recognized as affected by LDDoS attacks. In our preliminary LBNL dataset, we obtained 90 DUs out and out, and each DU has 50 TPs. The 14 detection units are thought to have

experienced low-rate scattered repudiation of the organization’s attacks, following our proposed two-phase gathering estimations. The TCP traffic in the 21st DU changed rapidly; the difference in TCP traffic was similar to the traffic that was pursued by LDDoS attacks. The 22nd DU resembled the 21st DU; they are both perceived as to have suffered LDDoS attacks. Figure 12 shows the packet analysis of TCP.

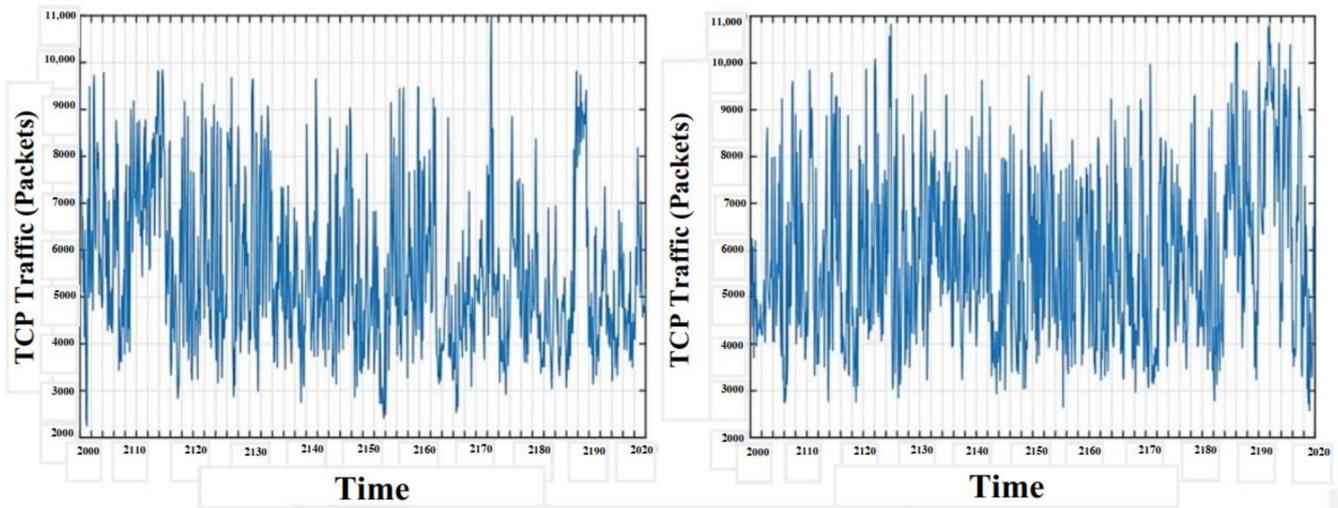


Figure 11. TCP traffic of the false positive detection units.

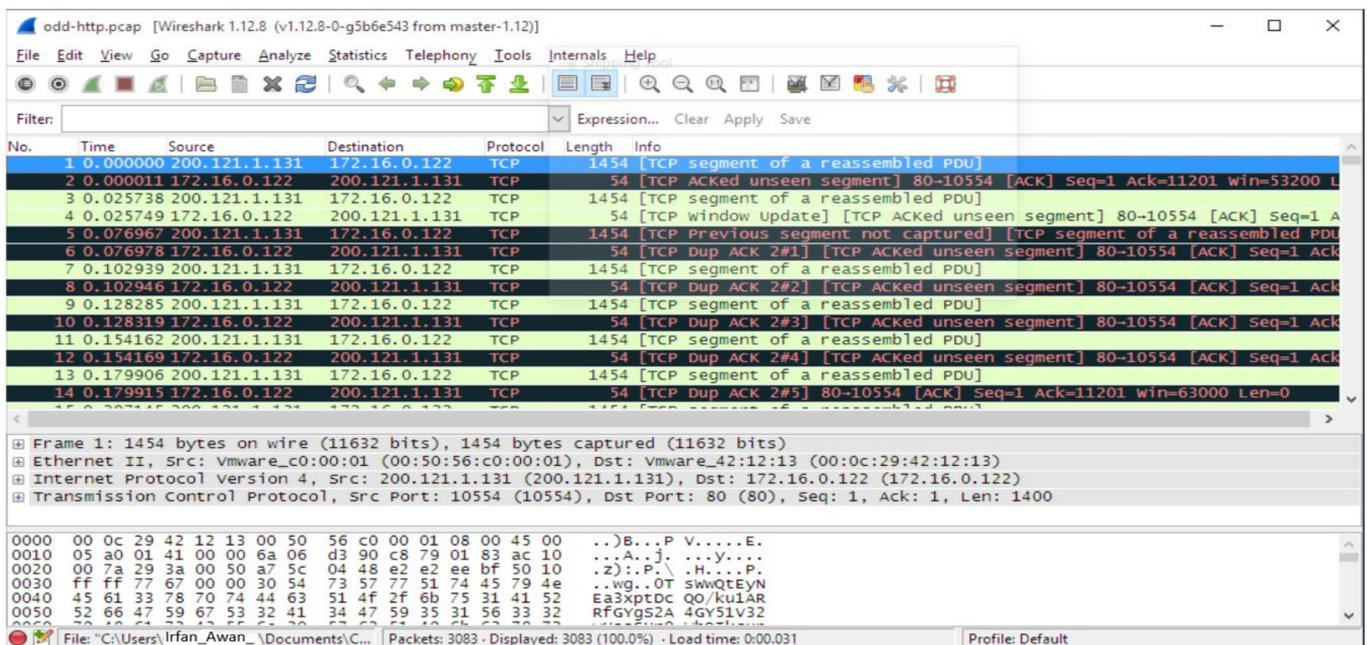


Figure 12. Packet analysis of Wireshark on Kali Linux.

### 5. Conclusions

In this paper, a two-step, clustering-based detection mechanism has been proposed to detect LDDoS attacks. The methodology was performed from TCP traffic with normal and abnormal procedures. From this viewpoint, the proposed method has proven to be efficient, in contrast to other privacy and security detection mechanisms. The analysis was performed with TCP traffic flowing normally. The results were performed by using the network simulator version 2 of NS-2. Two datasets were used, i.e., the LBNL and wide publicly available datasets. From that, it has been revealed that the proposed method produced a low rate of false positives. This research obtained its desired result, as compared

to previously proposed methods, such as EWMA and AEWMA. We benchmarked our work with previous EWMA and proved that our proposed mechanism works efficiently and has comparatively very low false positives, with a quick response detection of outliers of LDDoS attacks in a network.

During the experimental work, we observed that, with a better variation in the threshold and collaborative distance measures, we can further achieve a lower false-positive rate; by getting more detail on the TCP incoming header, we can detect the outliers of LDDoS attacks faster.

**Author Contributions:** Conceptualization, T.H.; funding acquisition, M.I.S. and I.U.K.; investigation and methodology, T.H., S.S.A. and N.A.; project administration, I.U.K.; resources, T.H., N.A. and S.S.A.; supervision, I.U.K., N.A. and S.S.A.; writing of the original draft, T.H. and M.I.S.; writing of the review and editing, T.H. and I.U.K.; software, M.I.S. and T.H.; validation, T.H. and I.U.K.; formal analysis, M.I.S., I.U.K. and S.S.A.; visualization, T.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** We would like to thank the SAUDI ARAMCO cybersecurity chair for funding this project.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** T.H., N.A. and I.U.K authors contributed equally to this work and are first co-authors.

**Conflicts of Interest:** The authors declared that they have no conflict of interest.

## References

1. Wu, Z.; Zhang, L.; Yue, M. Low-rate DoS attacks detection based on network multifractal. *IEEE Trans. Dependable Secur. Comput.* **2015**, *13*, 559–567. [CrossRef]
2. Wu, Z.; Wang, M.; Yan, C.; Yue, M. Low-rate DoS attack flows filtering based on frequency spectral analysis. *China Commun.* **2017**, *14*, 98–112.
3. Tang, D.; Chen, K.; Chen, X.; Liu, H.; Li, X. Adaptive EWMA Method based on abnormal network traffic for LDoS attacks. *Math. Probl. Eng.* **2014**, *2014*, 496376. [CrossRef]
4. Chen, H.; Meng, C.; Shan, Z.; Fu, Z.; Bhargava, B.K. A novel Low-rate Denial of Service attack detection approach in ZigBee wireless sensor network by combining Hilbert-Huang Transformation and Trust Evaluation. *IEEE Access* **2019**, *7*, 32853–32866. [CrossRef]
5. Zhou, L.; Liao, M.; Yuan, C.; Zhang, H. Low-rate DDoS attack detection using expectation of packet size. *Secur. Commun. Netw.* **2017**, *2017*, 3691629. [CrossRef]
6. Bhuyan, M.H.; Elmroth, E. Multi-scale low-rate DDoS attack detection using the generalized total variation metric. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 1040–1047.
7. 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.
8. Kuzmanovic, A.; Knightly, E.W. Low-rate TCP-targeted denial of service attacks and counter strategies. *IEEE/Acm Trans. Netw.* **2006**, *14*, 683–696.
9. Two Step Cluster Algorithm. Available online: [https://www.ibm.com/support/knowledgecenter/en/SSLVMB\\_24.0.0/spss/base/idh\\_twostep\\_main.html](https://www.ibm.com/support/knowledgecenter/en/SSLVMB_24.0.0/spss/base/idh_twostep_main.html) (accessed on 6 June 2022).
10. Rahman, M.U.; Rahman, Z.U.; Fayaz, M.; Abbas, S.; ShahSani, R.K. Performance analysis of tcp/aqm under low-rate denial-of-service attacks. In Proceedings of the 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–27 August 2016; Volume 3, pp. 1–5.
11. Toklu, S.; Şimşek, M. Two-layer approach for mixed high-rate and low-rate distributed denial of service (DDoS) attack detection and filtering. *Arab. J. Sci. Eng.* **2018**, *43*, 7923–7931. [CrossRef]
12. WIDE Project Datasets. Available online: <http://mawi.wide.ad.jp/mawi/> (accessed on 6 June 2022).
13. Silva, A.; Pontes, E.; Zhou, F.; Guelf, A.; Kofuji, S. PRBS/EWMA based model for predicting burst attacks (Brute Froce, DoS) in computer networks. In Proceedings of the Ninth International Conference on Digital Information Management (ICDIM 2014), Phitsanulok, Thailand, 29 September 2014–1 October 2014; pp. 194–200.

14. Wankhede, S.; Kshirsagar, D. DoS attack detection using machine learning and neural network. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBE), Pune, India, 16–18 August 2018; pp. 1–5.
15. Zhang, Y.; Shi, Y. A Slow Rate Denial-of-Service Attack Against HTTP/2. In Proceedings of the 2018 IEEE 4th International Conference on Computer and Communications (ICCC), Chengdu, China, 7–10 December 2018; pp. 1388–1391.
16. Alzaharani, S.; Hong, L. Detection of distributed denial of service (DDoS) attacks using artificial intelligence on cloud. In Proceedings of the 2018 IEEE World Congress on Services (SERVICES), San Francisco, CA, USA, 2–7 July 2018; pp. 35–36.
17. LBNL/ICSI Enterprise Tracing Project. Available online: <http://www.icir.org/enterprise-tracing/> (accessed on 6 June 2022).
18. Bhosale, K.S.; Nenova, M.; Iliev, G. The distributed denial of service attacks (DDoS) prevention mechanisms on application layer. In Proceedings of the 2017 13th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), Nis, Serbia, 18–20 October 2017; pp. 136–139.
19. 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](#)]
20. Zhou, W.; Jia, W.; Wen, S.; Xiang, Y.; Zhou, W. Detection and defense of application-layer DDoS attacks in backbone web traffic. *Future Gener. Comput. Syst.* **2014**, *38*, 36–46. [[CrossRef](#)]
21. Aladi, J.H.; Wagner, C.; Garibaldi, J.M. A simplified method of FOU design utilizing simulated annealing. In Proceedings of the 2015 IEEE International Conference on Systems, Man, and Cybernetics, Hong Kong, China, 9–12 October 2015; pp. 2255–2261.
22. Zhi-Jun, W.; Hai-Tao, Z.; Ming-Hua, W.; Bao-Song, P. MSABMS-based approach of detecting LDoS attack. *Comput. Secur.* **2012**, *31*, 402–417.
23. McGregory, S. Preparing for the next DDoS attack. *Netw. Secur.* **2013**, *2013*, 5–6. [[CrossRef](#)]
24. Bertino, E.; Islam, N. Botnets and internet of things security. *Computer* **2017**, *50*, 76–79. [[CrossRef](#)]
25. Yevsieieva, O.; Helalat, S.M. Analysis of the impact of the slow HTTP DOS and DDOS attacks on the cloud environment. In Proceedings of the 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkov, Ukraine, 10–13 October 2017; pp. 519–523.
26. Xiao, L.; Wan, X.; Lu, X.; Zhang, Y.; Wu, D. IoT security techniques based on machine learning: How do IoT devices use AI to enhance security? *IEEE Signal Processing Mag.* **2018**, *35*, 41–49. [[CrossRef](#)]
27. Shinde, P.J.; Chatterjee, M. A Novel Approach for Classification and Detection of DOS Attacks. In Proceedings of the 2018 International Conference on Smart City and Emerging Technology (ICSCET), Mumbai, India, 5 January 2018; pp. 1–6.
28. Park, T.; Cho, D.; Kim, H. An effective classification for DoS attacks in wireless sensor networks. In Proceedings of the 2018 Tenth international conference on ubiquitous and future networks (ICUFN), Prague, Czech Republic, 3–6 July 2018; pp. 689–692.
29. Kajwadkar, S.; Jain, V.K. A novel algorithm for DoS and DDoS attack detection in Internet of things. In Proceedings of the 2018 Conference on Information and Communication Technology (CICT), Jabalpur, India, 26–28 October 2018; pp. 1–4.
30. Maza, S.; Touahria, M. Feature selection for intrusion detection using new multi-objective estimation of distribution algorithms. *Appl. Intell.* **2019**, *49*, 4237–4257. [[CrossRef](#)]
31. Pu, C.; Song, T. Hatchetman attack: A denial of service attack against routing in low power and lossy networks. In Proceedings of the 2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), Shanghai, China, 22–24 June 2018; pp. 12–17.
32. Wu, X.; Tang, D.; Tang, L.; Man, J.; Zhan, S.; Liu, Q. A low-rate dos attack detection method based on hilbert spectrum and correlation. In Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Guangzhou, China, 8–12 October 2018; pp. 1358–1363.