Neural Computing and Applications (2024) 36:1–14
https://doi.org/10.1007/s00521-021-06389-6

S.I.: IMPROVING HEALTHCARE OUTCOMES USING MULTIMEDIA BIG DATA ANALYTICS

A statistical approach to secure health care services from DDoS attacks during COVID-19 pandemic

Zhili Zhou[1] · Akshat Gaurav[2] · B. B. Gupta[3,4,5] · Hedi Hamdi[6] · Nadia Nedjah[7]

Received: 11 May 2021 / Accepted: 26 July 2021 / Published online: 7 September 2021

Abstract
Over the course of this year, more than a billion people have been afflicted by the COVID-19 outbreak. As long as individuals maintain their social distance, they should all be secure at this period. Because of this, there has been a rise in the usage of different online technologies, but at the same time, there has also been a rise in the likelihood of different cyber-attacks. A DDoS assault, the most prevalent and deadly of them all, impairs an online resource for its users. Thus, in this paper, we have proposed a filtering approach that can work efficiently in the COVID-19 scenario and detect the DDoS attack. We base our proposed approach on statistical methods like packet score and entropy variation for the identification of DDoS attack traffic. We have implemented our proposed approach on Omnet++ and for testing its efficiency we have checked it with different test cases. Our proposed approach detects the DDoS attack traffic with 96% accuracy and can also clearly have differentiated the DDoS attack traffic from the flash crowd.

Keywords DDoS attack · Flash crowd · COVID-19 · Health care · Packet score · Clustering · Entropy

1 Introduction

COVID-19 is one of the biggest global crises, which caused approximately 43024 deaths [1] and forced millions of people from different countries to remain at their home [2]. COVID-19 primarily affects human life, but it also leads to a secondary thread in the IT industry. The main aim of the government and news agencies in this COVID-19 pandemic is to provide correct information to its citizens. Any wrong or delay in information may lead to a panic situation. Attackers use DDoS attacks [3] to affect the working of the government and news agencies to create panic. For example, the online portal of the Australian government (myGov) is down due to DDoS attacks, which creates a panic state among its citizens.

✉ B. B. Gupta
bbgupta@nitkkr.ac.in

Zhili Zhou
zhou_zhili@163.com

Akshat Gaurav
akshat.gaurav@ronininstitute.org

Hedi Hamdi
hhamdi@ju.edu.sa

Nadia Nedjah
nadia@eng.uerj.br

[1] Engineering Research Center of Digital Forensics, Ministry of Education, School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

[2] Ronin Institute, Montclair, New Jersey 07043, USA

[3] National Institute of Technology Kurukshetra, Haryana, India

[4] Asia University, 500, Lioufeng Rd, Wufeng 41354, Taichung, Taiwan

[5] Macquarie University, Sydney, NSW 2109, Australia

[6] Department of Computer Science, College of Computer and Information Sciences, Jouf University, Sakaka, Kingdom of Saudi Arabia

[7] State University of Rio de Janeiro, Rio de Janeiro, Brazil

Nowadays healthcare systems are mostly digitized and depend on information and technology. It helps doctors, nurses, and other researchers to understand different issues related to the patients quickly and efficiently. In COVID-19 time, attackers attacked these healthcare organizations so that they are not able to help people in this pandemic situation. For example, the United States HHS (Department of Human Health) faced DDoS attacks from an unknown source, hospitals in France, and the Czech Republic, which were working on the development of the COVID-19 vaccine were hit by DDoS attacks.

COVID-19 virus communicates from person to person due to which different governments encourage educational institutes to conduct online classes for students. In online classes, teachers and students interact through videoconferencing apps. Attackers conduct different attacks on these apps so that they are unavailable for their users. German's online education platform (Mebis) was flooded by fake traffic. Due to which students and teachers wear notable exchange information.

1.1 Different healthcare facilities during COVID-19 pandemic

Teleconsultation and digital healthcare play an important role during COVID-19. For this, an online portal has been created, which provides online counselling to the patients, through which the patient can interact with the doctors digitally to reduce the possibility of spreading the COVID-19 virus. Swash [4] is an app through which doctors and patients can interact, around 2000 doctors are registered with this app, and they can provide audio and video counselling to patients. "Test Your Self Goa" [5], this app is another app developed by the Goa Ministry of Health so that users can self-diagnose COVID-19 symptoms. It guides the patient about the correct method of self-quarantine, social disturbances, use of masks, etc. It also provides a list of doctors and hospitals nearby. Using these apps and online portals, COVID-19 patients receive timely counselling and appropriate treatment at home, reducing the burden on hospitals and helping to maintain social distance.

The data collected from COVID-19 patients can be used to predict the future behavior of the virus, and it also helps the patients maintain proper social distance from the infected patient. "Arogay Setu" [6] is an app that collects GPS and Bluetooth data to track a user's location and warns them if they get close to COVID-19 infected patients. Doctors and researchers at King's College London and St Thomas hospitals have developed a "Covid Symptom Tracker" app [7] that collects patient data and uses it for advanced research. The Indian Ministry of Electronics and Information Technology has developed the "COVID-19 Feedback" app [8] which collects data from users and identifies the locations that are most affected by viruses and which require more health facilities.

1.2 DDoS attack and flash crowd background for COVID-19 pandemic

In COVID-19 pandemic, attacks used a large number of compromised machines to generate enormous fake packets which in turn bring down the online healthcare website or app. This type of attack is called a DDoS attack. Attackers choose the DDoS attack [9] during the COVID-19 period because it consumes the network resources very quickly, so a small duration of DDoS attack can make the website or app unavailable for legitimate users. There is a rapid increase in the DDoS attacks in COVID-19 times compared to 2019, and there is a total 90% increase in DDoS attacks in the COVID-19 period compared to the last year. In Q1 of 2020, the average increase in the duration of DDoS attacks is 24% compared to Q1 of 2019. This shows that most of the attackers use DDoS tools to generate attack traffic. There are many tools that are available by which an attacker can start a DDoS attack like Trinoo [10], TFN [11], TFN2K [12], Mstream [13], Shaft [14], Knight [14]. Figure 1 represents the DDoS scenario in this; there are 'N' compromised machines which are represented as Attacker 1, Attacker 2,...., and Attacker N. These compromised machines are generating a large amount of traffic. Due to this, normal users are not benefiting from online services. Researchers have proposed several filtering techniques against DDoS attacks [15–19]. These filtering techniques can be classified into individual filtering techniques[20] and collaborative filtering techniques [20]. The collaborative filtering technique, unlike the individual filtering technique, uses a distributive approach for the identification and filtering of malicious packets. The advantage of using a distributive filtering technique for the detection of attack is that it reduces the space requirements of filters because now different filters shear their information and take decisions accordingly, there is no effect on the detection of malicious packets even if one or two filters crash. However, the main limitation of the collaborative filtering technique is that it cannot differentiate flash crowd [21, 22] from the DDoS attack traffic.

During COVID-19, most of the users are using online resources for checking their health conditions and different filtering methods can wrongly consider this traffic as attack traffic, this scenario is called flash crowd [23]. The flow characteristics of the flash crowd are similar to that of DDoS flow, it also overwhelms the server, which leads to the crash or slow response of the server, the same as that of the DDoS attack. Figure 1 represents the flash crowd scenario, there are 'N' legitimated users which are represented

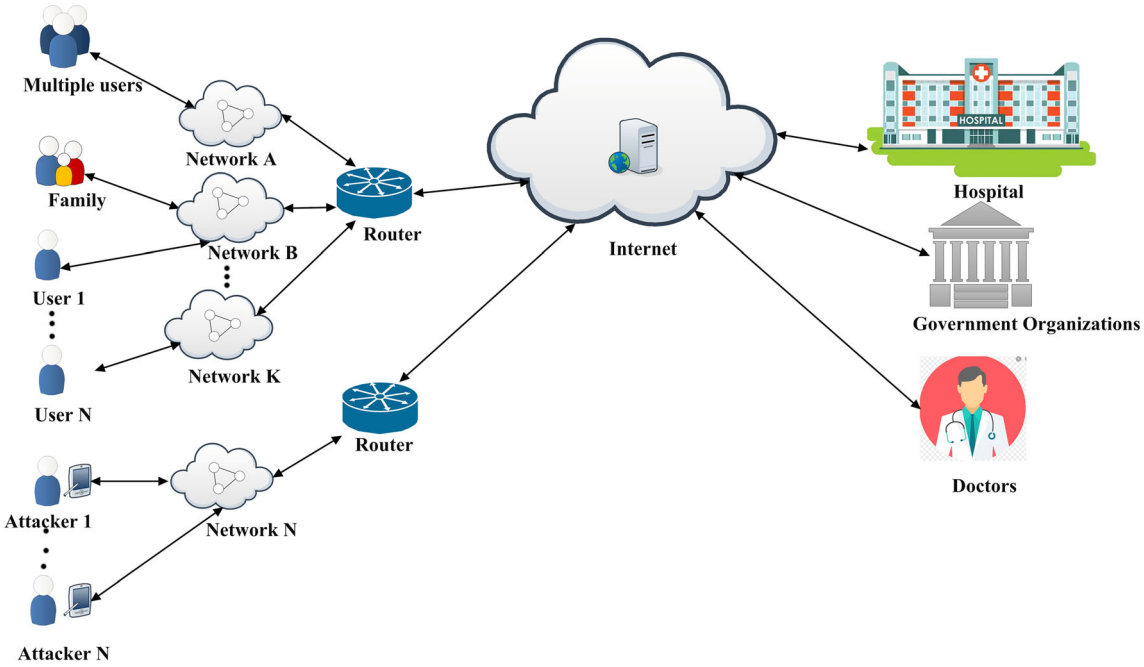

Fig. 1 Online use of resources during COVID-19

by user 1, user 2,..., user N. If all the N users try to access the online resource, then the router can assume the incoming traffic as DDoS traffic and filter it. Thus, it is always beneficial to differentiate DDoS attack traffic from flash crowd traffic. There are some differences between the DDoS attack and flash crowd, which are represented as follows.

(1) The rate at which traffic is generated during flash crowd and the DDoS attack is different. In the flash crowd scenario, the traffic flow depends upon the human interest, so its rate first gradually increases and reaches any peak value, and then it starts decreasing. However, this is not the case with DDoS attack traffic. As the main aim of DDoS attack traffic is to crash the victim, so its traffic flow increases exponentially and reaches its peak value very quickly, and then it decreases at a sharp rate.

(2) During the COVID-19 pandemic, users mostly search local healthcare resources, like nearby doctors or pharmacy stores. Therefore, the source IP addresses of traffic towards healthcare resources are less randomly distributed compared to the source IP addresses during the DDoS attack. As in the DDoS attack scenario, a large number of compromised devices want to access the resource and these devices could be from anywhere. Hence, if we calculate the entropy of source IP addresses, then its value is much higher in the case of the DDoS attack scenario than that in the flash crowd scenario.

The rest of the paper is organized as follows. Related work is represented in Sect. 2. Section 3 presents our proposed work in detail. Section 4 describes our proposed algorithm in detail; then Sect. 5 presents the simulation results. Finally, Sect. 6 presents the conclusion and future work.

2 Related work

After COVID-19 was declared an epidemic, researchers proposed new tools and applications that help people live a normal life. Most of these applications help the user to monitor their health status, such as Swash, Test Your Self Goa, and Aroge Setu. Reference [24] gives a survey of various mobile tracing apps used in the COVID-19 pandemic. The users health status is monitored through these mobile tracing apps and the government can detect locations that are severely affected by COVID-19. Due to COVID-19, there is a transformation of the development of different software products, this change is reviewed by the Reference [25]. Reference [26] presents the security point of view of applications developed in COVID-19 time. Reference [27] proposed a new approach "WeTrace" by which the privacy of the user is not compromised during sharing health-related data on mobile apps. Reference [28] proposed a lightweight authentication protocol that establishes trust between the user and a remote sensor node. Reference [29] proposed a CNN and ConvLSTM-based technique to diagnose COVID-19 from the patient's X-Ray report. However, all these apps are not free from traditional

cyberattacks like DDoS. Thus, we proposed an approach which can detect DDoS attacks efficiently. Our proposed approach uses a proactive and collaborative defense mechanism. We use a proactive mechanism because it continuously analyzes the traffic and if any anomaly is detected, then it raises the alarm. Thus, by using a proactive mechanism, the detection of attack at the early stage is possible. In our proposed approach, we use a proactive scheme named PacketScore which is given by [30], in this scheme the packet score of each incoming packet is calculated, and if the value is less than the predefined threshold, then the packet is discarded. Reference [30] use the PacketScore technique with individual filtering techniques that made that scheme enable us to detect more than one type of attack at the same time, but due to which the memory requirements for storing the packet scores increases. In order to reduce the memory requirement, we use it in a collaborative filtering technique. The collaborative filtering technique has more advantages than an individual filtering technique, and it is also more advantageous to implement a collaborative technique in a real-world scenario. In the past many years, researchers proposed many collaborative filtering techniques, from which some are as follows.

Reference [31] presented one of the earliest collaborative techniques to detect the DDoS attack. This model is named as 'Pushback technique'. According to this model, if any router faces congestion, then it requests the corresponding upstream router to limit its downstream traffic rate. The advantage of this method is that the bandwidths of downstream routers are saved as all malicious packets are filtered by upstream routers and they have to pass only the legitimate packets. However, if the attackers are uniformly distributed on the network, this model may not give the desired results. Reference [32] also proposed a collaborative and proactive technique, named "FireCol" to detect the DDoS attack. This technique forms a virtual ring of IDS (Intrusion Detection Systems) around the victim, and then these IDSs collaborate to detect the attack traffic according to the set of rules. To avail of the services of these IDSs, users must register with their ISP (Internet Service Provider). After registration, each user gets a unique ID, TTL (time of subscription), and supported capacity. Different users can set their detection rules and support capacity according to their requirements, but if the number of users increases, then this technique may overburden the IPS and reduce its response time. Reference [33] has proposed a novel method for the detection of flow table overload type of DDoS attacks in the SDN-based cloud. SNV gives many advantages to cloud computing, but SDN also makes it vulnerable to different types of attacks, such as flow table overload DDoS attacks. To counter this type of DDoS attack, the author firstly used a mathematical model to represent the flow table space, and then uses a novel method for sharing the flow table. However, this approach is only limited to a specific type of DDoS attack (flow table overload DDoS). Reference [17] has proposed a DDoS attack detection approach in cloud computing, but the proposed approach is based on individual filtering and depends on the packet arrival rate. The author proposed entropy-based techniques for DDoS attack detection in IoT environment [34] and VANET environment [35]. However, the proposed approach is based on individual filtering. In order to detect the spoofed packets, researchers proposed many methods [36], but the main limitation of these methods is that they cannot detect a new type of attack, and their detection time and memory usage is high.

All previously defined collaborative techniques are not purely using the distributive approach, either the single ISP controlling the working or the routers are not interacting completely to take the decision. However, [20] proposed a collaborative filtering method, ScoreForCore (SFC). This method uses a distributed approach for the detection of DDoS attacks, for this, each router selects the attribute pairs randomly that define the specific attack type, and then during the attack time, these values of attribute pairs are compared with the attribute pairs deviated during the attack. However, if the router does not have information about the most deviated attribute pairs, then it gets this information from its neighboring routers by broadcasting its request into the network. The main limitation of this model is that it uses a broadcasting technique, which increases the message complexity of the system and reduces the response time of the system. However, the common limitation of all the above techniques is that they cannot efficiently differentiate DDoS attack traffic from the flash crowd. Entropy-based and correlation-based are the techniques used by researchers to differentiate DDoS attack traffic from the flash crowd. Entropy-based methods measure the randomness of the occurrence of incoming packets and, according to the entropy value or the degree of randomness, it is decided whether the traffic is due to flash crowd or due to DDoS attack. Reference [37] uses the entropy-based method to detect different types of attacks in high-speed networks. This technique uses K means clustering. Incoming packets are clustered according to the packet size. Each cluster has a center C and radius R. If the packet size of the incoming packet is within the radius of the cluster, then it belongs to that cluster. However, the main limitation of this method is that it requires frequent training. Reference [38] uses clustering and entropy-based methods to differentiate DDoS attacks from the flash crowd. In this, the entropy of the incoming packet is compared with the entropy of the cluster group that generates it. Then by comparing these entropies, it is decided

that the traffic is due to flash crowd or by DDoS attack. The main limitation of this scheme is that it uses real-time data to train the model and due to the two-layer comparison, the processing speed of the model is reduced. Correlation-based methods analyze the relation among different attributes in the incoming packets. It uses the concept that if the traffic is generated from some artificial source, then there is a similarity among its attributes, which is not present if the traffic is generated by the human source. Many techniques use correlation-based methods for detecting the flash crowd, some of the popular techniques are as follows:

Reference [39] uses the Sibson distance between two incoming traffic, and if this distance is greater than a predefined threshold, then the incoming traffic is marked as attack traffic or flash crowd. However, it cannot detect DDoS attacks and the flash crowd at the same time. Reference [40] uses a probability matrix to differentiate the DDoS attacks from the flash crowd. It calculates the variation coefficient and similarity coefficient of the incoming traffic, and if these coefficients do not fulfill the desired criteria, then the respective flow is discarded. Reference [41] uses the correlation between different arrival rates to differentiate DDoS attacks and FC, for this, it calculates Pearson's correlation coefficient. However, for this method, the probability distribution of incoming traffic should be known, and this is also unable to detect DDoS attacks and flash crowds at the same time. Reference [42] tries improving the process given by [41] by calculating the correlation between the flow of the generated data and its source, as if the data is generated from software-controlled bots, then it shows a high degree of correlation. Therefore, if this correlation is analyzed, then malicious packets can be detected. However, if the attacker uses different software to generate the attack traffic, then this approach does not work. Reference [43] also uses the correlation between the source IP address of the incoming packets and the IP address of the packets whose incoming rate is higher than a predefined threshold value for the identification of attack packets. However, this process is also less effective when attackers frequently change the attack characteristics.

The main limitation of entropy-based methods is that there is no proper mechanism by which malicious packets are discarded. As in entropy-based methods, if the entropy of a set of packets is above the threshold, then all the packets of that set are discarded; hence, its detection rate of these methods is low. The correlation-based methods use the correlation between different attributes in the incoming packets, but if an attacker randomizes the attribute values, then the accuracy of this method is decreased. Hence, both of these methods are unable to differentiate flash crowd from DDoS attack traffic efficiently

3 Proposed approach

The characteristics of the flash crowd are similar to DDoS attacks, but as the flash crowd traffic is generated by legitimate users, filtering this traffic may lead to economic loss or credibility loss to the victim. Thus, in COVID-19 time, we required a method that can efficiently differentiate the flash crowd from DDoS. Many techniques have been developed to differentiate DDoS attacks from the Flash crowd. These techniques use information theory-based methods for detecting the flash crowd scenario. We found that the effectiveness of information theory-based methods is increased if they are used along with static methods like packet score method. Due to the use of the packet score method, analysis of individual packets is possible. Therefore, we focused our research to develop a filtering technique that uses the concept of information theory and statistical methods for the detection of DDoS and differentiating it from flash crowd. The main points of our proposed approach are as follows:

(1) The first router in the super-router combination extracts the selected attributes from the incoming packet; then according to these attributes the score of the packet is calculated.
(2) If the score of the packet is less than the predefined threshold, then the packet is considered as the legitimate packet, and it is forwarded to the next router.
(3) If the packet score is more than the predefined threshold, then the packet is considered the suspicious packet and it is forwarded to the entropy calculation module.
(4) In the entropy calculation module, the entropy for a specific duration is compared with the predefined threshold value, and if the entropy value is less than the threshold value, then the traffic is considered as flash crowd traffic.
(5) If the entropy value is more than the threshold value and if the marker's value in the packet is not set, then at first its marker value is set and then the packet is forwarded to the next router.
(6) But if the packet is in the suspicious category and its marker value is set, then that packet is considered a malicious packet, and it is discarded by the router.
(7) In this approach, both filters work collectively to identify the attack traffic. For sharing the information, they simply use the packet marking technique, in which the suspicious packets are marked by the packet marker, and if the marked packets are received by the packet analyzer, then they are considered malicious packets, and they are filtered by the router.

Figure 2 represents the work flow of the proposed approach. As represented in Figure 2, our proposed approach has three important modules; score calculation module, Threshold calculation module, and entropy calculation module, the details of these modules are given in the succeeding subsections.

3.1 Score calculation module

This module is the most important part of our module, as it is responsible for calculating the score value of the incoming packets. A packet score is calculated for each packet, and depending on the packet score value, the packet is declared as legitimated or malicious. To get more clarity on the method of calculation of packet score, let consider a scenario of a DDoS attack. During the attack, the incoming packets have many attributes, i.e., protocol type, TTL value, TCP flag. Let the attributes represented as A, B, and C and these attributes can take up to 'n' values

$$A = a_1, a_2, a_3....a_n$$
$$B = b_1, b_2, b_3....b_n$$
$$C = c_1, c_2, c_3....c_n$$

The attribute's value in the DDoS attack traffic is constant because most of the attacks are generated by DDoS attack tools. For example, in SQL slammer attack, different attribute values are as follows:

$$\text{protocol type} = UDP$$
$$\text{destination port} = 1434$$
$$\text{packet size} = between\ 371\ and\ 400\ bytes$$

Therefore, the total number of packets with attribute values a_1, b_1 *and* c_1 are:

$$\begin{aligned} P' = {} & P_1(A = a_1, b = b_1, C = c_1) \\ & + P_2(A = a_1, b = b_1, C = c_1) \\ & + P_3(A = a_1, b = b_1, C = c_1)..... \\ & + P_n(A = a_1, b = b_1, C = c_1) \end{aligned} \tag{1}$$

and total number of packets is given by:

$$\begin{aligned} P^t = {} & P_1(A = a_1) + P_2(A = a_2) \\ & + P_3(A = a_3)....P(A = a_n) \\ & + P_1(B = b_1) + P_2(B = b_2) \\ & + P_3(B = b_3)....P(B = b_n) \\ & + P_1(C = c_1) + P_2(C = c_2) \\ & + P_3(C = c_3)....P(C = c_n) \end{aligned} \tag{2}$$

then according to probability theory probability P^L that a packet having attributes $A = a_1,\ B = b_1,\ C = c_1$ is: Eq. 3 reduced as:

$$P^L = \begin{bmatrix} P_1(A = a_1, b = b_1, C = c_1) + P_2(A = a_1, b = b_1, C = c_1) + \\ 0 \quad P_3(A = a_1, b = b_1, C = c_1)..... + P_n(A = a_1, b = b_1, C = c_1) \\ P_1(A = a_1) + P_2(A = a_2) + P_3(A = a_3)....P(A = a_n) + \\ 0 \qquad P_1(B = b_1) + P_2(B = b_2) + P_3(B = b_3)....P(B = b_n) + \\ 0 \quad 0 \quad P_1(C = c_1) + P_2(C = c_2) + P_3(C = c_3)....P(C = c_n) \end{bmatrix} \tag{3}$$

$$P^L = \frac{P'}{P^t} = \text{Packet score} \tag{4}$$

P^L in the above equation is called the score of the packet. This score value is then stored in the scorebook. For each time window, the score value is stored in the scorebook. Then, it is compared with the threshold value. If the score value is more than the threshold value, then the packet is considered malicious and if the score value is less than the threshold value, then the packet is considered as the legitimated packet.

3.2 Threshold calculation module

This module explains the method used for calculating the threshold for the packet score value defined in the previous section. For the calculation of the threshold for $(i+1)th$ time window, the scores of the *ith* time window are extracted from the scorebook. The threshold is calculated by the load shedding algorithm [44]. According to the load shedding algorithm:

(1) Total incoming traffic for *ith* time window is $\psi(i)$.
(2) Total allowed traffic for *ith* time window is:

$$\phi(i) = \max\{\psi(i-1), \kappa\} \tag{5}$$

where κ is max. traffic stored in the score book

$$\text{Threshold value } \theta(i+1) = 1 - \frac{\psi(i)}{\phi(i)} \tag{6}$$

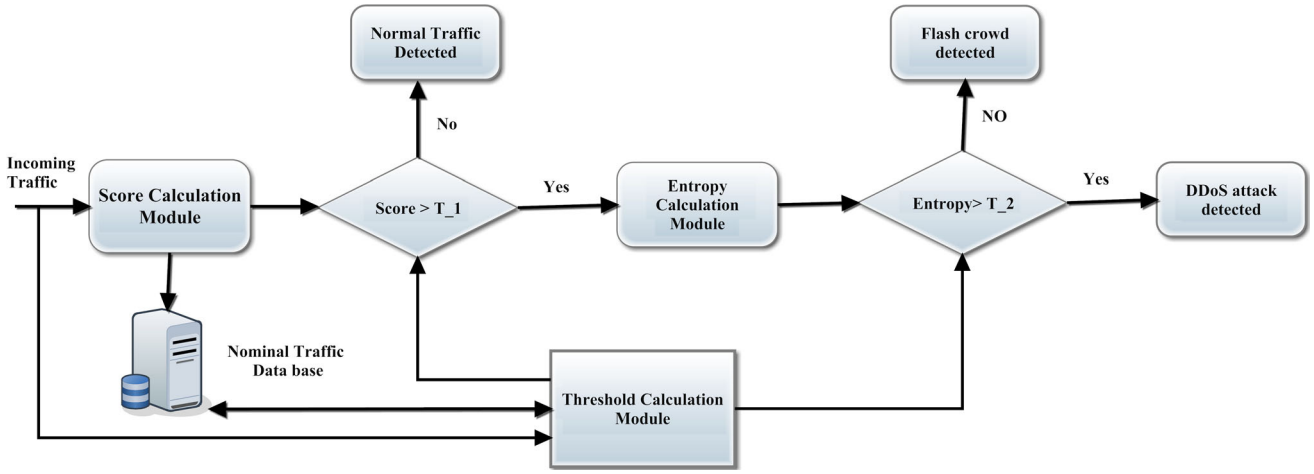

Fig. 2 Work flow of the proposed approach

3.3 Entropy calculation module

Definition 1 Probability of discrete random variable X is given in Equation 7.

$$P(x) = \frac{P(xn_i)}{\sum_{i=1}^{n} xn_i} \tag{7}$$

where P(x) represents the probability of a random variable X, which has 'n' possible values, i.e., $x_1, x_2 ... x_n$

Definition 2 Shannon defines the concept of entropy. It is the measure of the randomness of uncertainty of a random variable [45]. Let X be a random variable, then its Entropy H(X) is represented in the equation Eq. 8

$$H(X) = -\sum_{i=1}^{n} P_i \times \log(P_i) \tag{8}$$

where H(X) represents the Entropy of a random variable X, which has 'n' possible values, i.e., $x_1, x_2 ... x_n$ and each value has specific probability distribution $P_1, P_2, ..., P_n$

Definition 3 In our proposed approach, we group the incoming packets according to the destination address. Then, according to Eq. 7, we calculated the probability of occurrence of each cluster in a time window '$\triangle t$'.

$$P(C_i) = \frac{Cn_i}{\sum_{i=1}^{n} Cn_1} \tag{9}$$

where Cn_i represents number of packets in *ith* crystal.

Theorem 1 *If X is a random variable, which can take finite random values $x_1, x_2, ..., x_N$ then*

$$H(X) \leqslant \log(N) \tag{10}$$

Equality sign holds only if and only if the probability distribution of X is uniform, i.e., $P(x_i) = \frac{1}{N}$

Proof Suppose there are two set probability distribution $P_1 = p_1 + \varepsilon, p_2 - \varepsilon, p_3, ... p_N$ and $P_2 = p_1, p_2, p_3, ... p_N$ where ε is very small value. Entropy corresponds to P_1 and P_2 are H_1 and H_2

$$\begin{aligned} H_1 - H_2 = &- p_1 \log\left(\frac{p_1 + \varepsilon}{p_1}\right) - \varepsilon \log(p_1 + \varepsilon) \\ &- p_2 \log\left(\frac{p_2 - \varepsilon}{p_2}\right) + \varepsilon \log(p_2 + \varepsilon) \\ = &- p_1 \log\left(1 + \frac{\varepsilon}{p_1}\right) - \varepsilon\left(\log p_1 + \log\left(1 + \frac{\varepsilon}{p_1}\right)\right) \\ &+ \varepsilon\left(\log p_2 + \log\left(1 - \frac{\varepsilon}{p_2}\right)\right) \end{aligned}$$

As we know, for small values of x $\log(1 + x) = x + O(x^2)$, so the above equation is reduced as

$$-\varepsilon - \varepsilon \log p_1 + \varepsilon \log p_2 + O(\varepsilon^2) = \varepsilon \log \frac{p_2}{p_1} + O(\varepsilon^2)$$

which is positive when ε is small enough since $p_1 < p_2$. Therefore, the entropy increases if the randomness among the probability distribution decreases; hence the entropy is maximum if the probability distributions of random variables are uniformly distributed, i.e., $p_i = \frac{1}{N}$ and $H(X) \leqslant \log N$ □

Theorem 2 *The entropy value, $H(X)$ is stationary, i.e., its value at two different time windows is the same.*

Proof Most of the attackers use tools to initiate DDoS attacks. In these tools, a predefined program is used for spoofing the destination address of the attack packets. Therefore, if X_i represents all the spoofed addresses, then we can see that there is a linear function such that

$$X_i = f(x) = aX_i + b$$

where a and b are constants and $f(x)$ is a linear function.

Therefore, the probability of occurrence of each cluster is represented as:

$$p(x_i) = p(f(x_i));\ i \in i, 2, ...N$$
$$\textit{from Equation}\ 8,$$
$$H(X) = -\sum_{i=1}^{N} p(x_i)$$
$$H(X) = -\sum_{i=1}^{N} p(f(x_i))$$
$$H(X) = H(Y)$$

where $Y \in \{(x_1), f(x_2), ...f(x_N)\}$, hence we can say that the cluster entropy due to DDoS attack traffic is represented by a stationary stochastic process. □

Theorem 3 *Entropy, H(X), is a concave function.*

Proof Let a function $t(p) = -p \log p$ then

$$t' = -\log p - p \times \log e \times \frac{1}{p}$$
$$= -\log p - \log e$$
$$\textit{and}$$
$$t'' = -\frac{1}{p} \times \log p < 0$$

for all $x > 0$.

for a random variable $X \in \{x_1, x_2...x_n\}$ probability, $P(X)$ is given by Eq. 7 and entropy, $H(X)$, is given by Eq. 8.

so, we can write

$$H(x) = \sum_{i=1}^{N} t(P(x))$$

Thus, we can say that entropy is a concave function of probability $P(X)$ for every X. □

Lemma 1 *Cluster entropy at the time of DDoS attack is more than the flash crowd scenario.*

Proof During DDoS attack, a large amount of traffic is generated at a higher rate. Therefore, we can model it as a monotonically increasing convex function. Then by Jensen's inequality

$$Ef(x) \geq f(Ex) \tag{11}$$

where $f(x)$ is a probability distribution function for DDoS. During the flash crowd scenario, a large number of legitimated users connect to the server. However, the connection process is slow; hence, the probability distribution of flash crowd is represented as a monotonically increasing concave function. Then by Jensen's inequality

$$Ef'(x) \leq f'(Ex) \tag{12}$$

where $f(x)$ is a probability distribution function for flash crowd. If $P^1 = \{p_1^1, p_2^1...p_n^1\}$ represents probability of different clusters during flash crow scenario and $P^2 = \{p_1^2, p_2^2...p_n^2\}$ represents probability of different clusters during DDoS attack. During DDoS attack, the packets are more randomly distributed, so we can say that

$$P_i^1 < P_i^2,\ 1 \leq i \leq n \tag{13}$$

Therefore, from the above equations and the definition of entropy, we can say that

$$H^2(X) > H^1(X) \tag{14}$$

where $H^2(X)$ represents the entropy of DDoS attack and $H^1(x)$ represents the entropy of flash crowd scenario. □

4 Description of the algorithm

In this section, the algorithm which is used in each router at every time window is explained. Table 1 represents the terms used in the algorithm.

Our proposed algorithm is implemented on the router and activated according to the data rate, if the data rate is higher than the predefined data rate, then the router starts filtering the packets according to algorithm 1. The working of the algorithm is divided into three phases, the detail of each phase is given in the following subsections.

Table 1 Attributes used in algorithm

Term	Explanation
Pk	*Kth* incoming packet
Ai[k]	Kth attribute of *ith* packet
M[k]	Marker value of *kth* packet
H[k]	Entropy Value for *kth* packet
IP[k]	Reduced IP address
G	Clustering group
IP_{mask}	Mask used to select the *kth* packet in group G
Si[k]	Score of *kth* attribute of *ith* packet
Threshold_1	Threshold for packet score
Threshold_2	Threshold for Entropy measurement

Algorithm 1: Router side operation

```
Input Incoming Packet P_k
Output: Weather Packet is Malicious or Legitimated
Start
if D_c > D_n then
    for Each incoming Packet P_k do
        Ai[k] ⇐ Pk ; /* Store the attribute value
        M[k] ⇐ Pk ; /* Store the marker value
        IP[k] ⇐ P_kip * IP_mask
        if IP[k] belongs to group G = G_1; G_2; ... G_n then
            G_ci = G_ci + 1
        end
        else
            Add new group G_i → G
        end
        Calculate the packet score S_i(k)
        Calculate the entropy H_i(k) ;
        if H_i[k] > Threshold_1 then
            if S_i[k] < Threshold_2 then
                return Packet is Legitimate
            end
            else
                if M[k] is set then
                    return Packet is Malicious
                end
                else
                    Set the marker value in P_k
                end
            end
        end
        else
            Flash crowd detected
        end
    end
end
End
```

4.1 IP address extraction phase

In this phase, the source's IP address is extracted from the incoming packet and grouped. In order to group the IP addresses, we used the subnet masking method represented in [10]. In the subnet masking method, all the IP address which belongs to the same subnet comes under the same group. The grouping process is explained as follows:

$$IP_{\text{mask}} = 255.255.255.255$$
$$G_i = IP_i * IP_{\text{mask}}$$

where G_i represent the group in which the *i th* IP address belongs. $G \in \{G_1, G_2, G_3, ...G_n\}$ where G is the collection of all the groups. If the group G_i, to which the *k th* packet belongs, is present in G, then the count G_i is increased by one, but if G_i not belongs to G, then a new group is formed and added to the list G. This process is represented as follows:

$$\text{if } G_i \in G, \text{ then } G_{c_i} = G_{c_i} + 1$$
$$\text{else } G_i \longrightarrow G;\ G_i \text{ added to the list } G$$

4.2 Packet score and entropy calculation phase

In this phase, according to Definition 1, Definition 2, Definition 3, and the score calculation method defined in Sect. 3, group entropy and packet score of each packet are calculated.

H_t = Entropy at tth time period

$S_i[k]$ = Packet score of kth attribute of ith packet

4.3 Packet filtering phase

In this phase, the previously calculated entropy and packet scores are compared with the threshold values. According to lemma 1, the group entropy H_t at the time of DDoS attack is more than the group entropy at the time of the flash crowd.

$$H_{t_{\text{Normal traffic}}} < H_{t_{\text{Flash crowd}}} < H_{t_{DDoS\ \text{attack}}}$$

Therefore, if the entropy is more than the threshold value ($Threshold_1$), then the traffic is due to the DDoS attack, then we compare the packet score ($S_i[k]$) of the packets with the second threshold value ($Threshold_2$). If the value of $S_i[k]$ is less than $Threshod_2$ and the marker value ($M_i[k]$) of the respective packet is set, then that packet is considered as the malicious packet. Thus, in this phase, the identification and filtering of the malicious packet take place.

$$\text{if } H_t > Threshold_1 \text{ and } S_i[k] < Threshold_2 \text{ , Packet is legitimated}$$
$$\text{if } H_t > Threshold_1 \text{ and } S_i[k] > Threshold_2, \text{Packet is malicious}$$
$$\text{if } H_t < threshold_2 \text{ and data rate is high, flash crowd is detected}$$

5 Results and discussion

The proposed approach is implemented on the Omnet++ discrete event simulator. The simulated environment details are given in Table 2.

In order to measure the accuracy of our proposed approach, we test the filtering process in two different test scenarios. In test scenario 1, we analyzed the working of our proposed approach in the presence of DDoS attack traffic. In test scenario 2, we analyzed the working of our proposed approach in the presence of flash crowd.

5.1 Test scenario 1: DDoS attack is present

In this test scenario, the attacker nodes are generating malicious packets for different intervals of time. We have used four test cases to analyze this scenario. In each test case, attackers generate a massive number of attack packets

Table 2 Simulation Environment

Attributes	Value
Simulated area	500 × 500 meters
Simulation time	100 seconds
Communication channel	IEEE 802
Attacker data generation rate	Exponential(0.172) packet/seconds
Normal node data generation rate	Exponential(0.152) packet/seconds
Type of nodes	Malicious nodes, legitimated nodes, routers
Router Queue capacity	100

that start at 1 second and stop at 20 seconds, 30 seconds, 50 seconds, and 60 seconds, respectively. For each test case, we calculate the packet score value, if the packet score value exceeds the threshold value, then that packet is considered a malicious packet. According to our proposed approach, during DDoS attack scenarios, most packets are generated by attackers, so the packet score value is higher than the packet score value during the duration of any attack, as we see in Fig. 3 the same thing has happened in our test cases. In Fig. 3, we can see that in each test case, the threshold value is changing with the packet score value. Therefore, all packets whose packet score has a lower value are considered threshold value malicious packets. From the diagram below, we can see that our model correctly identifies the DDoS attack traffic and then filters them.

5.2 Test scenario 2: flash crowd is present

In this test scenario, the selected legitimated nodes generate packets at a higher rate for different intervals of time. We have used four test cases to analyze this scenario. In each test case, legitimate users generate a massive number of data packets that start at 1 second and stop at 20 seconds, 30 seconds, 50 seconds, and 60 seconds, respectively. For each test case, incoming packets are clustered according to their source IP address, and then the entropy of each cluster is calculated. According to our proposed model, if the entropy value is more than the threshold value, then we assume that the traffic is due to the flash crowd. For the evaluation of test cases, we take the threshold value as 2.5. For test case 1 (Fig. 4) and test case 2 (Fig. 4), the entropy value is more than 2.5 and for test case 3 (Fig. 4) and test case 4 (Fig. 4) the entropy value is less than 2.5, so we can easily say that for test cases 1 and 2 traffic is generated by DDoS and for test case 3 and case 4 traffic is generated by the flash crowd. Hence, our proposed model correctly differentiates DDoS attack traffic from the flash crowd.

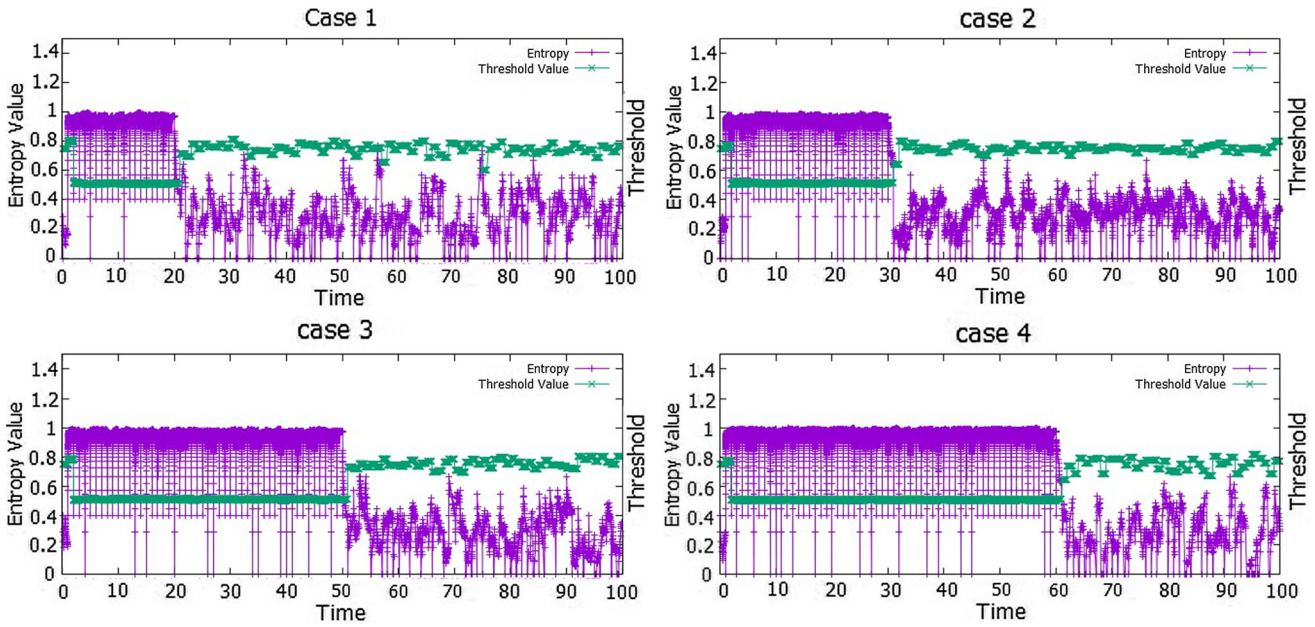

Fig. 3 Variation of packet score with threshold

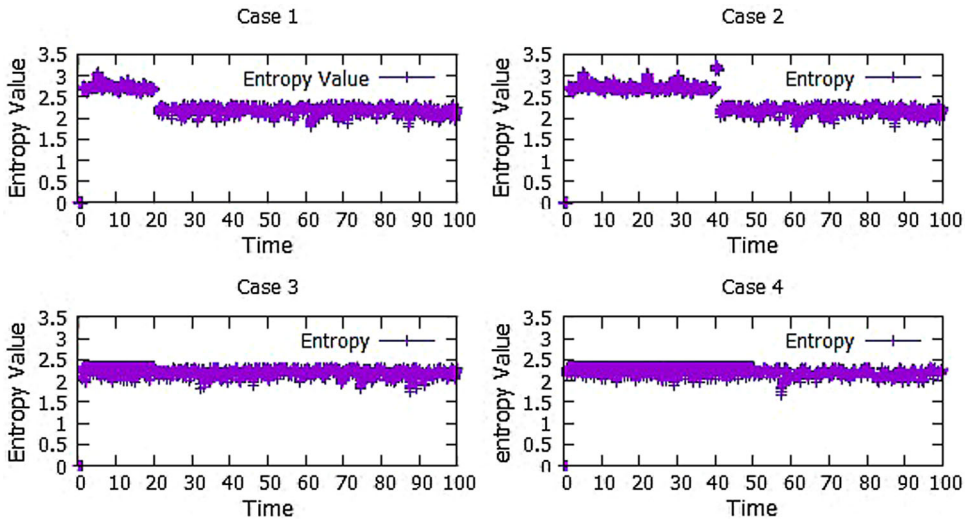

Fig. 4 Variation of entropy value

5.3 Statistical analysis of proposed approach

For statistical analysis of our proposed scheme, we consider ten test cases and then calculate the following statistical values:

(1) Precision (PL)—It is the percentage of legitimate packets from the total packets that reached the destination. In our proposed approach, the precision value is more than 0.93, which means our proposed approach is consistent in detecting the attack traffic.

$$\mathrm{PL} = \frac{TP}{TP + FP}$$

Where TP is the number of legitimate packets that pass through the filter and FP is the number of malicious packets that pass through the filter.

(2) Recall—It gives the percentage of legitimate packets that reach the destination. In our proposed approach, the recall value is more than 0.97, which means that our proposed approach correctly differentiates non-attack packets from attack packets.

$$\mathrm{Recall} = \frac{TP}{TP + FN}$$

Where FN is the number of legitimated packets filtered by the algorithm.

(3) True Negative Rate (TNR)—This gives the percentage of malicious packets that were filtered out by the router. From the results in Fig. 5, it is clear that most of the attack packets are filtered by our proposed approach.

$$\mathrm{TNR} = \frac{TN}{TN + FP}$$

Where TN is the number of malicious packets filtered out by the algorithm

(4) Negative Predictive Value (NPV)—It represents the percentage of malicious packets from the total filtered out packets. As the NPV value is high in our proposed approach, it means the majority of malicious packets are identified by our proposed approach.

$$\mathrm{NVP} = \frac{TN}{TN + FN}$$

5.4 State-of-the-art comparisons

To show the effectiveness of our proposed approach, we compare it with the existing collaborative filtering technique, ScoreForCore (SFC).

(1) Precision—For comparison of the accuracy of our proposed approach with the SFC method, we use ten test cases and the result is represented in Fig. 6. From Fig. 6, it is clear that our proposed approach more accurately detects the flash crowd than the SFC method.

(2) Message Complexity—SFC method uses query packets to shear information among different routers. The query packets are broadcast in the network. In the SFC approach, the query message travels up to three hops away from the source, and if the receiver of the query message has the score information, then it again uses the broadcasting method to share the information. Therefore, in the worst-case scenario, the total number of packets generated by SFC are $W \times X \times Y$ (where W, X, Y) which are the routers that are one-hop, two-hop, and three hops away, these are more than our proposed approach because our approach uses the unicast method. This comparison is represented in Fig. 6, so if three routers are at one

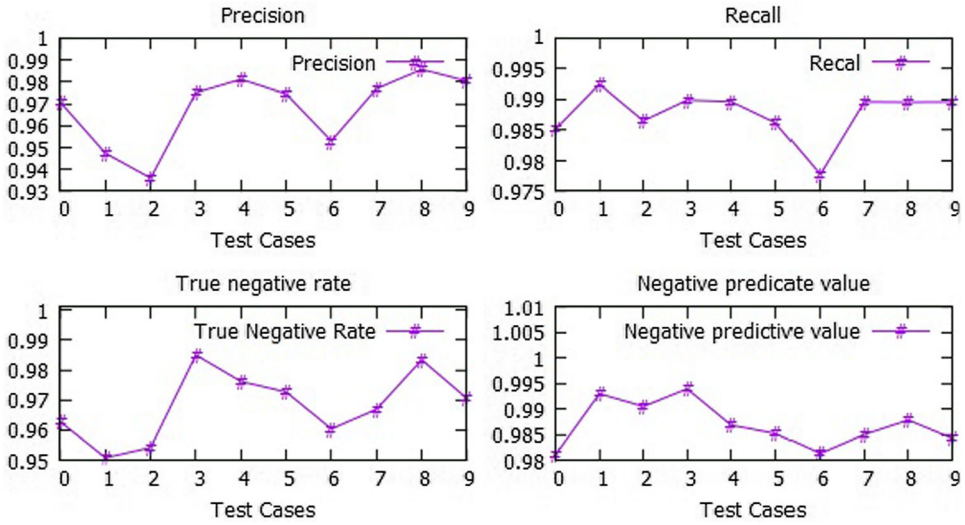

Fig. 5 Statistical analysis

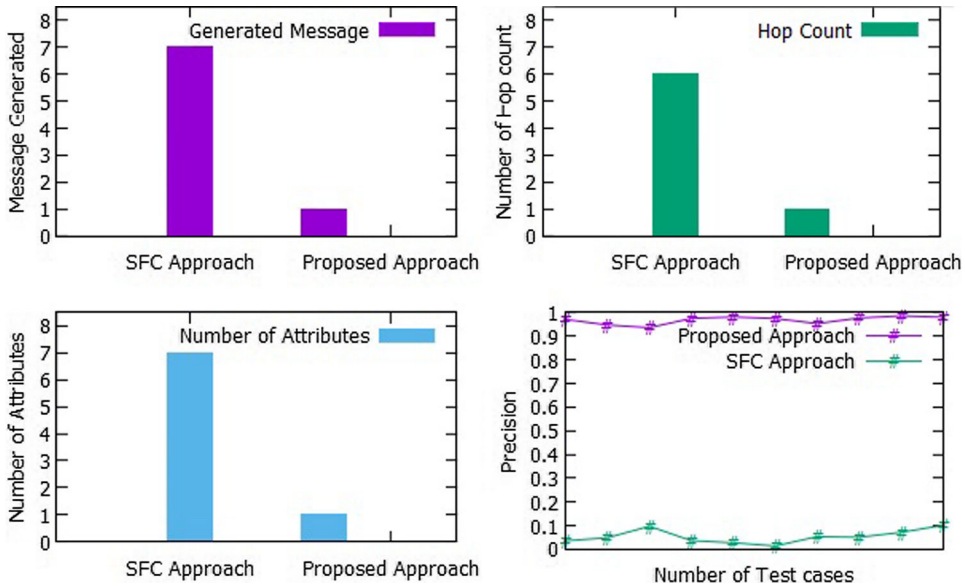

Fig. 6 Comparison of proposed approach with SFC [20] approach

hop, two hops, and three hops away, then for the SFC method a total of seven messages are generated (six for the query and one for the reply), but in our proposed approach only one message is sufficient to shear the information.

(3) Running Time—The SFC method uses a query and reply method for the communication of information. SFC method takes approximately 6 hops time for the exchange of information among the routers in the worst-case scenario. However, our proposed approach uses an unicast method for transferring the information between the routers. Hence, the time complexity of our proposed approach is less than compared to SFC. The detail of the running time calculation in our proposed approach is as follows:

$$\begin{aligned}&\text{Let } N = \text{Total number of attributes analysis}\\&n = \text{number of attributes analysis by one router}\\&TM = \text{Time to make the packet}\\&T = \text{Time to analyze the one attribute}\\&TL = \text{Time taken by the packet to move between two routers}\\&\text{Time complexity} = N * T + T_M + T_L + (N - n)T\end{aligned} \tag{15}$$

(4) Space Complexity—ScoreForCore stores the packet score for all the single attributes and a single pair of attributes, which defines the specific attack scenario during the idle period. However, our proposed approach stores only the score of its selected attributes, so in this method, the space required for

Table 3 Comparison of our proposed approach with other existing techniques

Author	DDoS attack detection	Flash crowd detection	Distributed approach	Hop count	Control message traffic	Filtering time	Storage space	Third-party control
[20]	✔	✗	✗	High	High	High	High	✗
[33]	✔	✗	✗	–	–	Medium	–	–
[17]	✔	✗	✗	–	–	Medium	–	✔
[37]	✔	✗	✗	–	–	Medium	Medium	✗
[30]	✔	✗	✔	–	High	High	High	✔
Our approach	✔	✔	✔	Low	Low	Low	Low	✗

storing the user information is less. Therefore, if there are N attributes, then SFC requires a space to store $(N+1)$ attributes; however, our proposed approach stores a single attribute in this case. For example, if there are six attributes, then ScoreForCore stores the score of seven (six single and one pair) attributes, and our proposed approach stores the score of a single attribute. Hence, we can say that Super-router is required less space for storing the attributes. Figure 6 represents this scenario.

The comparison of our proposed with other recent works is represented in Table 3. In Table 3 Hop count and the filtering time is calculated by using Eq. 15. Storage space in Table 3 represents the space complexity of the approach.

6 Conclusion

In the aftermath of the COVID-19 epidemic, the way that people work has completely altered. This makes it easier for hackers to spread fear, and they do it by using multiple cyberattacks. A DDoS assault is often used by cyber attackers because to its ease of implementation and the potential to completely absorb all of the resources of the target system. The goal of the DDoS assault is to crash the victim's system or take away its computing power. In the presence of the flash crowd, which is a circumstance in which massive amounts of traffic are created by genuine users, the DDoS assault becomes more difficult to detect. Since this is true, it is always a significant research problem to effectively and precisely identify DDoS assaults. Because of the similarities between DDoS attacks and the flash mob, it is almost impossible to tell the two apart. In this paper, we proposed an approach, which detects DDoS attacks efficiently and differentiates the DDoS attack from the flash crowd. The proposed approach uses the packet score method for the separation of DDoS attack traffic from the normal traffic. Moreover, the variation in the entropy of the packet's source IP address is used to differentiate flash crowd from normal traffic and DDoS attack traffic. Simulation results showed that the proposed approach has precision value, true-negative value, and negative predicate value greater than 95%, indicating the high efficiency of our proposed approach. Hence, our proposed approach provides an efficient solution to all the online resources facing the DDoS attack problem in this COVID-19 pandemic.

This study is focused on DDoS attack detection using statistical methods, our future studies should concentrate on the development of deep learning techniques like self-supervised learning for the detection of DDoS attack.

Funding This work is supported in part by the National Natural Science Foundation of China under Grant 61972205, in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund, and in part by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET) fund, China.

Declarations

Conflicts of interest Author declares no conflicts of interest

References

1. W.H. Organization (2020) Who coronavirus disease (covid-19) dashboard [Online]
2. Sohrabi C, Alsafi Z, O'neill N, Khan M, Kerwan A, Al-Jabir A, Iosifidis C, Agha R (2020) World health organization declares global emergency: a review of the 2019 novel coronavirus (covid-19). Int J Surg 76:71. https://doi.org/10.1016/j.ijsu.2020.02.034
3. Jelena Mirkovic PR (2004) A taxonomy of ddos attack and ddos defense mechanisms. ACM SIGCOMM Comput Commun Rev 34(2):39. https://doi.org/10.1145/997150.997156
4. Swasth. https://www.swasth.app/home (2020)
5. Chaturvedi GJA, Kalyani S (2020) Reliability and effectiveness of Indian covid-19 mobile apps. J Critical Rev 7(14):1296–1305

6. Jhunjhunwala A (2020) Role of telecom network to manage covid-19 in india: Aarogya setu. Trans Indian National Acad Eng 5(2):157. https://doi.org/10.1007/s41403-020-00109-7
7. Lochlainn MN, Lee KA, Sudre CH, Varsavsky T, Cardoso MJ, Menni C, Bowyer RC, Nguyen LH, Drew DA, Ganesh S, du Cadet JL (2020) Key predictors of attending hospital with COVID19: an association study from the COVID symptom Tracker APP in 2,618,948 individuals. medRxiv
8. Bajpai MWN, Biberman J (2020) ICT initiatives in India to combat COVID-19, Columbia academic commons
9. Gaurav AKSA (2017) Super-router: a collaborative filtering technique against ddos attacks, International Conference on Advanced Informatics for Computing Research pp. 294–305
10. Dittrich D (1999) The DoS project's 'trinoo'distributed denial of service attack tool
11. Criscuolo PJ (2000) Distributed denial of service: trin00, tribe flood network, tribe flood network 2000, and stacheldraht ciac-2319. California Univ Livermore Radiation Lab
12. Barlow WTJ (2000) Tfn2k an analysis. Axent Security Team 13(2):21
13. Dittrich D, Weaver G, Dietrich S, Long N (2000) The mstream distributed denial of service attack tool
14. Gupta BB, Joshi RC, Misra M (2012) Distributed denial of service prevention techniques. arXiv preprint arXiv:1208.3557
15. Stergiou CL, Psannis KE, Gupta BB (2020) Iot-based big data secure management in the fog over a 6g wireless network. IEEE Int Things J 8(7):5164–5171
16. Chhabra M, Gupta B, Almomani A (2013) A novel solution to handle DDOS attack in MANET
17. Shidaganti GI, Inamdar AS, Rai SV, Rajeev AM (2020) Scef: a model for prevention of ddos attacks from the cloud. Int J Cloud Appl Comput 10(3):67–80
18. Al-Qerem A, Alauthman M, Almomani A, Gupta B (2020) Iot transaction processing through cooperative concurrency control on fog-cloud computing environment. Soft Comput 24(8):5695
19. Mishra A, Gupta BB, Peraković D, Yamaguchi S, Hsu CH (2021) In: 2021 IEEE International Conference on Consumer Electronics (ICCE), pp. 1–6. https://doi.org/10.1109/ICCE50685.2021.9427772
20. Kalkan FAK (2016) A distributed filtering mechanism against ddos attacks: scoreforcore. Comput Netw 108:199
21. Sunny Behal MS, Kumar K (2018) D-face: an anomaly based distributed approach for early detection of ddos attacks and flash events. J Netw Comput Appl 111:49. https://doi.org/10.1016/j.jnca.2018.03.024
22. Jung MRJ, Krishnamurthy B (2002) Flash crowds and denial of service attacks: characterization and implications for cdns and web sites, Proc. 11th international conference on World Wide Web pp. 293–304
23. Gaurav A, Singh AK (2017) Entropy-score: a method to detect DDoS attack and flash crowd. In: 2017 2nd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT). IEEE, pp 1427–1431
24. Ahmed N, Michelin RA, Xue W, Ruj S, Malaney R, Kanhere SS, Seneviratne A, Hu W, Janicke H, Jha SK (2020) A survey of COVID-19 contact tracing apps. IEEE Access 8:134577–134601
25. Pashchenko D (2021) Fully remote software development due to covid factor: results of industry research (2020). Int J Software Sci Comput Intell (IJSSCI) 13(3):64
26. Magklaras G, López-Bojórquez LN (2021) A review of information security aspects of the emerging COVID-19 contact tracing mobile phone applications. International symposium on human aspects ofinformation security and assurance. Springer, Cham, pp 30–44
27. De Carli A, Franco M, Gassmann A, Killer C, Rodrigues B, Scheid E, Schoenbaechler D, Stiller B (2020) WeTrace–a privacy-preserving mobile COVID-19 tracing approach and application. arXiv preprint arXiv:2004.08812
28. Masud M, Gaba GS, Alqahtani S, Muhammad G, Gupta BB, Kumar P, Ghoneim A (2020) A lightweight and robust secure key establishment protocol for internet of medical things in COVID-19 patients care. IEEE Int Things J
29. Sedik A, Hammad M, Abd El-Samie FE, Gupta BB, Abd El-Latif AA (2021) Efficient deep learning approach for augmented detection of Coronavirus disease. Neural Comput Appl, 1–18
30. Kim Y, Lau WC, Chuah MC, Chao HJ (2006) PacketScore: a statistics-basedpacket filtering scheme against distributed denial-of-service attacks. IEEE Trans Dependable Secure Comput 3(2):141–155
31. Mahajan R, Bellovin SM, Floyd S, Ioannidis J, Paxson V, Shenker S (2002) Controlling high bandwidth aggregates in the network. ACM SIGCOMM Comput Commun Rev 32(3):62–73
32. Jérôme Francois RB, Aib I (2012) Firecol: a collaborative protection network for the detection of flooding ddos attacks. IEEE/ACM Trans Netw 20(6):1828. https://doi.org/10.1109/tnet.2012.2194508
33. Bhushan K, Gupta BB (1985) Distributed denial of service (ddos) attack mitigation in software defined network (sdn)-based cloud computing environment. J Ambient Intell Humanized Comput 10(5):1985–1997
34. Gaurav A, Gupta BB, Hsu CH, Yamaguchi S, Chui KT (2021) In: 2021 IEEE International Conference on Consumer Electronics (ICCE) IEEE, pp. 1–5
35. Gaurav A, Gupta BB, Castiglione A, Psannis K, Choi C (2020) International Conference on Computational Data and Social Networks. Springer, Berlin, pp 386–397
36. Al-Nawasrah A, Almomani AA, Atawneh S, Alauthman M (2020) A survey of fast flux botnet detection with fast flux cloud computing. Int J Cloud Appl Comput 10(3):17–53
37. Qin CWX, Xu T (2015) Ddos attack detection using flow entropy and clustering technique, 11th International Conference on Computational Intelligence and Security (CIS) pp. 412–415
38. Monika Sachdeva GS, Kumar Krishan (2016) A comprehensive approach to discriminate ddos attacks from flash events. J Inf Security Appl 26:8. https://doi.org/10.1016/j.jisa.2015.11.001
39. Yu S, Thapngam T, Liu J, Wei S, Zhou W (2009) Discriminating DDoS flows from flash crowds using information distance. In: 2009 Third international conference on network and system security. IEEE, pp 351–356
40. Li K, Zhou W, Li P, Hai J, Liu J (2009) Distinguishing DDoS attacks from flash crowds using probability metrics. In: 2009 Third international conference on network and system security. IEEE, pp 9–17
41. Thapngam T, Yu S, Zhou W, Beliakov G (2011) Discriminating DDoS attack traffic from flash crowd through packet arrival patterns. In: 2011 IEEE conference on computer communications workshops (INFOCOM WKSHPS). IEEE, pp 952–957
42. Xiao P, Qu W, Qi H, Li Z (2015) Detecting DDoS attacks against data center with correlationanalysis. Comput Commun 67:66–74
43. Baishya RC, Hoque N, Bhattacharyya DK (2017) DDoS attack detection using uniquesource ip deviation. Int J Netw Secur 19(6):929–939
44. Kasera CLMKAHS, Pinheiro J (2001) Fast and robust signaling overload control, Proceedings Ninth International Conference on Network Protocols. ICNP pp. 323–331
45. Witten E (2020) A mini-introduction to information theory. La Rivista del Nuovo Cimento 43(4):187. https://doi.org/10.1007/s40766-020-00004-5