International Journal of Innovative Computing, Information and Control
Volume **20**, Number **1**, February **2024**
 ISSN 1349-4198
pp. **31–46**

AN EFFECTIVE METHOD FOR DIFFERENTIATING BETWEEN DDOS ATTACKS AND FLASH CROWDS

Ruoyu Yan* and Yingfeng Wang

College of Computer and Information Engineering
Henan University of Economics and Law
No. 180, Jinshuidong Road, Zhengzhou 450002, P. R. China
yfwang@huel.edu.cn
*Corresponding author: ryyan@huel.edu.cn

Received April 2023; revised August 2023

Abstract. *Accurately distinguishing Distributed Denial of Service (DDoS) attacks from Flash Crowds is an essential task in current cyber security research. However, most existing methods face several issues such as difficult data acquisition, complex data processing, and low detection efficiency. DDoS attacks and Flash Crowds can cause a significant increase in the number of new source IP addresses. To filter out normal network traffic, an anomaly detection method based on Hurst parameter in the occurrences of new source IP addresses is proposed. Further discrimination of DDoS attacks and Flash Crowds is carried out based on the variance in the distribution of new source IP addresses caused by these anomalies. For this purpose, the study proposes a variance ratio statistical method based on cross entropy. The final experimental results demonstrate that the proposed approach outperforms other approaches in terms of significantly improving the true positive rate, precision, and overall accuracy, while reducing both the false positive and false negative rates in a real network environment. Additionally, the method is found to enhance the detection and recognition speed and alleviate the complexity of real-time traffic processing by employing fewer traffic features.*
Keywords: Cyber security, Distributed Denial of Service (DDoS), Flash Crowd, Hurst parameter, Cross entropy

1. **Introduction.** Denial of Service (DoS) attacks are strategically designed to prevent legitimate users from accessing shared services or resources on the Internet. When these attacks originate from multiple sources (bots), they are known as Distributed Denial of Service (DDoS) attacks. In DDoS attacks, the involvement of multiple attacking nodes significantly amplifies the intensity, thereby complicating defense strategies. DDoS attacks pose a main threat to web servers, with SYN Flooding attacks emerging as the predominant attack. SYN Flooding attacks constitute nearly 90% of all DDoS attacks [1]. These attacks closely resemble the Flash Crowd phenomenon, where an event such as the launch of a new Apple product triggers a sudden surge in legitimate user traffic to a website, thus causing performance degradation or service disruption. Given the divergent motivations behind these two network anomalies, it is crucial to differentiate them accurately, in order to take appropriate countermeasures. For instance, in the case of a DDoS attack, it is necessary to promptly determine the attack path [2], locate the source of the assault, and block the attack traffic upstream without delay to reduce the damage inflicted upon web services. In case of a flash crash, balancing user accessibility can be achieved server-side by implementing measures, such as setting reasonable concurrent connections and modulating IP access frequency.

DOI: 10.24507/ijicic.20.01.31

Previous studies have primarily focused on three aspects – analysis of user behavior, host-side testing, and analysis of traffic characteristics – for differentiating between DDoS attacks and Flash Crowds. The analysis of user behavior distinguishes between the two through differences in the visiting time of web pages, frequency of mouse clicks, visiting of hot pages, and packets inter-arrival time [3-6]. Although its detection rate is relatively high, it often requires analysis data at the application layer, which can be difficult to obtain. Furthermore, the models used often require complex training and real-time updates for model parameters. On the other hand, host-side testing method requires deployment on the protected server side [7,8]. Its computational complexity is minimal; but its operability is limited, ultimately impeding legitimate user access. However, the analysis of traffic characteristics predominantly involves capturing network traffic, extracting relevant attributes such as distribution of traffic on various attributes, "flow" distance, and utilizing information entropy or other techniques to differentiate DDoS attacks from Flash Crowds [9-16]. For example, Behal and Kumar conducted a comparative study of various measures, including generalized entropy, information distance, and Shannon entropy, to ascertain their effectiveness in differentiating DDoS attacks from Flash Crowds [9]. They concluded that generalized entropy and information distance pose more advantages, offering a lower false alarm rate. Additionally, they proposed an effective ϕ-entropy method for distinguishing between the two types of traffic [10], which, however, may be difficult to detect common DDoS attacks with large packet sizes. In their research, David and Thomas utilized Tsallis entropy to measure the concentration of destination IP address, enabling the detection of DDoS attacks and Flash Crowds. They also leveraged relatively fixed packet sizes to differentiate between the two traffic types, achieving high detection efficiency in their experiments [11]. However, this approach may not be effective when the attacks involve varying packet sizes. In another study, Jiang et al. employed five different flow characteristics to identify different anomalies in an SDN (Software Defined Network) network. They introduced ϕ-entropy to increase the information distance and implemented the K-Nearest Neighbor (KNN) algorithm to differentiate between DDoS attacks and Flash Crowds [12]. This approach necessitates the capture of additional flow features, as well as a substantial amount of computational resources. Additionally, the accuracy of the labeled samples used for supervised learning classification can significantly influence the final recognition outcomes. Similarly, Sekhar et al. employed a distinct machine learning algorithm, namely a novel Deep Neural Network (DNN), to tackle this particular issue [13]. Their approach also necessitates the accumulation of substantial flow feature data for training the classification model. The experimental findings demonstrate that this methodology has attained remarkably high levels of accuracy. However, it is worth noting that it encounters analogous challenges as the method in [12]. In contrast, Tao and Yu detected abnormal flows by measuring the information entropy of upstream router flows, subsequently utilizing the Sibson information distance of abnormal flows in the downstream router to discriminate between DDoS attacks and Flash Crowds [14]. However, deploying this technique across at least three routers hinders its practical application in certain network environments, and the appropriateness of parameter settings is a critical factor in detection and discrimination efficiency. Yu et al. proposed the use of a flow correlation coefficient to evaluate the similarity between two flows by grouping packets with the same destination IP address, in order to distinguish between DDoS attacks and Flash Crowds [15,16]. However, this approach is not always effective when the network attack scale is sufficiently large. While these network traffic-based methods are generally simple to implement and practical, their detection efficiency is often limited. Therefore, the article aims to address the issue of achieving high detection efficiency while minimizing the number of network traffic characteristics used.

It is widely recognized that DDoS attack tools frequently fabricate numerous source IP addresses to create attack packets repeatedly. Consequently, a substantial number of new source IP addresses arise during an attack. Flash Crowd events, on the other hand, result in an upsurge of new source IP addresses, as many users who have not visited a website in a prolonged time span visit simultaneously. As a solution, according to previous experience [17-19], during the detection phase, a time series of the occurrences of newly emerged source IP addresses in the network is built. Subsequently, by evaluating the time series via self-similarity analysis, DDoS attacks and Flash Crowds can be spotted effectively. Secondly, the method for distinguishing between DDoS attacks and Flash Crowds involves analyzing the change trend in the occurrence of new source IP addresses. Specifically, when a DDoS attack takes place, the number of new source IP addresses will increase suddenly, leading to major alterations in their distribution in the neighboring time interval. On the other hand, during a Flash Crowd event, these IP addresses tend to rise gradually, with minimal changes in their distribution. Therefore, during the differentiation phase, we can compare the distribution differences in the occurrence of new source IP addresses between the two types of anomaly. Previous studies [20,21] reveal that entropy represents an effective tool to achieve this goal.

The main contributions of this paper can be summarized as follows. The study presents an effective two-phase differentiation mechanism that employs an analysis of network traffic's self-similarity and changes in traffic distribution to detect and differentiate DDoS attacks from Flash Crowds. During the detection phase, the mechanism calculates the frequency of new source IP addresses in the monitored network, applies a Whittle estimator [22] to determining the self-similarity parameter and confidence interval, and detects anomalies by comparing them with normal Hurst parameter thresholds. Compared to conventional self-similarity methods, this mechanism markedly reduces the misjudgment rate. In the differentiation phase, the mechanism measures the distribution change of the occurrence of new source IP addresses using cross entropy, and employs a statistical approach to differentiate between the two types of anomalies. Compared to the traditional information entropy method, this mechanism improves the positive rate and accuracy considerably. Furthermore, this mechanism only extracts network traffic through routers and avoids complex modeling and extraction of application layer data, ensuring that legitimate user access is not affected. Experimental results demonstrate that the mechanism has simple calculations, excellent real-time performance, and high detection efficiency, based on minimal traffic characteristics.

The rest of this paper is organized as follows. Section 2 introduces a novel detection and differentiation scheme. In Section 3, we present the anomaly detection method based on self-similarity, while Section 4 focuses on the differentiation method based on cross entropy. We provide experimental results and comparison findings in Section 5. Lastly, our conclusions are drawn in Section 6.

2. **Detection and Differentiation Framework.**

This study presents a framework, illustrated in Figure 1, comprising six modules, i.e., traffic collection, feature extraction, Hurst parameter estimation, anomaly detection, time series construction, and statistical differentiation. The functionality of each module is detailed as follows.

1) Traffic collection: Firstly, the traffic collection module enables the collection of original data packets from the router's traffic through a collection program run on a server. This approach reduces the impact on the network's regular operating functionality. Additionally, winpcap development toolkit serves as the underlying development package.

2) Feature extraction: The feature extraction program extracts the source IP address from data packets and maintains a tally of the number of occurrences of each new source

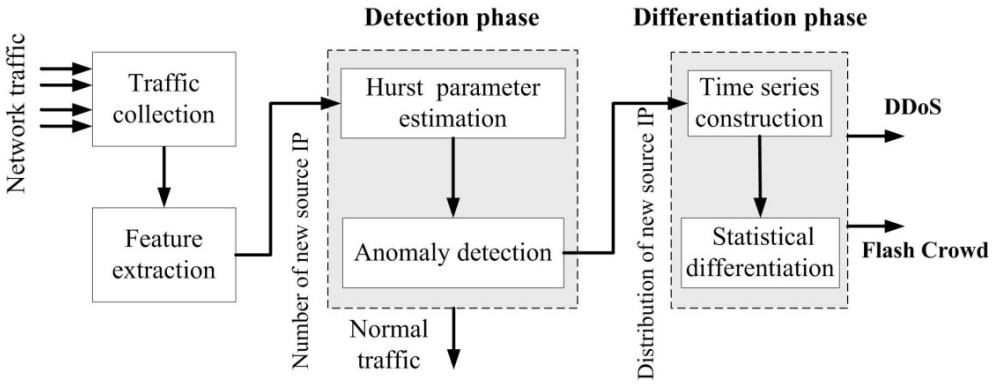


FIGURE 1. Detection and differentiation framework

IP address during each time interval by comparing to the historical source IP address database.

3) Hurst parameter estimation: The Hurst parameter and its confidence interval are estimated by a Whittle estimator [22] based on the data derived from the occurrences of new source IP addresses in the current period. This current estimate enables the method to detect anomalies and updates the normal Hurst parameter thresholds.

4) Anomaly detection: The 24-hour threshold data for the normal Hurst parameter is computed in advance by collecting data from regular network traffic in the monitored network over a span of several weeks. This dataset comprises the normal Hurst parameter and its corresponding confidence interval. To detect anomalies, the currently estimated Hurst parameter value is compared to the normal Hurst parameter thresholds. If the current value falls outside the confidence interval of the normal Hurst parameter at the corresponding time, it suggests the occurrence of a DDoS attack or Flash Crowd. In contrast, if the current value falls within the confidence interval, the normal Hurst parameter thresholds are updated.

5) Time series construction: When traffic anomalies are detected during the anomaly detection phase, the time period is broken down into smaller intervals. For each interval, the number of occurrences of each source IP address is counted, and a cross entropy time series is constructed.

6) Statistical differentiation: The difference between the waveforms of cross entropy time series generated by DDoS attacks and Flash Crowds is distinguished using statistical analysis of variance ratio. This method incorporates a sliding window mechanism to determine the current type of anomaly by observing whether the variance ratio statistic deviates from the threshold. This analysis helps to identify the type of anomaly and take appropriate action to mitigate it.

3. Anomaly Detection Based on Self-Similarity.

3.1. **Definition of self-similarity.** Suppose Y_n $(n = 1, 2, 3, \ldots)$ is a discrete random process, if

$$Y_i^{(m)} = \frac{1}{m} \sum_{k=(i-1)m+1}^{im} Y_k \tag{1}$$

then $Y_n^{(m)}$ is the m-order aggregation process of Y_n, and the corresponding m-order autocorrelation function is $\rho^m(k)$. If Y_n has the same autocorrelation function as its corresponding m-order aggregation process $Y_n^{(m)}$, that is

$$\rho^m(k) = \rho_k \quad (m = 1, 2, 3, \ldots) \tag{2}$$

then this generalized stationary stochastic process Y_n $(n = 1, 2, 3, \ldots)$ is considered to be self similar, which means that $Y_n^{(m)}$ and Y_n have the same second-order statistical properties. For a generalized stationary self similar process, its autocorrelation function satisfies

$$\rho_k = H(2H-1)k^{2H-2}, \quad k \to \infty \tag{3}$$

In the formula, the Hurst exponent H $(0.5 < H < 1)$ serves as a self-similarity coefficient. This means that as the H value increases, the degree of self-similarity in a system also increases. In the context of network traffic, various traffic indicators display self-similarity traits. However, anomalous traffic behavior frequently results in the loss of self-similarity in network traffic, resulting in an H value of less than 0.5.

3.2. **Self-similarity detection method.** In a typical scenario, the access patterns of network users demonstrate periodicity to a certain degree. To account for this periodicity and minimize its impact on the Hurst value, normal network traffic must be processed during different time periods throughout the day [23]. To establish a time series, this study collected two weeks' worth of normal traffic and quantified the frequency of new source IP addresses for each time interval. Using the approach outlined in [22], the Hurst parameter and its 98% confidence interval for different time periods were computed. The final step involved determining the average Hurst parameter and confidence interval within the same time period of a 24-hour day, utilizing the two-week traffic dataset to establish the threshold dataset of normal Hurst parameter in its initial state.

In the detection phase, the Hurst parameter estimation module computes the current Hurst value for detection and threshold update based on the time series of occurrences of new source IP addresses. This value serves as an input to the anomaly detection module, which compares it with the normal Hurst parameter threshold to determine if an exception exists. An exception is detected if the value falls outside the confidence interval of the normal Hurst value in the corresponding 24-hour time interval. Conversely, if the value falls within the range of the confidence interval, the normal Hurst parameter thresholds are updated via the update algorithm outlined in Algorithm 1.

Algorithm 1: Updating of the normal Hurst parameter thresholds
Variable definitions:
New_H_i: Hurst value calculated at current time interval i.
C_New_H_i: Upper limit of the 98% confidence interval of New_H_i.
F_New_H_i: Lower limit of the 98% confidence interval of New_H_i.
H_i: Normal Hurst value at time interval i.
C_H_i: Upper limit of the 98% confidence interval of H_i.
F_H_i: Lower limit of the 98% confidence interval of H_i.
Initialization: In the first two weeks of monitoring, calculate the initial vectors H, C_H and F_H of the normal network traffic.
Input: New_H_i, C_New_H_i, F_New_H_i, H_i, C_H_i, F_H_i
Output: H_i, C_H_i, F_H_i
Step 1: If ($F\_H_i \leq New\_H_i \leq C\_H_i$), go to step 2, else go to step 3.
Step 2: $H_i = 0.5H_i + 0.5New\_H_i$, $C\_H_i = 0.5C\_H_i + 0.5C\_New\_H_i$, $F\_H_i = 0.5F\_H_i + 0.5F\_New\_H_i$.
Step 3: Update H_i, C_H_i and F_H_i in thresholds dataset of normal Hurst parameter.

4. Differentiation Method Based on Cross Entropy.

4.1. Definition of cross entropy.

The cross entropy of order α is defined as

$$L_\alpha(P,Q) = \frac{1}{1-\alpha}\log\sum_{i=1}^{N}\frac{p_i^\alpha}{q_i^{\alpha-1}} \tag{4}$$

In Equation (4), P and Q are discrete random variables, while p_i and q_i are their respective distribution functions. A lower cross entropy value indicates that more information can be gained from a point of observation, thereby enabling better differentiation between P and Q. This significant property of cross-entropy provides a theoretical basis for using this function to identify network traffic anomalies. To determine the sensitivity of cross entropy to the detection of abnormal network traffic, each α $(0 < \alpha < 1)$ value is applied in the experiment. Based on experimental results, when α is set to 0.5, the detection effect is considerably better. Thus, this study uses $\alpha = 0.5$ to ensure that the cross entropy related to variables P and Q is symmetric. Equation (4) can thus be rewritten as Equation (5) to always result in non-negative values.

$$L_{0.5}(P,Q) = -2\log\sum_{i=1}^{N}\sqrt{p_i q_i} \tag{5}$$

In general, the network remains stable and unchanged over an extended period of time, resulting in a small calculated cross entropy. Unfortunately, certain anomalous behaviors, like DDoS attacks, can lead to sudden changes in the distribution of occurrences of new source IP addresses over a short time period. These changes can cause significant shifts in cross entropy values.

4.2. Variance ratio differentiation method.

When the anomaly detection method determines that there is an anomaly in a particular time interval, the method is promptly initiated. The process entails the following steps.

1) The time series construction module segments the abnormal time interval into smaller unit times, T. The number of occurrences of each newly detected source IP address for each T is tallied, sorted in descending order, and counted. Concurrently, the number of dissimilar new source IP addresses is also counted at the same time.

2) The time series construction module computes the cross entropy value for each adjacent unit time, P and Q, based on the new source IP address distribution. This process generates a cross entropy time series for the abnormal time interval. In order to calculate the entropy value, the calculation length is determined as the minimum N between the number of new source IP addresses in P and Q. The probability p_i and q_i of each new source IP address in P and Q is then calculated, followed by the calculation of the cross entropy value in accordance with Equation (5).

3) The statistical differentiation module leverages the sliding time window method shown in Figure 2 to differentiate the cross entropy time series. The sample variance value obtained from the window size can be indicative of the variance of the entire sample. Increasing the size of the history window yields a closer approximation of the sample variance value, thereby increasing the accuracy. However, excessively large window sizes may increase the system's storage and computing overhead. Therefore, it is crucial to strike a balance between accuracy and system overhead when selecting the window size.

4) To obtain the statistical ratio at current time t, the cross entropy variance, $DetV(t)$, is calculated within the discrimination window ($DisWin$), while the history window ($HisWin$) is used to calculate the $HisV(t)$ variance. The statistical ratio at time t, denoted by $ratio(t)$, is then computed as the square of the ratio of $DetV(t)$ to $HisV(t)$. This

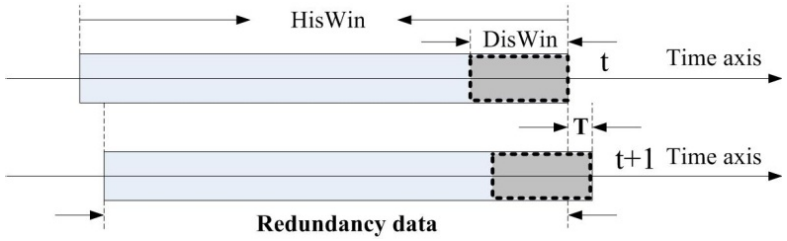


FIGURE 2. Variance ratio differentiation based on sliding time window

$ratio(t)$ provides a measure of how much the cross entropy in the current discrimination window deviates from that in the historical window at time t.

5) If $ratio(t)$ at current time is greater than the threshold $ratio_{threshold}$, it is considered as a possible DDoS attack during the abnormal time interval. If $ratio(t)$ is less than or equal to the threshold, it is regarded as a Flash Crowd.

The threshold $ratio_{threshold}$ is determined as follows.

First, calculate the mean $\mu = \frac{\sum ratio(i)}{n}$ of all variance ratios $(ratio(1), ratio(2), \ldots, ratio(n))$ computed during the abnormal time interval.

Then, calculate the standard deviation d of all variance ratios.

Next, calculate the absolute deviation $D_i = |ratio(i) - m|$, $i = 1, 2, \ldots, n$, for each value of the variance ratios between the $ratio(i)$ value and the median value m.

Obtain the maximum absolute deviation $D_{\max} = \max(D_1; D_2; \ldots; D_n)$.

Finally, the threshold $ratio_{threshold}$ is calculated as $ratio_{threshold} = \mu + D_{\max} - d$.

5. Experimental Results and Analysis.

5.1. **Experimental environment.** This study aims to test the effectiveness of the proposed method using a network environment depicted in Figure 3. Due to the lack of normal traffic data over an extended period, the team constructed an experimental platform consisting of three routers and three networks. In this platform, each computer within the networks can communicate with the Internet. To collect traffic data from Network 1, Router 1 mirrors the traffic to a collection server. A web server is present in Network 1 to enable web access services, while 50 experimental computers are randomly situated in Networks 2 and 3. Additionally, a large number of work computers are present to carry out various tasks in all three networks, and both internal and external computers can access the web server at any given time. To expand the scale of the experiment, five virtual hosts are created within each experimental computer. Tools for SYN flooding attacks, along with software simulating regular user access to the web server, are installed in each virtual host. In order to induce a Flash Crowd, the software utilizes an event modeling approach described in [24].

In this experiment, the first step is to allow Network 1 to run normally for two weeks, during which time the 24-hour threshold data for the normal Hurst parameter is computed to represent typical user access behavior. Following this, a SYN flooding attack is implemented, with each virtual host generating connection requests at intervals of 10 or 20 milliseconds. For each interval, 50 DDoS attack samples, varying in length and generated at different time periods, are randomly produced. Additionally, 50 Flash Crowd samples are also generated accordingly. The generated abnormal traffic has a duration of approximately 5 to 30 seconds, accounting for about 8% to 16% of the total traffic during the abnormal period, and is distributed randomly throughout a week. The data set for detection and identification is collected based on the traffic in Network 1, as well as the flow data in Routers 2 and 3.

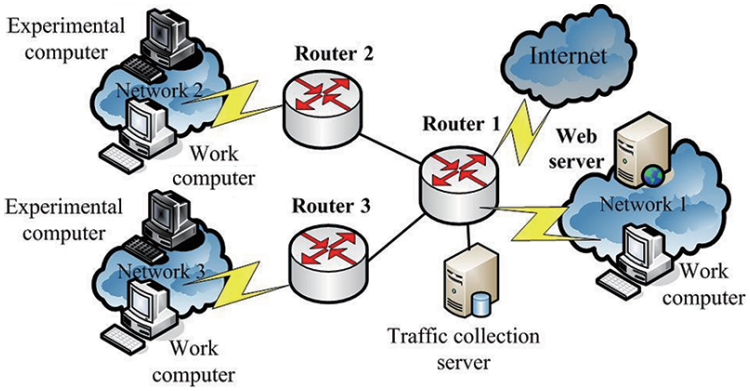


FIGURE 3. Experimental network topology

We extracted the packet header feature of new source IP to calculate the detection metrics. Other IP packet header features such as source port, destination IP, and packet inter-arrival time can also be utilized for anomaly detection. However, in our present experiments, we only selected the new source IP feature, which proved sufficient for detecting DDoS attacks and Flash Crowds. While the anomaly pattern may remain concealed within the collected packets, it is still possible to predict it based on the traffic feature. Before anomaly detection, we used two weeks of normal traffic to generate the threshold for the Hurst parameter. Despite the fact that network traffic is higher on weekdays than on weekends, and that there is a daily traffic pattern with lower traffic at night compared to daytime, we found that the Hurst parameter values calculated at different time periods showed small differences, indicating strong self-similarity. However, the two types of generated samples displayed abnormalities in the Hurst parameter and exhibited distinct characteristics in the distribution change of new source IP. Throughout the experiments, the Hurst parameter is calculated every minute, and the traffic is sampled every 20 milliseconds. Cross entropy is calculated at intervals of 200 milliseconds, and the variance ratio differentiation method is employed with a discrimination window size of 20 and a history window size of 100.

5.2. **Analysis of anomaly detection results.** Figures 4 and 5 display the computed Hurst parameter and its respective normal range during a DDoS attack or Flash Crowd. The solid lines represent the upper and lower limits of the 98% confidence interval of the normal Hurst parameter. The asterisk within the figures represents the correlation between the presently computed Hurst parameter (ordinate value) and the normal Hurst parameter at the specific time interval (abscissa value).

The results of our study, as shown in Figure 4, indicate that during a DDoS attack, the calculated Hurst parameter is 0.563, significantly lower than the confidence interval of normal Hurst values. This signifies that even though the Hurst parameter still exhibits some degree of similarity, it can be accurately identified as an exception using our method. Similarly, as indicated by Figure 5, during a Flash Crowd event, the computed Hurst parameter is 0.653, also lower than the confidence interval of normal Hurst values. This suggests that even though the self-similarity of Flash Crowd traffic is slightly different from the normal situation, it can still be identified as an exception using our method. To conclude, during a DDoS attack or Flash Crowd event, the Hurst parameter may not be less than 0.5. Consequently, the traditional approach of judging exceptions based solely on the Hurst value being less than 0.5 may lead to missed judgments. Therefore, the key advantage of the proposed method is its ability to avoid missed judgments, which ultimately improves the accuracy of detection.

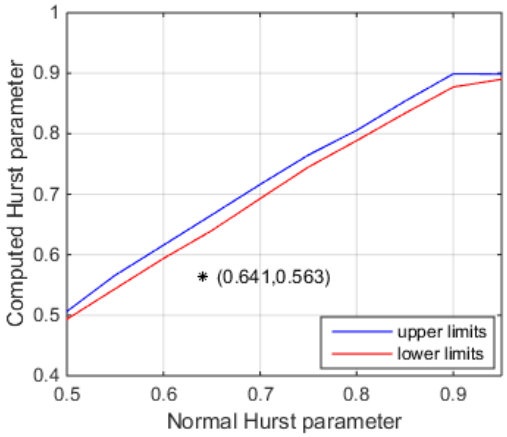


FIGURE 4. Divergence of Hurst parameter during a DDoS attack

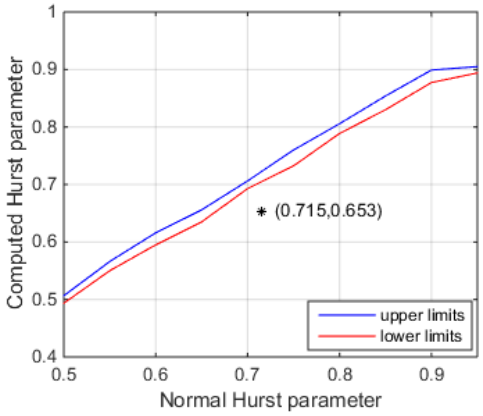
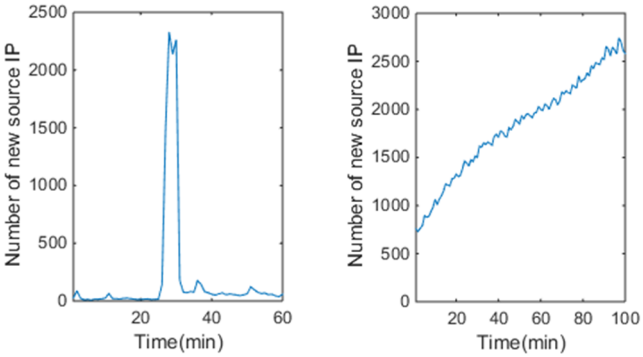


FIGURE 5. Divergence of Hurst parameter during a Flash Crowd

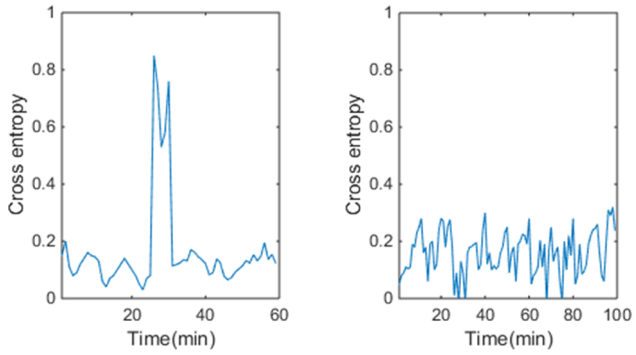
5.3. **Analysis of differentiation results.** The detection phase is capable of identifying anomalies, but it lacks the ability to differentiate between the two types of anomalies. Hence, there arises a need for a differentiation phase, which helps detect DDoS attacks. Figure 6 depicts the changepoint plots for DDoS attacks and Flash Crowds, indicating a marked increase in the number of new source IP addresses in both the CAIDA [25] and WORLD Cup [26] datasets. Additionally, Figure 7 illustrates the trend of cross entropy during a DDoS attack or a Flash Crowd event, thus providing valuable insights into network traffic behavior.

The figure provided in Figure 7(a) reveals that prior to the DDoS attack, the entropy values remained consistently low and exhibited a random pattern of change. However, when the attack began at time point 25, the entropy value significantly and continuously increased for a brief duration before returning to its normal state. This pattern is in alignment with the trend observed in Figure 6(a), wherein the number of new source IP addresses also rose and fell during the attack. In Figure 7(b) it is evident that the cross entropy value remained low and showed random fluctuations during the Flash Crowd. In contrast to Figure 6(b), the number of new source IP addresses appeared to gradually increase, making it challenging to distinguish them from DDoS attacks. It is evident that cross entropy, a measure of the level of change in new source IP address aggregation, is better suited in detecting these anomalies when compared to the statistics of the number of new source IP addresses.



(a) DDoS attacks (b) Flash Crowds

FIGURE 6. Trends of DDoS attacks and Flash Crowds in the number of new source IP addresses



(a) DDoS attacks (b) Flash Crowds

FIGURE 7. Trends of DDoS attacks and Flash Crowds in cross entropy

5.4. **Comparison and analysis with methods based on other entropy.** The performance of this proposed method on the generated dataset is evaluated and the results are presented in Table 1. Analyzing Table 1 reveals two key findings. 1) The anomaly detection method is highly effective and can efficiently identify the majority of samples belonging to the two different types of anomalies. 2) Utilizing the cross entropy method to discern the detected anomalies led to notable outcomes – as the intensity of the DDoS attacks increased, the likelihood of false identification as Flash Crowd decreased considerably.

TABLE 1. Detection and differentiation results of the proposed method

		Differentiation results		Detected/All
		Flash Crowd	DDoS	
Anomaly	1 request/20 ms DDoS	6	43	49/50
	2 requests/20 ms DDoS	2	47	49/50
	Flash Crowd	45	3	48/50
	Total	53	93	146/150

In this article, Shannon entropy is utilized as the fundamental entropy measure for benchmarking. During comparative experiments, Shannon entropy is implemented directly instead of using cross entropy, followed by the application of the variance ratio statistical differentiation method for identification purposes. At the same time, we also

compared the ϕ-entropy-based method proposed in [10], which is a recent and highly effective entropy-based method. During the experiment, we adopted the same method as described in the literature to choose the generalized parameter α of 0.5, the time window of 5 seconds and the sampling period of 100 seconds. And we computed the standard deviation of ϕ-entropy values by monitoring the normal network traffic under consideration using different window sizes of $t = 0.1$, 0.5, 1, and 5 seconds. The resulting findings are presented in Table 2 and Table 3, which incorporate six key metrics: True Positive Rate (TPR), precision, Overall Accuracy (OA), F1-score, False Positive Rate (FPR), and False Negative Rate (FNR). TPR denotes the proportion of correctly identified anomalies of a specific type to the total count of samples for that type. Precision describes the proportion of correctly identified anomalies of a specific type to the total count of identified anomalies for that type. OA estimates the proportion of correctly identified anomalies of their corresponding type to the total number of samples. F1-score is a harmonic mean of precision and OA. Specifically, FPR denotes the proportion of normal samples misrepresented as abnormal to the total number of identified abnormal samples. In contrast, FNR signifies the ratio of misclassified abnormal samples as normal to the total number of known abnormal samples. Overall, Table 2 and Table 3 demonstrate the following results.

Table 2. Detailed results comparison between three kinds of entropy

Anomaly type	Cross entropy		Shannon entropy		ϕ-entropy	
	TPR	Precision	TPR	Precision	TPR	Precision
1 request/20 ms DDoS	86%	93.5%	78%	86.7%	86%	91.5%
2 requests/20 ms DDoS	94%	94%	88%	88%	92%	92%
Flash Crowd	90%	91.8%	84%	84.8%	88%	92.6%

Table 3. Overall results comparison between three kinds of entropy

Method	OA	Precision	FPR	FNR	F1-score
Cross entropy	90%	92.5%	3.4%	2.7%	91.2%
Shannon entropy	83.3%	85.6%	3.4%	2.7%	84.4%
ϕ-entropy	88.7%	92.4%	2.8%	4%	90.5%

1) The detailed TPR and precision values reveal that the cross entropy method outperforms the Shannon entropy based method in two categories of anomaly detection. Moreover, the proposed method yielded an improvement of almost 7% in overall precision and 6.7% in OA. 2) In scenarios where the DDoS attack rate is low, the proposed method exhibits superiority over Shannon entropy based method in all aspects. This can be primarily attributed to the fact that Shannon entropy only accounts for the spatial distribution of network traffic, while the cross entropy method takes into account the temporal and spatial distribution of network traffic, which thereby enhances its ability to identify low-rate attacks. 3) Additionally, our method has much higher F1-score than the other two entropy based methods, which is closer to 1, indicating that our method achieves a better balance between precision and OA. 4) Based on various metrics, our approach is not inferior to the ϕ-entropy based method but rather outperforms it in terms of OA, FNR, and F1-score value. Furthermore, due to the utilization of more flow characteristics in the compared method, it lacks the same level of real-time performance as our approach.

5.5. **Comparison and analysis with the method in [14].** Based on the experimental results presented in [14], the reference method demonstrates a remarkable capability for detecting and recognizing anomalous network flows. The method entails an initial computation of the flow entropy of network flows in the upstream router, and then identifies the primary abnormal flows whenever a significant decrease in flow entropy is detected. Subsequently, the method calculates the Sibson information distance between the primary abnormal flows. If the resulting value of the distance measure is found to be in proximity to 0, then a DDoS attack is concluded to have occurred; otherwise, a Flash Crowd event is inferred. The experimental parameters used in this reference method include several key factors. Firstly, to minimize computational time, all network flows are sorted by data volume, and only the top 500 network flows are selected for calculating flow entropy. Additionally, a decline threshold of 0.03 is employed for detecting abnormal flows in the upstream router. Secondly, to calculate the Sibson information distance, the primary abnormal flows identified are sampled at intervals of 100 ms for a period of 600 samples. Finally, an information distance threshold of 5×10^{-4} is used to identify a DDoS attack, which is triggered when the current calculated information distance falls below this threshold. These specific parameters are crucial for ensuring the accuracy and effectiveness of the reference method in detecting and identifying both DDoS attacks and Flash Crowd events.

The present study employed the reference method to detect and identify DDoS attacks and Flash Crowd events in the generated dataset. Table 4 displays the detailed detection and differentiation outcomes of the reference method, while Table 5 presents the comparison results of both the proposed and reference methods. In Table 5 Area Under Curve (AUC) is a metric defined as the area under the Receiver Operating Characteristic (ROC) curve, which is used to measure the performance of a binary classifier. The higher the AUC value, the better the classifier. The results presented in Table 4 and Table 5 reveal several key findings. Firstly, the reference method detects fewer abnormal samples than the proposed method outlined in Table 1. This is attributed to the fact that the reference method utilizes the declining range of flow entropy to detect abnormal flows, which may not be sufficiently sensitive to sudden or minor fluctuations. As a result, it fails to detect abnormal flows that constitute a relatively small proportion of the overall network traffic. Additionally, the reference method appears to identify network flows that transmit large files as the primary source of abnormal activity, resulting in a higher number of false positives compared to the proposed method. Seconly, the results demonstrate that the proposed method has a higher overall accuracy compared to the reference method,

Table 4. Detection and differentiation results of [14]

		Differentiation results		Detected/All
		Flash Crowd	DDoS	
Anomaly	1 request/20 ms DDoS	2	44	46/50
	2 requests/20 ms DDoS	2	45	47/50
	Flash Crowd	41	3	44/50
	Total	45	92	137/150

Table 5. Results comparison between the proposed method and [14]

Method	OA	Precision	FPR	FNR	F1-score	AUC
Our method	90%	92.5%	3.4%	2.7%	91.2%	90.3%
Method in [14]	86.7%	94.9%	7.3%	8.7%	90.6%	88.5%

with notably lower false positive and false negative rates. However, the reference method exhibits superior precision, despite its greater tendency to misclassify normal samples as abnormal. This suggests that the reference method has greater reliability in detecting two abnormal behaviors. In contrast, our method has much higher AUC and F1-score value than the reference method, indicating that our method can better classify these two types of anomalies and have a better balance between precision and OA. Furthermore, the deployment of the reference method across at least three routers is often impractical in real-world network environments, limiting its ease of use compared to the proposed method.

5.6. **Comparison and analysis with the method in [11].** In [11], the approach involves first extracting all data flows within the network, then counting the occurrences of different packet sizes and destination IP addresses for each data flow, which are used to establish two detection statistics. Specifically, when DDoS attacks or Flash Crowds occur, a large number of packets in the network will have the victim host's destination IP address, resulting in a more centralized distribution of destination IP addresses. This distribution is measured by calculating the Tsallis entropy, which detects the occurrence of DDoS attacks or Flash Crowds. Additionally, differences in packet size distribution are used to distinguish between the two events. The experiment was conducted with the following parameter settings. 1) A sampling interval of 1 second was used for both traffic characteristics. 2) The size of the sliding window was set to 60, or 1 minute. 3) The β value used to set the threshold was set to 0.5.

The detailed detection and differentiation outcomes of the reference method for the generated dataset are presented in Table 6, while the comparison results of both methods are displayed in Table 7. These tables show the following. 1) In comparison with Table 1, the number of abnormal samples detected by the reference method is also lower than the proposed method in this study, and its false positive rate is considerably higher. On the one hand, when the Web server's normal traffic in Network 1 is high, the concentration of the destination IP address might be high; on the other hand, when the threshold value of the Tsallis entropy is high, many peaks are observed. These reasons might contribute to the high false positive rate of the reference method because it uses Tsallis entropy to calculate the concentration of the destination IP address. 2) The proposed method exhibits superior overall accuracy, precision, false positive rate, and false negative rate when compared to the reference method. A comparison between the values in Table 1 and Table 6 confirms that the reference method tends to incorrectly identify DDoS attacks as Flash

Table 6. Detection and differentiation results of [11]

		Differentiation results		Detected/All
		Flash Crowd	DDoS	
Anomaly	1 request/20 ms DDoS	8	39	47/50
	2 requests/20 ms DDoS	4	44	48/50
	Flash Crowd	45	2	47/50
	Total	57	85	142/150

Table 7. Results comparison between the proposed method and [11]

Method	OA	Precision	FPR	FNR	F1-score
Our method	90%	92.5%	3.4%	2.7%	91.2%
Method in [11]	85.3%	90.1%	8.5%	5.3%	87.6%

Crowds. This happens because the comparison method operates under the assumption of a relatively consistent size of DDoS attack packets. However, during the experiment, the size of attack packets was subject to randomness. 3) Our method has much higher F1-score value than the reference method, which is closer to 1, indicating that our method achieves a better balance between precision and OA.

6. **Conclusions.** Upon closer analysis, it has been observed that both DDoS attacks and Flash Crowds result in a significant increase in the occurrences of new source IP addresses. Taking this into account, we propose leveraging the Whittle estimator to calculate Hurst parameters and their confidence intervals, in order to detect these two types of anomalies, and subsequently establish a threshold dataset for normal Hurst parameter. Additionally, our analysis indicates a marked variation in the dispersion of new source IP addresses between DDoS attacks and Flash Crowds. Consequently, we introduce a variance ratio statistical method based on cross entropy, designed specifically to distinguish between the two types of anomaly. In conclusion, the results of our experiments carried out in a practical network environment demonstrate that the proposed detection and differentiation mechanism, which combines self-similarity and cross entropy, can effectively distinguish between DDoS attacks and Flash Crowds. Furthermore, the mechanism presents several advantages, including a straightforward calculation process, high real-time performance, and superior detection efficiency, all of which can be achieved by leveraging minimal traffic characteristics. However, given the increasingly diversified and covert nature of DDoS attacks on Web servers, single-traffic-feature-based measurements may not be sufficient to detect subtle variations in such attacks. Therefore, future research aims to examine and analyze the extraction of more comprehensive and versatile traffic features to further improve identification accuracy.

REFERENCES

[1] J. Kang, Z. Zhang and J. B. Ju, Protect e-commerce against DDoS attacks with improved D-WARD detection system, *Proc. of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service*, Hong Kong, pp.100-105, 2005.

[2] Y. Wang, Z. Sun and Y. Han, Network attack path prediction based on vulnerability data and knowledge graph, *International Journal of Innovative Computing, Information and Control*, vol.17, no.5, pp.1717-1730, 2021.

[3] B. L. Xie, S. Z. Jiang and Q. S. Zhang, Application-layer DDoS attack detection based on request keywords, *Computer Science*, vol.40, no.7, pp.121-125, 2013.

[4] R. Saravanan, S. Shanmuganathan and Y. Palanichamy, Behavior based detection of application layer distributed denial of service attacks during flash events, *Turkish Journal of Electrical Engineering & Computer Sciences*, vol.24, no.12, pp.510-523, 2016.

[5] F. Y. Wang, S. F. Cao, J. Xiao et al., Method of detecting application-layer DDoS based on the out-linking behavior of Web community, *Ruan Jian Xue Bao/Journal of Software*, vol.24, no.6, pp.1263-1273, 2013.

[6] O. Tinubu, A. Sodiya and O. Ojesanmi, A behavioral model for characterizing flooding distributed denial of service attacks, *International Journal of Information Technology*, vol.15, no.2, pp.955-964, 2023.

[7] K. S. Park, V. S. Pai, K. W. Lee et al., Securing web service by automatic robot detection, *2006 USENIX Annual Technical Conference*, pp.255-260, 2006.

[8] M. Walfish, M. Vutukuru, H. Balakrishnan et al., DDoS defense by offense, *ACM SIGCOMM Computer Communication Review*, vol.36, no.4, pp.303-314, 2006.

[9] S. Behal and K. Kumar, Detection DDoS attacks and flash events using information theory metrics – An empirical investigation, *Computer Communications*, vol.103, no.5, pp.18-28, 2017.

[10] S. Behal and K. Kumar, Detection of DDoS attacks and flash events using novel information theory metrics, *Computer Networks*, vol.116, no.7, pp.96-110, 2017.

[11] J. David and C. Thomas, Discriminating flash crowds from DDoS attacks using efficient thresholding algorithm, *Journal of Parallel and Distributed Computing*, pp.79-87, 2021.

[12] W. T. Jiang, Y. Gu, D. N. Ren et al., DDoS attacks and flash crowds detection based on flow characteristics in SDN, *Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition)*, vol.31, no.3, pp.420-426, 2019.

[13] C. H. Sekhar, K. Venkata Rao and M. H. M. Krishna Prasad, Deep neural network empowered bi-directional cross GAN in context of classifying DDoS over flash crowd event on web server, *Multimedia Tools and Applications*, https://doi.org/10.1007/s11042-023-15030-8, 2023.

[14] Y. Tao and S. Yu, DDoS attack detection at local area networks using information theoretical metrics, *The 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, Melbourne, VIC, Australia, pp.233-240, 2013.

[15] S. Yu, W. L. Zhou, W. J. Jia et al., Discriminating DDoS attacks from flash crowds using flow correlation coefficient, *IEEE Transactions on Parallel and Distributed Systems*, vol.23, no.6, pp.1073-1080, 2012.

[16] A. A. A. Dasilva, L. S. Silva, E. L. Bezerra et al., A proposal to distinguish DDoS traffic in flash crowd environments, *International Journal of Information Security and Privacy (IJISP)*, vol.16, no.1, pp.1-16, 2022.

[17] L. Liu, Y. Yue, Z. J. Wu et al., LDoS attack detection method based on traffic classification prediction, *IET Information Security*, vol.16, no.2, pp.86-96, 2022.

[18] R. K. Deka and D. K. Bhattacharyya, Self-similarity based DDoS attack detection using Hurst parameter, *Security and Communication Networks*, vol.9, no.17, pp.4468-4481, 2016.

[19] G. Kaur, V. Saxena and J. P. Gupta, Detection of TCP targeted high bandwidth attacks using self-similarity, *Journal of King Saud University – Computer and Information Sciences*, vol.32, no.1, pp.35-49, 2020.

[20] Y. T. Zhao, W. B. Zhang, X. Feng et al., A classification detection algorithm based on joint entropy vector against app-layer DDoS attack, *Security and Communication Networks*, vol.2018, no.5, pp.1-8, 2018.

[21] M. Sachdeva, K. Kuma and G. Singh, A comprehensive approach to discriminate DDoS attacks from flash events, *Journal of Information Security and Applications*, vol.26, no.2, pp.8-22, 2016.

[22] H. Kettani and J. A. Gubner, A novel approach to the estimation of the Hurst parameter in self-similar traffic, *The 27th Annual IEEE Conference on Local Computer Networks*, Tampa, FL, USA, pp.160-165, 2002.

[23] A. Tayfun, B. Suleyman, E. K. Melike et al., Periodicity-based anomalies in self-similar network traffic flow measurements, *IEEE Transactions on Instrumentation and Measurement*, vol.60, no.4, pp.1358-1366, 2011.

[24] S. Bhatia, G. Mohay, D. Schmidt et al., Modelling web-server flash events, *The 11th IEEE International Symposium on Network Computing and Applications*, Cambridge, MA, USA, pp.79-86, 2012.

[25] CAIDA, *DDoS 2007 Attack Dataset*, https://catalog.caida.org/dataset/ddos_attack_2007, Accessed on 2022-02-15.

[26] *ITA. WorldCup98*, http://ita.ee.lbl.gov/html/contrib/WorldCup.html, Accessed on 2022-02-15.

Author Biography

Ruoyu Yan received his M.S. degree in Computer Science in 2004 from Beijing Jiaotong University and Ph.D. degree in System Engineering in 2010 from Xi'an Jiaotong University. He is an associate professor of College of Computer and Information Engineering at Henan University of Economics and Law, Zhengzhou, China. His research interests focus on network security, information security and network measurement.

Yingfeng Wang received her M.S. degree in Computer and Applied Technology in 2004 from Xi'an Petroleum University and Ph.D. degree in Electronic Science and Technology in 2011 from Xi'an University of Electronic Science and Technology. She is an associate professor of College of Computer and Information Engineering at Henan University of Economics and Law, Zhengzhou, China. Her research interests include multi core system architecture, software and hardware collaborative scheduling, and high-performance computing.