Malware Traffic Classification Based on Recurrence Quantification Analysis

Zheng-Zhi Tang[1,2], Xue-Wen Zeng[1], Zhi-Chuan Guo[1], and Man-Gu Song[1]

(Corresponding author: Zhi-Chuan Guo)

National Network New Media Engineering Research Center & Institute of Acoustics, Chinese Academy of Sciences[1]
Beijing 100190, China
(Email: guozc@dsp.ac.cn)
University of Chinese Academy of Sciences, Beijing 100049, China[2]

(Received Sept. 17, 2018; Revised and Accepted Mar. 2, 2019; First Online July 16, 2019)

Abstract

To characterize the behavioral characteristics of different malware traffic more intuitively and identify malware traffic more accurately, a novel analysis and identification method based on recurrence property of malware traffic is proposed. According to the real malware traffic sequences generated by different malwares, a high-dimensional phase space of the malware traffic sequences is constructed, and then the recurrence properties of the state trajectories of malware traffic are analyzed to reveal their inherent behaviors. By analyzing feature vector acquired by Recurrence Quantification Analysis (RQA) statistically and being combined with machine learning, malware traffic can be well identified. Comparing with the traditional method which uses the common flow statistical features, the proposed method has higher classification accuracy (about 96.55%) using fewer features.

Keywords: Recurrence Plots; Recurrence Quantification Analysis; Recurrence Property; Malware Traffic

1 Introduction

The rapid growth of network traffic has enriched the Internet content, while the network security issue has become increasingly prominent. Viruses, Trojans, worms and other malicious software hidden in the network not only effect the service quality of Internet Service Providers (ISP), but also pose great challenges in the field of data security and privacy protection of Internet users for the cloud computing [1,2] or cloud storage service [10,14], and even threaten national security. Therefore, the detection and classification of malware behavior has become the focus of current researchers.

At present, the detection of malware behavior mainly focuses on the detection of behavioral characteristics of malware itself [4], but the general malware itself has strong concealment and can hardly detect. Therefore, it is possible to detect and analyze the malware traffic behavior. The most important purpose of the analysis for network traffic behavior is to detect and discover some abnormal behavior of network traffic. Currently, the data attributes used to detect abnormal behavior are mainly statistical measurements of network traffic at different composition sizes. The detection methods for extracting these data attributes can be categorized into two main categories as follows [6]:

1) The dimension values of the network packet header are taken as data attributes directly, such as source/destination IP, source/destination port, protocol type, packet length and time of the packet;

2) The statistical characteristics of network traffic are used as data attributes, such as the traffic bytes between two hosts in a fixed time, the number of packets, the number of flows, and traffic entropy.

This paper discusses the detection and identification of malware from the perspective of traffic classification, and classifies the traffic generated by malware during network communication to identify malware traffic. In work of [21], the authors used the first 784 bytes of each session to form a 28*28 image, and then combined the convolutional neural network classifier to classify the malware traffic. In work of [13], the authors presented a novel malware classification method based on clustering of flow features and sequence alignment algorithms for computing sequence similarity, which represents network behavior of malware. However, the flow features used by authors include the IP address and port number, which are not rigorous. In work of [20], the authors used deep learning techniques to malware classification by their binary files. In work of [23], the authors demonstrated how ELIDe identifies malware within network traffic based on partially trained malware signature patterns that have significant weighted values within the classifier's weight vector. In work of [8,19], the authors used common flow statistical features to classify network protocols or applications and achieve good results. But they did not involve malware traffic. Currently, some researchers point out that the Internet is a complex network system, and its traffic behavior has nonlinear, non-stationary and other chaotic characteristics [3, 7]. In work of [24], the authors applied non-linear theory to analyze the traffic behavior of normal network applications, revealing the inherent characteristics of network behavior for different normal network applications. But nearly there is no research on the application classification issue by using recurrence property. In this paper, we use non-linear theory to analyze the inherent characteristics of malware traffic behavior, and use recurrence quantification analysis to extract the features of normal application traffic and malware traffic. Then it is combined with machine learning to classify and identify.

The main contributions of this paper are as follows:

- We propose a flow feature extraction method based on recurrence quantification analysis for malware traffic or normal traffic classification;
- For malware traffic or normal traffic, we firstly obtain TCP or UDP flows according to the five-tuple. For TCP or UDP flows, we obtain fixed-length sequences of packet size. Then we extract feature vectors by using recurrence quantification analysis on these sequences. Finally, the feature vectors are used as input of machine learning to classify;
- We directly apply common flow statistical feature based classification methods to the malware traffic classification. We extract flow feature set from raw network capture by using open source Netmate tool.
- We carry out many experiments on the machine learning to evaluate the performance of flow feature extraction method proposed. We compared the flow features that we extracted with the flow features that are commonly used to traffic classification in term of classification accuracy. In the case of same feature number for two methods, the proposal outperforms 11.99% the common technique in term of classification accuracy.

The rest of this paper is organized as follows. In Section 2, we will elaborate on recurrence plots and recurrence quantification analysis. In Section 3, we give a detailed description of proposed flow feature extraction algorithm based on recurrence quantification analysis and establish an analytical framework combined with machine learning to evaluate its performance. In Section 4, we explain the experimental process and analyze the experimental results. Finally, Section 5 concludes the work and analyzes possible future studies.

2 Recurrence Plots and Recurrence Quantification Analysis

The recurrence plots analysis method was first proposed by Eckman *et al.* in 1987 [5]. It is an important method for visualizing the periodicity, chaos and non-stationarity of time series by recurrence analysis on phase space. At present, it is mainly used for qualitative analysis of nonlinear dynamic systems and suitable for short time series. It also can reveal the internal structure of time series and give prior knowledge about similarity and predictability.

2.1 Phase Space Reconstruction

For the time series of chaotic systems, both the calculation of chaotic invariants or the establishment and prediction of chaotic models are carried out in phase space. Therefore, a phase space reconstruction is a very important step in chaotic time series processing. The phase space reconstruction is to reconstruct the state motion trajectory of the phase space system of the original time series by mapping the one-dimensional time series to the high-dimensional phase space. There are two main methods for phase space reconstruction: derivative reconstruction and coordinate delay reconstruction, which were proposed by Packard *et al.* in 1980 [18]. In the study of chaotic time series, the phase space reconstruction method of coordinate delay is widely used. Assuming an original one-dimensional time series $\{x_1, x_2, ..., x_n\}$, then each row vector of the m-dimensional phase space vector obtained by the phase space reconstruction is:

$$\mathbf{X}_i = \{x_i, x_{i+\tau}, ..., x_{i+(m-1)\tau}\} \tag{1}$$

where $i = 1, 2, ..., n-(m-1)\tau$ and τ is the delay time. It can be seen from Equation (1) that the choice of two parameters, embedding dimension m and delay time τ, is crucial for phase space reconstruction. Only by properly selecting the embedding dimension and delay time can the characteristics of the original system be accurately characterized. At present, there are many calculation methods for embedding dimension and delay time. We adopt the method that is most commonly used by researchers. We use false nearest neighbors (FNN) for the calculation of embedding dimension and the calculation of delay time uses mutual information (MI) [12].

2.2 Recurrence Plots

The recurrence phenomenon represents the recurrence of the phase space trajectory to a certain state, which is a fundamental property of deterministic dynamical systems, that is, the evolutionary pattern of the system state motion trajectory appears periodic recursive phenomenon [24]. According to the recurrence phenomenon, Eckmann *et al.* proposed the concept of recurrence plots [5]. Through the method of recurrence plots analysis, the motion trajectory of the phase points in the high-dimensional phase space can be visually represented in two-dimensional space.

The recurrence plots consist of white and black points in a two-dimensional square matrix. The white dots in the two-dimensional square matrix indicate that the two phase points are far away, and the black dots indicate that the two phase points are close. The mathematical expression of the recurrence plots is:

$$R_{i,j} = \Theta(\epsilon - ||\mathbf{X}_i - \mathbf{X}_j||), i, j = 1, 2, ..., n - (m-1)\tau \quad (2)$$

where $R_{i,j}$ is a recurrence matrix element, when $R_{i,j} = 0$, it is represented as a white point on the recurrence plots, and when the value is 1, it is represented as a black point on the recurrence plots. $\Theta(x)$ is a Heaviside function, its value is 1 when the variable is greater than or equal to 0, and 0 when the variable is less than 0. $||x||$ is the Euclidean norm of the vector. ϵ is a pre-set threshold distance, and the choice of ϵ is critical for calculating recurrence plots. If its value is too large, the number of black points in the recurrence plots will be large, and its value is too small, which makes the white area in the recurrence plots large. In this paper, we choose 10% of the maximum diameter of the phase space as ϵ value, which called rule of thumb [16].

2.3 Recurrence Quantification Analysis

The recurrence plots are only qualitative and intuitive to show the recurrence property of the state motion trajectory of nonlinear systems. In the research, it is more desirable to quantitatively analyze. Recurrence Quantification Analysis (RQA) quantifies the characterization of recurrence plots by quantitative parameters, which are proposed by Zbilut *et al.* [22]. In this paper, the six typical RQA features, namely, recurrence rate (RR), determinism (DET), linemax (LMAX), entropy (ENT), laminarity (LAM), and trapping time (TT) are extracted to characterize the malware traffic or benign traffic. These features are used to characterize and describe the intrinsic characteristics of different malware traffic or benign traffic behaviors. Classification and identification are then performed based on the differences between these features. Herein, we give the definitions for RR, DET, and ENT. The detailed definitions for LMAX, LAM, and TT were stated in [15].

The recurrence rate (RR) represents the ratio of the recurrence point number to the entire phase point number, reflecting the density of the recurrence points. The recurrence rate is proportional to the periodicity of the time series. The formula is as follows:

$$RR = \frac{1}{N^2} \sum_{i,j=1}^{N} R_{i,j} \quad (3)$$

where N is the number of phase points and $R_{i,j}$ is the recurrence matrix element. The recurrence rate characterizes the recurrence degree of the system.

The determinism (DET) represents the ratio of the number of recurrence points which parallel to the main diagonal line segment to the number of total recurrence points in the recurrence plots. The determinism is positively correlated with the periodicity and predictability of the time series. The formula is as follows:

$$DET = \frac{\sum_{l=l_{min}}^{N} lP(l)}{\sum_{i,j=1}^{N} R_{i,j}} \quad (4)$$

where l_{min} is the minimum diagonal segment length (generally 2) and $P(l)$ is the frequency of line segments of length l that parallel to the main diagonal. The determinism can be used to quantify the certainty of the system.

Entropy (ENT) represents the Shannon entropy of the 45° diagonal length probability distribution in a recurrence plots. The formula is as follows:

$$ENT = -\sum_{l=l_{min}}^{N} P(l) \ln P(l) \quad (5)$$

where $P(l)$ is the distribution probability of the main diagonal segment with length l, and l_{min} is the initial value of the length in the diagonal structure (generally 2). Entropy can be used to indicate the complexity of system certainty.

3 Classification Method Based on Machine Learning

The Gradient Boosting Decision Tree (GBDT) algorithm is known to improve the performance of a single classifier by combining several base classifiers that outperform every independent one. Now, it performs well in various data mining and machine learning methods. Furthermore, GBDT also performs well among solution methods for class imbalance problems [9]. In this part, we firstly propose a flow feature extraction algorithm based on recurrence quantification analysis. Then we propose the malware traffic classification method combined with GBDT.

3.1 Feature Extraction Algorithm

Algorithm 1 describes the complete flow feature extraction algorithm based on recurrence quantification analysis. The raw packets are processed to obtain the TCP or UDP flows. A flow is defined as all packets that have the same 5-tuple, i.e. source IP, source port, destination IP, destination port and transport protocol. Then all the flows of a normal or malicious application are combined into a PCAP file in order of timestamps, and the obtained PCAP file is processed to obtain a sequence Q of packet sizes. According to the length of initialized RQA sequence is l, the algorithm intercepts RQA sequence samples to extract features from sequence Q in order. Finally, the RQA method is used to obtain the feature set for the sequence Q.

3.2 Classification Method Process

Figure 1 shows the method of malware traffic classification process combined with GBDT. As shown in Figure 1,

Algorithm 1 Feature extraction

```
Begin
Initialize the $RQA\ sequence$ length $l$.
Input raw packets and obtain UDP or TCP flows.
Combine all packets in flows in order of timestamps.
Obtain packet size sequence $Q$ from ordered packets.
$L \Leftarrow$ Get sequence $Q$ length
$n \Leftarrow \lfloor \frac{L}{l} \rfloor$
while $n > 0$ do
    Intercept $RQA\ sequence$ from sequence $Q$ in order
    $Feature \Leftarrow$ RQA($RQA\ sequence$)
    $n \Leftarrow n - 1$
end while
return $Feature$
End
```

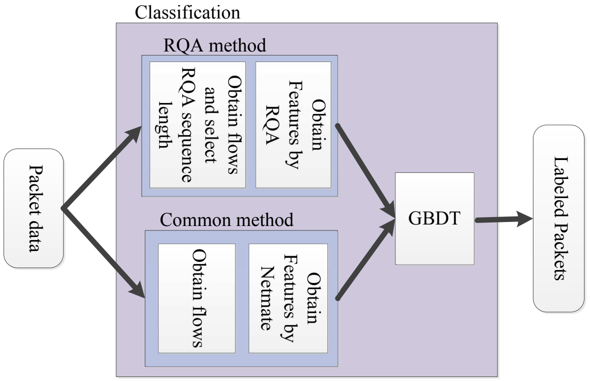

Figure 1: The method of malware traffic classification process combined with GBDT

the raw packets input can be selected by RQA method or commonly used method for feature extraction. Herein, the common feature extraction method is used for comparison experiment. As seen from Figure 1, the raw packets will be processed to obtain the flows, and the length of RQA sequence will be initialized in RQA method. Then the RQA method or the Netmate tool is used to extract the corresponding features respectively. Finally, the extracted features are used as input of the trained machine learning GBDT model for classification. Each of feature vectors input identifies a corresponding category label. In all experiments, we use the GBDT algorithm in the Scikit-learn for multi-classification.

4 Experiment and Result

To demonstrate the advantage of proposed flow feature extraction method based on the recurrence quantification analysis. We compare the flow features that we extracted by RQA with the common flow statistical features which are representative in traffic classification in term of classification performance by experiments.

4.1 Description of Dataset

The dataset used in the experiments is randomly selected from the CTU dataset, USTC-TFC2016 dataset [21] and VPN-nonVPN dataset [8]. A total of 10 normal application traffic and malware traffic capture are randomly selected here. The dataset is shown in Table 1.

Table 1: Malware traffic and benign traffic dataset

Malware traffic		**Benign traffic**	
Name	*Source*	*Name*	*Source*
Zeus	CTU	Hangouts	VPN-nonVPN
Miuref	CTU	HTTPS	CTU
Trickbot	CTU	P2P	CTU
Sennoma	CTU	SFTP	VPN-nonVPN
Artemis	CTU	SMB	USTC-TFC2016

4.2 Experiment Setup and Evaluation Metrics

The experimental platform is DELL R720 server which is equipped with CentOS release 7.3 operate system. The CPU is a 16-cores XeonE5620 2.40 GHz, and the memory is 16 GB. In all experiments, the classifier is GBDT algorithm and we carry out a grid search on parameter space to achieve the best classification accuracy with GBDT parameters are *random_state*=10, *n_estimators*=400, *max_depth*=6. In this paper, four evaluation metrics are used: accuracy (A), precision (P), recall (R), f1 value (F1). Accuracy is used to evaluate the overall performance of a classifier. Precision, recall and f1 value are used to evaluate performance of every class of traffic.

$$A = \frac{TP+TN}{TP+FP+FN+TN} \qquad P = \frac{TP}{TP+FP}$$
$$R = \frac{TP}{TP+FN} \qquad F_1 = \frac{2PR}{P+R} \tag{6}$$

where TP is the number of instances correctly classified as X, TN is the number of instances correctly classified as Not-X, FP is the number of instances incorrectly classified as X, and FN is the number of instances incorrectly classified as Not-X.

4.3 Common Flow Statistical Features for Classification

In the open literature, Moore *et al.* presented one of the earliest results about classification of network flows into protocol categories by using flow features combined with supervised machine learning [17]. The authors studied discriminators, attributes primarily derived from network flows and the feature set consists of 248 statistical characteristics. Follow-up research in this area mainly focused on the selection of flow feature sets and the machine learning methods. In this paper, we extract the common flow statistical feature sets from raw network capture by using open source Netmate tool. The extracted feature set consists of 44 common flow statistical features. We only use 40 common flow statistical features that remove the IP address and port number. A detailed description of the common flow statistical features extracted by Netmate can be found on the Netmate official website.

Herein, the Netmate tool is used to process the data set in Table 1, and the number of flow feature samples of each normal or malicious application is shown in Table 2. In order to reduce the impact of class imbalance on the classification results, the data is under-sampled for the class with a large number of samples, and the SMOTE method is applied to the class with a small number of samples. The final preprocessed result is as shown in Table 2.

As seen in Table 3, for the class imbalance samples, the classification accuracy after samples undersampling is higher than that processed by the undersampling and SMOTE combination method. The classification accuracy after samples undersampling outperforms 1.9%. However, undersampling only reduces the number of classes with a large number of samples and the number of classes with a small number of samples is still small. So the class imbalance is still obvious. This can be seen from Table 2. As seen in Table 4, due to the SFTP sample is the least, the recall of SFTP is only 50% and F1-score is only 67%. The SFTP identification result is very poor. The overall classification result is good by using the combination of undersampling and SMOTE. Except for the precision mean, the recall mean and the F1-score mean are higher. In view of the importance for the small class identification and the suggestion that if the training sample size is too large, a combination of SMOTE and undersampling is an alternative [9]. Finally, a comprehensive consideration is given to the use of samples processed by a combination of under-sampling and SMOTE in follow-up experiments.

4.4 Recurrence Quantification Analysis Based Flow Features for Classification

In this section, the flow features extracted by recurrence quantification analysis are performed through experiments. The reason for selecting the sequence of packet sizes is that the continuation of packet size on the timeline can well show the inherent characteristics of network behavior, such as periodicity, data transmission characteristics and so on for the malware traffic or benign traffic. The difference of network behavior characteristics will also lead to difference in the characteristics of the non-linear dynamic system of network traffic. Therefore, the recurrence analysis for the sequence of malware traffic packet size or benign traffic packet size can reveal its unique network behavior characteristics. They can be classified by machine learning methods based on their unique characteristics.

Herein, the raw packets are processed to obtain samples by Algorithm 1 in Section 3.1. The sequence length of each normal or malicious application is obtained by random sampling method as shown in the Table 5. Then, a subsequence of length n (n = 40, 60, 80, 100 in this paper) is taken as one sample, and finally the number of samples of each normal or malicious application in the case of sub-sequences with different lengths is shown in Table 5.

4.4.1 Embedding Dimension and Delay Time

In Section 2.1, it is mentioned that the false nearest neighbor method and mutual information method are used to calculate the embedding dimension and delay time respectively. According to the principle of embedding dimension is determined by the false nearest neighbor method in [12], by increasing the size of embedding dimension one by one, and then calculating the proportion of adjacent errors under each embedding dimension, the first embedding dimension which makes the proportion close to 0 (less than 0.05) or the proportion of adjacent errors no more reduce is the best embedding dimension. The calculation formula for the adjacent error is as follows:

$$r_i = \frac{||\mathbf{X}_{j+1} - \mathbf{X}_{i+1}||}{||\mathbf{X}_j - \mathbf{X}_i||} \tag{7}$$

where $\mathbf{X}_i$,$\mathbf{X}_j$ is the phase point in the m-dimensional phase space, $\mathbf{X}_{i+1}$,$\mathbf{X}_{j+1}$ is the phase point in the $m+1$-dimensional phase space, r_i is the ratio of the distance from the phase point $\mathbf{X}_{j+1}$ to the phase point $\mathbf{X}_{i+1}$ and the distance from the phase point $\mathbf{X}_j$ to the phase point $\mathbf{X}_i$, if r_i is greater than the determined threshold r (generally 2), then the phase point $\mathbf{X}_i$ and the phase point $\mathbf{X}_j$ are adjacent errors.

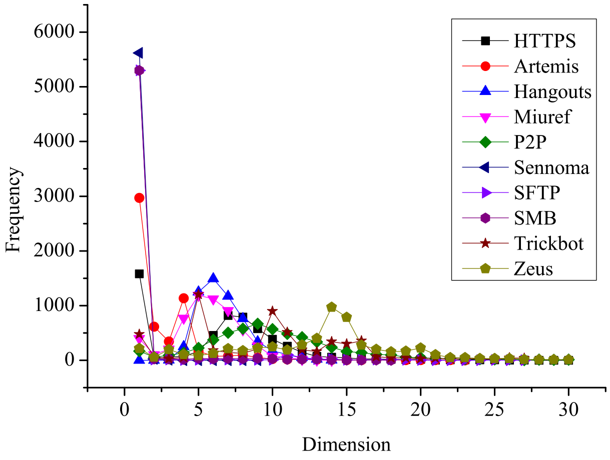

Figure 2: Frequency of the embedding dimension for subsequence samples with length 80

Figure 2 shows the frequency of embedding dimension distribution for subsequence samples with length 80. The

Table 2: Number of flow feature samples for malware traffic and benign traffic

Name	**Origin**	**Undersampling**	**Undersampling+SMOTE**
Zeus	227236	5681	5681
Miuref	4837	4837	4837
Artemis	221758	5687	5687
P2P	2209	2209	4418
HTTPS	6573	5651	5651
SFTP	24	24	5760
Sennoma	653	653	5877
SMB	214	214	5564
Trickbot	94515	5628	5628
Hangouts	1357	1357	5428

Table 3: Classification accuracy of class imbalance after different processing

Method	**Undersampling**	**Undersampling+SMOTE**
Accuracy	0.9678	0.9488

Table 4: Precision, recall, F1-score for class imbalance after different processing

Method	**Undersampling**			**Undersampling+SMOTE**		
Evaluation	*Precision*	*Recall*	*F1-score*	*Precision*	*Recall*	*F1-score*
Miuref	1.00	1.00	1.00	1.00	1.00	1.00
SMB	1.00	1.00	1.00	1.00	1.00	1.00
Zeus	0.97	0.97	0.97	0.97	0.95	0.96
Trickbot	1.00	1.00	1.00	1.00	1.00	1.00
P2P	0.86	0.83	0.84	0.85	0.82	0.84
Hangouts	0.97	0.96	0.96	0.99	0.78	0.87
SFTP	1.0	**0.5**	**0.67**	0.99	0.97	0.98
HTTPS	0.93	0.94	0.93	0.94	0.95	0.95
Artemis	1.00	1.00	1.00	1.00	1.00	1.00
Sennoma	0.96	0.97	0.97	0.80	0.99	0.95
Average	0.969	0.917	0.934	0.954	0.946	0.955

Table 5: The number of samples of each normal or malicious application

Name	**Length**	**40**	**60**	**80**	**100**
Zeus	446400	11446	7566	5650	4509
Miuref	446700	11453	7571	5654	4512
Artemis	445700	11428	7554	5641	4502
P2P	445500	11423	7550	5639	4499
HTTPS	445100	11412	7544	5634	4495
SFTP	444700	11402	7537	5629	4491
Sennoma	446400	11446	7566	5650	4509
SMB	444900	11407	7540	5631	4493
Trickbot	446000	11435	7559	5645	4505
Hangouts	443900	11382	7523	5618	4483

final embedding dimension for each normal or malicious application is the embedding dimension with the largest distribution frequency. The final embedding dimension statistics are shown in Table 6. As seen from Table 6, the embedding dimension of each normal or malicious application remains basically unchanged in the case of subsequence samples with different lengths.

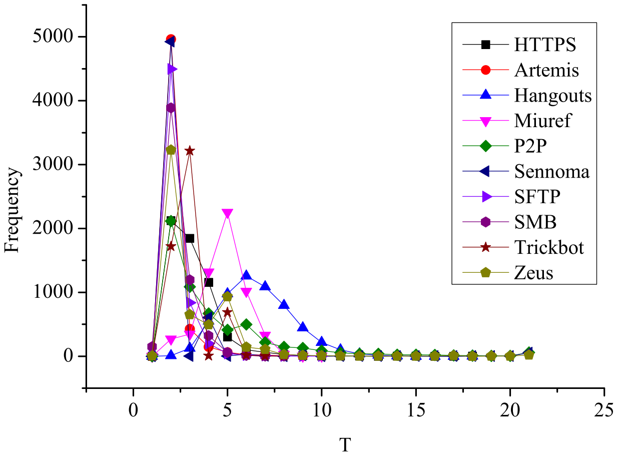

Figure 3: Frequency of the delay time for subsequence samples with length 80

The calculation formula for delay time by the mutual information method is as follows [12]:

$$S = -\sum_{i,j} P_{i,j}(t) \ln \frac{P_{i,j}(t)}{P_i P_j} \tag{8}$$

where P_i and P_j are the probabilities of the points falling into the segments i and segments j in the traffic sequences respectively, $P_{i,j}(t)$ is the probability that the two points with the interval time t fall into the segments i and segments j respectively. The mutual information under each delay time is calculated by the formula, and the delay time corresponding to the first mutual information with local minimum value is the optimal delay time τ.

Figure 3 shows the frequency of the delay time distribution for subsequence samples with length 80. The final delay time for each normal or malicious application is the delay time with the largest distribution frequency. The final delay time statistics are shown in Table 6. As seen from Table 6, the delay time of each normal or malicious application remains basically unchanged in the case of subsequence samples with different lengths.

According to the embedding dimension and delay time, the recurrence plots of each normal or malicious application can be calculated by Formula 2. Then the recurrence quantification analysis method is applied to each recurrence plots, and finally, RR, DET, LAM, ENT, LMAX, and TT are obtained to form the feature vectors of each normal or malicious application. The feature vectors of each normal or malicious application are used as input of GBDT to classify. Figure 4 is the classification accuracy for each normal or malicious application in the case of subsequence samples with length 40, 60, 80, 100, respectively. As shown in Figure 4, as the subsequence length increases, the classification accuracy also increases. When the length is 80, the classification accuracy reaches a maximum value 96.55% and then begins to decrease. This shows that when the subsequence length is 80, the inherent unique characteristics of each normal or malicious application can be well represented by recurrence quantification analysis. In follow-up experiments, we will use the subsequence samples with length 80.

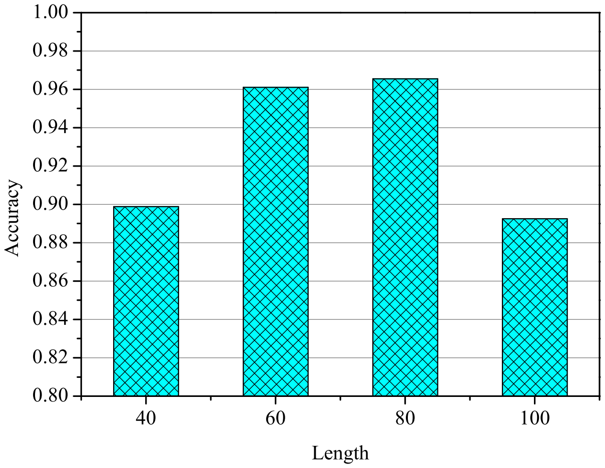

Figure 4: Classification accuracy of normal or malicious applications in the case of subsequences with different lengths

4.5 Comparison

In this section, we will compare the flow features that we extracted by recurrence quantification analysis with the flow statistical features that are commonly used to traffic classification in term of classification performance. Since there are only 6 kinds of flow features extracted by the proposed method, however, the type number of common flow statistical features extracted by the Netmate tool is 40. Herein, we carry out an experiment for choosing the number of common flow statistical features that can reach the best classification accuracy.

Figure 5 is a classification accuracy using different numbers of common flow statistical features randomly selected. It can be seen from the Figure 5 that as the number of random common flow statistical features increases, the classification accuracy increases, while the rising rate becomes slowly. When all the 40 features are used, the classification accuracy reaches the maximum value 94.53%. In follow-up experiments, we compare the 6 and all 40 common flow statistical features randomly selected with the 6 flow features extracted by recurrence quantification analysis in term of classification accuracy respectively.

Table 6: Embedding dimensions and delay time for different length subsequence samples of each normal or malicious application

Name	Parameter	40	60	80	100
HTTPS	*Dimension*	1	1	1	1
	Delay time	2	2	2	2
Artemis	*Dimension*	1	1	1	1
	Delay time	2	2	2	2
Hangouts	*Dimension*	5	5	6	7
	Delay time	5	5	6	6
Miuref	*Dimension*	1	5	5	7
	Delay time	5	5	5	5
P2P	*Dimension*	1	8	9	11
	Delay time	2	2	2	2
Sennoma	*Dimension*	1	1	1	1
	Delay time	2	2	2	2
SFTP	*Dimension*	1	1	1	1
	Delay time	2	2	2	2
SMB	*Dimension*	1	1	1	1
	Delay time	2	2	2	2
Trickbot	*Dimension*	5	5	5	5
	Delay time	3	3	3	3
Zeus	*Dimension*	1	14	14	15
	Delay time	2	2	2	2

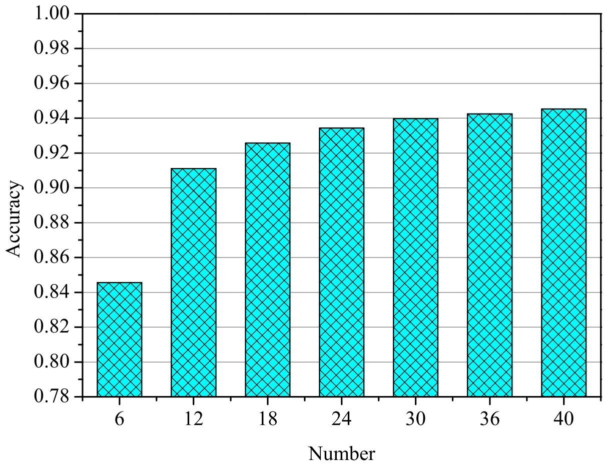

Figure 5: Classification accuracy corresponding to different numbers of common flow features randomly selected

Figure 6 shows the comparison of classification accuracy for the 6 and all 40 common flow statistical features randomly selected with the 6 flow features extracted by recurrence quantification analysis. The left column indicates the classification accuracy by using common flow statistical features, which is abbreviated as CFF for the convenience of description. The right column represents the classification accuracy of the 6 flow features extracted by recurrence quantification analysis. Also for the convenience of description, we abbreviate it as RQA. As shown in Figure 6, when the number of RQA and CFF features is 6, the classification accuracy of RQA is obviously better than CFF, and it outperforms 11.99%. When using all 40 common flow statistical features, the classification accuracy is much higher indeed, but the classification accuracy is still lower than that using 6 flow features extracted by the recurrence quantification analysis. The RQA outperforms 1.67%.

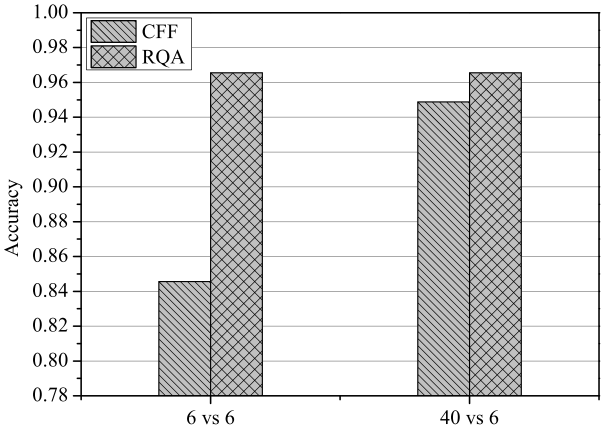

Figure 6: Comparison of classification accuracy between 6 flow features extracted by RQA and random 6 or 40 common flow statistical features

Table 7 shows the precision, recall, F1-score for 6 flow features extracted by RQA and all 40 common flow sta-

Table 7: Precision, recall, F1-score for 6 flow features extracted by RQA and all 40 common flow features

Method	**RQA**			**CFF**		
Evaluation	*Precision*	*Recall*	*F1-score*	*Precision*	*Recall*	*F1-score*
Miuref	1.00	1.00	1.00	1.00	1.00	1.00
SMB	0.92	0.92	0.92	1.00	1.00	1.00
Zeus	0.98	0.98	0.98	0.97	0.95	0.96
Trickbot	0.99	1.00	0.99	1.00	1.00	1.00
P2P	0.97	0.98	0.98	**0.85**	**0.82**	**0.84**
Hangouts	0.99	0.99	0.99	0.99	**0.78**	**0.87**
SFTP	0.96	0.95	0.96	0.99	0.97	0.98
HTTPS	0.91	0.93	0.92	0.94	0.95	0.95
Artemis	0.94	0.93	0.93	1.00	1.00	1.00
Sennoma	1.00	1.00	1.00	**0.80**	0.99	0.95
Average	0.966	0.968	0.967	0.954	0.946	0.955

tistical features. As seen from Table 7, the precision mean, the recall mean, and the F1-score mean of the RQA method are better than the CFF method. When using the RQA method proposed, the classification precision, recall, and F1-score of each normal or malicious application are stable at more than 91%. However, when using the CFF method, some evaluation values of P2P, Sennoma, and Hangouts are significantly less than 90%. From this, it can be concluded that the RQA method proposed has obvious advantages over the CFF method.

4.6 Analysis and Discussion

From the above experiments, obviously, the proposed flow features extracted by RQA performs better than the common flow statistical features extracted by Netmate tool. The proposed RQA method not only has higher classification accuracy, but also has better accuracy mean, recall mean, and F1-score mean. Moreover, the classification precision, recall, and F1-score of each normal or malicious application are stable at more than 91%. The most important thing is that the RQA method proposed uses only 6 flow features, but it performs better than the CFF method by using 40 common flow statistical features. Since only 6 flow features are used, fewer features mean less time consumption for training and classification. Therefore, the proposed RQA method is efficient and possibilities for real-time online classification.

For the classification of malware traffic, in the latest work of [21], the authors used the first 784 bytes of each session to form a 28*28 image, and then combined the convolutional neural network classifier to classify the malware traffic. Finally, its classification accuracy can reach about 99%. In this paper, we do not take the work of [21] as a comparison, mainly because the proposed RQA method is not similar to the method of [21] in principle. In the early work of [13], the authors presented a novel malware classification method based on clustering of flow features and sequence alignment algorithms. However, the authors took into account the IP address and port number in the flow features, which is not rigorous enough. In order to make a more scientific and fair comparison, we chose the current common flow statistical feature based traffic classification method [8, 19]. We use it directly on the malware traffic classification and compare the classification results with the proposed RQA method.

5 Conclusions and Future Work

Malware detection is an active and hot research area in network security issue, which governs identification performance. Motivated by identifying malware through traffic generated by malware communication, we propose a novel flow feature extraction method based on recurrence quantification analysis for malware traffic or normal traffic classification. Our goal is to reduce the high time consumption due to excessive flow features in classification and improve classification performance. The key characteristic of flow feature extraction method based on recurrence quantification analysis is to extract feature vectors by using recurrence quantification analysis on these sequences of packet size. The raw packets are processed to obtain the TCP or UDP flows. Then all the packets of a normal or malicious application are combined into a large PCAP file in order of timestamps, and the obtained PCAP file is processed to obtain a sequence of packet sizes. Finally, the feature vectors extracted by RQA are used as input of machine learning to classify. The sensitivity of this algorithm against different situations is studied. Experiments on the machine learning to evaluate the performance of proposed flow feature extraction algorithm verify that it has fewer flow features but higher classification accuracy than that using the common flow statistical features.

In the future, we will increase the types of malware traffic and benign traffic, and implement experiments in real-time systems, such as real-time data collection and analysis system [11], to evaluate the classification accuracy of the proposed flow extraction method based on RQA over a longer period.

Acknowledgments

This work was supported by The Next-Generation Broadband Wireless Mobile Communications Network National Science and Technology of Major Projects (No. 2017ZX03001019). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers.

References

[1] D. S. AbdElminaam, "Improving the security of cloud computing by building new hybrid cryptography algorithms," *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, pp. 40–48, 2018.

[2] M. H. R. Al-Shaikhly, H. M. El-Bakry, and A. A. Saleh, "Cloud security using markov chain and genetic algorithm," *International Journal of Electronics and Information Engineering*, vol. 8, no. 2, pp. 96–106, 2018.

[3] N. Bigdeli and M. Haeri, "Time-series analysis of tcp/red computer networks, an empirical study," *Chaos Solitons and Fractals*, vol. 39, no. 2, pp. 784–800, 2009.

[4] Z. Chen, Q. Li, P. Zhang, and P. Feng, "Signature selection for kernel malware based on cluster analysis (in chinese)," *Journal of Electronics and Information Technology*, vol. 37, no. 12, pp. 2821–2829, 2015.

[5] J. P. Eckmann and D. Ruelle, "Fundamental limitations for estimating dimensions and lyapunov exponents in dynamical systems," *Physica D Nonlinear Phenomena*, vol. 56, no. 2-3, pp. 185–187, 1992.

[6] Y. Fu, H. Li, X. Wu, and J. Wang, "Detecting apt attacks: a survey from the perspective of big data analysis (in chinese)," *Journal on Communications*, vol. 36, no. 11, pp. 1–14, 2015.

[7] K. Fukuda, "Observations and possible causes of phase transition phenomena in internet traffic," *Ipsj Magazine*, vol. 45, pp. 603–609, 2004.

[8] D. G. Gerard, L. A. Habibi, M. M. S. IsIam, and G. Ali, "Characterization of encrypted and vpn traffic using time-related features," in *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP 2016)*, pp. 404–414, 2016.

[9] H. Guo, Y. Li, S. Jennifer, M. Gu, Y. Huang, and B. Gong, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems With Applications*, vol. 73, pp. 220–239, 2017.

[10] M. S. Hwang, T. H. Sun, and C. C. Lee, "Achieving dynamic data guarantee and data confidentiality of public auditing in cloud storage service," *Journal of Circuits Systems and Computers*, vol. 26, no. 5, 2017.

[11] D. S. Jiang, L. Xue, W. X. Kai, and L. C. Mei, "Design of real-time data collection and analysis system based on spark streaming (in chinese)," *Network New Media*, vol. 6, no. 5, 2017.

[12] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis.* New York: Cambridge University Press, 2004.

[13] H. Lim, Y. Yamaguchi, H. Shimada, and H. Takakura, "Malware classification method based on sequence of traffic flow," in *Proceedings of the 1st International Conference on Information Systems Security and Privacy (ICISSP 2015)*, pp. 230–237, France, 2015.

[14] C. W. Liu, W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of attribute-based access control with user revocation in cloud data storage," *International Journal of Network Security*, vol. 18, no. 5, pp. 900–916, 2016.

[15] N. Marwan, N. Wessel, U. Meyerfeldt, A. Schirdewan, and J. Kurths, "Recurrence plot based measures of complexity and their application to heart-rate-variability data," *Phys. Rev. E*, vol. 66, p. 026702, 2002.

[16] Thiel M Marwan N, Romano M C, "Recurrence plots for the analysis of complex systems," *Physics Reports*, vol. 438, no. 5, pp. 237–329, 2007.

[17] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *Proceedings of the International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2005)*, pp. 50–60, Banff, Alberta, Canada, 2005.

[18] N. H. Packard, J. P. Crutchfield, and J. D. Farmer, "Geometry from a time series," *Physical Review Letters*, vol. 45, no. 9, pp. 712–716, 1980.

[19] A. Pektas and T. Acarman, "Identification of application in encrypted traffic by using machine learning," in *Proceedings of the 5th International Conference on Man-Machine Interactions (ICMMI 2018)*, vol. 659, pp. 545–554, 2018.

[20] R.K. Rahul, T. Anjali, V. K. Menon, and K. P. Soman, "Deep learning for network flow analysis and malware classification," *Communications in Computer and Information Science*, vol. 746, pp. 226–235, 2017.

[21] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Proceedings of the 31st International Conference on Information Networking (ICOIN 2017)*, pp. 712–717, Da Nang, Vietnam, January 2017.

[22] C. L. Webber and J. P. Zbilut, "Dynamic assessment of physiological system and state using recurrence plot strategies," *Appl Physiol*, vol. 76, pp. 965–973, 1994.

[23] K. F. Yu and R. E. Harang, "Machine learning in malware traffic classifications," in *Proceedings of IEEE Military Communications Conference (MILCOM 2017)*, pp. 6–10, 2017.

[24] J. Yuan, J. Wang, Q. Li, and X. Chen, "Recurrence based nonlinear analysis for network application traffic (in chinese)," *Journal of Tsinghua University (Science and Technology)*, vol. 54, no. 4, 2014.

Biography

Zheng-Zhi Tang a Ph.D. candidate in signal and information processing from National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences (IACAS) and University of Chinese Academy of Sciences. His research interests include Network Security, Information Safety and ML (Machine learning).

Xue-Wen Zeng received the B.Sc. degree from Shanghai Jiao Tong University, Shanghai, China, and the M.Sc. and Ph.D. degrees in signal and information processing from Institute of Acoustics, Chinese Academy of Sciences (IACAS), Beijing, China. He is currently working at National Network New Media Engineering Research Center, IACAS as a research professor. His research interests include network new media technology, media information security, multimedia communication, digital broadcasting and signal processing.

Zhi-Chuan Guo received the B.Sc. degree in optical technology and photoelectric instrument from Wuhan University, Wuhan, China, and the Ph.D. degree in electronic circuit and system from University of Science and Technology of China, Hefei, China. He is currently working at National Network New Media Engineering Research Center, IACAS as an associate research professor. His research interests include network new media technology and FPGA hardware acceleration technology.

Man-Gu Song received the B.Sc. degree in computer science and technology from the Tianjin University of Technology and Education, Tianjin, China, and the M.Sc degree in Electronics and Communication Engineering from the School of Microelectronics, Chinese Academy of Science, Beijing, China. She is currently working at National New Media Engineering Research Center, IACAS as a research assistant. Her current research interests include FPGA hardware accelerate technology and research on national secret algorithm.