Hindawi
Security and Communication Networks
Volume 2019, Article ID 2674684, 8 pages
https://doi.org/10.1155/2019/2674684

Research Article

A Feature Extraction Method of Hybrid Gram for Malicious Behavior Based on Machine Learning

Yuntao Zhao,[1,2] Bo Bo,[1] Yongxin Feng,[1] ChunYu Xu,[1] and Bo Yu[1]

[1] *School of Information Science and Engineering, Shenyang Ligong University, Shenyang 110159, China*
[2] *College of Information Science and Engineering, Northeastern University, Shenyang 110819, China*

Correspondence should be addressed to Yongxin Feng; fengyongxin@263.net

Received 12 October 2018; Accepted 20 January 2019; Published 4 February 2019

Guest Editor: Jiageng Chen

With explosive growth of malware, Internet users face enormous threats from Cyberspace, known as “fifth dimensional space.” Meanwhile, the continuous sophisticated metamorphism of malware such as polymorphism and obfuscation makes it more difficult to detect malicious behavior. In the paper, based on the dynamic feature analysis of malware, a novel feature extraction method of hybrid gram (H-gram) with cross entropy of continuous overlapping subsequences is proposed, which implements semantic segmentation of a sequence of API calls or instructions. The experimental results show the H-gram method can distinguish malicious behaviors and is more effective than the fixed-length n-gram in all four performance indexes of the classification algorithms such as ID3, Random Forest, AdboostM1, and Bagging.

1. Introduction

With the development of computer and Internet technology, *mal*ware soft*ware* (*malware*) such as Trojans, viruses, and worms also constantly mutates, self-renews, and becomes one of most serious threats to Cyberspace. Malware is a program that performs malicious tasks on a computer system, which implements control by changing or destroying process execution flow. In recent years, there are endless security incidents from malware. For example, in May of 2017, WannaCry [1], one of the devastating ransomware which plagued over 150 countries and traversed all continents, spared no industry niche owing to the indiscriminate nature of the attack [2]. At 7:10 PM Eastern Time in October 21, 2016, hackers manipulated millions of “broilers” to paralyze the server of the DNS supplier of Dyn through the Mirai malware in a hijacking attack, which led to the collapse of well-known American websites, including Twitter, Paypal, Spotify, Netflix, Airbnb, Github, Reddit, and New York Times, and half of the United States fell into a disconnected state. From these events, we can see that malware detection is extremely urgent.

At present, malware detection methods are divided into static detection and dynamic detection [3–5]. Based on the static method, features are extracted from the original codes or files such as PE files, binary code, and disassembled code. Without running, the static features are represented as a series of coding that is the only identity representation and can distinguish from other software. For example, antivirus software products (e.g., Symantec and Kaspersky) mainly use the signature-based method of detection [6, 7], which is unique for known malware so that its samples can be correctly classified with a small error rate [8]. But the static method is usually vulnerable to obfuscation technology. For example, hacker makes tiny changes in malware variants, such as adding null instruction (NOP) in noncritical areas, changing instruction jump mode, which will produce new signature-based codes. If the virus library is not updated and preserves the new codes, the antivirus software will not be able to detect these variants. Different from the static method, the dynamic features are extracted under runtime environment (e.g., sandbox or honeypot), where malicious acts and operations are not hidden or discounted. For example, the dynamic representations of malicious behaviors may be from a sequence of API calls or instructions in a virtual machine environment. Compared with static method, the dynamic one does not need reverse engineering such as decompilation and decryption. Though it consumes a lot of running time and storage space, the dynamic method is more resilient to obfuscation technology [9, 10].

As a subset of AI (artificial intelligence), machine learning, developed to allow robots and computers to learn autonomously, is used to better serve the corresponding business needs and has achieved great success. Through the approximation of complex function and nonlinear fitting, machine learning, such as SVM, Naive Bayes, and decision trees, has been used for model to detect malicious codes [7, 11]. Ye [12] and others use a sequence of API calls as the behavioral characteristic of malware and develop the Intelligent Malware Detection System (IMDS) scanning malware based machine learning. Li et al. [13] propose an approach for extracting the dynamic features of malicious code semantics. The method extracts the dynamic features of malicious codes in virtual environment so as to achieve the purpose of protecting physical machine. The experimental analysis adopts the machine learning methods such as decision trees, KNN, and SVM. Zhou et al. [14], based on the isomorphism of sensitive API call graph, propose a method which is used to construct malware family features with graph similarity metric.

In this paper, we propose a features extraction method of hybrid gram for malicious behavior with cross entropy based on machine learning. Inspired from semantic segmentation of NLP (Natural Language Processing), the API sequences of malware are essentially kinds of context and are as rigorous as NLP on semantic and structural features. Therefore, a sequence of high correlation API can be used to represent the malware behavior. Moreover, in order to select the malware features, the feature vector is extracted from the sequences of API calls by sliding time window of different gram. Considering that the vectors are very sparse and high dimensional at this time, nonsignificant features should be excluded. Furthermore, through calculating cross entropy of continuous overlapping subsequences in hybrid gram, the new feature vector is chronologically integrated and selected. Finally, the machine learning method is applied to classify and detect malware samples. We adopt the dataset from VXHeavens website [15] to train the model. The experiment results show that H-gram method effectively distinguishes malicious behaviors.

The rest of the paper is organized as follows. In Section 1 the API-related knowledge is introduced, as well as dynamic track and capture in a virtualized environment. Section 2 elaborates the process of extracting and quantifying n-gram semantic features and also an adaptive variable-length n-gram feature selection method. In Section 3, we describe the process of converting the API sequences processed by n-gram into information entropy, namely, the feature quantization. In Section 4, we use the machine learning method to verify the experimental data analysis. Finally, the full paper is summarized.

2. Win32 API Call Mechanism and Feature Selection

In the Windows operating system, user applications rely on interfaces provided by dynamic link libraries such as kernel32.dll, advapi32.dll, user32.dll, and gdi32.dll to access the hardware and system resources. Windows API call mechanism is shown in Figure 1.

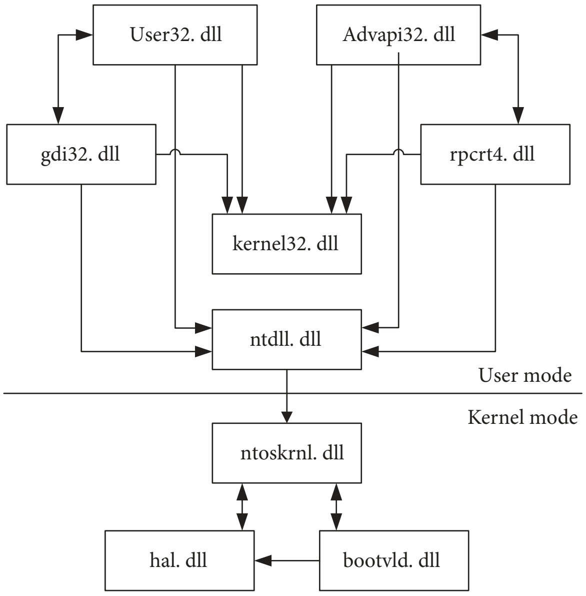

Figure 1: Windows API call mechanism.

This interface is called the Win32 API. For example, when a user program calls the Win32 API function of reading file, the process jumps to the NtReadFile function in the kernel state entry “ntdll.dll” firstly. Then the NtReadFile function calls the service routine in kernel mode, which is also named NtReadFile. Almost every program need directly calls the native API (kernel mode). If you want to monitor the program, the best way is to directly monitor its API calls. API function itself is not divided into the malicious or the benign. In other words, the malware uses the normal API function to achieve its own malicious purpose. The same API can be called by either the malicious or the benign. Only in terms of these sequences of API calls with context information can the diversity, between the malware behaviors and the benign ones, be discriminated effectively. However, due to the large number of API functions, it is not possible to describe the behavioral characteristics of samples in the actual operation through tracking all the APIs at all time. Therefore, in the paper we use the API hook technology to dynamically capture the API call sequence generated by the samples under the virtualized environment. After analysis and summary, six kinds of malicious behavior are obtained. The six kinds of behavior of API call sequences generated by each test sample are captured in chronological order. The API feature selection process is shown in Figure 2.

3. Feature Selection Model

The n-gram model has been successfully applied to the field of text analysis, which has improved the accuracy of similarity measure between texts. The API sequences of programs are essentially kinds of text and are more rigorous than the text

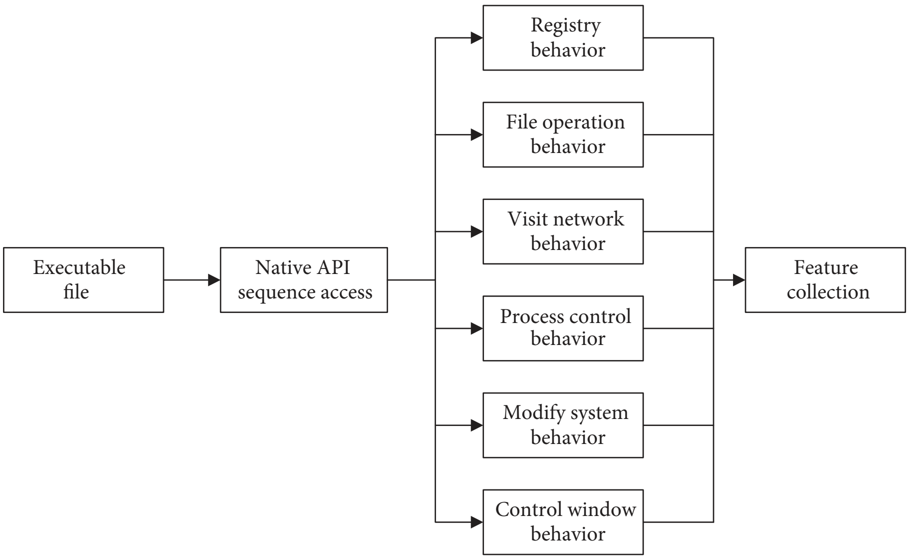

Figure 2: API feature selection process.

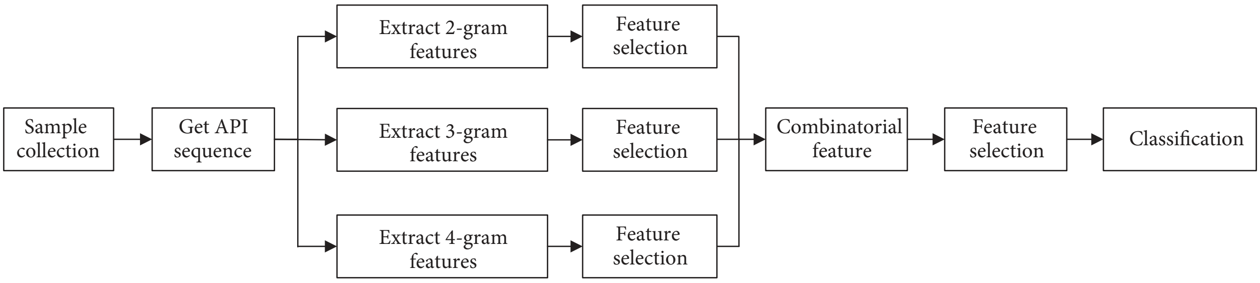

Figure 3: A model of feature selection of hybrid gram.

on semantic features and structural features. So n-gram can also be used for malware features analysis and selection. Without knowing which subsequences have representative semantic information, we extract new features from API call sequences by sliding fixed time windows. API call sequences are represented as $\{\cdots, x_{t-2}, x_{t-1}, x_t, x_{t+1}, x_{t+2}, \cdots\}$. The n-gram model, which satisfies Markov hypothesis as formula (1), can produce partial overlapped continuous subsequences.

$$\begin{aligned} &P\left(x_t \mid x_{t-1}, x_{t-2}, \ldots, x_{t-N}, x_{t-N-1}, \ldots, x_{t-N-M}\right) \\ &\quad = P\left(x_t \mid x_{t-1}, x_{t-2}, \ldots, x_{t-N}\right) \end{aligned} \tag{1}$$

For the example of 2-gram, the formula can be expressed as the following.

$$P\left(x_t \mid x_{t-1}, x_{t-2}, \ldots, x_{t-N}\right) = P\left(x_t \mid x_{t-1}\right) \tag{2}$$

For the example of 3-gram, the formula can be expressed as the following.

$$P\left(x_t \mid x_{t-1}, x_{t-2}, \ldots, x_{t-N}\right) = P\left(x_t \mid x_{t-1}, x_{t-2}\right) \tag{3}$$

For the API call sequence, if the semantic segmentation is performed by 3-gram as an example $\{x_1, x_2, x_3, \cdots, x_n\}$, a short sequence of consecutive partially overlapping sequences is generated as the following.

$$\left\{\left(x_1, x_2, x_3\right), \left(x_2, x_3, x_4\right), \ldots, \left(x_{n-2}, x_{n-1}, x_n\right)\right\} \tag{4}$$

From the above the formula (4), a new sequence set is as follows, where $s_1 = (x_1, x_2, x_3), s_2 = (x_2, x_3, x_4), \ldots, s_{n-2} = (x_{n-2}, x_{n-1}, x_n)$.

$$S_{\text{3-gram}} = \left\{s_1, s_2, \ldots, s_i, \ldots, s_{n-2}\right\} \tag{5}$$

The set element of s_{i} represents a short sequence of 3-gram $(x_{\mathrm{i}}, x_{\mathrm{i+1}}, x_{\mathrm{i+2}})$. Among them, it is relatively easy to extract n-gram segmentation items, but the semantics features of n-gram segmentation are not as obvious as those of the real code. So the amount of the length of the sliding time window of n-gram is a very important issue. A small value would ignore the structure and order of the process context, and an oversize value would reduce the similarity between the calls. Therefore, we use the same procedure to experiment with a hybrid n-gram to preserve as much semantic information as possible.

A model of feature selection of hybrid n-gram is proposed and shown in Figure 3. First of all, we collect representative samples of malware and benign software and put each sample into the analysis environment for a period of time and then record the dynamic API sequence of each. Secondly, we extract the semantic features with a variable N value of n-gram from API call sequence of each sample and generate feature segments such as 2-, 3-, and 4-gram and so on. The weight value of each short characteristic sequence can be represented by information entropy.

As the feature dimension of the API sequence is very huge, the selection method is used to reduce the dimension of features and extract the valid features. Due to dimension reduction, some semantic information is lost, while some semantic information is retained. By combining the features selected by hybrid n-gram, the complementarity between feature types is achieved, and more semantic information is retained as much as possible. After hybrid n-gram, the number of features is still large. As some features are redundant, the feature selection method is used to establish the effective feature subset and then do it again. After the above process, we can get the effective features that can distinguish between the malware and the benign software well. Finally, we use a variety of classification algorithms to detect the malware and verify the validity of the method of feature selection. A model of feature selection of hybrid n-gram is shown in Figure 3.

4. Feature Selection Method Based on Joint Entropy

The most obvious behavior difference between malware and normal software is that the malware needs to accomplish its own illegal goal, such as the realization of hidden, self-deleting, unrecognized payload and so on. These behavior features are not in the normal software. Therefore, different API sequence features have different information entropy. It happens that the entropy will change when the malware is illegally performed on the host computer.

In the paper, a novel feature selection of hybrid n-gram with cross entropy is proposed. Two variables C_i and s_j are set. C_i represents the number of the category that this behavior belongs to. s_j represents the j$^{\text{th}}$ short sequence of n-gram formed from formula (4)~(5). Firstly, the behaviors of API calls are converted to the sequence of n-gram such as $\{s_1, s_2, \ldots, s_j\}$. In same category the occurrence number of each short sequence of n-gram is counted, which is expressed as $a_{i,j}^{(k)}$. The sign k stands for the category label. The sign i stands for the number of samples. The sign j stands for the serial number of each short feature sequence of n-gram. The relationship of category and feature sequence of samples is shown as Table 1.

The purpose of feature selection is to obtain a kind of characteristics with the ability to classify data, which can improve the efficiency of classification learning. If there is no obvious difference between the results of classification with one feature and the results of random classification, this feature has no classification ability. Discarding the feature will not affect the classification accuracy. In the process of identifying the malware, the cross entropy of API calls is used to extract important features.

```
for n=1:N; %the value of n with n-gram
    calculate H(D);
    for j=1:$M$
        calculate $H(D \mid s_j)$
        ga($s_j$)=calculate $g(D, s_j)$
    end
    for k=1:K
        s_max(n,k)=max{ga($s_1$), ga($s_2$),. . .,ga($s_M$)}
        delete s_max(n,k) from{ ga($s_1$), ga($s_2$),. . .,ga($s_M$)}
    end
end
establish feature selection set { s_max(n, k)}
```

Algorithm 1: The algorithm procedure.

Let D be the train data set. |D| represents sample capacity, namely, the amount of total samples. There are K kinds of API behaviors from C_1 to C_k, and $i = 1, 2, \cdots K$. $|C_i|$ is the amount of samples that belong to the C_i category. So $\sum_i^K |C_i| = |D|$. The feature set of API sequences of n-gram is represented as S_{n-gram}.

$$S_{n-gram} = \left\{s_1, s_2, \ldots, s_j\right\} \tag{6}$$

Firstly, the empirical entropy of the data set D is calculated.

$$H(D) = -\sum_{i=1}^{K} \frac{|C_i|}{|D|} \log_2 \frac{|C_i|}{|D|} \tag{7}$$

Secondly, the empirical conditional entropy of the characteristic s_j for data set D is expressed as the following formula.

$$H\left(D \mid s_j\right) = -\sum_{i=1}^{K} p\left(C_i, s_j\right) \log_2 p\left(C_i \mid s_j\right) \tag{8}$$

In the above formula, $p(C_k, s_j)$ is expressed as the following.

$$p\left(C_k, s_j\right) = \frac{\sum_i^{|C_k|} a_{i,j}^{(k)}}{\sum_{i,j,k} a_{i,j}^{(k)}} \tag{9}$$

Also $p(C_i \mid s_j)$ is expressed as the following.

$$p\left(C_k \mid s_j\right) = \frac{\sum_i^{|C_k|} a_{i,j}^{(k)}}{\sum_{i,k} a_{i,j}^{(k)}} \tag{10}$$

Thirdly, the cross entropy of the characteristic s_j for data set D can be calculated.

$$g\left(D, s_j\right) = H(D) - H\left(D \mid s_j\right) \tag{11}$$

The top (maximal) M of $g(D, s_j)$ is selected. Then the feature set of S_{n-gram}:$\{s_1, s_2, \cdots, s_M\}$ is extracted, whose elements match the top M of $g(D, s_j)$ one by one. Finally, the total data set of feature selection such as $\{S_{2-gram}, S_{3-gram}, \cdots, S_{n-gram}\}$ is established. The key procedure with the hybrid n-gram algorithm is shown as Algorithm 1.

Table 1: The relationship of category and feature sequence of samples.

sample	category	s_1	s_2	……	s_j
Sample-c1-1		$a_{1,1}^{(1)}$	$a_{1,2}^{(1)}$		$a_{1,j}^{(1)}$
Sample-c1-2	C_1	$a_{2,1}^{(1)}$	$a_{2,2}^{(1)}$	……	$a_{2,j}^{(1)}$
……		…	…		…
Sample-c1-$\|C_1\|$		$a_{\|C_1\|,1}^{(1)}$	$a_{\|C_1\|,2}^{(1)}$		$a_{\|C_1\|,j}^{(1)}$
Sample-c2-1		$a_{1,1}^{(2)}$	$a_{1,2}^{(2)}$		$a_{1,j}^{(2)}$
Sample-c2- 2	C2	$a_{2,1}^{(2)}$	$a_{2,2}^{(2)}$	……	$a_{2,j}^{(2)}$
……		…	…		…
Sample-c2-|C2|		$a_{\|C_2\|,1}^{(2)}$	$a_{\|C_2\|,2}^{(2)}$		$a_{\|C_2\|,j}^{(2)}$
……	……	……	……	……	……
Sample-ci-1		$a_{1,1}^{(i)}$	$a_{1,2}^{(i)}$		$a_{1,j}^{(i)}$
Sample-ci-2	C_i	$a_{2,1}^{(i)}$	$a_{2,2}^{(i)}$	……	$a_{2,j}^{(i)}$
……		…	…		…
Sample-ci-|Ci|		$a_{\|C_i\|,1}^{(i)}$	$a_{\|C_i\|,2}^{(i)}$		$a_{\|C_i\|,j}^{(i)}$

Table 2: Malware categories and quantity.

Malware category	Backdoor	P2P-Warm	Warm	Trojan	Virus	Total
Amount	138	56	72	162	158	586

5. Experimental Analysis

5.1. Evaluation Indicators. In order to evaluate the method of the hybrid n-gram, three main indicators are used to measure the performance of the classifier through different feature extracted, including true positive rate, false positive rate, and the accuracy. The performance of the classifier can also be evaluated by using a ROC (receiver operating characteristic) curve, whose vertical axis represents true positive rate and whose horizontal axis is false positive rate. The area under the ROC curve (AUC) is a more comprehensive index for evaluating the classifier. The value of AUC is usually between 0.5 and 1.0. A larger value of AUC generally indicates that the performance of the classifier is better.

5.2. Experimental Data Acquisition. We collected 932 experimental samples. The benign software samples are collected from the Windows XP system directory and PE format EXE files, including different types of software, such as graphics software, system tools, multimedia software, and office software, whose amount is a total of 346. All benign software samples are detected by 360 anti-virus software. Malware samples were collected from the VXHeavens website [15], a total of 586, including viruses, worms, Trojans, and backdoor programs. The distribution of malware is shown in Table 2.

The Windows XP system is installed in the VMware virtual machine and then run the samples. API Monitor is taken as a hook routine to capture the native API sequences that the samples constantly call during execution. The sample run time is 120 seconds. For the vast majority of malware programs, 120 seconds is enough for them to execute all the processes, including a large number of cycles calling. Each sample intercepts the top 1000 API sequences as extraction feature. API sequences are recorded for each sample and not directly processed with the machine learning algorithm. In the paper, we select 100 feature sequences with the highest cross entropy of the fixed-length (N = 2, 3, 4) n-gram short sequences, respectively.

The Integrated 300 features after the initial selection are still too much, which is high dimension for the classification. The feature selection algorithm needs to select the most relevant subset of the features. In this paper, the features generated by hybrid n-gram (N=2, 3, 4) model are adopted. By features reduction of APIs segmentation by hybrid n-gram, the features of short sequences of different lengths are obtained. The combined number of features is 154 that is still more for the classification learning. Further, by adjusting the threshold value of cross entropy of the feature selection method again, we get 28 features that will eventually be used for categorical learning. After the above process, the features of low dimension with effectively distinguishing between malware and benign software are obtained.

5.3. Experimental Results Analysis. The final features that are used as the input of the classification algorithm and model are obtained. The experiment adopts four kinds of machine learning classification algorithms including ID3, Random Forest, AdboostM1, and Bagging. All four algorithms use an implementation version of the open source machine learning platform WEKA [16].

In the paper, we use 10-fold cross-check experiment method and apply the above four classification algorithms. The comparison results of the novel proposed method and other methods are shown in Table 3. We also use the above four classification algorithms to test the selected features of API call sequence.

Table 3: Malware detection experiments based on feature selection.

Feature representation	Features Quantity	Classification algorithm	TPR /%	FPR /%	Accuracy /%	AUC
2-gram	36	ID3	85.9	14.3	87.5	0.863
		Random Forest	86.3	12.8	86.6	0.850
		AdboostM1	82.9	16.3	80.2	0.808
		Bagging	83.9	15.8	82.3	0.826
3-gram	53	ID3	86.3	15.7	85.0	0.841
		Random Forest	94.1	13.7	92.0	0.971
		AdboostM1	91.2	14.7	93.0	0.956
		Bagging	91.2	12.8	87.5	0.931
4-gram	65	ID3	87.9	14.3	95.8	0.868
		Random Forest	96.8	6.2	93.1	0.98
		AdboostM1	90.9	9.1	87.5	0.974
		Bagging	93.9	7.0	92.0	0.957
Hybrid n-gram with cross entropy	28	ID3	96.8	6.3	92.5	0.963
		Random Forest	97.8	5.1	96.8	0.983
		AdboostM1	97.8	5.1	96.8	0.983
		Bagging	97.6	5.2	96.8	0.897

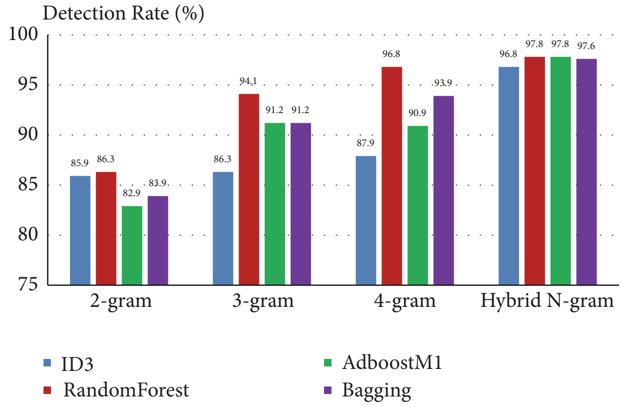

Figure 4: Comparison of detection rate in four methods.

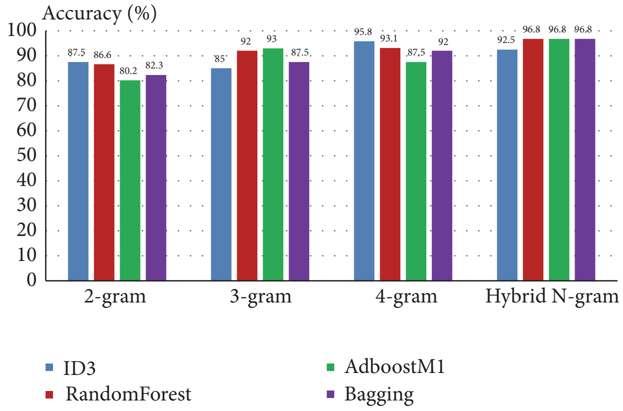

Figure 5: Comparison of accuracy in four methods.

In order to compare the effectiveness and generalization of feature selection method of API call sequence such as 2-gram, 3-gram, 4-garm, and hybrid gram with cross entropy, the experimental chooses four evaluation indexes, including detection rate, accuracy, false positive rate, and AUC value. Comparison of detection rate is shown in Figure 4. Comparison of accuracy is shown in Figure 5. Comparisons of false positive rate and AUC value are shown in Figures 6 and 7.

As can be seen from Figure 4, test results of 2-gram are less than the other three results, which illustrate that the extracted feature sequence is not obvious enough and leads to a low degree of discrimination. Of the four classification algorithm methods, Random Forest detection performs the best. The experimental results showed that the detection rate of the proposed method of H-gram is higher than that of the other three methods.

As can be seen from the comparison of accuracy of Figure 5, the overall accuracy rate is on the rise along the direction of the horizontal axis. The overall gap is small. The accuracy of 4-gram and H-gram is higher than that of 2-gram and 3-gram feature extraction. The H-gram method is slightly higher than the 4-gram method.

As can be seen from the comparison of false positive rates in Figure 6, 2-gram remains the weaker feature, with the highest false positive rate reaching 16.3%. The 4-gram and the proposed method of H-gram have achieved a relatively low false positive rate of 6.2% and 5.1%, respectively, with Random Forest algorithm. The method of H-gram has achieved the lowest false positive rate 5.1%. The false positive rate drop is more obvious with H-gram.

It can be seen from the comparison of AUC values in Figure 7 that the AUC value of the 2-gram is still lower than the other three feature selection methods. The difference of

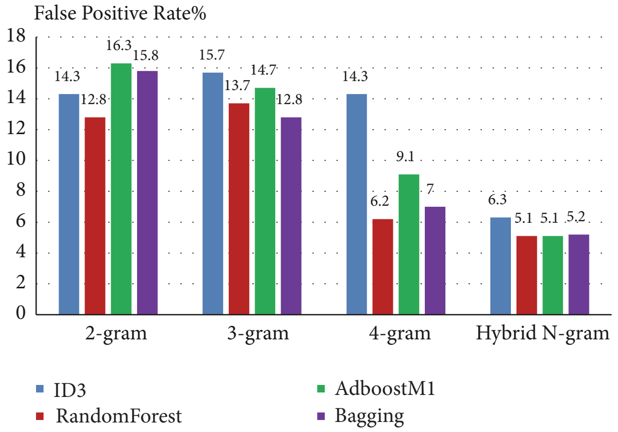

Figure 6: Comparison of false positive rate in four methods.

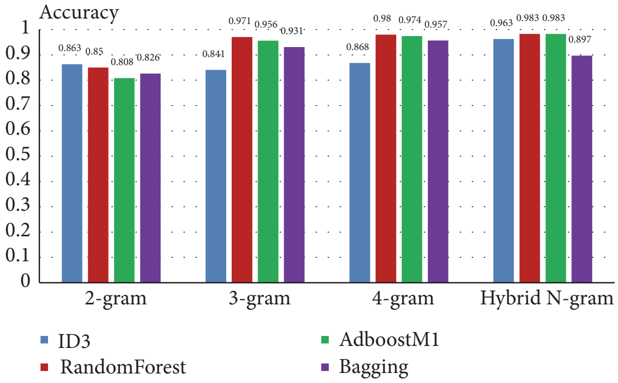

Figure 7: Comparison of AUC value in four methods.

the AUC values for other three methods is similar, and the method of H-gram is slightly better than the other three selection methods. The AUC value is up to 0.983, which is close to the optimal AUC value of 1.

6. Conclusion

As cyberspace becomes the fifth dimension of human life, the network security is getting more and more attention. Dynamic analysis method of malware behavior has become the focus of research. The past analysis methods are mainly by capturing all API functions of malware running, which describe the malware behavior with semantic segmentation of fixed parameters. So the previous methods are not only large amount of information, but also high redundancy. In the paper a novel feature selection method of hybrid H-gram with joint cross entropy is proposed. Based on the dynamic behavior tracking and feature analysis of native API in virtual environment, the proposed method is of adaptive variable length n-gram. At the same time the joint cross entropy is introduced to select the features of API sequence. Compared with the results of the experiments on semantic short sequence of the fixed-length n-gram, the proposed method is more effective in all four performance indexes of four classification algorithms (ID3, Random Forest, AdboostM1, and Bagging).

Data Availability

The dataset is real and available from VXHeaven website as mentioned in reference [15].

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Authors' Contributions

Yuntao Zhao is the main contributor of this work, given that he originated the idea, provided the general design, and wrote most of the paper. Bo Bo and others contributed to the implementation and testing of the experiment. All authors read and approved the final manuscript.

Acknowledgments

This work was supported by China Postdoctoral Science Foundation (2016M590234), General Project of Liaoning Provincial Department of Education (LG201611), Postdoctoral Fund of Shenyang Ligong University, Project of Applied Basic Research of Shenyang (18-013-0-32), 2017 Distinguished Professor Project, Liaoning Nature Science Foundation (20180551066), and Program for Liaoning Distinguished Professor.

References

[1] A. McNeil, "How did the WannaCry ransomworm spread?," 2017, https://blog.malwarebytes.com/cybercrime/2017/05/how-did-wannacry-ransomworm-spread/amp/.

[2] T. Webb and S. Dayal, "Building the wall: Addressing cybersecurity risks in medical devices in the U.S.A. and Australia," *Computer Law and Security Review*, vol. 33, no. 4, pp. 559–563, 2017.

[3] D. Tumer, S. Entwisle, M. Fossi et al., *Symantec Internet Security Thread Report*, Symantec Corporation, 2014.

[4] C. Xu, *Research on The Automatic Classification Method Based on The Behaviors of The Malicious Software*, Xiangtan University, 2014.

[5] R. Jing, *Research And Implementation of Malicious Code Detection System*, University of Electronic Science and Technology of China, 2010.

[6] J. Huang and A. L. Swindlehurst, "Secure communications via cooperative jamming in two-hop relay systems," in *Proceedings of IEEE Conference on Communications Society*, pp. 524–528, 2010.

[7] R. Wang, D. G. Feng, Y. Yang et al., "Semantic based behavior feature extraction and detection method for malicious code," *Journal of Software*, vol. 23, no. 2, pp. 378–393, 2012.

[8] D. Kirat, G. Vigna, and C. Kruegel, "Bare cloud: bare-metal analysis-based evasive malware detection," in *Proceedings of the 23rd USENIX Security Symposium*, 2014.

[9] X. Liu, *Research on Malware Analysis and intelligent technology detection based on machine learning*, Xiangtan University, 2014.

[10] T. Dube, R. Raines, G. Peterson, K. Bauer, M. Grimaila, and S. Rogers, “Malware target recognition via static heuristics,” *Computers & Security*, vol. 31, no. 1, pp. 137–147, 2012.

[11] D. W. Chen, P. Tang, and S. T. Zhou, “Malware detection model based on sandbox,” *Computer Science*, 2012.

[12] Y. Ye, D. Wang, T. Li, D. Ye, and Q. Jiang, “An intelligent PE-malware detection system based on association mining,” *Journal of Computer Virology and Hacking Techniques*, vol. 4, no. 4, pp. 323–334, 2008.

[13] M. Li, X. Q. Jia, R. Wang et al., “A method of feature selection and modeling for malicious code,” *Computer Application And Software*, no. 8, pp. 266–271, 2015.

[14] H. Zhou, W. Zhang, F. Wei, and Y. Chen, “Analysis of android malware family characteristic based on isomorphism of sensitive API call graph,” in *Proceedings of the 2nd IEEE International Conference on Data Science in Cyberspace, DSC 2017*, pp. 319–327, Guangdong, China, June 2017.

[15] “VXHeaven,” 2018, http://vxheaven.org/.

[16] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers Inc., San Francisco, Calif, USA, 2011.

Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
Advances in Multimedia
Advances in Civil Engineering
Journal of Control Science and Engineering
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration