Malware Detection Using Semantic Features and Improved Chi-square

Seung-Tae Ha[1], Sung-Sam Hong[1], Myung-Mook Han[1]

[1] IT convergence engineering, Gachon University, South Korea
azure105@naver.com, sungsamhong@gmail.com, mmhan@gachon.ac.kr

Abstract

As advances in information technology (IT) affect all areas in the world, cyber-attacks also continue to increase. Malware has been used for cyber attacks, and the number of new malware and variants tends to explode in these years, depending on its trendy types. In this study, we introduce semantic feature generation and new feature selection methods for improving the accuracy of malware detection based on API sequences to detect these new malware and variants. Therefore, one of the existing feature selection methods is chosen because it shows the best performance, and then it is improved to be suitable for malware detection. In addition, the improved feature selection method is verified by using the Reuter dataset. Finally, the actual API sequences are extracted from the given malware and benign, and the proposed feature generation and selection methods are used to generate a feature vector. The performance is verified through classification.

Keywords: API sequence, Feature selection, Malware detection

1 Introduction

As IT technologies evolve, they have affected all areas globally. As a result, cyber-attacks also continue to increase. Most of cyber-attacks are made for attackers' political purposes or monetary purposes. Malware is used for various types of attacks to achieve attackers' objectives such as APT, DDoS, personal information stealing, etc. Recently, a particular type of malware tends to increase rapidly in accordance with the trend. According to the Symantec 2015 Internet Security Threat Report (ISTR), ransomware attacks more than doubled in 2014, from 4.1 million in 2013, up to 8.8 million [1]. According to the monthly statistics of ransomware attacks detected from 2013 to 2014, it can be known that trendy ransomware attacks have increased explosively since a particular point in time. The explosive growth of such certain types of malware is related to its variants based on obfuscation and executable compression techniques which are used to avoid their detection and to make the analysis difficult. Signature-based detection is commonly used for anti-virus software currently to identify malware. The signature-based detection registers unique binary signatures of malware and then detects the malware by checking the signature existence. This method means that more malware attacks leads to more signatures. It becomes very time-consuming to generate and register signatures for various types of malware. Therefore, there is a need for a new malware detection method in order to respond efficiently and quickly to such new malware and variants.

For this reason, there have been studies on malware detection using behavior-based malware feature definition recently. Most of these studies are based on extraction of API sequences and perform malware classification using machine learning techniques. In [3], static analysis is performed by generating an API Call Graph from the Control Flow Graph of a portable executable. After that, N-grams are applied to the API Call Graph to generate a feature vector which is used for malware detection through machine learning methods. N-gram is used to model behavior [4]. In [5], dynamic analysis is performed to extract API call sequences from which a feature vector is generated by using the feature selection method based on *N-grams* and *Odds ratio (OR)*. The study also employs classification techniques such as *Naïve Bayes, Support Vector Machine (SVM), etc*. In [6], an experiment is performed to find optimal API sequence length and combination for malware classification using the API sequences.

This study aims at improving malware classification whose accuracy is higher than the existing studies. In previous studies, an *n-gram* of a particular size has been applied in the feature generation step and the *Odds ratio* has been applied in the feature selection step. There are some Windows APIs with different function names due to a slight difference even though they perform the same operation. In this case, applying only an *n-gram* in the feature generation step may represent the same behavior but different features. In addition to the *OR*, the *Chi-square* technique can be used to achieve a high performance in the feature

*Corresponding Author: Seung-Tae Ha; E-mail: azure105@naver.com
DOI: 10.3966/160792642018051903023

selection. Therefore, in this paper, we perform behavior-based API name integration in the feature generation step and apply an improved *Chi-square* technique in the feature selection step to obtain a higher detection performance than ever before.

This paper is organized as follows. Section 2 discusses related work and describes the Microsoft Windows application programming interface (API) and *Chi-square*. Section 3 explains semantic feature generation and *improved Chi-square*. Section 4 shows experiments on malware classification using the proposed methods, where the Reuter dataset and actual malware and normal programs are given. And proves the proposed method through experiment results and explains the experiment results. Finally, Section 5 draws the conclusions.

2 Background

2.1 Related Work

For the method to detect based on the behavior of malicious code, there is a method using the API that the malicious code calls for. For this study, the method using the API is divided into three: Mapping API call [7-9], API call graph [2-3] and API call sequence [5-6]. Mapping API call method obtains the API information by extracting the IAT (Import Address Table) through static analysis. API information is mapped according to the malware behavior steps. For example, Alazab [7] divided the API into six steps of malware behavior (Search Files to Infect, Copy/Delete Files, Get File Information, Move Files, Read/Write Files, Chage File Attributes).

API Call Graph method obtains the API information by extracting the IAT through static analysis for malicious code, and displays this in the Control Flow Graph to use. This method, in general, is used to detect the variations trough similarity analysis. Faruki [3] proposed the method that generates the features by applying n-gram to detect using the partial graph of API Call Graph, and uses them.

Uppal [5] generated the features by using n-gram in API call sequence like Faruki used. In the research of Uppal, the generated features were used all as input value. Faruki applied the feature selection by using Odds Ratio which is used for document classification, to improve the classification accuracy. Both researches all compared 1-gram, 2-gram, 3-gram and 4-gram and used 4-gram which is better in performance.

Choi [6] conducted the study to find the most significant size of n-gram as the size of n-gram affects the classification accuracy in the feature generation phase of the existing research. In the corresponding research, she used various sizes of n-gram as well as the specific size of n-gram complexly.

Mapping API call and API Call Graph methods requires static analysis. Since static analysis is time-consuming method, we used API call sequence. The novelty of our research is as follows. We integrate the API having the same meaning in the collected API call sequence as one name of API. We generate the features by applying various sizes of n-gram from 1 to 5 which was used in the existing study [10]. We used *Improved CHI* suitable for data set which is composed in asymmetric size by two kinds in the feature selection phase to improve the classification accuracy of malicious code. Table 1 shows the comparison with previous related studies.

Table 1. Comparison of existing research

Authors	Improved Step	Methods
P. Faruki	Feature Generation	Applied N-gram to API call graph
D. Uppal	Feature Selection	Applied Odds Ratio to select features
Ji-yeon Choi	Feautre Generation	Compared variable length of N-gram features
Proposed Method	Feature Generation & Selection	Semantic Feature Generation & Improved CHI

2.2 Microsoft Windows API

Microsoft Windows uses various APIs from a number of *DLL*s. All the APIs are to invoke APIs within *kernel32.dll* and *ntdll.dll*. The architecture of Windows [11] is divided into user mode and kernel mode. The APIs within *kernel32.dll* serve as open entry points or wrapped functions that allow users to use. *ntdll.dll* contains the Native API which isn't open to the end users as a document, and these functions are actually handled in kernel mode. Therefore, common API calls are eventually redirected to the corresponding Native API calls. Since monitoring the entire Native API causes severe overheads, it is necessary to monitor only some parts. Furthermore, information of the entire Native API isn't available so that this study focuses on monitoring only malware-related APIs within *kernel32.dll*, including the registry, memory, files, processes, etc.

2.3 Chi-square [12]

Text classification involves very high-dimensional feature spaces. As the feature dimensionality gets greater in text classification using machine learning techniques, the learning speed gets slower. The classification performance will also be dropped. There is a need for dimensionality reduction process to deal with such a problem. That is, it is necessary to use a feature selection method is required to select meaningful features in the dimensionality reduction step. There are the following feature selection methods: *Document Frequency (DF), Information Gain (IG), Mutual Information (MI), Chi-square (CHI), Odds ratio (OR)*, etc. [13-14].

A *Chi-square* measures the degree of relationship

between the keyword t and the category c to obtain the importance of the keyword t in document classification. A is the number of documents of the category c containing the keyword t. B is the number of documents of other category (not c) containing the keyword t. C is the number of documents of the category c not containing the keyword t. D is the number of documents of other category (not c) not containing the keyword t. N refers to the total number of documents. The *Chi-square* statistic between the keyword t and the category c is defined as follows:

$$X^2(t,c) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)} \quad \textbf{(1)}$$

3 Feature Generation Methods

In this section, we propose a semantic feature generation method and a new feature selection method using *Improved CHI*.

3.1 Semantic Feature Generation

An API sequence can be considered as a document composed of words representing the behavior. Feature generation has a significant impact on classification accuracy in the document classification. According to the study conducted by *Gabrilovich* [15], he proposed a method for addressing the *synonymy* and *polysemy* problems in natural language processing to improve the performance of document classification. API functions don't show polysemy unlike natural languages, but they can indicate synonymy when they perform the same operation despite their different names. For example, in case of the functions used to compare two strings, "*_stricmp*" and "*_strcmpi*" are provided with different names for version compatibility although they are identical. Furthermore, there are two functions indicating the path to the executable file: *'GetCurrentDirectory'* and *'GetModule FileName'*. If functions showing synonymy are used in the feature generation step as they are, the operation will be performed for each individual feature showing identical behavior but having a different name in the feature selection step. This represents a limitation to meaningful feature selection.

In this paper, Microsoft's *Windows API Index* is used to grasp the meaning of each API and then to integrate the names of APIs representing the same behavior. Moreover, the *1~5-gram* method applied in previous study [10] is used for feature generation. The procedure for *semantic feature generation* is as follows.

- First create a conversion table based on the meaning of each API.
- Use the table to convert the API names included in the API sequences.
- Generate the *1~5-gram* features from the converted API sequence.

3.2 Improved Chi-square

An API sequence can be considered as a formalized document in which the number of API function names are listed. Therefore, malware classification using API sequences is similar to a formalized text classification. In the text classification, feature selection has a significant impact on classification accuracy as much as classification methods and the performance of feature selection varies depending on characteristics of the classification target. In this paper, we propose an *improved Chi-square* method for feature selection in order to improve the accuracy of malware classification.

There are various feature selection methods for text classification. There are experiments on various feature selection methods to compare their performance [13, 16]. In [13], The Chi-square statistic shows the highest classification accuracy among *Document Frequency*, *Information Gain*, *Mutual Information*, *Chi-square statistic*, and *Term Strength*. In [16], *Odds ratio* shows the highest classification accuracy among *Document Frequency*, *Information Gain*, *Chi-square statistic*, and *Odds ratio*. In general, text classification uses two or more categories. The performance of these methods can be shown differently in our experiment using two categories indicating the presence or absence of malware. Therefore, the Reuter-21578 dataset [17] commonly used for text classification is divided into two categories, and the high- performance feature selection methods identified in the previous experiments are applied. After that, a comparative performance analysis of the methods is done by using *sequential minimal optimization (SMO)* which is one of the classification methods. Half of the Reuter-21578 dataset have been used for our experiment. The dataset has 24,329 features. And each number of entities for categories are 2,369 and 8,420. A particular category usually contains a large number of entities in learning of malware and benign so that the ratio of malware to benign is set to 1: 4 approximately.

As shown in Figure 1, the experiment results indicate that the Chi-square statistic has the highest classification accuracy in two categories. The *Chi-square* statistic considers the observed values of the entire categories as well as the relatively observed values between categories; therefore, it generally gives higher scores to high-value features. If two categories are used and a larger number of entities exist in a particular category as the above experiment, a feature which frequently occurs in categories having fewer components will be given unusually larger weight. For example, we assume that the dataset composed of total 10,000 documents is composed of *Category 1* having 2,000 objects and *Category 2* having 8,000 objects respectively. Among the features displayed in the dataset, when *Feature A* and *B* have the same observation value as Table 2, if we perform the *CHI*

operation, we will be able to get the same result value as Table 3. The observed rate in *Category 1* for *Feature A* and the observed rate in *Category 2* for *Feature B* is same as 1/20 and the observed rate in *Category 2* for *Feature A* and the observed rate in *Category 1* for *Feature B* is same as 1/4. However, according to the operation result displayed in Table 3, *Feature B* that is generated with high rate in the small size obtained the weighted value higher more than double compared with *Feature A*. This result creates the problem that the features generated a lot in the small size in the dimension compression phase are only selected.

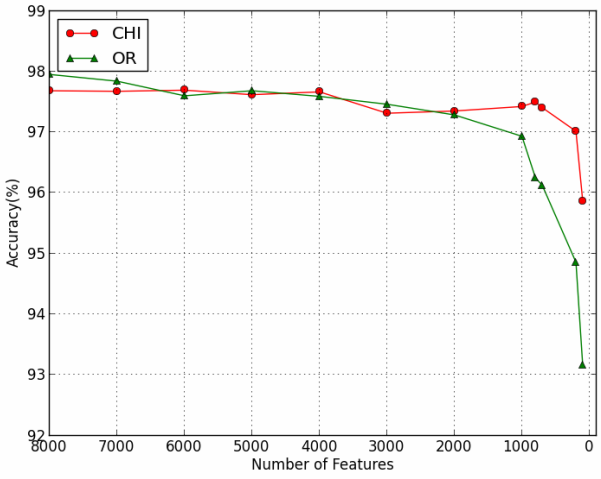

Figure 1. Comparison of existing high performance methods

Table 2. Feature example

	Category 1 (2000)	Category 2 (8000)
Feature A	100	2000
Feautre B	500	400

Table 3. CHI value of feature A and B

	Feature A	Feature B
CHI value	385.77	781.44

As a result, the accuracy of classifying the categories containing fewer components will drop in case of dimensionality reduction. And the classification accuracy of the smaller category will get lower.

In this paper, a new method is proposed to overcome such a limitation. That is, the larger category with many entities is compressed at the same rate to have an equal number of entities between the categories in the *CHI* operation.

Accordingly, the observed values are also compressed in the proposed method. A is the number of documents of the category c containing the keyword t. B is the number of documents of other category (not c) containing the keyword t. C is the number of documents of the category c not containing the keyword t. D is the number of documents of other category (not c) not containing the keyword t. N refers to the total number of documents. The proposed method can be defined by the following equation:

$$B' = B(\frac{A+C}{B+D}), D' = D(\frac{A+C}{B+D})$$
$$X^2 = \frac{(A+B'+C+D')\times(AD'-CB')^2}{(A+C)\times(B'+D')\times(A+B')\times(C+D')} \quad (2)$$

4 Experiment

In this section, the performance of *semantic feature generation* is compared with that of *n-gram feature generation* in the classification for malware detection. The *Reuters-21578* data set is used to compare the performance of existing feature selection methods (*Chi-square, Document Frequency, Information Gain, Mutual Information and Odds ratio*) with that of the *improved Chi-square* and to verify the excellence of each method. After that, both of the proposed methods are applied to verify their excellence in terms of classification accuracy, compared to methods used in the previous studies.

4.1 Experiment Environment

The experiments were performed on Intel Core i5 3GHz machine, 8GB RAM, Windows 7 Enterprise (64bit) operating system.

API monitor. To create the data set by extracting API call sequence from the sample of malicious code and normal program, we used API Monitor v2 [18] which is a monitoring tool in the virtual environment of Windows XP using the Virtual box 4.3.12 [19]. At this time, we perform the unpacking for each malicious code to get the significant API call sequence.

Dataset. The test data set for malware detection is VX heaven [20] 165 of the 270 normal programs and malware in virtual environments through dynamic analysis utilizing API Monitor was created by extracting the API call sequence Dataset received from. The number of each is the same as shown in Table 4 and Table 5.

Table 4. Number of Malware samples

Malware	Number of samples
Rootkit	35
Trojan	45
Virus	40
Worm	45

Table 5. Number of Benign samples

Benign software	Number of samples
Application	175
Device Driver	35
Utility	60

The *Reuters-21578* dataset contains 90 different categories and consists of classified Reuters news documents. The number of documents in each category differs in size. To measure the performance of the proposed method for asymmetric categories, the data set (a total of 10789 documents) used for the experiments has been divided into the "*Acq*" category with 2369 documents and a set of other categories with 8420 documents.

4.2 Semantic Feature Generation

The performance of semantic feature generation has been evaluated in comparison with the performance of *n-gram*-based *feature generation*. The test dataset used for performance comparison consists of malware and API call sequences of normal programs.

Figure 2 shows the experimental model used to comparatively analyze the classification accuracy of the semantic feature method. First, extract API call sequences from a normal program and malware, and then apply the proposed semantic feature method and the n-gram method. After that, use the *Chi-Square* for feature selection, and apply *sequential minimal optimization* algorithm via Weka [21] for performance comparison.

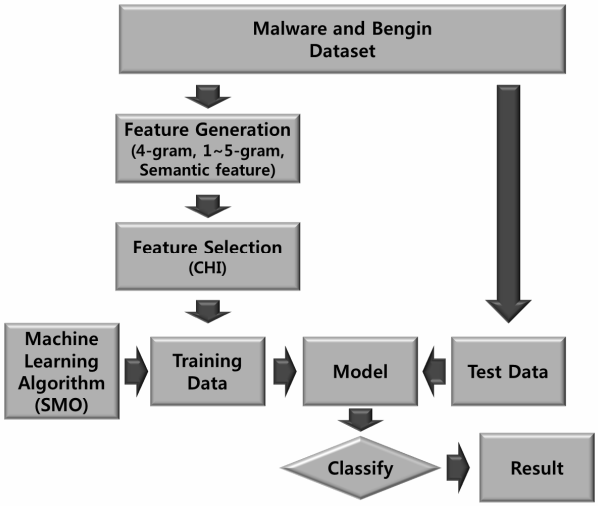

Figure 2. Semantic feature experimental model

According to the comparative analysis of classification accuracy in multiple dimensions, 1~5-gram feature and Semantic Feature has shown a high degree of classification accuracy in low number of features as shown in Figure 3. 1~5-gram feature and Semantic Feature has shown almost same classification accuracy in most dimensions. However, Semantic Feature has shown the highest classification accuracy (96.32%) in 10000 features.

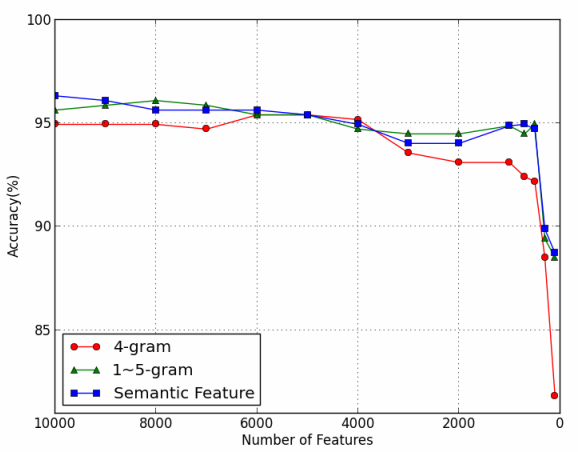

Figure 3. Comparison using *SMO*

4.3 Improved Chi-square

We have performed two experimental verifications to evaluate the performance of the *improved Chi-square*. In first experiment, the *improved Chi-square* statistic performance is verified by using the Reuter-21578 dataset. In second experiment, we evaluate the performance with malware and benign dataset. The experiment on malware classification using the proposed method can be conducted as follows. First, extract API Sequences from the malware and benign, generate the feature vector and then perform feature selection using the proposed method (refer to Section 4.3.1). In the feature selection phase, we also used *Chi-square, Document Frequency, Information Gain, Mutual information and Odds Ratio* to compare their performance with proposed method. After that, use *SMO* to measure the accuracy based on the selected features (refer to Section 4.3.2).

Reuter-21578 dataset. The Figure 4 shows the experiment model to verify the performance of *Improved CHI*. In this experiment, to verify the performance of the proposed method, we used the reuter-21578 dataset which is used widely to measure the performance of document classification. We created the feature vector by using the existing feature selection method and the proposed method.

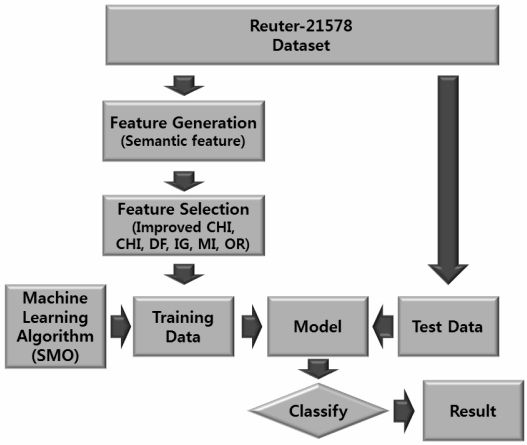

Figure 4. Improved CHI experimental model

The number of features of Reuter-21578 dataset is 24329. *CHI*, *DF*, and *Improved CHI* are available for the operation of weighted value for all features. However, *IG*, *MI*, *OR* will have the unlimited value when the observation value is 0 or maximum value. As these values cover more than the half, these features were excluded for this experiment. In conclusion, the available features through the operation of weighted value for *IG*, *MI*, *OR* are about 8000. Thus, the maximum number of features in the experiment for comparison was set as 8000.

The Figure 5 shows a comparative performance analysis of the conventional *CHI* method*, DF, IG, MI, OR* and the proposed method. The proposed method shows a higher degree of classification accuracy than the conventional *CHI*. In Figure 5, the proposed method in dimension reduction process up to about 1000 shows no big difference in performance with *CHI*. But, it is found that when the dimension is more reduced, the proposed method shows higher classification accuracy than *CHI*.

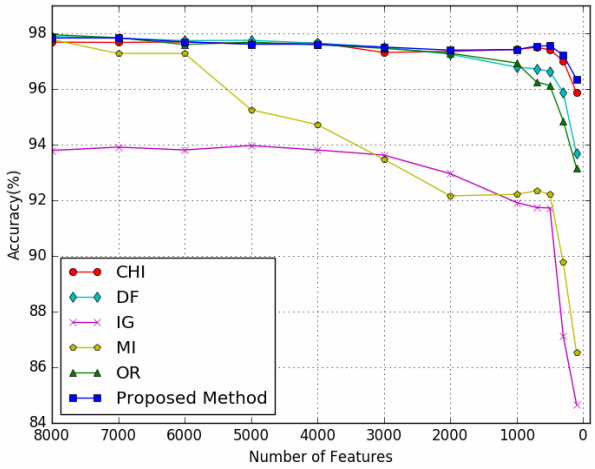

Figure 5. Comparison of *improved CHI* and other methods

Table 6 shows the classification accuracy for Acq composed of 2369 in the classification through the above SMO. The proposed method showed higher accuracy except for 5000 and 20000. The more the dimension is reduced, the higher accuracy the proposed method showed relatively.

Table 6. Accuracy for small category (reuter-21578 dataset)

Number of features	CHI (%)	Improved CHI (%)
100	88.65	**90.59**
500	93.50	**94.47**
1000	93.75	**93.96**
3000	93.67	**94.22**
5000	94.17	94.13
7000	94.26	**95.10**
10000	94.68	**95.02**
15000	94.93	**95.06**
20000	95.31	95.19

Malware, benign dataset. 165 malware samples and 270 benign samples are used for our experiment. There are static and dynamic methods for extracting API sequences of benign and malware. We performed a dynamic analysis using the API Monitor [15] tool on Windows XP. Since monitoring all the APIs causes severe overheads, we only monitor APIs related to malware activities in this study, including the registry, files, the system, etc.

Generate feature vector. N-grams are applied to increase the classification accuracy for the observed API sequences. N-grams proposed in the previous study [10], and N-grams are applied from 1-gram to 5-gram in order to generate a feature vector. As a result, 217,778 features are generated in this step. And each feature rank is computed by using the *CHI, DF, IG, MI* and *OR* methods and the proposed *Improved Chi-square* method.

Classification. *SMO* in Weka is used for a comparative performance analysis of the proposed method. These supervised learning algorithms produce a model based on the given training set and the performance is verified on the given test set. 10-fold cross validation is used to improve the statistical reliability of performance measurement because the amount of experimental data is not as sufficient as that of actual data.

According to the comparative analysis of classification accuracy in multiple dimensions, the proposed method has shown a high degree of classification accuracy in most dimensions as shown in Figure 6. As shown in Table 7, the proposed method also improves classification accuracy in the smaller category with fewer entities so that it can overcome the existing limitation previously mentioned. This is because the effect of the equal observation size between the categories in the *CHI* operation.

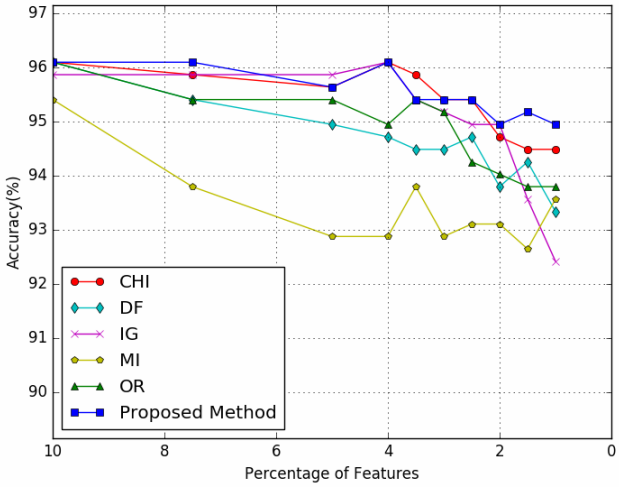

Figure 6. Classification accuracy of Malware and Benign dataset

Table 7. Accuracy for small category (Malware and Benign dataset)

Percentage of features (%)	CHI (%)	Improved CHI (%)
1	95.76	**96.36**
2	96.97	**97.58**
3	96.97	**97.58**
4	98.18	98.18
5	97.58	97.58
7	97.58	**98.18**
10	98.18	98.18

4.4 Evaluation of the Proposed Methods

In this phase, two proposed methods are all used to compare the performance with the existing research. The process of monitoring the sample of malicious code and normal program in the virtual environment and creating the API call sequence is same as the previous experiment. Then, through *4-gram and OR* [5], *1~5 gram and CHI* used in the existing research [10] in the created API call sequence, the feature vector is created. And then the feature vector is created through *Semantic feature and Improved CHI* which is the proposed method and the classification is performed using a *SMO* to compare the accuracy, respectively.

As the method using the *4-gram and OR* displays the unlimited operation result when it has the observed value necessary for the operation in *OR* operation is 0 or maximum value, we excluded this for the feature. Thus, among total 61,663 features, about 6,000 which correspond to 10% can be used.

The result shows a comparative analysis of the existing feature vector generation method using the 4-gram with *OR*, 1~5-gram with *CHI* and the proposed method using *Semantic Feature* with *Improved CHI*.

Figure 7 shows a performance comparison of 4-gram with *OR*, 1~5-gram with *CHI* and proposed methods through the *SMO*-based classification. To make a performance comparison of the selected features, we have compared the classification accuracy in accordance with the number of selected features. According to the comparative analysis of classification accuracy in multiple dimensions, Proposed Methods has shown a high degree of classification accuracy in low number of features. Figure 7 also shows highest classification accuracy of proposed methods (96.32%) in 10000 features.

As shown in Table 8, the proposed methods improve classification accuracy in the smaller category with fewer entities so that it can overcome the existing limitation previously mentioned.

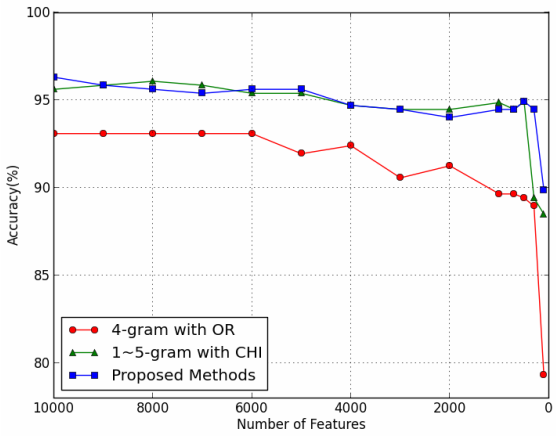

Figure 7. Performance comparison with existing methods

Table 8. Accuracy for small category (Malware and Benign dataset)

Number of Features	CHI (%)	Proposed Method (%)
100	80	**82.42**
300	84.23	**94.54**
500	96.36	96.36
700	96.36	96.36
1000	94.54	94.54
2000	94.54	**95.75**
3000	94.54	**95.15**
4000	96.36	96.36
5000	95.75	95.75
6000	96.96	96.96
7000	97.57	97.57
8000	97.57	97.57
9000	98.18	97.57
10000	97.57	**98.18**

5 Conclusion

In this paper, we propose a new feature generation and selection method for improving the detection and accuracy of malware.

In order to handle the problem of API synonymy in the feature generation step, the APIs showing the same behavior have been converted to have a single name. As a result, the proposed method has made it possible to classify things which couldn't be classified with conventional feature generation methods.

In feature selection, we apply the *Chi-square* statistic to malware detection using API sequences because it has shown the highest classification accuracy in the existing studies. *CHI*-based malware detection has a limitation in handling the smaller category; however, the proposed method using the improved *CHI* can solve the classification accuracy problem. The proposed method shows a higher degree of classification accuracy in the smaller category than the conventional ones, and it also improves the overall classification accuracy substantially.

For future research, there will be studies on malware classification and accuracy improvement as well as

malware detection. In addition, we plan to study how to select the effective number of the features required to build a machine learning model for malware detection in consideration of speed and classification accuracy.

Acknowledgement

A preliminary version of this paper was presented at ICONI 2015, and was selected as an outstanding paper. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2015R1D1A1A01060874)

References

[1] Symantec Corporation, *Symantec Internet Security Threat Report*, Vol. 20, April, 2015.

[2] J.-S. Lee, K.-C. Jeong, H.-J. Lee, Detecting Metamorphic Malwares Using Code Graphs, *2010 ACM Symposium on Applied Computing*, Sierre, Switzerland, 2010, pp. 1970-1977.

[3] P. Faruki, V. Laxmi, M. S. Gaur, P. Vinod, Mining Control Flow Graph as API Call-grams to Detect Portable Executable Malware, *5th International Conference on Security of Information and Networks*, Jaipur, India, 2012, pp. 130-137.

[4] C.-M. Chiu, S.-S. Hung, J.-J. Tsay, K.-K. Fan, D.-C. Tsaih, Online Opponent Modeling for Action Prediction, *Journal of Internet Technology*, Vol. 15, No. 7, pp. 1209-1215, December, 2014.

[5] D. Uppal, R. Sinha, V. Mehra, V. Jain, Malware Detection and Classification based on Extraction of API Sequences, *2014 International Conference on Advances in Computing, Communications and Informatics*, New Dehli, India, 2014, pp. 2337-2342.

[6] J.-Y. Choi, H.-S. Kim, K.-I. Kim, H.-S. Park, J.-S. Song, A Study on Extraction of Optimized API Sequence Length and Combination for Efficient Malware Classification, *Journal of the Korea Institute of Information Security and Cryptology*, Vol. 24, No. 5, pp. 897-909, October, 2014.

[7] M. Alazab, S. Venkataraman, P. Watters, Towards Understanding Malware Behaviour by the Extraction of API Calls, *2010 Second Cybercrime and Trustworthy Computing Workshop*, Ballarat, Australia, 2010, pp. 52-59.

[8] A. M. Al-Bakri, H. L. Hussein, Static Analysis Based Behavioral API for Malware Detection Using Markov Chain, *Computer Engineering and Intelligent Systems*, Vol. 5, No. 12, pp. 55-63, December, 2014.

[9] M. Alazab, S. Venkatraman, P. Watters, M. Alazab, Zero-day Malware Detection based on Supervised Learning Algorithms of API call Signatures, *9th Australasian Data Mining Conference*, Ballarat, Australia, 2011, pp. 171-182.

[10] S.-T. Ha, M.-M. Han, Malware Detection Using API Sequences of Variable Lengths, *16th International Symposium on Advanced Intelligent Systems*, Mokpo, Korea, 2015, pp. 227-234.

[11] S. Roman, *WIN32 API Programming with Visual Basic*, O'Reilly Media, 1999.

[12] R. L. Plackett, Karl Pearson and the Chi-Squared Test, *International Statistical Review*, Vol. 51, No. 1, pp. 59-72, April, 1983.

[13] Y. Yang, J. O. Pedersen, A Comparative Study on Feature Selection in Text Categorization, *14th International Conference on Machine Learning*, Nashville, TN, 1997, pp. 412-420.

[14] M. Rogati, Y. Yang, High Performing Feature Selection for Text Classification, *11th International Conference on Information and Knowledge Management*, McLean, VA, 2002, pp. 659-661.

[15] E. Gabrilovich, S. Markovitch, Feature Generation for Text Categorization using World Knowledge, *19th International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland, 2005, pp. 1048-1053.

[16] A. G. Chin, *Text Databases & Document Management*, IGI Publishing Hershey, 2001.

[17] D. D. Lewis, Reuters Ltd, Carnegie Group Inc., *Reuters-21578 Text Categorization Test Collection Distribution 1.0* (Version 1.3), http://www.daviddlewis.com/resources/testcollections/reuters21578/.

[18] R. Batra, *API Monitor v2 (Version Alpha-r13)*, http://www.rohitab.com/downloads.

[19] Oracle, *Virtualbox (Version 4.3.12)*, https://www.virtualbox.org.

[20] vxheaven.org team, Malware Samples, *Computer Virus Collection*, http://vxheaven.org/vl.php.

[21] University of Waikato, *Weka (Version 3.4)*, http://www.cs.waikato.ac.nz/ml/weka.

Biographies

Seung-Tae Ha received the Bachelor degree in Computer Media from Gachon University, Korea in 2014 and Master degree Computer Engineering from Gachon University, Korea in 2016. He is currently a researcher for Big Data, Data Mining and Information Security in Information Security Lab of Gachon University.

Sung-Sam Hong received the Bachelor degree in Computer Science from Gachon University, Korea in 2009 and Master degree Computer Engineering from Gachon University, Korea in 2011 and Doctor degree Computer Engineering from Gachon University, Korea in 2016. He is currently a researcher professor for Big Data, Data Mining and Information Security in Information Security Lab of Gachon University.

Myung-Mook Han received M.S. degree in Computer Science from New York Institute of Technology in 1987 and Ph.D. degree in Information Engineering from Osaka City University in 1997, respectively. From 2004 to 2005, he was a visiting professor at Georgia Tech Information Security Center (GTISC), Georgia Institute of Technology. Currently, he is a professor in the Department of Computer Engineering, Gachon University, Korea.