Proceeding of the IEEE
International Conference on Information and Automation
Shenzhen, China June 2011

A Feature Selection Method for Malware Detection

Qingshan Jiang
Shenzhen Institutes of Advanced Technology
Chinese Academy of Sciences
Shenzhen, Guangdong Province, China
qs.jiang@siat.ac.cn

Xinxing Zhao and Kai Huang
Xiamen University,
Software School
Xiamen, Fujian Province, China
xx.zhao@siat.ac.cn

Abstract **: Due to the serious network security problems in recent years, a large number of malware features have been emerged, which leads to increasing time-complexity and space-consumption for malware detection systems. Moreover, irrelevant and redundant features may decrease the detection rate. Feature selection, as an important data mining phase and technology, can effectively reduce the redundant and irrelevant features in the original large feature space, thereby can increase the detection rate and reduce the false positive rate for malware detection model. This paper proposes a class driven correlation based on feature selection method, which can select corresponding features for different classes of data respectively. Then this method uses correlation based feature selection method to eliminating redundant features. Experimental results indicate that the approach can not only reduce the complexity of malware detection system, but also increase the detection rate as compared to other methods.**

Index Terms: **Malware; Feature Selection; Data Mining; Correlation Measure**

I. INTRODUCTION

Malware is considered as computer virus in general [1]; it includes the traditional computer virus, as well as all computer programs that intentionally destroy computer operation system and data [2]. With the development of internet and software technology, the malware has been increasingly emerging in recent years. The traditional feature codes based on scanning techniques have failed to satisfy detection requirements, with the explosion of the amount and complexity of malware. In order to overcome the defects of feature coding detection technology and to realize real-time automatic detection of vast amounts of malicious software, the data mining based malware detection technologies (such as neural network, decision tree, support vector machine (SVM), associative classification, clustering, etc.) have been widely used in malware detection system [3-5].

In malicious software, the file features characterize software behaviour, and the detection system needs to extract various feature types from the file, such as strings, API (Application Programming Interface), instruction sequences, etc. Normally, the quantity of file characteristics for every feature type is very large [2,4,5]. These large amounts of features increase detection system workload, thus decrease the instant response ability of the system to new file detection. And also, malicious software often contains a large amount of irrelevant, redundant code to escape from being detected, which leads to low detection precision[2,5]. In addition, in the training data set for detection model, the sample's distribution is highly unbalanced; generally speaking, malware is far less than normal software [6]. This requires malware detection system must be able to filter out irrelevant or redundant features, to select the most important features to improve the accuracy and efficiency, and even to perform feature-filtering effectively on unbalance training data.

Zheng et al. [5] proposed a method on selecting corresponding effective features in different data types to address the unbalanced distribution problem. Yu et al. [7] adopted the correlation analysis method, through the measurement of the correlation among the features, while merging strong features with strong correlation, finally one or more as the representative of a subset is left, in order to achieve the purpose of feature selection. However, these feature selection algorithms for unbalanced data need to provide feature selection parameter in every category, where no satisfactory access exists; and the measurement-based method cannot effectively apply to the unbalanced training data training, and the efficiency still requires improvement.

This paper exploits function-based instruction fragments as file feature [3], and based on this, advances a connection metric Feature Selection method -- CDCBF (Class Driven Correlation based Feature Selection) that can be applied to unbalanced data. Experiments show that our algorithm gains great improvement in operation efficiency, and can effectively filter out redundancy, and unrelated features. Moreover, it is shown that our algorithm can improve the malware detection accuracy.

The remainder of the paper is organized as follows: section II introduces related work; section III presents the CDCBF algorithm in details; the section IV gives experimental results and analysis; finally, summary is given in section V.

II. RELATED WORKS

In malware detection system, no matter what kind of feature extraction method is adopted, document analysis method or testing strategy, the process is similar. Generally, it mainly involves two stages：one is the training phase, the other is the forecasting stage [2,4,5]. Detection model generation is the first step in the process. The specific procedures[2] are shown in figure 1:

From the training data, features are extracted, and feature selection is performed, so a file feature database is generated, which is used for detection model construction and selection.

978-1-61284-4577-0270-9/11/$26.00 ©2011 IEEE

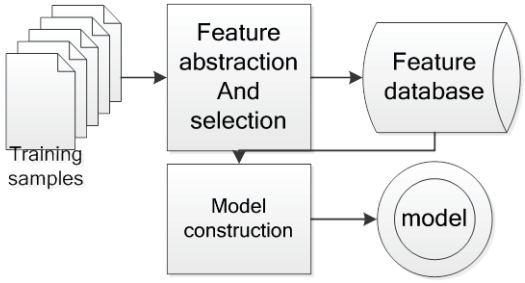

Fig.1 The flowchart of training phase in malware detection system

After the detection models are constructed, they are used to forecast the unknown sample, as the specific flow shown in figure 2:

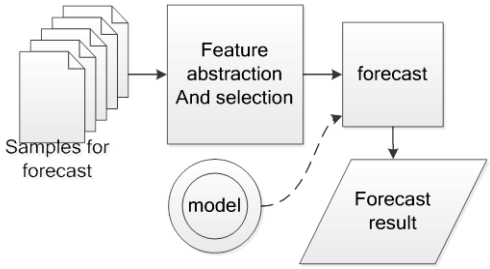

Fig.2 The flowchart of forecasting phase in malware detection system

For malware detection system, in the building process, no matter in what stage, feature selection is an extremely important step. Feature selection algorithms, according to whether using classification performance of the algorithm as the feature evaluation standard, can be divided into two types [2]: filter method and wrapper method. In order to guarantee efficiency and easy modelling for the malware detection system, this paper uses Filter method for feature selection [8].

A. Information Gain Based Feature Selection Methods

The information Gain, short as *IG*, is defined as follows:

$$IG(t) = \sum_{c\in\{c_i,\bar{c}_i\}} \sum_{t'\in\{t,\bar{t}\}} P(t',c)\log\frac{P(t',c)}{P(t')P(c)} \quad (1)$$

where IG(t) reflects how much information share a word provides for the entire classification, $P(\bar{t})$ says feature non–appearance probability, $P(C_i \mid t)$ signifies the probability of a sample belonging to C_i when t feature appears; $P(C_i \mid \bar{t})$ indicates the probability of a sample belonging to C_i when t feature doesn't appear.

This method selects the feature with the maximum or minimum value as candidate feature according to calculation of the statistics size. Nevertheless, the IG computes information proportion on the whole data set for each feature t, not focusing on different categories for the corresponding information share, which cannot perform feature selection in different categories.

B. Unbalanced Data Feature Selection Method (DFSF)

Reference [7] proposed a method of solving unbalanced data feature selection--DFSF, using the following formula:

$$S_{NEW}(t,c_i) = S(t,c_i)S_{OLD}(t) \quad (2)$$

where:

$$S(t,c_i) = sign(P(t,c_i)P(\bar{t},\bar{c}_i) - P(t,\bar{c}_i)P(\bar{t},\bar{c}_i))$$

$S_{OLD}(t)$ is calculated by formula (1), while

$$sign(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

can define the relationship between feature t and the classification, namely, positive correlated or negatively correlated. Combining Formula (1) and (2), we can get the following formula:

$$SIG(t,c_i) = S(t,c_i)IG(t) \quad (3)$$

Through this formula, the value of information gain of each feature in every classification can be calculated, now the DSFS algorithm process is expounded combining the malware detection, in this case, where contains two kinds of data: C_M means malicious software, C_B represents the normal software.

Algorithm name: DSFS feature selection [7]

Input: a feature set F, selection parameter l, $l=l_1+l_2$;

Output: new feature set F_{New};

Step 1. Calculate weight value of each feature through $SIG(t,c_M)$,select l_1 features of the biggest weight as positive correlation feature set F^+;

Step 2. Calculate weight value of each feature through $SIG(t,c_B)$,select l_2 features of the biggest weight as negative correlation feature set F^-;

Step 3. Combine features sets, where $F_{New} = F^+ + F^-$;

Step 4. Output new features set F_{New}

The critical issue of this method is the ratio between l_1 and l_2, reference[7] exploits two methods: one is to adopt the whole space search, the other is the Wrapper[7] method. However, these two schemes involve iterative calculation, which leads to high time complexity, thus they cannot effectively solve the feature selection problem in malware detection.

C. Association Analysis

In [9], feature that is highly correlated with classification but not highly correlated with other features is considered as effective feature. Its method (FCBF) procedure flow is shown in figure 3:

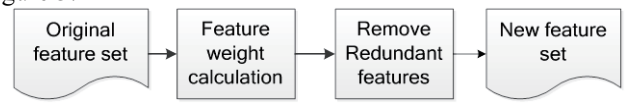

Fig.3 Feature Selection in Algorithm FCBF

To find the features of the classification, algorithm FCBF introduces "entropy" concept from the information theory. As for a feature X, its entropy is defined as follows [8]:

$$H(X) = -\sum_{x \in X} P(x) \log P(x)$$

Moreover, the conditional entropy of feature X related to feature Y is defined as:

$$H(X \mid Y) = -\sum_{y \in Y} P(y) \sum_{x \in X} P(x \mid y) \log P(x \mid y)$$

where $P(y)$ is prior probability of each component from variable Y, $P(x \mid y)$ indicates posteriori probability of x related to y. In order to avoid relative error problem when facing multi-valued type data, FCBF adopts SU (Symmetrical Uncertainty) method, defined as follows:

$$SU(X,Y) = 2 \cdot \frac{H(X) - H(X \mid Y)}{H(X) + H(Y)}$$

where the domain for SU is [0, 1], when $SU = 1$, it says two features are completely associated, whereas when $SU = 0$, it indicates two features are independent from each other.

The main ideas of FCBF is that if correlation between a feature t and another feature f is greater than it to the classification, which will show feature t in the dataset is redundant feature relative to f. The procedures of this algorithm are as follows:

Algorithm name: FCBF feature selection [9]

Input: feature set F, threshold parameters tp,

Output: new feature set F_{New};

Step 1. Calculate correlation SU (f, c) between each feature and the classification, according to the parameter tp, choose features satisfying $SU(f,c) > tp$ as candidate set F_{Cdd};

Step 2. Put features in F_{Cdd} in descending order according to $SU(f,c)$ values,

Step 3. Pick out feature f_p from F_{Cdd} according to the order, and add feature f_p into F_{New};

Step 4. Calculate $SU(f_p,f)$ of feature f_p with other feature f, if $SU(f_p,f) > SU(f,c)$, remove feature f from F_{Cdd},

Step 5. Repeat Step3 ~ Step 4 until F_{Cdd} is empty.

This method adopts a two-stage feature selection, the first stage is to screen out features classification related, and the second phase will remove redundant features. Nevertheless, this method does not consider relationship between feature and classification during the first stage, making the feature selection of unbalanced data characteristic easily fall into the local optimal solution.

III. ASSOCIATION METRIC FEATURE SELECTION METHOD FOR UNBALANCED DATA

In view of the both advantages and existing problems in the above algorithms, based on the approach treating function based instruction fragments as file features in [3], this paper advances a Association Metric Feature Selection methods - CDCBF (Class Driven Correlation based Feature Selection) which can be applied for unbalanced data. This method combines the advantages from DSFS and FCBF algorithm, and concentrates on the specific requirements of malware detection for the corresponding improvement. The specific flow is shown in figure 4:

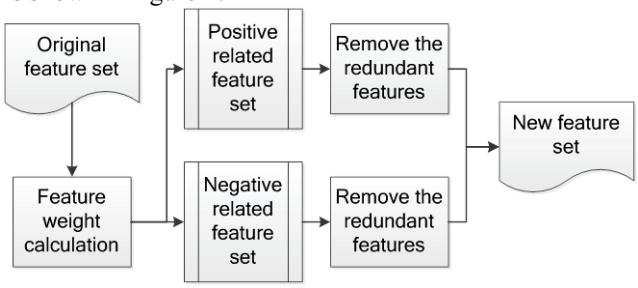

Fig.4 Feature Selection Algorithm CDCBF

The CDCBF method, after calculating feature weights, implements feature selection respectively for different classification, again conducts redundant feature filtering on the foundation of each subset.

Aimed at the unbalanced data feature selection problem, the DSFS algorithm thoughts is imported, that is, the corresponding important features are selected separately from malicious software and normal file, in addition, a method to automatically determine the proportion of positive and negative correlation is presented.

After selecting positive correlation and negative correlation features, association metric is carried out to corresponding features in these two subsets, where the redundant features is filtered out. Through the set division, efficiency of the algorithm is improved, which also ensures features of different classification will not be filtered out because of their strong relevance.

A. *Feature Proportion Selection*

In order to automatically determine the proportion of positive to the negative correlation, this paper puts forward an approach according to file samples distribution of the training set. Supposed the total number of candidate features is l, according to the inverse number ratio between malicious software and normal software in the training set, the candidate features number of positive or negative correlation is determined. That is to say, if the total number of feature candidates in the training set is 100 with 60 malicious software and 40 normal files, then the selected features related to malicious software must not be more than 40, while features related to normal files must not be more than 60, according to the inverse ratio. The specific formula is as following:

$$l_M = l - l \times \frac{P(c_M)}{P(c_M) + P(c_B)} = l - l_B \tag{4}$$

where l_M means the number for *Malware* classification, l_B is for *Malware* classification, which can effectively control unbalanced classification problem in the training data, the concrete effect of this method is verified by experiment in the section IV.

B. *Algorithm CDCBF*

The CDCBF algorithm firstly calculates weight of each feature through formula 3, while this statistic is associated with classification. This paper mainly aims at the binary classification problem, and it is positive classification related when the calculation result is positive (in this application malicious software is positive related) while negative for the negative classification. Again according to the various types of samples distribution in training set, the features with strong relevance to other features are selected respectively to compose several new feature subsets; In order to reduce redundant ones in the feature set, each feature subset employs association analysis based selection method, which extracts several most representative features from each subset to compose new feature set. Specific procedures of this algorithm are as follows:

Algorithm name: CDCBF feature selection
Input: a feature set F, selection number l,
Output: new feature set F_{New};
Step 1. Screen out positive related features: calculate $SIG(f, C_M)$ value of each feature to C_M, according to formula (4) then calculate l_M, select l_M features from F with the greatest SIG value to constitute positive correlation feature set F^+,
Step 2. Use the same method to generate F^-,
Step 3. Put the features in F^+ and F^- in descending order according to the SIG value,
Step 4. According to the order, pick out f_p from F^+ and add it to F^+_{New},
Step 5. Calculate $SU(f_p,f)$ value of f_p with other feature f in F^+ respectively, If $SU(f_p,f) > SU(f,c)$, removed feature f from F^+,
Step 6. Repeat Step 4 ~ 5 until the F^+ is empty.
Step 7. Use Step 4 ~ 6 to generate F^-_{New}, get the final result: $F_{New} = F^+_{New} + F^-_{New}$.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In order to validate the improved effectiveness of feature selection algorithm, SVM[14] is used respectively for training with the traditional feature selection algorithm and with the algorithm in this paper, while the training result under SVM model is taken as the efficiency index of the algorithm.

A. Experimental Data

So as to determine the parameters of CDCBF algorithm and to verify its effectiveness, this paper adopts the malicious software package from VxHeavens[15] and normal PE file collected from Windows XP system as experimental data. From malicious software package and normal file data collection, three pieces of data without intersection are abstracted to constitute the following several training data set. The detail is shown in table I.

Table I experimental data set details

	TDS1	TDS2	TDS3
Malware	2500	500	4000
Benign	2500	4500	1000

In the feature subsets validity evaluation, LIBSVM[14] is used to evaluate its performance. To make the data format fit LIBSVM requirements, function based instruction fragment in [3] is used to stand for the file and converted to VSM(Vector Space Model)[11]. Each feature in VSM has only 0 or 1 value, saying whether the file contains the instruction fragment. For example, from the file training set 10 features are extracted, while VSM value of one file is v=<1,0,1,1,1,0,1,0,0,1>, where 1 says the feature existed in the file v but 0 means not.

B. Evaluation

To evaluate the performance of various methods, this paper adopts harmonic average $F1$ and detection rate as performance evaluation indexes[12]. $F1$ is defined as follows:

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

Where, *Recall* and *Precision* separately mean recall and precision.

Detection rate is an important index of malware detection system performance evaluation[13], defined as follows: Detection rate (RR) = detected malware quantity/ total malware quantity .

This index is mainly to measure malicious software detection ability for detection system, but it does not consider the rate of false report of the system.

C. Experimental Environment and Settings

Experiment environment is Pentium4 CPU 3.00 GHz, 1.00 GB memory, Windows XP operating system. Algorithm adopts the c++ language, in VC9 platform for realization. In order to prove the improvement in efficiency and redundancy reduction of the algorithm, this paper designs two groups of test, as shown in table II:

Table II experimental Settings

experiment	purpose	data
1	Determine the feature quantity selection parameters l for algorithm	TDS1 TDS2 TDS3
2	Validate result of feature selection	TDS1 TDS2 TDS3

Experiments 1 first determines the experimental needed parameters, then on this basis, the efficiency and feature selection operation result of the improved CDCBF are compared with the traditional algorithm in experiments 2.

D. Experiment Result

An experiment 1 is to determine the parameter of the feature selection method in CDCBF: namely the feature quantity *l* during selection. However, because *l* value setting mainly aims at dimension reduction in the first step in CDCBF, while in the second feature selection step redundant features will be removed, the final feature quantity tends to be smaller than *l* after CDCBF completing feature selection.

In order to validate this parameter, it first chooses to carry out experiment on balanced data set TDS1, and the experimental results are shown in table III.

Table III parameter of feature selection method in CDCBF and the TDS1 experimental results

l	time	quantity	*F*1	*RR*
50	**140.9s**	46	91.9%	92.2%
100	161.3s	93	93.9%	94.5%
500	189.4s	452	96.1%	96.4%
1000	211.0s	879	**98.3%**	**98.5%**
2000	231.0s	1631	97.0%	97.9%
5000	247.1s	4175	97.1%	98.0%
/	258.3s	ALL	96.3%	97.8%

The experimental results from table 3 show that the CDCBF algorithm on TDS1 datasets performs the best when *l* is1000, which improves not only the efficiency but also the detection rate.

Next on the TDS2 and TDS3, experiments validate the difference with different *l* value on unbalanced datasets. Experimental results are shown in table IV and table V.

Table IV parameter of feature selection method in CDCBF and the TDS2 experimental results

l	Time	Quantity	*F*1	*RR*
50	**132.5s**	43	65.4%	55.7%
100	173.4s	98	67.8%	57.1%
500	197.1s	471	72.4%	59.0%
1000	233.1s	901	**82.9%**	**65.6%**
2000	254.0s	1754	80.1%	65.1%
5000	265.1s	4008	72.9%	58.3%
/	273.3s	ALL	70.1%	54.1%

Table V parameter of feature selection method in CDCBF and the TDS3 experimental results

l	time	quantity	*F*1	*RR*
50	**121.9s**	35	63.7%	81.3%
100	167.8s	84	66.3%	83.0%
500	185.4s	468	70.2%	86.1%
1000	249.1s	895	**79.9%**	**93.9%**
2000	263.4s	1799	74.7%	87.4%
/	297.2s	ALL	65.2%	83.4%

The data in Table 4 & 5 illustrates it works the best when *l* is 1000, the same as the case on the balanced dataset, which also shows *l* value will not change with data distribution.

The above three groups of data display that taking *l* for 1000 is the best. Therefore, *l* value in this paper is 1000 by default for other related experiments.

Experiments 2 mainly aims to verify the effectiveness of CDCBF feature selection algorithm, including detection efficiency and effect, which will be compared with the mentioned IG[8]、DSFS[7] and FCBF[9].

The parameter *tp* in the FCBF is set to 0.9, and positively related and negative related features ratio employs the same method in DSFS as in CDCBF method. The remaining methods, which need the feature selecting quantity setting, get the same *l* value to be 1000.

The experimental result of TDS1 is shown in table VI, which shows that through CDCBF feature selection more irrelevant and redundant features can be removed compared to other methods, and when using SVM training production model, time efficiency is improved, as well as SVM training indexes (*F*1 and RR) raised. Because TDS1 is balanced distributed data set, so other algorithms have also achieved a good result.

Table VI experiment result comparison on TDS1

Feature selection method	time	*F*1	*RR*
IG	262.1s	92.2%	94.1%
DSFS	234.2s	96.9%	98.0%
FCBF	228.4s	97.4%	97.7%
CDCBF	**211.0s**	**98.3%**	**98.5%**

The following group of data is the result comparison of various feature selection methods on TDS2. Due to the high normal files proportion on TDS2, so, the generated model shows a tendency of normal file judgment, and the RR value will relatively be lower. Specific data is shown in table VII.

Table VII experiment result comparison on TDS2

Feature selection method	time	*F*1	*RR*
IG	248.9s	70.1%	52.3%
DSFS	264.4s	80.2%	63.7%
FCBF	244.9s	81.4%	**65.9%**
CDCBF	**233.1s**	**82.9%**	65.6%

Because the TDS2 is unbalanced data set, so index value of every method reduced. CDCBF relative to other methods is the best referring to *F*1, but the *RR* indicators is not all the best, though very similar to FCBF method. Meanwhile, with CDCBF, SVM training efficiency is still the best method.

One last group data with TDS3 and TDS2 is similar on unbalanced data, because malware from TDS3 occupies 80%, so generated SVM model after training in this group of data has tendency of malicious software judgment, so, *RR* value is higher relative to TDS2 in this group of data. The specific data is shown in table VIII.

Table VIII experiment result comparison on TDS3

Feature selection method	time	*F*1	*RR*

IG	279.1s	74.1%	86.5%
DSFS	259.2s	77.5%	90.3%
FCBF	264.9s	76.0%	92.7%
CDCBF	**249.1s**	**79.9%**	**93.9%**

Data in table VIII shows improved CDCBF feature selection method achieves a better effect compared with other methods on unbalanced datasets.

The three groups of data of experiments 2 illustrates that, in order to achieve a good model training result, it is better to keep data distribution in training set balanced, if data set is imbalanced, its generated model in the prediction will shows judgment tendency to file that has a quantity advantage.

At the same time, three groups of data also verified that CDCBF feature selection approach could effectively remove unrelated and redundant features, obtaining time efficiency improvement in model training, and achieving better performance within model detection.

V. CONCLUSION AND FUTURE WORK

Based on the malware detection system requirements and taking function-based instructions clips as malicious software features, this paper proposes a kind of malware detection feature selection algorithm -- CDCBF. This algorithm works well to select the relative important features for different file types, and can automatically determine feature selection scale in different file classification to remove not related features; On the basis of filtering result, CDCBF performs respective analysis on different classification feature subsets as well as the calculation of association between the features in the subset to filter out the redundant ones. Experiments have proved that CDCBF algorithm can improve the operation efficiency of training algorithm in the detection model, while irrelevant and redundant ones are effectively removed from the feature set compared with other algorithms. More strict mathematical formula for the feature selection ratio will be the future research content.

ACKNOWLEDGEMENTS

This work is supported by the Shenzhen New Industry Development Fund under grant No.CXB201005250021A, the National Natural Science Foundation of China (No. 10771176), National Science Foundation of Guangdong Province (No. 2008A090300017).

REFERENCES

[1] Fred Cohen. Computer Viruses.CA: University of Southern California,1985.

[2] Nwokedi Idika and Aditya P. Mathur. A Survey of Malware Detection Techniques. CA: West Lafayette: Department of Computer Science, Purdue University, 2007: 3-10.

[3] Kai Huang, Yanfang Ye and Qingshan Jiang. " ISMCS: An Intelligent Instruction Sequence based Malware Categorization System", ASID,2009.

[4] Dmitriy Komashinskiy and Igor Kotenko. "Integrated Usage of Data Mining Methods for Malware Detection". Information Fusion and Geographic Information Systems, 2009, vol 7, pp 343-357.

[5] Muazzam Ahmed Siddiqui. Data Mining Methods for Malware Detection.CA: Orlando, Florida: University of Central Florida, 2008.

[6] Karsperky-Labs. Kaspersky Security Bulletin: Statistics 2008[R]. CA: 2009.

[7] Zhaohui Zheng, Xiaoyun Wu, Rohini Srihari. "Feature selection for text categorization on imbalanced data". ACM SIGKDD Explorations Newsletter, 2004,vol 6, no 1.

[8] Pawlak.Z, Rough Sets and Intelligent data analysis, CA: Computer and Information Science,2002

[9] Lei Yu , Huan Liu. "Efficient Feature Selection via Analysis of Relevance and Redundancy". The Journal of Machine Learning Research, 2004, vol 5, pp. 1205-1224.

[10] Guangzhi Qu, Salim Hariri and Mazin Yousif. "A New Dependency And Correlation Analysis for Features". IEEE Transactions On Knowledge And Data Engineering, 2005, Vol. 17, No. 9.

[11] J.Han and M.Kamber. Data mining: concepts and techniques, 2rd edition. CA: Morgan Kaufmann, 2006.

[12] Muazzam Ahmed Siddiqui. Data Mining Methods for Malware Detection. CA: Orlando, Florida: University of Central Florida, 2008.

[13] Jon Oberheide, Evan Cooke and Farnam Jahanian. "CloudAV: N-Version Antivirus in the Network Cloud". 2008, 17th USENIX Security Symposium, pp. 91-106.

[14] LIBSVM -- A Library for Support Vector Machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[15] Vx Heavens, http://www.vx.netlux.org.

[16] Yanfang Ye, Qingshan Jiang, Weiwei Zhuang. Associative Classification and Post-processing Techniques using in Malware Detection System[C]. International Conference on Anti-counterfeiting, Security, and Identification (ASID 2008).

[17] Yanfang Ye, Dingding Wang, Tao Li, and Dongyi Ye. IMDS: Intelligent malware detection system[C]. In Proccedings of ACM International Conference on Knowlege Discovery and Data Mining (SIGKDD), 2007, On page(s): 1043-1047.

[18] Yanfang Ye, Dingding Wang, Tao Li, Dongyi Ye, and Qingshan Jiang. An intelligent pe-malware detection system based on association mining[J]. J Comput Virol, January 2008, On page(s): 323-334.

[19] Yanfang Ye etc. SBMDS: an interpretable string based malware detection system using SVM ensemble with bagging. Journal in Computer Virology, Volume 5, Number 4, November, 2009.