Survey on Malicious Code Intelligent Detection Techniques

Bo Yun ZHANG, Xi Ai YAN, De Quan TANG

Deptment of Computer Science and Technology, Hunan police College, Changshan, Hunan 410138, China
hnjxzby@hotmail.com

Abstract. This paper describes the research status and progress of malicious code intelligent detection techniques. It introduced the research background of the malicious code detection. Then the classified and analyzed research work from the theoretical research on malicious code detection, malicious code static detection and dynamic detection, integrated detection and based on biological immune detection techniques are described in detail. The contrastive analysis of different detection methods principle and further study on the future directions are discussed.

1. Introduction

Malicious code is a general term for a variety of hostile or intrusive software, including viruses, worms, Trojans, Rootkit, backdoors, botnets, spyware, and so on. Malicious code may steal the information of computer users and privacy in the user's knowledge; it is possible to control illegal computer systems and cyber source, the destruction of computer and network credibility, integrity and availability, so as to control for non interest for malicious code. According to Symantec released the "2016 Internet Security Threat Report" [1], during 2015 the company captured 430 million new malicious code, a daily average of 1 million 170 thousand new malicious code released to the Internet, an increase of 36% over 2014. 360 Internet Security Center released the "2016 China Internet Security Report" [2] display, during 2016 360 Internet Security Center intercepted PC end new malware samples 190 million, Android platform malware samples added 14 million 33 thousand. Such a large number of malicious codes have become the biggest security threat to the Internet, which has seriously affected the information security of all countries in the world. Because the current malicious code detection methods have more limitations, it is urgent to study new effective malicious code detection methods to ensure the security of network space.

For malware analysis and prevention, it must first collect malicious code related information. By collecting the performance of the program in different aspects and levels, it provides information support for the malicious analysis of the program. In information collection, the main technologies used include semantic enhancement, API monitoring, integrity checking, code auditing and virtual execution, and so on. In order to determine the malice of the program, it is necessary to extract the features of the program information that can distinguish the malicious character of the program, and use it as the basis of the malicious decision of the program. The research achievements at home and abroad, the formation characteristics mainly include: the system call feature, normalized code feature and N-gram feature, control flow graph (CFG: Control Flow Graph) features, instruction sequence features and characteristics and so on file format[3]. The malicious behavior decision algorithm compares the known malicious features with the program to be detected, and finally determines the malicious nature of the program. In recent years, machine learning, data mining, graph theory,
immunology and sociology in different fields such as the results of cross application of malicious code security, researchers have proposed many heuristic intelligent detection methods.

The first section of this paper introduces the abstract theory of computer viruses, which is of great significance to guide the practice of malicious code prevention. The second section introduces the malicious code detection technology based on static features, and makes a brief analysis of its advantages and disadvantages. The third section sums up the malicious code detection technology based on dynamic features. The fourth section introduces the malicious code detection technology based on fusion features, and points out that the multi-feature fusion detection method is better than the single feature detection method in general. The fifth section introduces the malicious code detection technology based on biological immune technology, the sixth section introduces the future research trends, and the seventh section summarizes the research.

2. Research on Malicious Code Detection Theory

Abstract theory of computer viruses is of great significance for understanding computer viruses, studying the basic nature and mathematical characteristics of computer viruses, and guiding the practice of anti-virus. Since the first computer virus based on the abstract theory of Turing machine, some important theories were proposed and studied, and obtained some important results. There are two most important conclusions: one is about all possible computer viruses the un-decidability of [4], another is the existence of computer virus detection cannot be [5]. Zuo Zhihong [6] studied the computational complexity of computer viruses. It is proved that there is a computer virus in theory, and its infection process has any large computational complexity, and the existence of non-decidable computer viruses is proved. They have no minimal detection process. Diomidis Spinellis proved that detecting a bounded length variant virus is a NP complete problem, [7]. From the abstract theory of computer viruses, it is necessary to use different heuristic knowledge to identify malicious code from different angles and improve the accuracy of judgment.

3. Malicious Code Static Detection Techniques

The detection rate of malicious code static detection method can provide testing environment safer and faster, but is easily affected by the packers and obfuscation technology, before analysis is usually required to decrypt shelling, and standardized processing. Table 1 presents the features and related references used in various static detection techniques.

| Classification feature | Acquisition method | Detection method | Reference |
|------------------------|--------------------|------------------|-----------|
| The byte code n-grams | converted directly into 16 Decimal classification | Ref.[8,9] |
| File format | format parsing classification | Ref.[10,11] |
| Gray image features | convert files to gray images Cluster | Ref.[12,13] |
| Function call graph | disassembly discriminate distance metric learning | Ref.[14] |

N-grams method is widely used in Natural Language Processing, in the detection of malicious code, n-grams expression refers to the adjacent elements of the n sequence, and the elements can be byte, instructions, or other information software function. The references [8][9] uses byte sequence n-grams as a feature, and supervised learning or semi-supervised learning is used to train the detection model. There are some exceptions to malicious code and the format of infected executable files; they are the key to detecting malicious code.

The references [10][11] proposed detection method to detect malicious code executable file format based on the information, this kind of method is usually not affected by the deformation or
polymorphism of obfuscation, feature extraction without traversing the entire executable file, extract features faster, but this method does not extract decision software behavior code section and data section as the feature information.

Binary executable files can be represented as binary strings of 0 and 1, each byte exactly corresponding to one pixel, and binary executable files can also be considered as a grayscale image. The references [12][13] maps binary programs to grayscale images, and uses image processing methods to detect known malicious or malicious code variants. This kind of method is effective for the packers or shell samples, high detection efficiency.

Kong et al. [14] disassemble the samples and generate the function call graphs of the samples from the assembly code. For each function, 6 types of features are extracted. For each type of feature, discriminate distance metric learning algorithm is used to clustering, and finally use the weight combination to get the detection model.

Malicious code static method is easy to be affected by the packers and malicious code obfuscation technology; part of the anti-disassembly technology, resulting in a static method is invalid.

4. Malicious Code Dynamic Detection Techniques
Software exists in binary files in a static way, dynamically stored in memory. Through static analysis and detection software, its behavior is not necessarily safe. The essence of dynamic method can accurately identify the malicious behavior, is still valid for packers, deformation, polymorphism, mixed sample. Table 2 presents the features, detection methods and references that are involved in different dynamic detection techniques.

| Classification feature | Acquisition method | Detection method | Reference |
|------------------------|--------------------|------------------|-----------|
| Variable length API subsequences | dynamic API sequences | classification | Ref.[18-21] |
| Operation code n-grams | dynamic instruction sequence control flow graph, data flow graph, system call graph, function call diagram subgraph | K-means | Ref.[15,16,17] |
| Graph | Subgraph isomorphism/ match | Ref.[22-29] |

4.1. Detection Method Based on Operation Code n-grams
The document [15][16] uses the operation code n-grams as the feature, and then the machine learning classifier algorithm is applied to construct the detection model. Pai et al. [17] also use the sequence of operations code as features, construct the model using the hidden Markov method (HMM), then evaluate the degree of similarity of samples, and finally use K-means and EM algorithm to cluster.

4.2. Detection Method Based on API Call Sequence
Nair et al. [18] through dynamic analysis malicious code and its variant API call sequence, extracted from this kind of malicious code sharing API call short sequence as signature, can achieve the same class of malicious code detection.

Chen and Fu [19] obtained API call sequence through dynamic analysis of malicious code, API calls short sequence traversal API call has the same length, and then put these short sequences into vector, as the malicious code signature.

Firdausi et al. [20] proposed a malicious code detection method for dynamic behavior, which monitors the system calls of samples, and then converts the system call reports into vector space
models. The method uses two feature extraction methods: (1) use binary 0 or 1 to represent the value of the feature; (2) use the frequency of the system call to represent the value of the feature. By using a variety of machine learning classification algorithms to deal with two features, the system calls frequency feature representation to achieve slightly better results.

Ahmed et al. [21] perform malicious code detection by mining spatial and temporal information in the dynamic API call sequence. Spatial information refers to the statistics of the parameters and return values of API calls, including mean, variance, entropy, minimum and maximum values. Temporal information refers to the transfer probability of the API call sequence.

The above dynamic detection method using system call sequences of fixed length, but it is difficult to determine the short length of sequence is reasonable, even if an optimal length value, also lost a lot of semantic information is the length of the other.

4.3. Detection Method Based on Graph

The researchers conducted a preliminary study on the detection of malicious code based on the graph, these methods will be expressed as the software control flow chart, data flow chart, system call graph, function call graph, and then through the detection of similarity measurement, data mining and machine learning methods to achieve malicious code.

Bruschi et al. [22] proposed a malicious code detection method based on control flow graph matching. The subgraph isomorphism algorithm was used to determine whether the signature of the signature library was a subgraph of the sample to be checked. Since subgraph isomorphism problem is a NP complete problem, the efficiency of this method is low, and it is impossible to determine whether subgraph isomorphism exists when the scale of graph nodes reaches a certain number.

Bonfante et al. [23] improved the above method. The control flow diagram of malicious code was transformed into tree automata, and then matched. This method improved the efficiency of the above methods to some extent.

Cesare et al. [24] use an executable file as a control flow graph for each set of control flow graph, extract key semantic control statements, control flow based on the expressed as a string, and then use the K subgraph (k-subgraphs) or n-gram features are extracted from the string, using feature filtering algorithm selected the most relevant features as the feature vector samples, the final calculation of each signature detecting sample feature vector and the malicious code signature library standardized compression distance, when the distance is less than the specified threshold, judging for malicious code.

Yu and Islam et al. [25] also propose a software reliability evaluation method based on data stream. This method evaluates the credibility of the software by analyzing the data flow in the stack, stack, register, memory and so on.

Karbalaie et al. [26] proposed to detect malicious code mining system based on call graph; the method firstly converts the system call sequence as a graph, and then uses the algorithm structure model graph mining (gSpan) mining frequent connected subgraphs. The shortage of this method is that the algorithm of graph mining is complex and inefficient.

Carrera et al. [27] expressed the executable file as a function call graph, control flow graph and then use the function node, other auxiliary nodes; nodes for all nodes according to the matching proportion of similarity, then the samples are classified into the corresponding class application system clustering algorithm.

Xu et al. [28] malicious code is expressed as a function call graph, where each node represents a function for the machine instruction sequence, machine instruction sequence, and then use the auxiliary graph node function, approximate graph edit distance between two specimens.

Kostakis et al. [29] expressed malicious code as a function call graph, in order to compare the similarity between samples, the sample graph edit distance calculation of the method, the graph edit distance computation belongs to NP complete problems, poor efficiency of the method, and cannot calculate the large scale map.

The above methods are explored from different perspectives to solve the problem of malicious code
detection, malicious code detection method is proposed based on graph of different ideas, has made many constructive achievements, but there are still some problems to be solved: (1) part of the method efficiency is not ideal, for more nodes within a limited time to complete the graph still need to be studied, the time complexity is polynomial algorithm; (2) part of the graph representation size is fine, resulting in graph size and high complexity, still need to improve the graph that, can effectively describe the software, and improve the storage and matching efficiency; (3) due to confusion over the malware variants that figure may change, some methods cannot detect the malicious code.

5. Malicious Code Detection Technology Based on Fusion Features

Researchers have proposed a variety of malicious code detection methods based on machine learning. These methods represent executable files as features at different levels of abstraction, and use these features to train classifiers in order to intelligently detect unknown malicious code. All of these methods with high accuracy to detect unknown malicious code, but each method has some disadvantages and limitations. Classification algorithm in different feature view has inductive bias, no one has the high performance classification algorithm for any classification problem, and so the researchers proposed various detection methods of integration feature type. Table 3 presents the classification features and related references used in different integration detection techniques.

| Table 3  Malicious Code Detection Technology Based on Fusion Features |
|---------------------------------------------------------------|
| **Dynamic features** | **Static feature** | **Detection methods** | **Detection accuracy (%)** | **Reference** |
| Dynamic sequence API | operation code n-grams | Multi-classifier | 96.22 | Ref.[30] |
| Program behavior API call sequence | Static DLL, API PE format information | SVM-AR | - | Ref.[32] |
| API call sequence | BKS/multi-weight | 98.7 | Ref.[31,33] |
| Dynamic API operation code n-grams | Static API byte code n-grams | Multi-classifier | 97.87 | Ref.[34] |
| | | Multi-classifier | 99.00 | Ref.[35] |

Integrated classification method first from the data set select a sufficient variety of data extraction and feature selection, and then use the sub classifiers suitable for training, determine the number of sub optimal classifier, and then use a certain strategy combination of each classifier, in order to obtain better results of classification. The method of constructing integrated learning is shown in Figure 1.

Santos et al. [30] first proposed static and dynamic feature detection method. The static characteristic is the operation code sequence of n-grams, the dynamic characteristic is the system call, then it mixes the static and dynamic characteristics, using a variety of machine learning classification algorithm, training can distinguish benign software and malicious code model. Experimental results show that the proposed method achieves 96.22% accuracy, the rate is better than the result of using static or dynamic features alone.

Guo et al. [31] divided the API calls into 7 broad classes, constructed a base classifier using each of the API sequences of each class, and then integrated the 7 base classifiers using the BKS algorithm. The method achieves a detection rate of 98.7%, but there is a 9.1% false positive rate. The method with the majority voting method and the best base classifiers are compared, the result is slightly better
than that of the two algorithms, the main disadvantages of this method is to use the same algorithm to construct the base classifier, and the complementarities between features is not strong.

Lu et al. [32] proposed fusion detection method of malicious code static characteristics and dynamic characteristics, the method of extracting import section of the DLL and API as the static characteristics, the use of 6 tools manually extracted 12 common dynamic behavior characteristics of malicious code (such as creating a file, DLL injection, hidden services, open ports) then, the combination of static and dynamic characteristics, the use of a new integrated learning algorithm SVM-AR training classification model proposed by the author. The experimental results show that the accuracy of classification algorithm is improved obviously after combination feature, and the SVM-AR algorithm is superior to the commonly used ensemble learning algorithms (such as Bagging and Boosting).

Krawczyk et al. [33] extract PE file headers and import section API calls as features, the original feature set is divided into multiple feature subsets, and then build a base classifier based on each feature subset. This method has not integrated all base classifiers, but the use of evolutionary algorithms to select a subset of base classifiers, assigning a weight to the selected base classifier, using weighted voting integrated subset of base classifiers selected. The method uses unbalanced data sets to classify experiments, and achieves better results than common ensemble learning methods.

Ozdemir et al. [34] proposed Android malware detection fusion method of static and dynamic characteristics, the method to extract the features of the 4 types: static Native API, dynamic Native, static Dalvik API Byte API and dynamic Dalvik. 4 types of features are used to train different base classifiers, and then the selective ensemble method is used to fuse the base classifiers. The method achieves 97.87% accuracy and outperforms all base classifiers. The deficiency of the method is that the 4 types of features are API information and lack the diversity of feature types.

Bai Jinrong et al. [35] proposed new method to detect malicious code based on multi-view integrated learning, application of a variety of integrated learning method to fuse byte code operation code n-grams, characteristics of n-grams, the format information from the data layer and feature layer, model layer, layer fusion strategy is effective integration of the above features. Experimental results show that this method is based on early malicious code detection, the highest detection rate of new malicious code is 94.6%, and the maximum detection rate is 99% based on the small training set to detect 20 times of the large test set.

The above research shows that the multi-feature fusion detection method is better than the single feature detection method in general, and it is better than the single classifier method. Malware usually forges features similar to benign software to escape detection of antivirus software. However, it is difficult for malware to forge multiple abstract features at the same time to avoid detection. Fusing multiple types of features can not only improve the accuracy of the detection method, but also improve the robustness of the detection method.

6. Detection Methods Based on Biological Immunity

Biological immune system and prevent the malicious code encountered (Biological Immune System: BIS) problem is very similar, BIS uses a series of immune mechanism effectively solves the problem, so the simulation of artificial immune system with the immune mechanism (Artificial Immune System) is a direction to solve the problem.

S.Forrest is the founder of the research of artificial immune system, in 90s do a lot of pioneering work[36], in 1994, she proposed a computer virus detection method using immune negative selection mechanism, the immune mechanism applied to virus detection, computer virus immune system after this (CVIS: Computer Virus Immune System) research flourish development is the most important work of [38] computer virus immune model ARTIS S.Homeyr and IBM [37] universal framework laboratory proposed by Kephart et al. ARTIS is the development of S.Forrest's early work, BIS related concepts and mechanisms are well represented in the model, such as self, non self, the detector (detector), self-tolerance, clone selection, self-tolerance and co-stimulation etc..

The design thought most of the subsequent release of CVIS from ARTIS, such as the computer
virus detection system T.Okamoto and Ishida [39], with P.Harmer and G.B.Lamout computer virus immune architecture agent technology [40] and Rune Schmidt Jensen combined with computer virus immune model of HMM technology [41].

The computer virus immune model and ARTIS IBM Laboratory of different time, efficiency and practical applications based on the consideration, it adopted some biological immune mechanism but not fully simulated BIS, combined with the use of trap technology and automatic feature analysis techniques in which.

There are many researches in China, among them, the research team of Professor Li Tao of Sichuan University has done a great deal of research work in computer immunology, [42]. Now the application of main problems of the immune system is time efficiency problems, so in the future research work is mainly to solve the problem, reasonable design excellent algorithms, the balance mechanism more to reduce the time complexity.

7. Future Research Trends
This paper analyzes the research status of the above aspects at home and abroad, researchers explore the solution of malicious code detection problems from different perspective, proposes different ideas of the malicious code detection method, has made many constructive achievements, but there are still some challenging problems need to be solved:

(1) Although the new malicious code detection method based on the characteristics of the code can only identify less, but because of high false negative rate, false alarm rate is almost 0, short detection time and low system overhead, this method is still the main method of anti-virus software use. Heuristic based detection methods can generally identify some unknown malicious code, but there is the possibility of false positives, rarely deployed directly on the user's computer side. Many intelligent methods in the detection time, detection rate, false positive rate, system overhead and other aspects of the actual deployment of anti-virus software cannot meet the requirements, there is an urgent need to study new methods to meet the above indicators;

(2) Based on machine learning and data mining methods, cross validation is used to evaluate the performance of the detection method, and the training set and the test set are the mixture of the old and new samples. In the real scene detection, detection method of malicious code detection based on the existing sample at the same time, the finite sample detection of unknown samples based on the huge, so the existing detection methods of the experimental results is too optimistic, need to consider more real scene detection, detection method is effective;

(3) Malicious code detection is very complicated problem, and all kinds of detection methods have their own advantages and disadvantages. Some methods alone cannot solve the malicious code detection problem completely. At present, there is an urgent need to integrate many complementary detection methods to improve the detection rate of anti-virus software, reduce the false positive rate, shorten the detection time, and not increase the user's computer system overhead.

8. Conclusion
On the basis of full investigation and analysis, this paper summarized the research progress of malicious code intelligent detection. It summarized and analyzed research work from the theoretical research, the static detection of malicious code detection technology, dynamic detection technology, integrated detection technology and biological immune detection technology and other aspects of the existing research work, it analyzed the principle of different kinds of malicious code detection technology, and comparing their advantages and disadvantages, finally introduced the future research trend.

Acknowledgments.
This work was partially supported by the National Natural Science Foundation of China under Grant No.61471169.
References

[1] Paul W, Ben N, Kavitha C, et al. Symantec internet security threat report 2016, http://www.symantec.com, 2016.4.

[2] 360 Internet Security Center. 2016 China Internet Security Report . http://zt.360.cn/1101061855.php?did=1101062514 & did=490278985.2017.2.

[3] Mihai Christodorescu, Somesh Jha, Christopher Kruegel. Mining Specifications of Malicious Behavior[A]. In: Proceedings of the 6th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering[C], Dubrovnik,Croatia, 2010: 5-14

[4] F. Cohen, Computer Viruses: Theory and Experiments. Computers and Security 6(1) (Feb.1987) pp. 22-35.

[5] D. M. Chess and S. R. White. An undetectable computer virus. Virus Bulletin Conf., Sept. 2000, [Online]. Available: http://www. research. ibm.com/ antivir us/ SciPapers/ VB2000DC.pdf, June 2002.

[6] ZUO Zhi-hong, ZHU Qingxing, Zhou Mingtian, On the time complexity of computer viruses, IEEE transaction on information theory, 51(8), 2005.

[7] Diomidis Spinellis. Reliable Identification of Bounded-Length Viruses Is NP-Complete. IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 49, NO. 1, JANUARY 2003. pp280-284

[8] Kolter J Z, Maloof M A. Learning to detect malicious executables in the wild. [C]. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004: 470-478.

[9] Santos I, Penya Y K, Devesa J, et al. N-grams-based File Signatures for Malware Detection. Proceedings of the 2009 International Conference on Enterprise Information Systems (ICEIS), 2009, 9: 317-320.

[10] Shafiq M Z, Tabish S M, Mirza F, et al. Pe-miner: Mining structural information to detect malicious executables in realtime. Recent advances in intrusion detection, Springer Berlin Heidelberg, 2009: 121-141.

[11] Bai J, Wang J, Zou G. A Malware Detection Scheme Based on Mining Format Information. The Scientific World Journal, 2014.

[12] Nataraj L, Karthikeyan S, Jacob G, et al. Malware images: visualization and automatic classification[C]. Proceedings of the 8th international symposium on visualization for cyber security. ACM, 2011: 4.

[13] HAN Xiao-guang, QU Wu, YAO Xuan-xia, et al. Research on malicious code variants detection based on texture fingerprint. Journal on Communications, 2014, 35(8):125-135.

[14] Kong D, Yan G. Discriminant malware distance learning on structural information for automated malware classification[C]. Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2013: 1357-1365.

[15] Moskovitch R, Feher C, Tzachar N, et al. Unknown malcode detection using OPCODE representation[M] //Intelligence and Security Informatics. Springer Berlin Heidelberg, 2008: 204-215.

[16] Shabtai A, Moskovitch R, Feher C, et al. Detecting unknown malicious code by applying classification techniques on opcode patterns. Security Informatics, 2012, 1(1): 1-22.

[17] Pai S, Di Troia F, Visaggio C A, et al. Clustering for malware classification. Journal of Computer Virology and Hacking Techniques, 2016: 1-13.

[18] Nair V P, Jain H, Golecha Y K, et al. MEDUSA: MEtamorphic malware dynamic analysis using signature from API[C]. Proceedings of the 3rd International Conference on Security of Information and Networks. ACM, 2010: 263-269.

[19] Chen F, Fu Y. Dynamic detection of unknown malicious executables base on api interception[C]. Database Technology and Applications, 2009 First International Workshop on. IEEE, 2009: 329-332.
[20] Firdausi I, Lim C, Erwin A, et al. Analysis of machine learning techniques used in behavior-based malware detection[C]. Advances in Computing, Control and Telecommunication Technologies (ACT), 2010 Second International Conference on. IEEE, 2010: 201-203.

[21] Ahmed F, Hameed H, Shafiq M Z, et al. Using spatio-temporal information in API calls with machine learning algorithms for malware detection[C]. Proceedings of the 2nd ACM workshop on Security and artificial intelligence. ACM, 2009: 55-62.

[22] Bruschi D, Martignoni L, Monga M. Detecting self-mutating malware using control-flow graph matching. Detection of Intrusions and Malware & Vulnerability Assessment. Springer Berlin Heidelberg, 2006: 129-143.

[23] Bonfante G, Kaczmarek M, Marion J Y. Architecture of a morphological malware detector. Journal in Computer Virology, 2009, 5(3): 263-270.

[24] Cesare S, Xiang Y, Zhou W. Control flow-based malware variant detection. IEEE Transactions on Dependable and Secure Computing, 2014, 11(4): 307–317.

[25] Yu D, Islam N. A typed assembly language for confidentiality. Programming Languages and Systems. Springer Berlin Heidelberg, 2006: 162-179.

[26] Kolbitz C, Comparetti P M, Kruegel C, et al. Effective and Efficient Malware Detection at the End Host. USENIX security symposium, 2009: 351-366.

[27] Carrera E, Erdélyi G. Digital genome mapping–advanced binary malware analysis. Virus bulletin conference. 2004, 11.

[28] Xu M, Wu L, Qi S, Xu J, Zhang H, Ren Y, Zheng N. A similarity metric method of obfuscated malware using function-call graph. Journal of Computer Virology and Hacking Techniques, 2013, 9(1): 35–47.

[29] Kostakis O, Kinable J, Mahmoudi H, Mustonen K. Improved call graph comparison using simulated annealing. Proceedings of the 2011 ACM Symposium on Applied Computing, ACM, 2011, 1516–1523.

[30] Santos I, Devesa J, Brezo F, et al. OPEM: A dynamic-static approach for machine learning based malware detection[C]. International Joint Conference CISIS’12-ICEUTE’12-SOCO’12 Special Sessions. Springer Berlin Heidelberg, 2013: 271-280.

[31] Guo S, Yuan Q, Lin F, et al. A malware detection algorithm based on multi-view fusion. Neural Information Processing, Models and Applications, Springer Berlin Heidelberg, 2010: 259-266.

[32] Lu Y B, Din S C, Zheng C F, et al. Using multi-feature and classifier ensembles to improve malware detection. Journal of CCIT, 2010, 39(2): 57-72.

[33] Krawczyk B, Woźniak M. Evolutionary Cost-Sensitive Ensemble for Malware Detection. International Joint Conference SOCO’14-CISIS’14-ICEUTE’14, Springer International Publishing, 2014: 433-442.

[34] Ozdemir M, Sogukpinar I. An Android Malware Detection Architecture based on Ensemble Learning. Transactions on Machine Learning and Artificial Intelligence, 2014, 2(3): 90-106.

[35] Bai, Jinrong, and Junfeng Wang. Improving malware detection using multiview ensemble learning. Security and Communication Networks 9.17 (2016): 4227-4241.

[36] Forrest S., Perelson A.S., Allen L.R, and Cherukuri. Self-Nonself Discrimination in a Computer. Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy, Los Alamitos, 1994

[37] Hofmeyr S., Forrest S. Architecture for an artificial immune system. Evolutionary Computation. 2000,8(4):443–473

[38] J.O. Kephart. A biologically inspired immune system for computers. Proc. of the 4th International Workshop on the Synthesis and Simulation of Living Systems. MIT Press, 1994

[39] Okamoto T. and Ishida Y. A Distributed Approach against Computer Viruses Inspired by the
Immune System. IEICE Trans. on Communication. 2000, E83-B(5):908-915

[40] Harmer P.K and Lamont G. B. Agent Based Architecture for a Computer Virus Immune System. GECCO 2000, Las Vegas, Nevada, USA, July 8, 2000

[41] Jensen R. S. Immune system for virus detection and elimination. Master's Thesis, Technical University of Denmark, DTU, 2002.

[42] Li Tao. Computer immunology. Publishing House of electronics industry, Beijing, 2004