The Software Gene-Based Test Set Automatic Generation Framework for Antivirus Software

Liang Bai[1], Yu Rao[1], Shiwei Lu[2*], Xu Liu[2], Yiyi Hu[2]
[1] CNCERT, Beijing, China.
[2] Shanghai Roar Panda Network Technology Co., Ltd.

* Corresponding author. Tel.: +86 18550862423; email: 1074614564@qq.com
Manuscript submitted June 21, 2019; accepted August 6, 2019.
doi: 10.17706/jsw.14.10.449-456

Abstract: After studying the existing test set generation methods of antivirus software and sample analysis methods based on manual experience, the paper proposes a software gene-based test set automatic generation framework for antivirus software. Most of current test set automatic generation frameworks have problems of unstable performance, time-consuming, and the fact that its test set cannot well reflect the density distribution character of the original dataset. In this paper, some improvements are made to resolve above problems. Experiment results show that the framework can efficiently generate the test sample set with the volume no more than one tenth of the original data set, meanwhile the distribution characteristics of the original dataset can be retained.

Key words: Automatic generation framework, clustering, malware, software gene.

1. Introduction

Malware and antivirus strategies evolve together in the form of mutual games [1], and improving the accuracy of identification of malicious samples has become the main work of research institutions. Different antivirus softwares have different ability to identify and definite malicious programs [2]. Therefore, constructing a high-quality test set can adequately evaluate the performance of antivirus software and help improve antivirus strategies [3], [4]. The test sample set should be characterized by appropriate volume, rich variety, and retention of the original dataset density distribution characteristics. For the test sets that obtained under different generation strategies, the identification accuracy of antivirus software may be different, so it doesn't reflect the performance of antivirus software.

Many industries provide standardized assessments of their work by setting or establishing assessment criteria. In the domain of cyber security, because of the rapidly evolution of malware variants and the usage of encryption shell protection technology, the malicious program sample database soon becomes out of date and invalidation, and then it becomes unsuitable for the study of new malicious programs. The original sample database needs to be continuously updated in order to ensure that the most up-to-date and valid test sample set is generated. Some scholars have obtained malicious sample test set and related data by AV-TEST or AV-Comparatives, to verify the performance of antivirus software [4]. Meanwhile, through the sandbox-based program dynamic characteristic detection technology, some researchers have collected the sample's behavioral characteristics and static analysis-based sample characteristics code, in order to get the test sample from them [5]. However, the rapidly evolution of malware variants and the constantly introduction of new technologies highlights the limitations of these traditional analysis methods, and

meanwhile, methods that rely too much on manual experience lag far behind rapidly updated malicious programs [6].

A software gene is a set of data characteristics in a software code, which is hereditary and unique, and contains people or organizations' characteristic. It carries with functional information, and is the minimum unit of information that is recessive and can fundamentally describe the characteristics of the software. As a new malware analysis solution, software gene technology is distinct from traditional technology based on characteristic detection. To some extent, software gene technology can resolve the problem of low identification accuracy caused by rapidly evolutional malware variants or code confusion [7]. The software gene technology defines and extracts the gene of software, and then identifies and classifies the malicious characteristics of the sample by the machine algorithm [8]. Finally, according to the software gene, new classification method and clustering result can be obtained by analyzing the detected sample. Thus, it is a feasible technical solution that constructing a antivirus software test sample set based on software gene technology.

In this paper, the main work is to propose the automatic generation framework of antivirus software test set based on software gene technology. Because the framework is not limited to the analysis of sample behavior characteristics by human experience to form a test sample set, its performance is stable. Combining software gene with the machine learning technology, the software gene-related mixed attribute clustering of the sample can be achieved through the mixed attribute data clustering algorithm automatically determined by the density-based clustering center, to divide the original data into multiple categories. Meanwhile, the efficiency of massive sample clustering process would also be improved in the non-negative matrix way. At the end, the massive data set is rationally sampled according to the non-uniform-based density deviation sampling algorithm, to retain the density distribution characteristics of the original data set and construct a standard test sample set with the volume no more than one tenth.

2. Related Work

In the construction of sample classification characteristics, Qian *et al.* [6]. analyzed the sample's attributes and characteristics information by observing the dynamic behavior of the sample in the sandbox and recording its detailed log information. However, this method relies on the performance of the sandbox, and the accuracy of the analysis results cannot be controlled autonomously. Although the method proposed by Pete *et al*. [9]. has high accuracy, it needs to run the sample and completely record the activity information, which severely limits the efficiency of the analysis. Zhao *et al.* [10]. obtained the sample operation code and API through analysis of the intermediate language, and then combined with the LSTM algorithm to conduct malicious identification and classification for the sample. With this method, the test sample set is only applicable to the program sample in the Windows platform, so it has no general meaning.

Scholars are committed to develop data sets technology that can express more sample characteristics with less data. Many authoritative research institutions are also actively promoting the progress of related work, such as AV-TEST and AC-Comparatives, both provide related open source projects consisting of a large number of malicious samples for research [11], [14]. Sommer and Paxson's method of constructing datasets is severely limited by the quality and source of the original data, and performs poorly in the process of engineering practice [12]. Rossow believes researchers need to be cautious about the construction and selection of data sets [13].

In addition to the sample analysis method, the sampling method used in the construction of small-scale test sample set is as important as the data source. The genetic algorithm-based roulette sampling algorithm adopted by Liang *et al*. [3]. has a high time complexity problem, and the sandbox-based behavioral characteristics detection scheme would result in inaccurate analysis because the sample did not fully

expose all behaviors. For the characteristics of non-uniform distribution, Chen et al. proposed the mixed attribute data clustering algorithm automatically determined by the density-based clustering center. By analyzing the mixed attributes of the sample set, selecting the corresponding distance measurement according to different attributes, and analyzing the density and distance distribution diagram of each point in the sample set, the clustering center was found [14]. Lvdan et al. proposed an improved density deviation sampling algorithm for non-uniformly distributed sample sets, and formed a new method that can better reflect the distribution characteristics of the original data set. This is a constructive contribution to the design of a reasonable antivirus software test sample set [13].

3. The Proposed Method

The software gene-based antivirus software test set automatic generation framework is divided into the following four steps. First, the original data set is prepared. Then, the software gene is extracted from the sample to be detected according to a particular method. Next, the similarity of each sample's genes is compared, so that samples with higher similarity can be clustered together. Finally, under the premise that the sample distribution density characteristics of the original data set are unchanged, a suitable number of samples are selected from each clustering as a representative to form a final test sample set.

3.1. The Extraction of Software Gene

A software gene is a binary segment of software, which carries functional information and is capable of consistently execution, non-invertible and minimally inseparable. The corresponding assembly instructions can be obtained by disassembling the detected samples. According to the principle of consistently execution, the software gene code segment can be obtained by cutting the assembly instructions. The so-called consistently execution principle means that under normal operation, regardless of any input, the code can be regarded as a whole, executed completely or not at all.

The sample file is disassembled and the corresponding assembly results are obtained. According to the method described in the literature [9], the software gene of the sample was extracted. The extracted software gene is written to a file for storage. The gene extraction process is shown in Fig. 1.

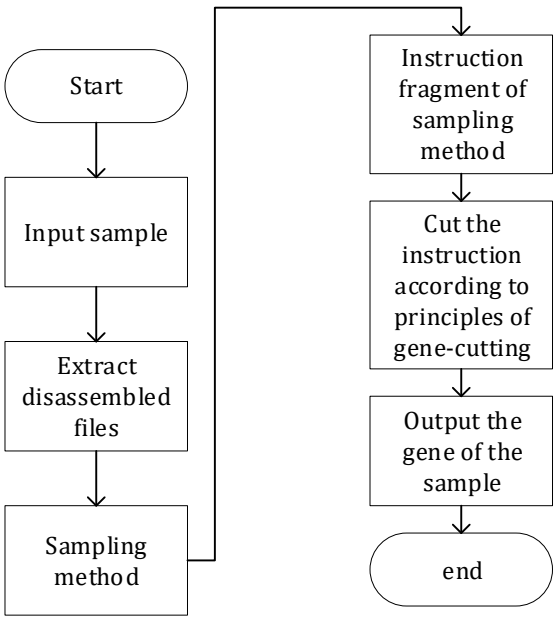

Fig. 1. Software gene extraction flow chart.

3.2. Software Gene Clustering

At present, the more common clustering methods include K-means clustering, hierarchical clustering, density-based clustering, radiation propagation clustering and spectrum clustering and etc. In the course of clustering analysis of sample genes, co-expressed sample clusters can be obtained. Within each cluster, the gene expression of the samples is very similar, indicating that the samples in the cluster have high homology. According to previous research by scholar, it is found that the mixed attribute clustering algorithm has a better performance in the distance measurement method in the clustering of complex sample space [17,19,20]. On the basis of improving the clustering quality and reducing the parameter dependence, the method proposed in the literature can automatically determine the number of clustering categories and find the clustering center [14].

The gene data has two dimensions, the gene amount of each sample and the number of each gene -- that is, the dimension r of the numerical attribute is 1, and the dimension q of the type attribute is 1. According to the test result of the UCI data set, the dominant factor α is set to 0.75, whereby it can be analyzed that the genetic data belongs to the balanced mixed attribute data. Let the sample gene data set be D. For any two samples in D, X_i and X_j, the distance between them is expressed as:

$$D(X_i, X_j) = \sum_{p=1}^{d} d^p(X_i, X_j)$$

After obtaining the distance between any two samples, the value range of distance dc at the optimal stage and the set C of the clustering center can be determined according to the ant colony clustering ACC algorithm. Combining the particle swarm algorithm, and setting the upper and lower limits of iterations and the particle take-off velocity, and finally the exact optimal phase distance dc, the number of clustering K and the set C of clustering center are calculated.

3.3. Generation of Test Sample Set

Density Deviation Sampling Method (DDS) is a new sampling method with perfect sampling results [22]. It determines the sample data according to the distribution characteristics of the row data [21]. The established sample better achieves the consistent distribution characteristics of the sample data and the original data, with high quality and strong anti-noise ability [21], [22]. Based on the density deviation sampling algorithm, the literature proposed an improved algorithm based on variable grid, and achieved better density deviation sampling effect by introducing grid element density and trigonometric function. In this paper, the method described in the literature [23] is adopted to sample the results of clustering in 3.2, to obtain the final test sample set.

After clustering the sample gene data set D, K populations are obtained. Each population is D_i, and the number of samples of each population is the density value of the population. The number of samples taken from each population is expressed as:

$$m_i\omega(m_i) = \frac{m_i^{1-(s-t\cos(\frac{m_i\pi}{2m}))}}{\sum_{i=1}^{K} m_i^{1-(s-t\cos(\frac{m_i\pi}{2m}))}}$$

m_i is the grid density, m is the number of samples, and t and s are constants between [0, 1]. Summarizing the samples from each population sample into a new set constitutes the final test sample set.

4. Experments

To verify the advantages of the proposed test set automatic generation framework, experiments and analysis were carried out based on the data from the VirusShare malware sample databased. The experimental environment was Windows 10 operating system, using Microsoft Visual Studio 2017 integrated development environment, 2.5Ghz Intel Core i5-7200U CPU and 8GB of RAM. VirusShare is a malware sample database that provides malicious code samples for security researchers and incident analysis, and 3,000 of them were selected for experimentation. According to the principle of cross-validation, the samples were divided into 10 groups on average. 9 groups of them were sequentially taken as experimental data, and 2700 samples were tested each time. According to the above clustering method, 10 rounds of sample clustering tests were performed. The number of test sample sets, the average time and the gene coverage rate are as shown in Table 1.

Table 1. Test Set Generation Experiment Information Statistical Table

Number of Experiments	1	2	3	4	5	6	7	8	9	10
Number of Test Sample	283	316	331	326	308	305	311	314	297	295
Average Time(s)	0.58	0.52	0.53	0.49	0.48	0.55	0.51	0.55	0.56	0.52
Gene Coverage Rate(%)	68.2	70.5	65.5	65.6	64.9	66.1	65.4	67.2	65.9	64.3

According to the malware killing report released by AV-Comparatives official website in November 2018, five well-known manufacturers were selected to test the sample set, and the killing rate is as shown in Table 2:

Table 2. Statistical Table of Identification Rate of Test Set by Safety Protection Software

	1	2	3	4	5	6	7	8	9	10
Kaspersky Lab	92.23%	93.99%	94.56%	92.94%	93.18%	93.44%	92.93%	93.95%	93.27%	92.88%
Avast	90.46%	93.04%	93.35%	92.33%	91.88%	92.79%	93.25%	93.63%	92.93%	92.54%
Avira	94.70%	93.99%	96.37%	95.71%	94.48%	94.75%	94.21%	95.22%	95.29%	95.93%
Bitdefender	93.29%	93.04%	91.54%	92.33%	92.86%	93.11%	92.93%	92.36%	93.60%	93.22%
Tencent	85.16%	87.03%	83.38%	82.82%	84.42%	85.57%	84.89%	85.99%	86.87%	86.78%

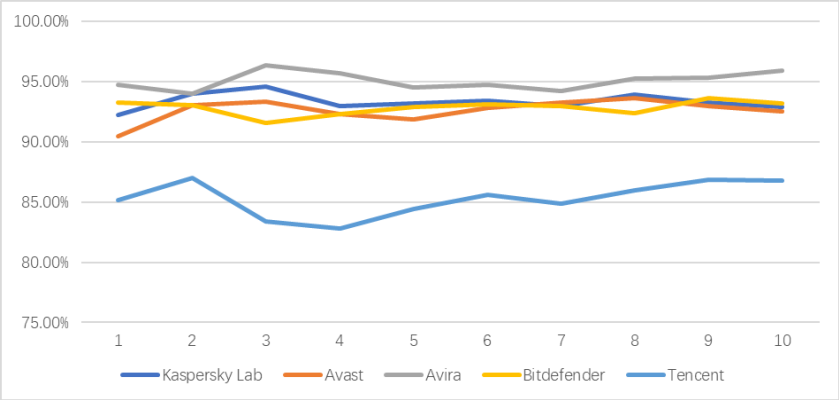

Fig. 2. Identification rate of test set by safety protection software.

After ten experiments, excluding the generation time of software genes, it only took 0.52 seconds on average to generate test sample sets, which was significantly improved compared with the methods of manual analysis and literature [4]. Five well-known international manufacturers have maintained a stable identification rate for ten groups of test sets. The variance of the identification rate in ten rounds of experiments conducted by five manufacturers in Table 2 is calculated as shown in Table 3.

Table 3. The Variance Table of the Identification Rate from 10 Experiments

Security Manufature	Kaspersky Lab	Avast	Avira	Bitdefender	Tencent
Variance of Test Results	0.0064	0.0086	0.0073	0.0056	0.0137

According to the information in Table 3, the performance of the proposed scheme and software-based antivirus software testing and automatic generation framework is stable.

5. Conclusions and Discussion

This paper analyzes the existing antivirus software test set generation method, and innovatively proposes the software gene-based antivirus software test set automatic generation framework. The framework is divided into three components, a software gene extraction module, a mixed attribute clustering module for software genes, and a sampling module. The ability of the framework to generate test sets is verified by experiments, and the generated test set is no more than one tenth of the original data set. The average time taken to process each sample is less than 1 second, which has the advantages of high efficiency and stability.

In this paper, on the basis of analyzing the existing methods of test set generation for antivirus software, an innovative framework of test set automatic generation for antivirus software based on software gene is proposed. The framework consists of three parts: software gene extraction module, software gene-related mixed attribute clustering module and sampling module. Experiments verifies the ability of the framework to generate test sets, and the volume of the test sets generated was not more than one tenth of the original data set. The average processing time of each sample is less than 1 second. The framework has advantages of high efficiency and stability.

Acknowledgment

We are grateful to AV-TEST and VirusTotal for providing us with the malware samples.

References

[1] Sevil, S., Emre, A., Ahmet, I. A., *et al*. (2018). Coevolution of mobile malware and anti-malware. *IEEE Transactions on Information Forensics and Security*, *10(13)*, 2563-2574.

[2] Han, J., Zhao, R. C., Shan, Z., *et al*. (2017). Analyzing and recognizing android malware via semantic-based malware gene. *Proceedings of the 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery.*

[3] Liang, G. H., Pang, J. M., Zheng, S., *et al*. (2018). Automatic benchmark generation framework for malware detection. *Hindawi Security and Communication Networks*.

[4] Jason, U., Xiaobo, Z., *et al.* (2015). Variant A Malware Similarity Testing Framework.

[5] Automated behavioral analysis of malware a case study of WANNACRY ransomware. *Proceedings of the IEEE International Conference on Machine Learning and Applications*.

[6] Chen, Y. H., Zheng, S., Liu, F. D., *et al*. A Gene-inspired malware detection approach.

[7] MalCommunity: A graph-based evaluation model for malware family clustering.

[8] Meng, X., Shan, Z., Liu, F. D., *et al.* (2018). Malware gene sequence vectorization representation and clustering. *Computer Engineering and Design*.

[9] Pete, B., Richard, F., Frederick, T., *et al*. (2018). Malware classification using self organising feature maps and machine activity data. *Computers and Security*, *73*, 399-410.

[10] Bing, L. Z., Jin, H., & Xi, M. (2017). A malware detection system based on intermediate language. *Proceedings of the IEEE International Conference on Systems and Informatics*.

[11] Wolf, G., & Khoshgoftaar, M. T. (2016). Using machine learning for network intrusion detection. *The*

Need for Representative Data.
[12] VirusShare. Retrieved from https://virusshare.com/
[13] AV-Comparatives, Malware Protection Test March.
[14] Comparatives. Retrieved from https://www.av-comparatives.org/wp-content/uploads/2017/04/avc_mpt_201703_en.pdf.
[15] Chen, J. Y., & He, H. H. (2015). Research on density-based clustering algorithm for mixed data with determine cluster centers automatically.
[16] Shi, J., Malik, J., & Normalized, C. (2000). Image segmentation. *IEEE Trans on Pattern Analysis and Machine Intelligence*.
[17] Ding, S. F., Hongjie, J., & Shi, Z. Z. (2014). Spectral clustering algorithm based on adaptive nyström sampling for big data analysis. *Journal of Software*, *25(9)*, 2037-2049.
[18] Wang, Y. B., Ma, J., Song, X. Q. (2018). Semi-supervised spectral clustering algorithm based on optimal projection. *Application Research of Computers*, *35(1)*, 97-100.
[19] Xue, L. X., Sun, W., Wang, R. G., *et al*. (2019). Spectral clustering based on density peak value optimization. *Application Research of Computers*.
[20] Online graph regularized non-negative matrix factorization for large-scale datasets.
[21] Chi, X. B., Jia, X. C., & Han, Q. L. (2012). Sampled data stabilization for takagi-sugeno fuzy systems using membership function deviations. *Proceedings of the IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*.
[22] Fu, P. G., & Hu, X. H. (2016). Biased-sampling of density-based local outlier detection algorithm. *Proceedings of the* 2016 12th International Conference on Natural Computation Fuzy Systems and Knowledge Discovery.
[23] Kittisak, K., Nittaya, K., & Sun, J. P. (2005). Density biased reservoir sampling for clustering. *Artificial Inteligence and Applications*, 95-100.
[24] Lv, D., Long, H., & Gao, J., *et al.* (2018). An improved alorithm for Density deviation sampling based on Uneven Data. *Software Guid*, 77-80.

Liang Bai is a senior engineer in CNCERT. He received the Ph.D. degree from the Department of Automation, Tsinghua University. His research interests include the research and development of network security technology for industrial Internet, Internet of Things and network security technologies for key information infrastructure.

Yu Rao is a senior engineer in Cncert, received the Ph.D. degree from the Department of Electronics, Tsinghua University. Her research interests include on the research and development of network security and data analysis technology.

Xu Liu is an internet security expert, one of the founder of the theory of software gene, external expert of Shanghai Institute of Data Analysis and Processing Technology.

Yiyi Hu has a master degree in computer science from National University of Defense Technology and a degree from Fudan University, chief executive officer of Shanghai Roarpanda Network Science & Technology Co,.Ltd, one of the founder of the theory of software, certified information security professional, certified information systems security professional.

Shiwei Lu is a lead researcher of Shanghai Roarpanda Network Science & Technology Co,.Ltd , one of the Initial researchers about software gene technology, specialize in deep learning and cyberspace security. He has a background in photoelectronic imaging, electronic countermeasures and network security.