MCSMGS: Malware Classification Model Based on Deep Learning

Xi Meng, Zhen Shan, Fudong Liu, Bingling Zhao, Jin Han, Jing Wang
State Key Laboratory of Mathematical Engineering and Advanced Computing
Zhengzhou, China
mengxixi0525@163.com

Hongyan Wang
Division of Information Theory
Aviation University of Air Force
Changchun, China

***Abstract*—As a major threat to cyber security, malware has been increasingly damaging national security. This paper proposes a malware classification model, i.e. MCSMGS model (Malware Classification Based on Static Malware Gene Sequences), that combines the static malware genes with deep learning methods. The model extracts the malware gene sequences that have both material attribute and informational attribute. Then it makes distributed representation for each malware gene to represent the intrinsic correlation and similarity. Finally, the SMGS_CNN (Static Malware Gene Sequences -- Convolution Neural Network) module is used to construct the neural network to analyze the malware gene sequences and realize malware classification. The experimental results show that the classification accuracy is greatly improved and up to 98% with the MCSMGS model. CNN model is more effective than the traditional SVM model.**

Keywords-malware gene sequence; intrinsic correlation; similarity; neural network; classification;

I. INTRODUCTION

With the development of Internet, malware has begun to flourish. On the one hand, the number of malwares grows rapidly. According to the China Internet Security Report, the number of new malwares was about 190 million in 2016 [1]. On the other hand, attackers often use polymorphism and deformation algorithms to generate the advanced variants. On May 12, 2017, WannaCry and its variants swept the globe. The attacks of advanced variants have caused great damage to national politics, economic and social security.

The existing signature-based detection system can detect the known malwares effectively, but it is invalid for unknown malwares and malware variants. Yet the advanced variants of malwares are even more destructive. The existing methods for malware behavior analysis mainly include dynamically tracking the execution trace of malwares and analyzing the system calls. However, dynamically capture the behavior sequences is flawed. The execution of malwares is restricted by execution environment. And when analyzing malwares dynamically, there is only one path of the whole malware structure diagram. Nowadays, deep learning technology has matured and made a major breakthrough. Deep learning is very different from traditional machine learning. The main difference is that deep learning methods extract features automatically by learning from the data and training the neural network rather than design features manually. Designing features manually requires the analysts to have prior knowledge, which often fails to obtain the essential features of data.

II. RELATED WORK

As a traditional analysis method, the signature-based malware detection methods could not deal with unknown malwares and malware variants [2].

Recognizing malwares based on the behavior, Tian et al. proposed a method to monitor the system API calls [3]. The method did not consider the malware behavior sequences. Qiao et al. extracted API calls dynamically combining with data mining techniques [4]. But dynamic execution of malwares to monitor the API calls is vulnerable to the environment constraints and the execution path is single.

With the rapid growth of malwares, the machine learning techniques became popular. M. Shankarapani et al. proposed a method using SVMs to classify malwares [5]. However the performance of SVMs mainly depends on the selection of the kernel function and there is no theoretical approach to select suitable kernel functions. In 2015, Jason et al. proposed a framework that identified malware family variants through similarity testing [6]. And Annachhatre et al. proposed a model based on the Hidden Markov Model (HMM) to identify malwares with K-means algorithm [7]. Similarly, Nataraj et al. proposed a method of classification based on image processing techniques with K-means algorithm [8-9]. The lack of the algorithm is that the selection of the value K must be pre-determined and lacks theoretical verification.

Inspired by biological theory, Rfique et al. used the genetic classification algorithm to develop new classifiers [10]. Later Pfeffer et al. used descriptive methods such as evolutionary analysis to describe malware families [11]. In 2017, Ding et al. proposed MGeT model based on malware genes. The experimental results showed the detection performance of the model is higher than that of HMM [12], N-GRAM [13] and DTW-SVM [14]. The model did not separate from the traditional machine learning methods and the classification accuracy need to be further improved.

In this paper, we propose a malware classification method that combines malware gene sequences extracted statically with deep learning methods. Recognizing malwares in the behavior sequence and using the ability of deep learning to achieve malware analysis and classification.

III. THE MCSMGS MODEL

In this section, we describe the MCSMGS (Malware Classification Based on Static Malware Gene Sequences)

978-1-5386-2209-4/17 $31.00 © 2017 IEEE
DOI 10.1109/CyberC.2017.21

model and detailed methods. The model is shown in Fig. 1. The model consists of the following three parts. The first part is responsible for malware gene extraction. As the input of the model, the Windows executables are disassembled. And the automatic extraction for batch malware gene sequences is achieved with recursive descent method. The second part generates distributed representation for the malware genes and expresses intrinsic correlation with the spatial distance. The third part implements malware classification with constructing suitable convolution neural network.

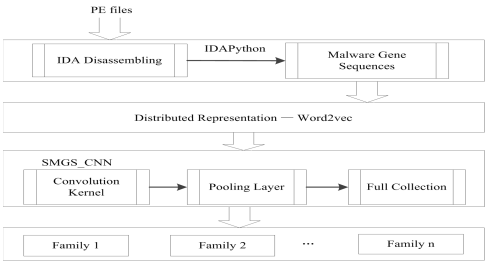

Figure 1. The MCSMGS model

A. Malware gene sequence extraction

Similar to the evolution of creatures, malware continues to evolve. The genetic theory has greatly promoted the development of biology. In this paper, we use the genetic theories of biology to analyze malwares.

Definition: software genes are code fragments of programs that carry functional information.

Since malware is a specific kind of software, we assume that software genes of malwares discussed in this paper are malware genes. Malwares achieves their malicious operations by calling API calls. Therefore, we focus on API call sequences to cognize malwares. In the model, we regard the API call sequences as the malware gene sequences.

First, we preprocess the collected PE files. The model converts them to the assembly codes using IDA Pro (Interactive Disassembler Professional) and generates IDB files. We write script files in IDA Pro using IDA Python to extract the API call sequences with recursive descent method. Thus, we obtain the API calls of the full path. The basic idea of the algorithm is shown in Algorithm 1:

Algorithm for extracting malware gene sequences

INPUT: IDB files of malware
OUTPUT: API call sequences of malware

```
BEGIN
Calculate the entry point address;
Calculating the starting and ending addresses;
WHILE current instruction address is lower than ending address DO
    Create dictionary structure to store the call and jump information;
    IF the current instruction is the call-function THEN
        Store the information of the calling function;
    ELSE IF current instruction is the jump instruction THEN
        Store the information of the jumping block;
    ELSE IF current instruction is the API-call instruction THEN
        Store the address and type of the instruction;
    END IF
END WHILIE
Create the dictionary structure storing invocation relationship ;
FOR each calling function or jumping block DO
    Linear scanning instructions;
    Store call, jump and API- call information ;
END FOR
From the entry point address , traverse call functions and jump blocks
and extract the API sequences with the Depth First Traversal method
END
```

B. Distributed representation of malware genes

In this section, we use word2vec method to make a distributed representation for each malware gene. The basic idea is that each malware gene is trained to a k-dimensional real vector, thus all the malware gene sequences were converted into an n×k two-dimensional matrix, where n is the length of the malware gene sequences. Each vector represents a point in the k-dimensional space, and each element of the vector is determined by repeatedly training and adjusting the weight for the gene sequences. The spatial correlation between the vectors is used to represent the semantic relevance and similarity of the malware genes.

In the paper, we consider API calls as the malware genes and express intrinsic correlation of API calls using word2vec method. API calls are functional, such as openfile(), writefile(), deletefile() etc.. All of these are API calls for file operations, which are semantically similar. However, most existing methods of matching similar API calls are artificial. With word2vec method, all the API calls are expressed effectively and the semantic similarity is expressed automatically.

C. Malware gene sequence extraction

In this section, we propose the SMGS_CNN (Static Malware Gene Sequences--Convolution Neural Network) module to classify malwares by constructing suitable convolution neural networks.

Initially, CNN achieved significant results in dealing with image issues. CNN contains two key components, i.e. convolution kernel and pooling layer. The convolution kernel is shown in Fig. 2, where the 6×6 two-dimensional matrix represents the original image, and the 4×4 square represents the convolution kernel. The kernel parameters are as (1,0,0,0; 1,1,0,1; 1,1,0,1; 1,1,1,0). After the convolution kernel traverses the image, a new feature image witch is represented as a 3×3 two-dimensional matrix is generated. Convolution kernel can effectively extract the feature by training parameters. Pooling layer extracts a new feature image by taking the average or maximum number from the local image.

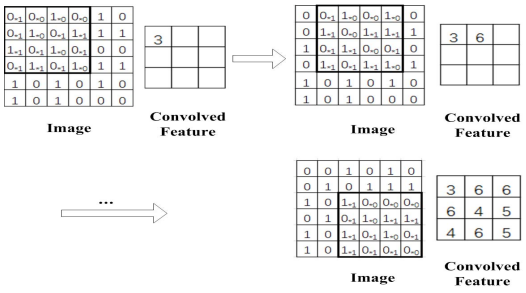

Figure 2. Convolution kernel

The SMGS_CNN module used in this paper is shown in Fig. 3:

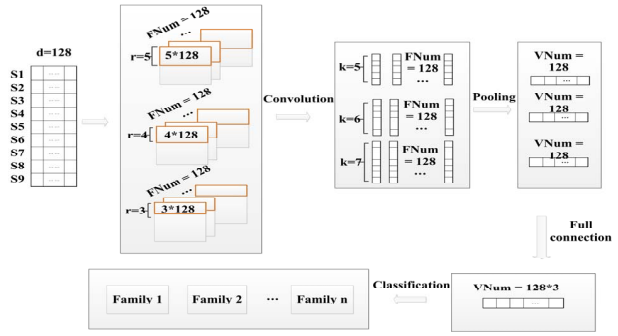

Figure 3. SMGS_CNN module

In the module, each malware gene is represented as a 128-dimensional real vector with word2vec method and each malware gene sequence is represented as an n×128 two-dimensional matrix in which each line represents a malware gene. This allows a malware gene to be treated as a row of pixels in the image. While constructing the convolution kernels, the length of the filter is set to 128 which is the number of the dimension of the malware gene, as the malware gene represented by a row of matrix is inseparable. The width of the filter is (3, 4, 5) and 128 convolution kernels are selected for each kind of the filter. That is, the convolution kernel can extract 128 different feature matrices for each kind of filter. The third layer is the pooling layer where we select the 1-max pool layer. Finally, we construct full connection layer with Softmax method and then get the classification probability.

IV. Experiment And Result Analysis

A. Data-sets and configurations

We choose the data-sets from VX Heavens for our experiment, which is an important malware database for malware research. The data-sets including Trojan and Backdoor consist of 5647 malwares from four families . The detailed information of the data-sets is shown in Table 1:

TABLE I. Data-Sets

Family Name	Sample Number
Backdoor.Win32.Bifrose	1422
Trojan-Downloader.Win32.Small	1074
Trojan.Win32.Obfuscated	1828
Trojan.Win32.Agent	1323

As the gene extraction module is based on IDA PRO, it is required to extract the malware genes under the Windows system, and the rest of the process is done under the Linux system. And neural networks is completed with TensorFlow framework.

The experiment begins with the decompilation of 5647 malwares. The API call sequences are then generated from each IDB file based on the script file in IDA Pro. Each API call is represented as a 128-dimensional real vector. As the lengths of the malware gene sequences are different, we take the longest length of the malware gene sequence as the fixed length n, and all the others fills with the blank characters. Each malware is represented as an n×128 two-dimensional matrix. The model uses three different sizes of convolution kernels. The number of training and testing samples account for 90% and 10% of the total sample number respectively.

In order to reflect the advantages of CNN model in high recognition accuracy, we use the classical traditional classification method SVM as a contrast. The SVM classification method is achieved by calling the LIBSVM toolkit developed by Associate Professor Lin Chih-Jen of Taiwan University in the context of MATLAB R2013a. The malware gene sequences are extracted from 5647 samples and distributed using word2vector. Malwares are finally classified by SVM method. The number of training and testing samples account for 90% and 10% of the total sample number respectively.

B. Experimental results and analysis

The experimental procedure consists of two parts, the training process and validation process. Fig. 4 shows the classification accuracy during the training process. The abscissa of the figure represents the number of iterations and the ordinate indicates the classification accuracy during the training process. Fig. 4 shows that with the increase of iterations, the accuracy of classification is increasing. After the number of iterations gets higher than 3500, the accuracy becomes higher than 98% and tends to be stable. Fig. 5 shows the variation of the loss function of the cross entropy during the training process. The loss function is used to describe the error between the actual output calculated by the model and the standard output. The smaller the loss function is, the batter the model is. The abscissa represents the number of iterations and the ordinate represents the loss function of the MCSMGS model during the training process. Fig. 5 shows that with the increase of iterations, the loss function is decreasing. When the number of the iterations is up to 3000, the loss function is maintained between 0 and 0.1 and tends to be stable, indicating the model converges.

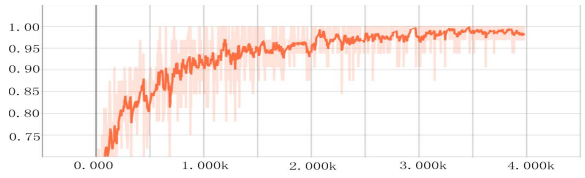

Figure 4. The variation of the accuracy during the training process

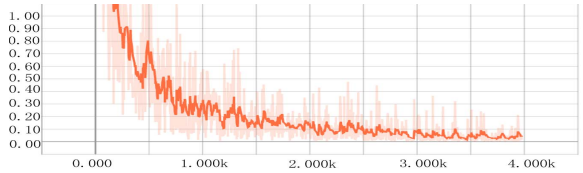

Figure 5. The variation of the the cross entropy loss function during the training process

Fig. 6 and Fig. 7 show the classification accuracy and the loss function of the MCSMGS model during the validation process, respectively. The abscissa of the figure represents the number of iterations and the ordinate indicates the classification accuracy and loss function of the MCSMGS model during the validation process, respectively. The figures shows that with the increase of the iterations, the accuracy during the validation process is increasing and the loss function is decreasing. When the number of iterations gets higher than 3000, the accuracy and loss function tend to be stable, while the classification accuracy is maintained between 98% and 100%, and the loss function is maintained between 0.05 and 0.1. After the loss function gets stable, the MCSMGS model converges and the classification accuracy reaches 98%. The MCSMGS model has the generalization to a certain extent. The experimental results show that the MCSMGS model is superior to the traditional malware classification method for classifying malwares. The classification accuracy of SVM model is 94.7% which is lower than the accuracy of MCSMGS model. The experimental comparison is shown in Table 2:

TABLE II. EXPERIMENTAL COMPARISON

Model	Feature	Word2Vector Dimension	Classification Accuracy
MCSMGS	Malware gene sequences	k= 128	98%
SVM	Malware gene sequences	k = 1	94.7%

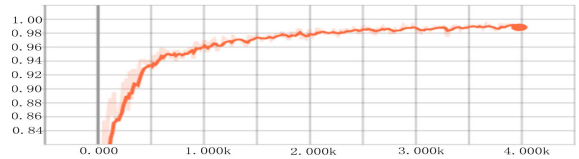

Figure 6. The variation of the accuracy during the validation process

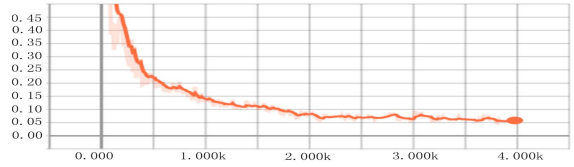

Figure 7. The variation of the the cross entropy loss function during the validation process

V. CONCLUSION

In the paper, we have proposed an efficient malware classification model (MCSMGS). The malware gene sequences have been extracted with the recursive descent method based on the programmable attribute of IDA Pro. The malware genes have been represented to express their intrinsic correlation with word2vec method. The malware classification has been finally completed by training SMGS_CNN module. The MCSMGS model deals with massive malwares based on the behavioral level, and the classification effect is superior to the traditional malware classification methods.

REFERENCES

[1] China Internet Security Report, http://research.360.cn/report/.

[2] Sung A H, Xu J, Chavez P and Mukkamala S. Static analyzer of vicious executables (SAVE)[C]// Computer Security Applications Conference, 2004. IEEE, 2004:326-334.

[3] Tian R, Islam R, Batten L and Versteeg S. Differentiating malware from cleanware using behavioural analysis[C]// International Conference on Malicious and Unwanted Software. IEEE, 2010:23-30.

[4] Qiao Y, He J, Yang Y and Ji L. Analyzing Malware by Abstracting the Frequent Itemsets in API Call Sequences[J]. 2013, 8137(1):265-270.

[5] Shankarapani M, Kancherla K, Ramammoorthy S and Mowa R. Kernel machines for malware classification and similarity analysis[C]// International Joint Conference on Neural Networks. IEEE, 2010:1-6.

[6] Upchurch J, Zhou X. Variant: a malware similarity testing framework[C]// International Conference on Malicious and Unwanted Software. IEEE Computer Society, 2015:31-39.

[7] Annachhatre C, Austin T H, Stamp M. Hidden Markov models for malware classification[J]. Journal of Computer Virology and Hacking Techniques, 2015, 11(2):59-73.

[8] Nataraj L, Karthikeyan S, Jacob G and Manjunath B S. Malware images:visualization and automatic classification[C]// International Symposium on Visualization for Cyber Security. ACM, 2011:1-7.

[9] Nataraj L, Yegneswaran V, Porras P and Zhang J. A comparative assessment of malware classification using binary texture analysis and dynamic analysis[C]// ACM Workshop on Security and Artificial Intelligence. ACM, 2011:21-30.

[10] Rafique M Z, Chen P, Huygens C and Joosen W. Evolutionary algorithms for classification of malware families through different network behaviors[C]// Conference on Genetic and Evolutionary Computation. ACM, 2014:1167-1174.

[11] Kirat D, Vigna G. MalGene: Automatic Extraction of Malware Analysis Evasion Signature[C]// ACM Sigsac Conference on Computer and Communications Security. ACM, 2015:769-780.

[12] Imran M, Afzal M T, Qadir M A. Using hidden markov model for dynamic malware analysis: First impressions.[M]// International Conference on Security and Privacy in Communication Networks. Springer International Publishing, 2014.

[13] Rieck K, Trinius P, Willems C and Holz_Affn T. Automatic analysis of malware behavior using machine learning[J]. Journal of Computer Security, 2011, 19(4):639-668.

[14] Jianwei D, Zhaoguo C, Yue Z, Hong S, Yubin G and Enbo S. MGeT: Malware gene-based malware dynamic analyses[C]// ICCSP '17 Proceedings of the 2017 International Conference on Cryptography, Security and Privacy, 2017:96-101.