Design of Malicious Code Detection System Based on Convolutional Neural Network

Yumeng Wu[1(✉)], Jianjun Zeng[2,3], Zhenjiang Zhang[1], Wei Li[4,5], Zhiyuan Zhang[1], and Yang Zhang[1]

[1] School of Electronic and Information Engineering, Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, China
{21120146,zhangzhenjiang,zhangzhiyuan,zhangyang1}@bjtu.edu.cn
[2] College of Intelligence and Computing, Tianjin University, Tianjin, China
jj@inchtek.ai
[3] Beijing InchTek Technology Co., Ltd., Beijing 100012, China
[4] The Classified Information Carrier Safety Management Engineering Technology Research Center of Beijing, Beijing, China
[5] Beijing Jinghang Computation and Communication Research Institute, Beijing, China

Abstract. With the rapid development of Internet of things, cloud computing, edge computing and other technologies, malicious code attacks users and even enterprises more and more frequently with the help of software and system security vulnerabilities, which poses a serious threat to network security. The traditional static or dynamic malicious code detection technology is difficult to solve the problem of high-speed iteration and camouflage of malicious code. The detection method based on machine learning algorithm and data mining idea depends on manual feature extraction, and can not automatically and effectively extract the deeper features of malicious code. In view of the traditional malicious code detection methods and the related technologies of deep learning, this paper integrates deep learning into the dynamic malicious code detection system, and proposes a malicious code detection system based on convolutional neural network.

Keywords: convolutional neural network · malicious code detection · network security · deep learning

1 Introduction

In recent years, malicious code and network attacks have become more frequent, and new threats have emerged. The increasingly serious information security problem not only makes enterprises and users suffer huge economic losses, but also makes the national security is facing a serious threat. Viruses multiply and iterate quickly [1]. It can easily escape traditional detection methods by changing their signature concealment behavior. In order to keep up with the increasingly frequent and rapidly evolving pace of malicious code changes and improve the speed of emergency response to malicious attacks, it is

D.-J. Deng et al. (Eds.): SGIoT 2022, LNICST 497, pp. 113–124, 2023.
https://doi.org/10.1007/978-3-031-31275-5_12

necessary to timely analyze the attack methods and characteristics of malicious code quickly and accurately.

The traditional malicious code detection model needs to be trained by manually extracting features. The number of malicious code and the content of feature extraction greatly affect the detection effect of the model. Traditional malicious code detection methods include analysis methods based on dynamic behaviour [2, 3] and static signature [4]. The working principle is to obtain the relevant feature information through static scanning or dynamic analysis, and then compare it with the existing feature library. The feature library is limited and needs to be updated in time. The existing feature library is difficult to deal with the current surge in malicious code detection. In addition, this method of feature comparison will occupy a lot of running memory and low detection efficiency. Therefore, the traditional malicious code detection technology has been unable to effectively resist the new threats and attacks on the computer system and the Internet [5].

In recent years, machine learning has developed rapidly, especially in the fields of computer vision [6] and natural language processing [7]. Data mining is a process of finding anomalies, patterns and correlations in large data sets to predict results. Therefore, it is a popular application field to mine the potential value in the field of big data and discover the relationship between data by using the set of data mining and machine learning [8, 9]. However, the detection method based on machine learning algorithm and data mining idea can not automatically and effectively extract the deeper features of malicious code, which depends on manual extraction. These shallow features can not fully and accurately describe malicious code, and feature extraction largely determines the results of malicious code detection, resulting in the low accuracy of malicious code detection.

Aiming at the shortcomings of machine learning based detection technology, the deep learning model can automatically extract deeper features of malicious code, which can more accurately describe malicious code [10]. Compared with the artificial dependence of machine learning detection, the deep learning model has self-learning ability, and can learn the characteristic differences between malicious code samples and normal samples, so as to better complete the task of malicious code detection.

Combined with the traditional dynamic analysis method of malicious code detection and CNN model, this paper proposes a malicious code detection system model based on convolutional neural network. The system uses the sandbox to extract the malicious code API call sequence, and then takes its one-hot vectorization as the characteristic input of the malicious code detection system. The neural network parameters are adjusted by the optimization algorithm. Finally, the usability and superiority of the system are verified by experimental analysis.

2 Related Work

Many scholars have begun to study malicious code detection technology. Tahan et al. [11] proposed a new automatic signature generation method, which is based on deep packet inspection and runs in real time. This method can be used for large-scale malware detection by ignoring the signatures in benign executable files. Signature based detection

is the most common method for commercial anti malware, but it can not identify new and unknown malware, so it is necessary to constantly update the malicious code signature database. With the rapid growth of the number and types of malicious code in recent years, this method relies on manual extraction, and the performance has become a big problem.

The behavior based malicious code detection method mainly collects the malicious behavior of malware instances and detects according to the collected behavior information. Firstly, it is necessary to perform dynamic analysis on relatively new malware data sets in a controlled virtual environment, and capture the API call information executed by malware instances. Firdausi et al. [12] extract the behavior information of malware in sandbox environment and generate behavior reports. These reports are preprocessed into sparse vector model, and then trained for classification through machine learning classification model. Burguera et al. [13] proposed a behavior based malware detection system crowdroid for Android, which dynamically analyzes the behavior of malicious code, and takes the behavior information extracted from the dynamic analysis as the feature of detecting malicious code on Android platform. Behavior based malicious code detection technology has the risk of being attacked by malicious code and occupies more resources.

In order to solve the shortcomings of traditional malicious code detection technology, research experts focus on the field of machine learning. Santos et al. [14] thought that the practical difficulty was that the detection method based on machine learning needed a large amount of labeled data, so they proposed llgc semi supervised learning to expand the training samples. The detection methods based on data mining [15] and traditional machine learning usually extract the features of malicious code and use the classifier algorithm of machine learning for detection and classification. The traditional classification model of machine learning can not effectively and automatically extract deep-seated features. It depends on manual feature extraction, and the experimental accuracy is low.

The above malicious code detection methods have some disadvantages. In recent years, the emerging deep learning methods can still show good detection effect in the case of insufficient feature extraction. Researchers are trying to apply the deep learning algorithm to the field of malicious code detection, which has research significance and development prospects at present. The advantages of deep learning, such as many feature dimensions, feature self-learning and large number of samples, mean that deep learning can play a great role in the field of malicious code detection. For example, feedforward neural network is used to analyze malicious code [16], and cyclic neural network is used to model system call sequence to build the language model of malicious code [17]. Most studies focus on the improvement of malicious code detection algorithm, and do not build a specific system model of malicious code detection. This paper draws lessons from the analysis methods of natural language processing, pays attention to the dynamic characteristics of malicious code, proposes a malicious code detection system model based on convolutional neural network, and makes an in-depth study on the feature extraction of malicious code, the feature vectorization representation of malicious code, and the detection and classification model of malicious code.

3 System Model

This section introduces the overall architecture of malicious code detection system based on convolutional neural network and the key technologies of feature extraction and neural network construction.

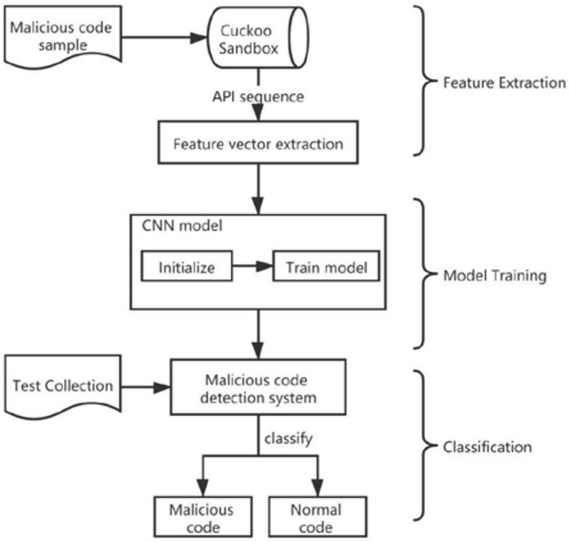

Fig. 1. Malicious code detection model framework

3.1 Overall Architecture

The model of malicious code detection system based on convolutional neural network is mainly divided into three stages: feature extraction, model training and classification detection. In the feature extraction stage, the API call log of malicious code is obtained through the open source automatic malware analysis tool cuckoo sandbox, and the API features are vectorized to obtain the feature vector. In the second stage, the convolutional neural network model is constructed and trained to the best state through the training set. In the third stage, the malicious code detection system detects unknown code and obtains malicious or benign classification results. Figure 1 is the overall framework of malicious code detection model based on CNN.

3.2 API Based Dynamic Behavior Capture

Sandbox. Sandbox is a lightweight virtual machine that can intercept system calls and restrict program execution in violation of security policies. Its core is to establish an execution environment with limited behavior. We put the sample program into the environment and run it. The path of file operation and registry operation in the sandbox will be redirected to the specified location of the sandbox. Some dangerous behaviors of the program, such as underlying disk operation and installation driver, will be prohibited by the sandbox, which ensures that the system environment will not be affected, The system state is rolled back after the operation. Thanks to the modular design and powerful scripting function of cuckoo, it can be used as an independent application or integrated into a larger framework. Cuckoo can be used to analyze windows executable files, DLL files, office files, URL and HTML files, VB scripts and other types of files.

In this paper, the open source automatic malware analysis sandbox cuckoo is built to automatically analyze and collect the behavior of samples in the isolated windows operating system. Cuckoo sandbox environment is mainly used for dynamic analysis of malicious code. It can execute and monitor malicious files in real time. The standard process of dynamic analysis of malicious code is to run PE files in an independent, transparent and secure analysis environment and monitor the dynamic behavior of samples. Through a variety of virtual sandboxes and the establishment of simulation technology to simulate the file running environment.

Cuckoo Sandbox mainly includes host machine (central management module) and guest machine (guest virtual machine module), which communicate with each other by establishing a virtual network. Host machine includes cuckoo sandbox software, virtual box software and various analysis components to manage the startup analysis, behavior monitoring and report generation analysis process of sandbox. Guest machine is an isolated environment where malicious code samples can be safely executed and analyzed, and finally report the analysis results to the central management module. Figure 2 shows the structure of Cuckoo Sandbox.

API Sequence Acquisition. First, put the code file sample into the cuckoo sandbox to get the analysis report in JSON format. The original JSON format API information includes: API type, API name, API parameter value, API return value, etc. Match it with the fields (category and API) to be extracted by the python script. If the matching is successful, the values corresponding to category and API in the API information are extracted and saved in the TXT format document as the original data of the feature vector.

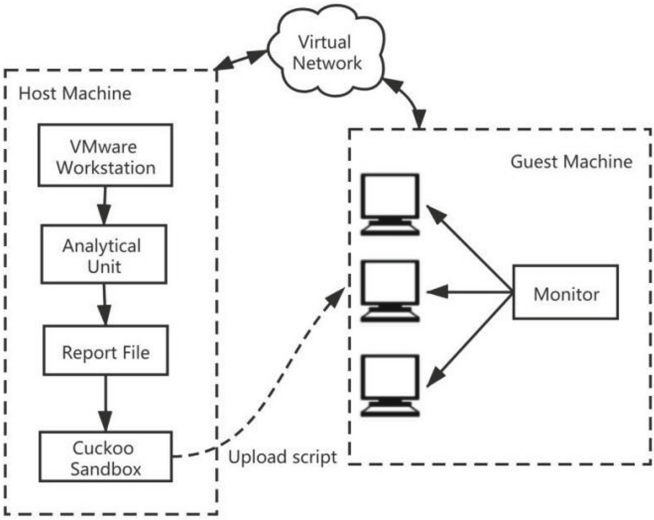

Fig. 2. Cuckoo Sandbox structure

3.3 Construction of Convolutional Neural Network

Convolution neural network is a deep feedforward neural network with the characteristics of local connection and weight sharing. The convolution layer uses convolution filter to extract features from data samples. In the field of image processing, convolution filter is mainly used to identify features from images. Similar to image, convolution filter is used to extract information and detect high-level features of short text in text processing. Because the logs containing malicious executable program instructions are composed of sequences, there are obvious similarities with natural language processing when selecting modeling methods. At present, convolutional neural networks are generally composed of convolution layer, convergence layer and full connection layer. Referring to the convolutional neural network structure, this paper uses the convolutional neural network to detect the malicious behavior of samples. Convolutional neural network architecture is shown in Fig. 3.

One-Hot Feature Vectorization. The convolution neural network model takes the word vector as the input of the input layer. The purpose of feature vectorization is to generate feature vectors for algorithm processing by using the API call information sequence obtained by sandbox. The whole process fully excavates sensitive information and maintains the behavior characteristics of malicious code. In the process of generating eigenvectors, one-hot model is selected.

One-hot coding is one of the methods of text vectorization, as shown in Fig. 4. One-hot encoding uses an n-bit status register to encode N states, and only one bit is valid. It associates the unique integer index i with each word, and then converts the index i into a binary vector with length n (n is the dictionary size, corresponding to the above n-bit status register). The characteristic of this vector is that only the ith element is 1 and the other elements are 0.

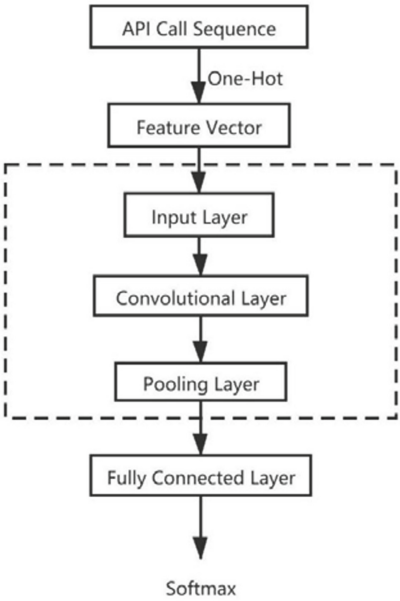

Fig. 3. Convolutional neural network architecture

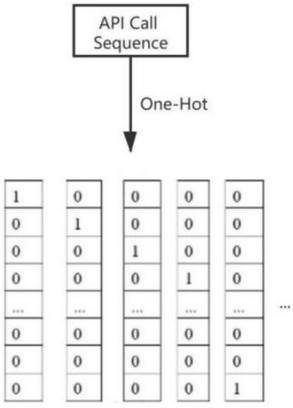

Fig. 4. One-hot Feature Vectorization

Convolution Layer Feature Extraction. Convolution layer mainly includes local perception and weight sharing. Different from the ordinary neural network, which designs the input layer and hidden layer as a fully connected form, the convolution layer of convolution neural network designs each hidden unit to connect only a part of the input unit.,as shown in Fig. 5. This local sensing structure can sense small areas, so as to reduce the number of parameters.

On the one hand, weight sharing enables the repeating unit to recognize the feature without considering the position of the feature in the visual domain. On the other hand,

weight sharing can effectively extract the feature and reduce the number of free variables to be learned.

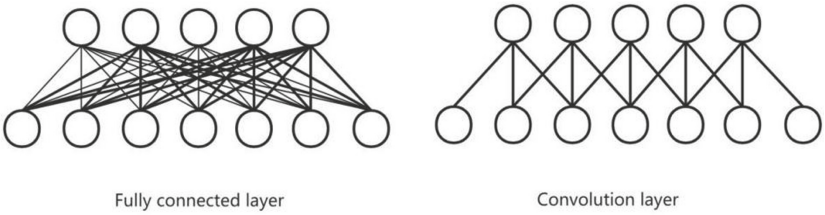

Fig. 5. Fully connected and convolution layer

The way of local perception and weight sharing has the disadvantage of insufficient feature extraction. Therefore, multiple convolution filters with different weights can extract different features for the input and convert the text information into two-dimensional image data. Through the convolution filter and the feature RE Extraction of the pooling layer, the most significant features are sent to the softmax classifier.

Max Pooling Characteristic Downsampling. The pooling layer takes the results of the local features extracted by the convolution layer as the input, and then extracts the most significant features, so as to reduce the dimension of the feature matrix and solve the problem of model over fitting. At the same time, the introduction of pooling also ensures the deformation invariance of the feature matrix. The pool layer mainly includes Max pooling and Average Pooling. Max Pooling calculates the maximum value in the image pooling window as the sampling value to represent the characteristics of the region; The Average Pooling layer calculates the weighted average value in the image pooling window as the sampling value to represent the characteristics of the region. The model in this paper adopts Max pooling (as shown in Fig. 6), which can reduce the output dimension while maintaining the important global information captured by the filter.

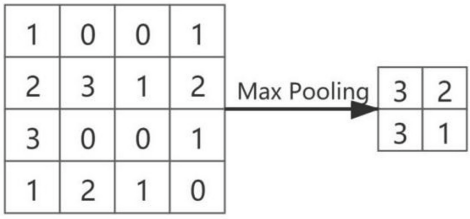

Fig. 6. Max Pooling sample (window2*2)

Model Optimization Algorithm. In this paper, SGD random gradient descent algorithm is selected as the optimization algorithm. After calculating the loss, the optimizer optimizes the constructed network model. In a complex neural network model, the optimizer optimization process changes the parameters of each layer of the network. In each

iteration, the parameter value stabilizes to the optimal value in the specified direction, and finally makes the loss (the proximity between the classified predicted value and the actual value) tend to be the minimum.

SGD random gradient descent algorithm can randomly use one sample for parameter optimization in each iteration and one sample for gradient descent in each update. Because the samples are random, the accurate gradient cannot be obtained. θTherefore, the loss function obtained in each iteration is generally close to the direction of the global optimal solution. Stochastic descent gradient algorithm SGD is a common optimization algorithm in general neural network models. The algorithm formula is as shown in Table 1.

Table 1. SGD algorithm

SGD algorithm

Loop{
For i in range(m):{
$\theta_j = \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x^{(i)}_j$
}
}

4 Simulation and Analysis

4.1 Experimental Environment

This paper selects the public website to download the windows malicious PE file set, a total of 2000 samples; In addition, Download 1000 samples of the normal sample set from Baidu app store, and finally a total of 3000 samples.

The environment configuration of cuckoo sandbox is shown in the Table 2.

Table 2. Configuration environment

Environment	Configuration
CPU	Intel(R)Core(TM)i7-8550U
Memory	8G
Operating System	Ubuntu 16.04
Software environment	Python2.7, Cuckoo2.0.4

Table 3. Confusion matrix

		Predicted class	
		y^ = c	y^ ≠ c
Real class	y = c	TP_c	FN_c
	y ≠ c	FP_c	TN_c

4.2 Evaluation Index

The confusion matrix is defined to represent the relationship between classification results and actual results, as shown in Table 3.

Define the accuracy rate, recall rate and F value according to the contents in the table. The accuracy rate of category c is the proportion of correct prediction in all samples predicted as category c.

$$\alpha_c = \frac{TP_c}{TP_c + FP_c} \tag{1}$$

The recall rate of category c is the correct proportion predicted in all samples with real label of category c.

$$R_c = \frac{TP_c}{TP_c + FN_c} \tag{2}$$

F measure is a comprehensive index, which is the harmonic average of accuracy rate and recall rate (generally, the value of β is 1):

$$F_c = \frac{\left(1 + \beta^2\right) \times \alpha_c \times R_c}{\beta^2 \times \alpha_c + R_c} \tag{3}$$

4.3 Experimental Analysis

The data set is randomly divided into two partitions of the same size as the training set and the test set. The experiment was repeated three times, leaving a different partition for testing each time. Finally, a reliable measurement method is obtained to measure the performance of the proposed convolutional neural network model on the whole data set. CNN is compared with NaiveBayes, MLP and SVM to detect the performance of the model.

The performance of the convolutional neural network model was quantitatively evaluated using three indicators: accuracy, recall and F. The experimental results are shown in Table 4. The overall results of CNN model in accuracy, recall and F ($\beta = 1$) are higher than those of other common machine learning algorithms, so CNN has more advantages than other machine learning algorithms.

Table 4. Comparison of model performance

Model	Accuracy	Recall	F1
CNN	0.93	0.91	0.92
NaiveBayes	0.83	0.72	0.77
MLP	0.91	0.89	0.90
SVM	0.89	0.84	0.86

5 Conclusion

Aiming at the shortcomings of traditional malicious code detection methods and machine learning methods, this paper introduces convolutional neural network into the traditional malicious code dynamic behavior detection system, and proposes a malicious code detection system based on convolutional neural network. The usability of the system and its superiority over other machine learning models are proved by experiments. In the next step, the feature extraction method and depth learning model parameters will be optimized to achieve better training effect.

Acknowledgement. This research is supported by the National Natural Science Foundation of China (NO. 62173026).

References

1. Zhang, Y., Li, B.: Malicious code detection based on code semantic features. IEEE Access **8**, 176728–176737 (2020)
2. Kang, B.B.H., Srivastava, A.: Dynamic malware analysis. In: van Tilborg, H.C.A., Jajodia, S. (eds.) Encyclopedia of Cryptography and Security, pp. 367–368. Springer, Boston (2011). https://doi.org/10.1007/978-1-4419-5906-5_846
3. Zhang, B., Qianmua, L., Ma, Y.: Research on dynamic heuristic scanning technique and the application of the malicious code detection model. Inf. Process. Lett. **117**(jan.), 19–24 (2017)
4. Sung, A.H., Xu, J., Chavez, P., et al.: Static analyzer of vicious executables (SAVE). In: 2004 20th Annual Computer Security Applications Conference. IEEE Computer Society (2005)
5. Han, L., Qian, M., Xu, X., et al.: Malicious code detection model based on behavior association. Tsinghua Sci. Technol. **19**(5), 508–515 (2014)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
7. Collobert, R., Weston, J., Bottou, L., et al.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**(8), 2493–2537 (2011)
8. Chayal, N.M., Patel, N.P.: Review of machine learning and data mining methods to predict different cyberattacks. In: Kotecha, K., Piuri, V., Shah, H., Patel, R. (eds.) Data Science and Intelligent Applications. Lecture Notes on Data Engineering and Communications Technologies, vol. 52, pp. 43–51. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-4474-3_5

9. Yang, H., Li, S., Wu, X., et al.: A novel solutions for malicious code detection and family clustering based on machine learning. IEEE Access **7**, 1 (2019)
10. Cui, Z., Xue, F., Cai, X., et al.: Detection of malicious code variants based on deep learning. IEEE Trans. Industr. Inform. 1 (2018)
11. Tahan, G., Glezer, C., Elovici, Y., Rokach, L.: Auto-Sign: an automatic signature generator for high-speed malware filtering devices. J. Comput. Virol. **6**(2), 91–103 (2010)
12. Firdausi, I., Erwin, A., Nugroho, A.S.: Analysis of machine learning techniques used in behavior-based malware detection. In: 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies, pp. 201–203 (2010)
13. Burguera, I., Zurutuza, U., Nadjm-Tehrani, S.: Crowdroid: behavior-Based malware detection system for android. In: Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices, pp. 15–26 (2011)
14. Santos, I., Laorden, C., Bringas, P.G.: Collective classification for unknown malware detection. In: Proceedings of the International Conference on Security and Cryptography, pp. 251-256 (2011)
15. Yang, Y., Yang, Z., Liu, X.: The algorithm of malicious code detection based on data mining. In: Green Energy and Sustainable Development I: Proceedings of the International Conference on Green Energy and Sustainable Development (GESD 2017) (2017)
16. Saxe, J., Berlin, K.: Deep neural network based malware detection using two dimensional binary program features. arXiv preprint arXiv:1508.03096 (2015)
17. Pascanu, R., Stokes, J.W., Sanossian, H., Marinescu, M., Thomas, A.: Malware classification with recurrent networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2015)