Detection technology of malicious code family based on BiLSTM-CNN

Guodong Wang, Tianliang Lu* and Haoran Yin

School of Police Information Engineering and Cyber Security People's Public Security University of China, Beijing, China

*Corresponding author e-mail: lutianliang@ppsuc.edu.cn

Abstract. The explosive growth of the number of malicious code makes it one of the important threats to network security. Among them, a new type of malicious code family accounts for a small part, and most of them are generated by mutation on the basis of the original family. Based on batch processing of newly added malicious code, this paper proposes a malicious code family detection technology combining malicious code visualization and deep learning. The malicious code executable file is directly converted into a grayscale image, and then the BiLSTM-CNN deep learning algorithm is used to detect the malicious code family. Experiments prove that the model has higher accuracy.

1. Introduction

The 2019 Symantec Global Intelligence Network data shows that an average of 142 million malicious code cyber attacks per day in more than 157 countries and regions around the world [1], which means that the Internet, which is closely related to our lives, is suffering serious security threats. Malicious code refers to the code that is artificially written or set and will cause harm to the network or system. Its types include computer viruses, worms, Trojans, botnets, ransomware, etc., which illegally invade the target network to steal information and data, damage the systems. In recent years, the number of malicious codes has exploded, in order to enhance the concealment of the malicious code, the malicious code mutates to generate malicious code variants with similar attack behaviors, but essentially belongs to the original malicious code family. By detecting the malicious code family, new malicious codes can be processed and prevented. Malicious detection technology methods can generally be divided into static detection technology, dynamic detection technology and hybrid detection technology.

Static detection refers not to run the malicious code family executable file directly, but to analyze the sample source code or the corresponding disassembly syntax and structure to extract the characteristics of the malicious code family to perform classification detection [2, 3]. Jiawei Su [4] et al. Proposed a novel lightweight method to detect DDos malware in IoT environments, extract malware images and use light convolutional neural networks to classify their families., the experimental classification accuracy of the two main malwares reached 81.8%. Yuntao Zhao [5] et al. Proposed a deep learning malicious code classification method based on texture visualization, which uses code mapping to divide and extract textures, and uses machine learning to classify malicious codes.

Dynamic detection requires that malicious code executable files are actually run in a simulated environment, and the malicious code family characteristics are extracted and classified by identifying their malicious operations and behaviors [6, 7]. Aziz Mohaisen [8] et al. Proposed a system for detecting
malicious operation sequences of malicious code, using mapping relationships to achieve dynamic tracking, and applying n-gram document classification technology to classify malware families. Mitsuhiro Hatada [9] et al. Used malicious behaviors to determine new types of malicious code, including malicious code specific functions and regular traffic functions, and applied cluster analysis to generate classifiers. Experiments have proved that the method has a higher accuracy rate in discovering new varieties of malware.

Hybrid detection methods need to combine static detection technology and dynamic detection technology, use dynamic detection to obtain the characteristics of malicious code, and then use static detection to detect and classify, or use static methods to extract the genetic characteristics of malicious code, and then use dynamic detection to identify unknown malicious operations [10, 11]. Di Xue [12] et al. Proposed a classification method based on probability scoring and machine learning, setting probability thresholds to connect static analysis and dynamic analysis, speeding up the static analysis process, and using dynamic analysis seems to have greater adaptability to fuzzy processing.

Faced with a large amount of malicious code, it is impractical to dynamically detect all malicious code, and more rapid and effective algorithms are needed to implement it. The malicious code visualization technology can quickly transform malicious code into pictures, and fully display the family characteristics of malicious code without harming the computer itself. Neural networks in deep learning are good at mining the deep features of pictures and can better classify malicious code families [13, 14]. This article uses B2M technology to visualize malicious code, directly convert the executable file of the malicious code into a grayscale image, and then combines the BiLSTM and CNN to learn the global and local features of the malicious code in the grayscale image. Finally, the softmax classifier is used to detect and classify the malicious code family.

2. Methods

2.1. System Description

The detection and classification of the malicious code family needs to extract the features of the malicious code family, and use the classification algorithm to learn the extracted features to achieve classification. This paper aiming at the traditional malicious code feature extraction technology cannot effectively and comprehensively extract malicious code features, traditional machine learning algorithms cannot focus on global features and local features at the same time, and the classification accuracy of malicious code families is too low et al.. Combined with malicious code visualization technology, a model combining long-term and short-term memory networks and convolutional neural networks is designed and implemented. The BiLSTM-CNN malicious code family detection model architecture is shown in Figure 1.

![Figure 1. Malicious code family classification model framework](image)

This model contains three parts: visualization of malicious code, BiLSTM and CNN. First, the malicious code visualization technology is used to transform the malicious code into a second-order grayscale image. The grayscale image input model traverses the global features through a BiLSTM, and
then inputs it into the CNN to extract the local features. And local feature input classifier to get classification results.

2.2. Malicious code visualization

Feature extraction of malicious code first requires visual processing of the malicious code. There are many characters in the source code of the malicious code sample that are not associated with the characteristics of the malicious code. If the malicious code features are directly extracted from the source code, the sample's obfuscated code needs to be processed, which increases the workload and may reduce the accuracy of detection. This article directly converts executable files into grayscale images, and treats binary executable files of malicious code as 8-bit unsigned integer sequences between 0-255 (including 0 and 255). Each value is interpreted as the intensity of the pixel, with 0 being black and 255 being white. The B2M algorithm was used to read the malicious code executable file, fixed 256 as the row width vector, and converted the executable file into a two-dimensional matrix. The two-dimensional matrix is represented as a grayscale image. The range of each element in the two-dimensional matrix is 0-255, corresponding to the value of each pixel of the grayscale image, and the malicious code is converted into a second-level grayscale image.

2.3. Bi-directional LSTM

The model uses a BiLSTM to extract global features, and trains forward and reverse time series at the same time. The output data contains contextual information. Recurrent neural network (RNN) neurons share parameters so that they are memorable and have certain advantages when learning the non-linear features of a sequence, but traditional RNNs can only have short-term memory, which deepens the neural network (in time or in space) will cause the gradient to disappear and the gradient to explode. LSTM uses gate control to combine short-term memory with long-term memory, and uses gradient descent to make the error gradient disappear as the time of the event increases. LSTM networks can only be trained and delivered unidirectionally, but in actual applications need to consider the front and back inputs to improve accuracy. The malicious code family detection model in this paper uses a two-way LSTM algorithm, the two-way LSTM in the model has two LSTMs connected to each other on the input sequence. The malicious code characteristics of each input will pass through the recurrent God network in the forward and reverse directions, which is a neural network, provide contextual global features to learn faster and more fully. BiLSTM is a neural network proposed to solve the lack of contextual connection of LSTM networks.

2.4. Convolutional Neural Networks

The convolutional layer of the convolutional neural network in this model is used as a feature extractor to divide the feature matrix of the malicious code family of the data set into several sub-matrices. Convolution operation of all Eigen submatrices in each convolution layer with the same weight matrix (convolution kernel), the convolution operation is used to train the data to extract the local features of the grayscale image. The convolution operation can extract the local abstract features of the malicious code family that cannot be understood by humans in the data set. After the convolutional layer, the pooling layer screens the features obtained from the convolution, reduces the number of features to reduce the amount of calculation, and at the same time can play a role in retaining the characteristics of the malicious code family and preventing overfitting.

The model in this paper uses the sigmoid function as the activation function of each convolution layer. The form of the sigmoid function (formula (1)) and its derivative function (formula (2)) is shown below. The Sigmoid function is smooth and sigmoidal easy to derive. It is very similar to the integral form of the normal distribution function. In the case of unknown malicious code family types, it is the most likely form of all probability distributions, so the sigmoid function is often used choose as activation function.

\[
S(x) = \frac{1}{1 + e^{-x}}
\]  
(1)
The model's convolutional neural network has two fully connected layers that are used to fuse the features extracted by the bidirectional recurrent neural network and the convolutional neural network. The malicious code family classification model in this paper belongs to a multi-classification model, and the features of each malicious code family are mutually exclusive, so the output value of the last fully connected layer is passed to the softmax classifier for classification.

This model uses the softmax classifier to output the standard model, and the form of the function is formula (3). Where \( \mathbf{\theta} \) and \( \mathbf{x} \) are column vectors, and \( \mathbf{\theta}^T \mathbf{x} \) may be replaced by a function \( f_i(x) \) about \( x \). The softmax function can make the range of \( P(i) \) between \([0,1]\). In classification and regression problems, usually \( \mathbf{\theta} \) is the parameter to be sought, and by finding \( \mathbf{\theta} \) that maximizes \( P(i) \) as the best parameter. Equation (4) represents the cross-entropy loss function, and \( \log \) represents the base-10 logarithm. \( q_i \) is the confidence of the i-type malicious code family of the prediction model, \( q_k = 1 \) indicates which family the malicious code sample belongs to, and if the sample is \( k \), \( q_k = 1 \) and other values are zero.

\[
P(i) = \frac{\exp(\mathbf{\theta}^T \mathbf{x})}{\sum_i \exp(\mathbf{\theta}^T \mathbf{x})}
\]

\[
F = -\sum_i q_i \log a
\]

3. Experiments and analysis

3.1. Collect samples
The experimental data set of the malicious code samples collected in this experiment is from the network malicious sample database VirusShare. The collected data sets include ELF, Winexe, CryptoRansom, and EK with a total of 9285 malicious executable files. After labeling the data, 80% of the executable files of each malicious code family are taken as the training set, and the remaining 20% are used as the test set. The data set composition is shown in Table 1.

| Malicious code family | Number of samples | Training set | Test set |
|-----------------------|-------------------|--------------|----------|
| Winexe                | 2359              | 1887         | 472      |
| CryptoRansom          | 2460              | 1968         | 492      |
| EK                    | 1832              | 1465         | 367      |
| ELF                   | 2634              | 2107         | 527      |
| Total                 | 9285              | 7427         | 1858     |

3.2. Data Preprocess
After collecting a sample of malicious code, shelling and cleaning may be performed on malicious code obfuscation techniques such as packing and deformation. In order to ensure the integrity of the information, this article fills the pictures transformed by the malicious code executable files in the experimental data set instead of intercepting them. The malicious code execution file selected by the model is less than 65k and greater than 20K. Because the images input by the convolutional neural network must be uniformly fixed in size, with 256 as the line width vector, 0-byte pairs are used for grayscale images with length vectors less than or equal to 256, fill it so that it produces a 256 × 256 square grayscale image. Figure 2 shows examples of some malicious code executable files from left to right from the CryptoRansom, Winexe, EK, and ELF families, it can be seen that the images of the same
family are similar. In this article, we use the B2M method to visualize the malicious code family binary files, this method can more fully extract the characteristic genes of the malicious code and obtain higher accuracy.

![Malicious code family gray map](image)

**Figure 2.** Malicious code family gray map

### 3.3. Malicious code family identification

After the malicious code sample is converted into a grayscale image with family characteristics, the grayscale image of the training set is put into the BiLSTM-CNN model for learning and training, and then the grayscale image of the test set is put into the trained model for detection and classification. Finally, the results obtained by using the training model and method in this paper are compared with the results of other machine learning classification algorithms. This article tested each family of malicious code. There are four main metrics: Accuracy, Precision, Recall and F1-Score.

In order to verify the validity of the model proposed in this paper, the results obtained in the experiments of this paper are compared with the experimental results of several other models. As shown in table 2.

| Classifier       | Accuracy | Precision | Recall | F1-Score |
|------------------|----------|-----------|--------|----------|
| SVM              | 71.26%   | 71.03%    | 68.52% | 69.75%   |
| Random Forest    | 78.64%   | 78.36%    | 79.43% | 78.89%   |
| CNN              | 82.18%   | 81.29%    | 81.67% | 81.98%   |
| RNN              | 82.69%   | 83.57%    | 82.87% | 83.22%   |
| CNN-BiLSTM       | 89.31%   | 89.16%    | 88.65% | 88.90%   |
| BiLSTM-CNN       | 88.87%   | 89.52%    | 89.09% | 89.30%   |

From the comparison of the experimental results of various machine learning algorithms in Table 2, it can be seen that the accuracy, precision and recall of the two deep learning algorithms CNN and RNN for malicious code family detection and classification are better than those of SVM and Random Forest. Among them, the RNN algorithm's detection accuracy rate is 82.69%, which is 11.43% and 4.05% higher than the two types of machine learning algorithms of SVM and Random Forest, the reason is that the RNN algorithm has certain advantages in the processing of sequence data. It can learn the deeper features of the malicious code family and achieve higher accuracy. The combination of the BiLSTM-CNN algorithm formed by combining the two algorithms of CNN and RNN achieves better results than the two. In terms of F1-Score, the BiLSTM-CNN algorithm combines the advantages of both local and global feature extraction to reach 89.30%, which is 7.32% and 6.08% higher than the CNN and RNN algorithms, respectively.

In addition, a comparative experiment was performed in this article, the CNN-BiLSTM model was obtained by swapping the order of CNN and BiLSTM. Comparative experiments under the same conditions show that the BiLSTM-CNN model is equivalent to the CNN-BiLSTM model in terms of accuracy, but in terms of Precision, Recall, and F1-Score, the BiLSTM-CNN model has better performance and achieves the best results among the above six machine learning algorithms. The reason is that in the application scenario of this article, the grayscale image is first extracted by the recurrent neural network to extract the context features, and then the convolutional neural network is used to extract the local features, which is more conducive to the detection of the malicious code family.
Experiments prove that the BiLSTM-CNN network model proposed in this paper is more accurate in classifying malicious code than some traditional machine learning algorithms.

4. Conclusions
The number of malicious codes has grown rapidly, and network security issues need to be addressed urgently. Aiming at the problems of traditional malicious code family classification algorithms, this paper uses B2M visualization technology to convert malicious code samples into two-bit grayscale images, and then uses the BiLSTM-CNN model to automatically learn features from image data sets, finally, the four malicious code families collected are detected and classified. In terms of feature extraction, this method saves the computer from being damaged by malicious code, saves a lot of resources and time, and extracts the family features of malicious code more comprehensively. In classification detection, the model combines the advantages of BiLSTM and CNN to extract malicious code family features from both global and local aspects, improving the accuracy of detection classification. From the experimental results, it can be seen that compared with traditional machine learning methods, the BiLSTM-CNN network model has a higher accuracy rate for detecting and classifying malicious code families when the same malicious code data set is classified and detected, and achieved better results.

Acknowledgments
This work was financially supported by the National Cryptography Development Fund (MMJJ20180108). Correspondence should be addressed to Tianliang Lu lutianliang@ppsuc.edu.cn.

References
[1] Symantec. Internet security threat report (ISTR) 2019 (24): 18-21.
[2] Likarish P, Jung E, Jo I. “Obfuscated malicious javascript detection using classification techniques” [C]. International Conference on Malicious and Unwanted Software. IEEE_ 2010: 47-54.
[3] MohammadFraiwan, RamiAl-Salman, NatheerK hasawneh, et al. “Analysis and Identification of Malicious JavaScript Code” [J]. Information Systems Security, 2012, 21 (1): 1-11.
[4] Jiawei Su, Danilo Vasconcellos Vargas, Sanjiva Prasad et al., “Lightweight Classification of IoT Malware based on Image Recognition” [J], CoRR abs/1802.03714 (2018).
[5] Yuntao Zhao, Chunyu Xu, Bo Bo et al., “MalDeep: A Deep Learning Classification Framework against Malware Variants Based on Texture Visualization” [J]. Security and Communication Networks Volume 2019, Article ID 4895984, 11 pages.
[6] Curtisinger C, Livshits B, Zorn B, et al. “ZOZZLE: fast and precise in-browser JavaScript malware detection” [C]. LTsenix Conference on Security. TJSENTX Association_ 2011: 3-3.
[7] Pan J, Mao X. “Obfuscated Malicious JavaScript Detection by Machine Learning” [C]. International Conference on Advances in Mechanical Engineering and Industrial Informatics. 2016.
[8] Aziz Mohaisen, Omar Alrawi, Jeman Park et al., “Network-based Analysis and Classification of Malware using Behavioral Artifacts Ordering” [J]. ICST Trans. Security Safety 5(16): e2 (2018).
[9] Mitsuhiro Hatada, Tatsuya Mori, “Finding New Varieties of Malware with the Classification of Network Behavior” [J]. IEICE Transactions 100-D (8): 1691-1702 (2017).
[10] Vitor M, Afonso, Ricardo A, et al. “A hybrid system for analysis and detection of Web-based client-side malicious code” [J]. Las.ic.unicamp.br, 2011.
[11] Jungho Kang, Sejun Jang, Shuyu Li et al., “Long short-term memory-based Malware classification method for information security” [J]. Computers & Electrical Engineering 77: 366-375 (2019).
[12] D. Xue, J. Li, T. Lv et al., “Malware Classification Using Probability Scoring and Machine Learning” [J]. IEEE Access, vol. 7, pp. 91641-91656, 2019.
[13]  Kim HJ Image-Based “Malware Classification Using Convolutional Neural Network” [J]. Advances in Computer Science and Ubiquitous Computing (CSA-CUTE 17). 2017, 474 (1): 1352-1357.

[14]  Sang Ni, Quan Qian, “using visualization Rui Zhang. Malware images and deep identification learning” [J]. Computers&Security. 2018, 77 (6): 871-885.