ISSN 2616-5775 Vol. 5, Issue 8: 95-100, DOI: 10.25236/AJCIS.2022.050814

Research and Application of Malware Classification Method Based on LSTM

Bochun Hu[a], Yuqing Yang[b], Jinghao Wei[c], Bin Wu[d]

School of Computer Science and Technology, Guizhou University, Guiyang, Guizhou, 550025, China
[a]867409997@qq.com, [b]cp.yqyang19@gzu.edu.cn, [c]827031815@qq.com, [d]806418267@qq.com

Abstract: *Information technology enhances efficiency and introduces the threat of malware. A typical means of destruction, malware affects the operational security of information systems and poses certain risks to human life, production, and social operations. One of the key research directions in the field of information security is the identification and classification of malware. Currently, mainstream malware analysis and identification methods fall into static and dynamic categories. As a result of obfuscated system calls, it is difficult to detect malware using static analysis, and traditional dynamic analysis methods based on local features are not sufficiently accurate. This paper combines sequence-based LSTM neural networks with learning algorithms for traditional API call features to study malware classification. Furthermore, this paper discusses the idea of hybrid classification based on LSTM and expands the research in this area to some extent. As a result of the research presented in this paper, dynamic analysis and classification of malware are expected to be more effective.*

Keywords: *LSTM; Malware; API call features*

1. Introduction

There are two sides to the development of information technology. On the one hand, it enhances the efficiency of the functioning of human society. On the other hand, it also provides a new avenue for external intrusion and destruction of the business. Malware is one of the most common threats to information systems. Information security practitioners face a challenging task in identifying and classifying malware due to ongoing iterative upgrades to its behavioral characteristics, such as its spreading, manifestation, and destruction methods. Information security practitioners face a challenging task in identifying and classifying malware due to ongoing iterative upgrades to its behavioral characteristics, such as its spreading, manifestation, and destruction methods. Traditional malware identification methods rely primarily on static feature identification. By analyzing malware samples manually, features such as internal instruction types and instruction numbers can be extracted. Due to the continuous enrichment of malware types and the continual development of packaging technology, the limitations of static analysis have become more apparent, and it is relatively difficult to reverse the malware's core code alone [1]. Therefore, dynamic extraction and analysis came into being. The main purpose of dynamic analysis is to extract its behavior patterns and summarize common identification features. However, the variety of malware emerges in an endless stream, making the identification mentioned above based on fixed characteristics less effective. Based on this situation, the malicious sample analysis method with machine learning as the core came into being. In this mode, the identification and classification of malware mainly rely on feature mining of the call sequence vector of the malware, and the artificial intelligence neural network is used to capture the features of many API calling behaviors of the malware to achieve a wider scope and more high recognition efficiency [1]. This paper mainly studies the application of neural networks in malware classification, especially the integrated application of API parameter vectorization technology based on LSTM, hoping to find a more efficient malware classification method by mining its hidden rules and characteristics.

2. Related Technology Overview

2.1 Deep Learning and LSTM

The concept of deep learning originated from human research and the simulation of human neural networks. It is expected that computers can autonomously capture the internal features and relationships

ISSN 2616-5775 Vol. 5, Issue 8: 95-100, DOI: 10.25236/AJCIS.2022.050814

of sample data such as sounds, images, and texts by building a multi-layer perceptual organization and strengthening machine learning. And finally, form a predictive model. In theory, the greater the depth, the stronger the recognition ability and the better the learning effect [2]. So far, deep learning has been applied in many fields, including image, semantic recognition, object extraction, etc.

RNN network is a typical deep learning network, and its typical feature is the loop. During the cycle, each feature participating in the cycle will retain its information in the hidden neurons and continue to enter the cycle with new information. In this way, each cycle will affect the result. Unlike the traditional neural network, there are connections between the nodes in the hidden layer, which enables the network to memorize the time sequence of information, which is also the origin of its short-term memory characteristics. The short-term memory characteristics of the RNN network are difficult to meet the prediction and learning environment that cares about the context, so LSTM was proposed and used today [2].

LSTM is an improved network formed to deal with the gradient disappearance problem of the RNN network, which can meet the demand for the long-term memory performance of the network during the training process. A key valve is set in LSTM, and the sigmoid forget gate. This valve determines whether the information is retained when the data is transmitted between the neural networks. Its principle can be seen in Figure 1.

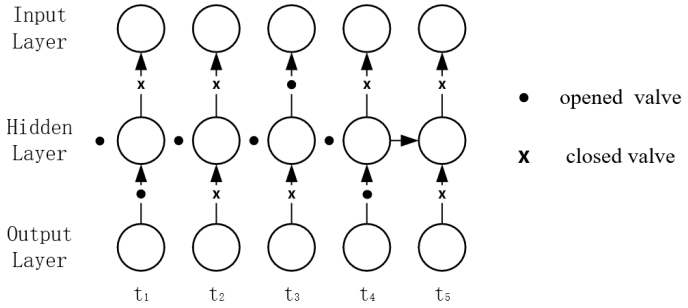

Figure 1: Schematic diagram of the principle of LSTM forgetting gate

2.2 API Parameter Vectorization Technology

API parameter vectorization technology essentially comes from the idea of API vectorization. System API calls are the key to analysis in the process of malware identification and classification. API calls are usually encoded and vectorized to meet the input requirements of a neural network. At the same time, the API call features are learned like semantic recognition, and then feature capture and prediction are performed. A common vectorization tool is the word embedding technique.

API parameter vectorization is a new research field formed based on the above ideas, and the related research is relatively weak. The traditional identification method only collects and processes the API call name for API calls generated by malware. However, in essence, a series of parameters are generated during the API call process, such as file path, call sequence, IP address, etc. [2]. These parameters also contain much information. For example, when the file deletion API is called, the type of filehandle or file address directly determines the danger of the file deletion API call: If the target of the parameter of the call is a common application file, the harm is relatively low, but the target is a system file, which may cause a system crash.API parameter vectorization is to vectorize this valuable parameter information and use a neural network for further processing.

3. Research and Application of Malware Classification Method Based on LSTM

In order to realize the malware classification method based on API parameter vectorization and LSTM neural network, the data preprocessing mode needs to be designed first, and the application and fusion method of API parameter vectorization technology should be studied at the same time. After that, it is necessary to construct a network for this classification and identification task and then implement the LSTM-based malware classification method.

ISSN 2616-5775 Vol. 5, Issue 8: 95-100, DOI: 10.25236/AJCIS.2022.050814

3.1 Data Preprocessing

Data preprocessing is very important for deep learning and neural network training. This time, the basic method of malware identification based on API calls must be combined with deep neural networks. The core idea is to use natural language processing ideas, treat API and its parameters as words, and treat the entire API calling process as malware. Treat it as a complete article.

Based on this idea, we first consider using the N-gram method to extract subsequences. N-gram uses the idea of the sliding window to construct a sequence of byte fragments of length N byte by byte. By performing frequency statistics on each byte segment and filtering it with the set rules, a keyword list, namely a gram list, is finally formed, and a feature vector space that can describe the text is finally formed [3].

Since the subsequences extracted by N-gram often have the problem of high dimensionality, it is necessary to reduce the dimensionality and remove redundant information in a certain way to improve training efficiency. This paper evaluates the amount of information brought by different subsequences to the final screening through the information gain feature to determine whether there is a reserved value. Equation 1 represents the information increment brought by the self-sequence feature T to the final classification C. Where VT represents the value of the subsequence T, if the subsequence appears in a certain sample, then V_T=1, otherwise V_T=0; $P(V_T, C)$ represents the ratio of T in the final classification C; $P(V_T)$ represents the T ratio in the whole sample; P(C) represents the ratio of category C in the entire sample. According to this idea, subsequences with higher information gain are more valuable for sample identification and classification so that the characteristic subsequences can be further screened according to this information.

$$IG(T) = H(C) - H(C|T) = \sum_{V_t \in \{0,1\}} \sum_{C \in C_i} P(V_T, C)\, log(\frac{P(V_T,C)}{P(V_T)P(C)}) \quad (1)$$

In the previous step, valid feature subsequences are screened out, and by connecting these valid feature subsequences in time sequence, the calling sequence of samples in time can be obtained. After the above processing, each element in the feature subsequence set of the final input model has typical classification features and can significantly reduce the training sequence's length and improve the training efficiency.

3.2 API Parameter Vectorization Technology Application

This paper proposes API parameter vectorization as a feature sequence generation method based on API vectorization. For different APIs, the number and length of the parameters vary. For example, the range of IP addresses is fixed, but the file path parameter has no fixed length and range. This makes vectorizing a difficult problem. This paper proposes the parameter hash information and word embedding technology as the core idea of parameter vectorization. First, the original data is used to construct a hash space through hash mapping, and then word embedding technology is used to embed words in the hash space to complete the parameterization. Vectorization process [4].

The specific technical path is as follows:

(1) reate an array h_vector, and use this array to create a hash space. In particular, to identify the occupancy in the space, all elements of the array are initialized with -1, indicating that the space is not occupied. After that, a mapping relationship is established between the parameters and the specific storage locations of the hash space (see Equation 2). Where $doHash$ is the method for calculating the hash value of the parameter w_j.

$$h\_vector[doHash(w_j)] = j \quad (2)$$

$$doHash(w_i) = dohash(w_j)\,,\ (i \neq j) \quad (3)$$

As described in Equation 3, there may be a hash occupancy conflict problem in this step, so a linear open addressing method is introduced here to deal with the problem of the same hash value, that is: if the hash value of the parameter w_j is found in the hash space h_vector. When the same item already exists, press down to find the vacant position and select occupy.

(2) Perform word embeddings. This paper uses the embedding lookup method to complete the vectorization further. The embedding lookup method converts discrete variables into vectors by random mapping. This method can distribute the vectors in an N-dimensional space, making it have the characteristics of hyperplane separability, which is more suitable for applying neural networks.

ISSN 2616-5775 Vol. 5, Issue 8: 95-100, DOI: 10.25236/AJCIS.2022.050814

(3) Perform parameter filtering. Among the parameters of API calls, the frequently appearing parameters often lack characteristics, and the effective information that can be provided for subsequent analysis is relatively insufficient. Therefore, some downsampling can be considered to improve the training efficiency of the model. In this paper, we set a threshold t and calculate the discarding probability of the parameter w by the method described in Equation 3 to realize the optimization of the parameter [4]. Where $f(w)$ is the calculation function of the occurrence frequency w, which can be expressed as Equation 4.

$$prob(w) = 1 - (\sqrt{\frac{t}{f(w)}} + \frac{t}{f(w)}) \tag{4}$$

$$f(w) = \frac{counter(w)}{\sum_{u \in D} counter(u)} \tag{5}$$

3.3 Construction of Residual Recurrent Neural Network Based on LSTM

As mentioned above, LSTM is an improved RNN network, and its cycle and selection forgetting features enable it to effectively deal with gradient descent and gradient explosion problems, especially suitable for processing data with time series. Since the core of the problem proposed in this paper is applying local features in the sequence, this paper proposes a residual recurrent neural network based on LSTM. Traditional neural networks are often unable to predict long-term sequences. Combining the long and short-term memory characteristics of LSTM can effectively record the temporal relationship between sequences in the context. By controlling the forgetting gate, more important information can be selectively memorized to provide more accurate support for subsequent predictions [5].

(1) Gate Mechanism of LSTM

The gates of LSTMs are the key to memory, which is a structure that controls the flow of information in and out of memory blocks, and then controls the extent to which information is remembered and propagated. In the loop process, if the previous input does not meet the threshold requirements, it will be discarded, equivalent to forgetting the information. If the threshold requirements are met, the matrix multiplication operation is performed with the current result, and the transfer is continued. Figure 2 depicts the structure of a typical memory block. The key input, output, and forget gates are described [6]. The black dots in the figure represent the multiplication operation, f represents the activation function, and the sigmoid function is used here. H is the output gate function, usually a sigmoid or tanh function [6]. The dotted line in the figure represents the weight information.

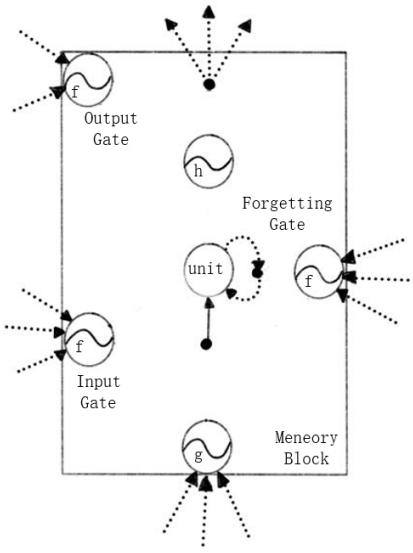

Figure 2: Schematic diagram of memory block structure

Define an input sequence $X = \{x_0, x_1, \cdots, x_t, \cdots, x_n\}$, where each element is the input information at the corresponding moment. At the same time, define $W_i, W_f, W_c, W_o, U_i, U_f, U_c, U_o$ as the weight matrix, and b_i, b_f, b_c, b_o as the bias vector, then at time t, the calculation of the input gate, the median value of the memory gate, and the forgetting gate are shown in Equation 6-Equation 8, the state value of the memory gate, the output of the output gate, and the output of the final hidden layer can also be

ISSN 2616-5775 Vol. 5, Issue 8: 95-100, DOI: 10.25236/AJCIS.2022.050814

respectively used Equation 9-Equation 11 to express.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{6}$$

$$C_t' = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{7}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{8}$$

$$C_t = i_i * C_t' + f_t * C_{t-1} \tag{9}$$

$$o_t = \sigma(x_t + U_o h_{t-1} + b_o) \tag{10}$$

$$h_t = o_t * tanh(C_t) \tag{11}$$

(2) Bidirectional Recurrent Network Based on LSTM

Since the increase in the number of LSTM layers means an increase in the abstraction level of high-level features, it also means higher time consumption. In order to balance efficiency and effect, a two-layer implicit LSTM model structure is selected here. As shown in Figure 3, the core of the LSTM-based model designed this time is constructed as multiple serial SA blocks containing n residual structures. To further remove redundant information accumulated during the convolution cycle, after n SA blocks, a max-pooling layer is set for downsampling [7]. Finally, the classification output for the input information is realized through a fully connected layer with the logical classifier.

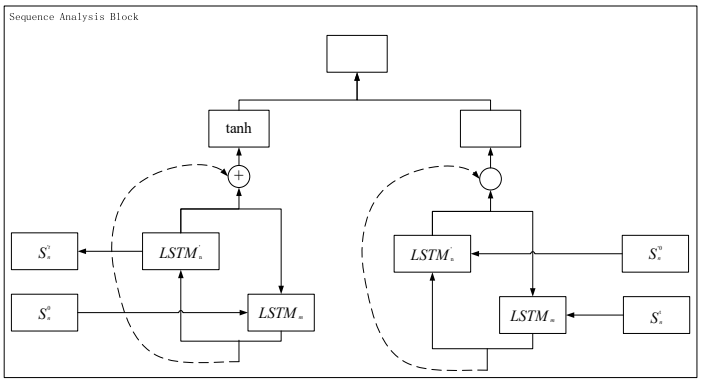

Figure 3: Core structure of the deep neural network

(3) Applications of residual connections

To further improve the classification effect, it is necessary to increase the network depth further. Residual connections are an essential technique for training deep neural networks. Introducing this technology can effectively avoid the problem that the performance of the deep network is inferior to that of the ordinary network. Due to the problem of gradient descent in the training process of deep networks, especially after repeated calculations with weights less than 1, the gradient may even disappear. By increasing the residual connection, the gradient can be transmitted forward, effectively avoiding the loss of gradient in the training process so that the deep neural network can achieve higher feature extraction performance [8].

In essence, this residual connection has been reflected in the recurrent bidirectional network designed above, as shown in Figure 3: A residual connection from input to output is set in each residual structure in the SA block, and the gradient of the input stage is passed forward, superimposed with the output of the bidirectional cyclic system, it is then superimposed with the output of the bidirectional recurrent structure and finally used as the output of the residual structure through the tanh activation function.

(4) Extraction and Classification of Hidden Features

In the extraction stage of the feature information of the hidden layer, this paper proposes to use the MaxPooling algorithm. MaxPooling is a nonlinear downsampling layer that reduces dimensionality and avoids overfitting. In order to facilitate the work of the classifier, it is necessary to use this layer to realize the output of the feature vector of equal length and to ensure that the feature information can be effectively preserved and transmitted under the premise that the dimension is controlled.

In addition, the problem of this research is malware classification, which belongs to the binary classification algorithm, so the Logistic classifier is finally selected to output the results of the entire network.

ISSN 2616-5775 Vol. 5, Issue 8: 95-100, DOI: 10.25236/AJCIS.2022.050814

3.4 Research on the Idea of Hybrid Classification Based on LSTM Network

In the previous paper, an LSTM-based residual recurrent neural network was proposed to solve the problem of malware classification. Within traditional machine learning, there are more effective methods to solve malware classification. The typical method is to classify API calls based on statistical features based on random forest [9]. In order to further improve the performance of the LSTM-based malware classification method designed this time, it can be considered to integrate the designed LSTM neural network model with the traditional random forest classification method to realize a hybrid classification scheme, which can be expressed as Formula 12-Formula 14.

The core idea of hybrid classification is:

(1)Use the LSTM and random forest networks to obtain their respective sample detection probabilities, denoted as P_{LSTM} and P_{RF} here.

(2)Calculate the confidence of the two detection methods, denoted as C_{LSTM} and C_{RF} here.

(3)The calculated probability is selected with confidence, and the final detection probability P is obtained.

This process can be expressed as Equation 12-Equation 14.

$$C_{LSTM} = min(abs(1 - P_{LSTM}), P_{LSTM}) \tag{12}$$

$$C_{RF} = min(abs(1 - P_{RF}), P_{RF}) \tag{13}$$

$$P = \begin{cases} P_{LSTM}, (C_{LSTM} \leq C_{RF}) \\ P_{RF}, (C_{LSTM} > C_{RF}) \end{cases} \tag{14}$$

4. Conclusion

This paper designs a residual cyclic network based on LSTM facing the problem of malware classification. The related methods and paths are constructed by designing data processing algorithms, residual network structures, and building neural network models. Compared with the traditional neural network classification method, the recurrent bidirectional network's expression ability improves the model's expression ability, and the residual structure's introduction further improves the deep network's training effect. In addition, to further improve the classification effect, this paper proposes a hybrid classification idea, using the confidence strategy to fuse the prediction results of LSTM and traditional random forest methods.

References

[1] Liwei Wang, 123, Jiankun Sun, 123, Xiong Luo, 123, Xi Yang. Transferable Features from 1D-Convolutional Network for Industrial Malware Classification [J]. Computer Modeling in Engineering & Sciences, 2022,130(2).
[2] Kakelli Anil Kumar, Kaustubh Kumar, Nag Lohith Chiluka. Deep learning models for multi-class malware classification using Windows exe API calls [J]. International Journal of Critical Computer-Based Systems,2022,10(3).
[3] Mahdavifar Samaneh, Alhadidi Dima, Ghorbani Ali. A. Effective and Efficient Hybrid Android Malware Classification Using Pseudo-Label Stacked Auto-Encoder[J]. Journal of Network and Systems Management,2021,30(1).
[4] Ring Markus, Schlör Daniel, Wunderlich Sarah, Landes Dieter, Hotho Andreas. Malware detection on Windows Audit Logs using LSTMs [J]. Computers & Security,2021(prepublish).
[5] Zhangjie Fu, 123, Yongjie Ding, Musaazi Godfrey. An LSTM-Based Malware Detection Using Transfer Learning [J]. Journal of Cyber Security, 2021, 3(1).
[6] Barath Narayanan Narayanan, Venkata Salini Priyamvada Davuluru. Ensemble Malware Classification System Using Deep Neural Networks [J]. Electronics,2020,9(5).
[7] John Wade. Arbitration of Matrimonial Property Disputes [J]. Bond Law Review,2019.
[8] Stegman Michael J. Matrimonial property in the American states: choice-of-law and conflict-of-laws issues[J]. Trusts & Trustees,2019,25(1).
[9] Juma Katarina. Family Law Digest: Matrimonial Property [M].African Books Collective:2009-01-01.