Malicious Domain Name Detection Based on Doc2vec and Hybrid Network

Donglin Ma, Shuhuan Zhang*, Fanqi Kong and Zhaoyang Fu
Department of Computer and Communication, Lanzhou University of Technology, Lanzhou 370050, Gansu, China.
Email: 599239440@qq.com

Abstract. In traditional malicious domain name detection algorithms, machine learning detection methods take a long time to extract features and the extraction methods are more complex; deep learning detection methods are easy to lose the semantic features of the entire domain name, and the detection results of a single neural network for different sample data sets are unstable. In response to the above problems, a malicious domain name detection method based on Doc2vec and hybrid network was proposed, and a new DLR (Doc2vec-LSTM-RNN) detection model was constructed. The word vector is constructed using the Doc2vec algorithm optimized on the basis of the Word2vec algorithm, which retains the semantic information of the domain name as a whole; and mixes the bidirectional LSTM network and the bidirectional RNN network in series to carry out deep-level feature extraction and enhance the robustness of the detection model; finally the Softmax function is used to output the classification result. Experimental data shows that this method has excellent performance in malicious domain name detection. Compared with traditional machine learning and deep learning detection methods, it has higher detection accuracy.

1. Introduction
The Domain Name System (DNS) as an infrastructure of the Internet [1], provides services for the conversion of domain names and IP addresses, routing of application layer data messages, and load balancing services for servers. Because DNS is widely used and lacks a security detection mechanism, it has become the main target of malicious domain names. Through malicious domain name detection technology, the malicious domain name appearing in the network can be detected in time and intercepted, which can effectively prevent malicious domain name attacks.

Among the many techniques for network attacks, botnets are the most used means by attackers, and the domain name generation algorithm (DGA) [2-3] adopted by them is a great challenge to security personnel. The main technical means is to continuously change the domain name of the command and control host (C&C) [4] to prevent detection. There are currently two main types of botnets. One is to use DGA technology to continuously and rapidly change the domain name and IP address of the C&C host; the other is to change the resolution server of the domain name on the basis of constantly changing the domain name, and realize the control of the botnet based on the registration of multiple parsing servers.

For malicious DGA domain names, there are currently two main categories of detection methods at home and abroad. The first is the detection of malicious domain names based on machine learning [5], which is mainly to manually extract features to establish feature equations [6], and then detect them by feature matching, which mainly includes the detection of domain name character statistical features and DNS traffic information [7]; the second is the detection of malicious domain names based on deep learning [8], which mainly solves the problem that cannot be circumvented by manual feature
extraction through the ability of deep neural network self-learning features, and its accuracy and false positive rate All aspects are better than traditional machine learning techniques, mainly including recurrent neural networks and convolutional neural networks.

A new malicious domain detection algorithm model DLR (Doc2vec-LSTM-RNN) is proposed in this article. Using the Doc2vec algorithm to vectorize the domain name text to obtain more complete semantic information, and then combine the LSTM and RNN neural networks to achieve more accurate detection of malicious domain names.

2. DLR Algorithm Model

First, the data set are preprocessed, and irrelevant content such as punctuation, special characters, and http headers etc. in the original domain name is removed through regular expressions; then, the Doc2vec algorithm which optimized on the basis of the Word2vec algorithm is used to convert the domain name to be tested into a vector space sequence expression is used as the embedding layer of the model, and preliminary feature extraction is performed at the same time.

Secondly, because of the single neural network has a weak ability to express features, Bi-LSTM network and Bi-RNN network are connected as the neural network layer of the model, to extract the depth features on the vector sequence of the domain name; preserve the association between forward and backward character sequences to make the semantics more fully expressed.

Finally, the output of the hybrid is linearly processed, and the probability distribution is calculated using the Softmax classifier to realize malicious domain detection. The algorithm model is shown in figure 1.

![Figure 1. DLR algorithm model.](image)

2.1. Doc2vec Algorithm

When using the neural network to process the domain name, the text sequence of the domain name should be vectorized first. In 2010, Mitchell and Lapata [9] proposed a semantic distribution model combination to generate phrase or sentence representation. Zanzotto et al. [10] also proposed an innovation on compositional distributional model and added an estimation method based on multiple independent variable regression model. In 2011, Yessenalina and Claire [11] proposed a phrase representation method of matrix space, which can be effectively used in the text of emotion analysis. In 2013, Tomas Mikolov et al. [12] improved the quality and training speed of the representation vector by using frequent words twice and learning more conventional words. These research methods simply use the weighted average of all words in the document to obtain vector representations, while more complex methods combine word vectors that follow a certain order in the parse tree. Both approaches have a disadvantage. The first is the loss of word order, and the second is the combination of analytic trees and word vectors that only apply to sentences because it relies on parsing.

In 2013, Thomas Mikolov et al. [13] proposed Word2vec algorithm, in which the famous Continuous Bag-of-Word (CBOW) Model structure is shown in figure 2:
The three words "Web", "of" and "science" in the domain name are respectively mapped into vectors, then spliced and averaged to obtain the output sequence. Inspired by Word2vec, Quoc Le and Tomas Mikolov [14] proposed Paragraph Vector in Google in 2014. In text recognition, the performance of Paragraph Vector can be improved by 30% compared with the word bag model. In addition, the paragraph vector of Doc2vec can be used in phrases, sentences or paragraphs, unlike the previous methods which have the limitations of application scenarios. Doc2vec algorithm is an unsupervised algorithm, which can learn feature representation of fixed length from text of different length. The algorithm predicts words within a document, enabling it to represent each document with a single vector. Paragraph Vectors can be used to predict the next word based on a given context sample from the paragraph. The original paper mentioned two models: Distributed Memory Model of Paragraph Vectors (PV-DM) and Distributed Bag of Words Version of Paragraph Vectors (PV-DBOW). The frame of the Distributed Memory Model of Paragraph Vectors (PV-DM) is shown in figure 3:

The framework is similar to the CBOW model framework in Word2vec. The difference is that through matrix D, additional paragraph segments are placed into a single vector. In this model, the concatenation or average result of this vector and three other context vectors is used to predict the fourth word. The paragraph vector represents contextual missing information and also acts as a memory for the topic of the paragraph. The advantage of paragraph vector is that it overcomes the disadvantage of word bag model. First, feature vector can inherit the semantic information of word.
Secondly, it considers word order like n-gram model [15], but the applicability of paragraph vectors is better than that of n-gram model due to the low generality caused by high-dimensional representation.

2.2. Hybrid Network
Since manual extraction of feature values is relatively labile and time-consuming, neural network is used instead of manually extracting domain name feature values. In this article, bidirectional LSTM network and bidirectional RNN network are used in combination.

2.2.1. Bi-LSTM layer. After the vectorization encoding, the character sequence is input into the bidirectional LSTM layer. The LSTM unit is controlled by a gate, figure 4 and figure 5 show the basic structure of the forward LSTM and backward LSTM respectively.

![Figure 4. Forward LSTM basic structure.](image)
![Figure 5. Backward LSTM basic structure.](image)

An LSTM unit consists of a forgetting gate, an input gate, and an output gate, allowing access states on long sequences to be stored within the neural network, thereby reducing gradient disappearance. There are two transmission states, one is c (cell State) and the other is h (hidden state). Usually, the output c2 is the c1 passed from the previous state plus some value, while h varies considerably from node to node. Firstly, the current input \(x^t\) of LSTM and the \(h^t_{t-1}\) passed down from the previous state are used for splitting training to obtain four state, as shown in equation (1) - equation (4):

\[
\begin{align*}
    z &= \tanh(W_h^x h_{t-1}) \\
    z^i &= \sigma(W_i^x h_{t-1}) \\
    z^f &= \sigma(W_f^x h_{t-1}) \\
    z^o &= \sigma(W_o^x h_{t-1}) \\
\end{align*}
\]

Where \(z^f, z^i\) and \(z^o\) are transformed into values between 0 and 1 by a Sigmoid activation function after the assembly vector is multiplied by the weight matrix, with 0 representing discard and 1 representing pass as a gated state. And \(z\) is converting the result to a value between -1 and 1 through a Tanh activation function.

For domain name detection, it can learn dependencies between multiple characters in a continuous or discrete sequence, linking the output of the last unit forward and backward as input to the next level.

2.2.2. Bi-RNN layer. In the internal structure of bidirectional recurrent neural network, each training sequence forward and backward can be regarded as two recurrent neural networks, and these two networks are connected to the same output layer. Figure 6 and figure 7 are the basic structures of forward and backward RNN respectively:
Figure 6. Forward RNN basic structure.

Figure 7. Backward RNN basic structure.

Where \( x \) is the input of the data under the current state, and \( h \) is the input of the previous node received. \( y \) is the output in the current node state, and \( h' \) is the output passed to the next node. The output \( y \) needs to be mapped to a linear layer using \( h' \) and classified using the Softmax function. This structure provides complete contextual information about the past and future at each point in the input sequence, both forward and backward acting on the result.

2.3. Softmax Layer
First, the forward and backward outputs of the bidirectional RNN layer are spliced together; then it is processed by linear transformation; finally, Softmax function is used to obtain the final probability distribution and realize malicious domain name detection. The input of the Softmax classifier is the result obtained from \( K \) different linear functions, and the output of multiple neurons is mapped into the interval \([0,1]\), which is represented by equation (5):

\[
P(y = j) = \frac{e^{x^T W_j}}{\sum_{k=1}^K e^{x^T W_k}}
\]  

Where \( P \) is the probability that the sample vector \( x \) belongs to the \( j \)th classification. Softmax regression model is used as a supervised classifier in deep learning in combination with cross entropy loss function. It can be expressed in equation (6):

\[
H(p, q) = \sum_x p(x) \log q(x)
\]  

Where \( p \) represents the distribution of real markers, \( q \) represents the predicted marker distribution of the trained model, and the cross entropy loss function can measure the similarity between \( p \) and \( q \). Another advantage of cross entropy as a loss function is that Sigmoid function can avoid the problem of decreasing learning rate of mean square error loss function in the case of gradient descent.

3. Experimental Design and Result Analysis

3.1. The Data Set
The data set of this experiment is divided into three parts: training set, verification set and test set. Select the top 100,000 domain names in Alexa website as the normal domain name set; The DGA domain names of five families including Banjori, Emotet, Gameover, Rovnix and Ramnit were selected from the malicious domain names published by 360 Security Lab, with 20,000 entries per
family. The normal domain name set was randomly divided into 5 groups of the same amount and mixed with the five family DGA domain names respectively. Then the training set, verification set and test set were divided according to the ratio of 3:1:1, and the comparative experiment was carried out.

3.2. Parameter Settings
The DLR detection model was constructed, after using the training set to train the data and fit the model, the verification set was used to adjust the model's super parameters. After many experiments, the batch size was set as 10, the training times were 100, the discard rate was 0.5, the number of LSTM embedded layers was 5, and the number of hidden layer nodes was consistent with the length of the vector, which was set as 100.

3.3. Experimental Environment
The experimental environment is shown in table 1:

| ENVIRONMENT | PARAMETER |
|-------------|-----------|
| CPU         | i9-10850K(4.70GHz) |
| GPU         | RTX3080 10GB |
| MEMORY      | 16GB |
| SYSTEM      | Windows 10 2004(64-bit) |
| IDE         | Pycharm |
| LANGUAGE    | Python3.8 |

3.4. Model Evaluation Criteria
The performance was evaluated by using recall, precision and F1. The calculation formulas are shown in equations (7), (8) and (9):

\[
Recall = \frac{TP}{TP+FN} \tag{7}
\]

\[
Precision = \frac{TP}{TP+FP} \tag{8}
\]

\[
F1 = \frac{2\times Precision \times Recall}{Precision + Recall} \tag{9}
\]

Where \(FN\) represents the number of malicious domain names misreported as legitimate domain names; \(FP\) represents the number of legitimate domain names misreported as malicious domain names; \(TP\) represents the total number of malicious domain names accurately detected by the algorithm. In equation (9), the results of recall and precision are combined, can be seen that the higher of the F1 value, the more effective the detection method will be.

3.5. Evaluation of Model Effectiveness
In order to verify the performance of the detection model DLR proposed in this paper, the Invincea-CNN detection model proposed in reference [16], the M-LSTM detection model proposed in reference [17] and the AN-LSTM detection model proposed in reference [18] were constructed under the same experimental environment, and the same data set was used for testing. The experimental results are shown in table 2 - table 6.

| Banjori     | Recall/% | Precision/% | F1/%   |
|-------------|----------|-------------|--------|
| Invincea-CNN| 91.62    | 92.13       | 91.88  |
| M-LSTM      | 92.32    | 91.47       | 91.90  |
| AN-LSTM     | 93.24    | 93.22       | 93.23  |
| DLR         | 96.67    | 96.25       | 96.46  |
Table 3. Test results for Emotet family domain name.

| Emotet     | Recall/% | Precision/% | F1/%  |
|------------|----------|-------------|-------|
| Invincea-CNN | 91.29    | 91.77       | 91.53 |
| M-LSTM     | 92.34    | 92.87       | 92.61 |
| AN-LSTM    | 93.60    | 93.25       | 93.43 |
| DLR        | 96.35    | 97.12       | 96.74 |

Table 4. Test results for Gameover family domain name.

| Gameover | Recall/% | Precision/% | F1/% |
|----------|----------|-------------|------|
| Invincea-CNN | 91.53    | 92.56       | 92.05 |
| M-LSTM   | 92.38    | 91.56       | 91.97 |
| AN-LSTM  | 92.94    | 93.65       | 93.30 |
| DLR      | 97.81    | 96.15       | 96.98 |

Table 5. Test results for Rovnix family domain name.

| Rovnix    | Recall/% | Precision/% | F1/% |
|-----------|----------|-------------|------|
| Invincea-CNN | 91.74    | 91.73       | 91.74 |
| M-LSTM    | 92.25    | 91.97       | 92.11 |
| AN-LSTM   | 93.22    | 92.85       | 93.04 |
| DLR       | 96.15    | 97.54       | 96.85 |

Table 6. Test results for Ramnit family domain name.

| Ramnit    | Recall/% | Precision/% | F1/% |
|-----------|----------|-------------|------|
| Invincea-CNN | 92.47    | 92.88       | 92.67 |
| M-LSTM    | 91.52    | 92.12       | 91.82 |
| AN-LSTM   | 92.39    | 91.48       | 91.93 |
| DLR       | 97.34    | 98.96       | 98.14 |

It can be clearly seen from table 2 – table 6 that, compared with several traditional malicious domain detection models, the DLR model proposed in this paper has greater advantages in recall rate, accuracy rate and F1 value. Among them, in the detection of malicious domain names of the Gameover family, the recall rate of the DLR model has the highest increase of 6.28%; and the highest accuracy rate increased by 7.48% for Ramnit family; and the highest F1 value increased by 6.32% for Ramnit family. This shows that for different DGA domain names, the detection performance of DLR model is relatively stable, and there is no shortcoming, has excellent robustness and strong practicability.

4. Conclusion
In this paper, a malicious domain name detection method based on Doc2vec and hybrid network is proposed. A new detection model DLR is constructed. Experiments have proved that the DGA domain name detection effect of this model is better than the traditional deep learning model. Its advantage lies in the use of Doc2vec algorithm for vector space embedding, which can retain the semantic information of the text to the greatest extent; at the same time, the bidirectional LSTM network and the bidirectional RNN network are connected in series to fully capture the internal connection of the
domain name text, and the feature value can be obtained more accurately. The detection accuracy rate has been significantly improved.

5. References
[1] Ahmed J, Gharakheili H H, Raza Q, Russell C and Sivaraman V 2020 Monitoring enterprise DNS queries for detecting data exfiltration from internal hosts J. Ieee T. Netw. Serv. Man. 17 265-79
[2] Mao J, Zhang J M, Tang Z and Gu Z L 2020 DNS anti-attack machine learning model for DGA domain name detection J. Phys. Commun-Amst. 40
[3] Can N V, Tu D N, Tuan T A, Long H V, Son L H and Son N T K 2020 A new method to classify malicious domain name using Neutrosophic sets in DGA botnet detection J. J Intell. Fuzzy Syst. 38 4223-36
[4] Xu C Y, Shen J Z and Du X 2019 Detection method of domain names generated by DGAs based on semantic representation and deep neural network J. Comput. Secur. 85 77-88
[5] Almashhdani A O, Kaiiali M, Carlin D and Sezer S 2020 MaldomDetector: a system for detecting algorithmically generated domain names with machine learning J. Comput. Secur. 93
[6] Fukushi N, Chiba D, Akiyama M and Uchida M 2020 Exploration into gray area: toward efficient labeling for detecting malicious domain names J. Ieice T Commun. E103B 375-88
[7] Truong D T, Tran D T and Huynh B 2020 Detecting malicious Fast-Flux domains using feature-based classification techniques J. J Internet Technol. 21 1061-72
[8] Yang L H, Liu G J, Dai Y W, Wang J W and Zhai J T 2020 Detecting stealthy domain generation algorithms using heterogeneous deep neural network framework J. Ieee Access 8 82876-89
[9] Mitchell J and Lapata M 2010 Composition in distributional models of semantics J. Cognitive Sci. 34 1388-429
[10] Zanzotto F M, Korkontzelos I, Fallucchi F and Manandhar S 2010 Estimating linear models for compositional distributional semantics Proc. 23rd Int. Conf. on Computational Linguistics (Beijing: Association for Computational Linguistics) pp 1263-71
[11] Yessenalina A and Cardie C 2011 Compositional matrix-space models for sentiment analysis Proc. Conf. on Empirical Methods in Natural Language Processing (Edinburgh: Association for Computational Linguistics) pp 172-82
[12] Mikolov T, Sutskever I, Chen K, Corrado G S and Dean J 2013 Distributed representations of words and phrases and their compositionality In Advances in neural information processing systems 26 3111-9
[13] Mikolov T, Chen K, Corrado G S and Dean J 2013 Efficient estimation of word representations in vector space J. Comput. Sci.
[14] Le Q and Mikolov T 2014 Distributed representations of sentences and documents Int. Conf. on Maching Learning pp 1188-96
[15] Selvi J, Rodriguez R J and Soria-Olivas E 2019 Detection of algorithmically generated malicious domain names using masked N-grams J. Expert Syst. Appl. 124 156-63
[16] Yang L H, Liu G J, Zhai J T, Liu W W, Bai H W and Dai Y W 2020 Improved algorithm for detection of the malicious domain name based on the convolutional neural network J. J Xidian University 47 37-43
[17] Huang S Q, Zhang D M and Yan B 2019 A detection algorithm of malicious domain based on LSTM and Multi-Head attention Mechanism J. Computer Engineering & Software 40 83-90
[18] Zhou K, Wan L and Ding H W 2020 Detection of malicious domain names based on AN and LSTM J. Computer Engineering and Applications 56 92-8