Comparison of two-classification models based on neural network for DGA domain name detection

Zhiping Li, Jian Chen, Rui Pan*
China Academy of Information and Communications Technology, Beijing, 100191, China

ABSTRACT

Many malware families could generate a huge number of pseudo-random domain names through DGAs (Domain Generation Algorithm). Using DGA domain name to take DDoS (Distributed Denial of Service) attacks makes network defenses more difficult. So detection of DGA domain name has become an important research in network security, and methods based on neural network have been explored. By extracting different character features of domain name in character-level word embedding, this paper compared the performance between CNN (Convolutional Neural Network) and Bi-LSTM (Bi-Directional Long Short-Term Memory) in two-classification of DGA domain name. Experiment results show that using character features including semantic features could improve the performance of neural network, and there is little difference between CNN and Bi-LSTM in DGA domain name detection.

Keywords: DGA domain name, CNN, Bi-LSTM, word embedding, semantic feature

1. INTRODUCTION

DNS (Domain Name System), which achieves the conversion between IP address and domain name, is the infrastructure of the Internet. With the development of the digital economy, the safety of DNS becomes more and more important. Meanwhile, many malware families use DGA (Domain Generation Algorithm) domain names to take DDoS (Distributed Denial of Service) attacks, which threatens to network security and even economy development[1].

A large number of pseudo-random domain names (hundreds to tens of thousands per day) could be generated through DGAs, which makes network defenses difficult[2-3]. So detection of DGA domain name has become an important research in network security.

Domain name is similar to natural language text. Zhang et al discovered the pseudo-randomness of DGA domain names[4]. Koh et al proposed a novel approach that combined context-sensitive word embeddings with a simple fully-connected classifier to perform classification of domain names based on word-level information[5].

Word-level word embedding needs to be pre-trained on a large unrelated corpus, which makes the model training time longer. Character-level word embedding could reduce the dependence on training data. Pan et al proved that using semantic features in Bi-LSTM (Bi-Directional Long Short-Term Memory) could improve the detection performance in multi-classification[6].

With DGAs, network attacks are developing rapidly. Deep learning algorithms have the advantage of automatic feature extraction, and methods based on neural network have been explored[7]. Saxe et al proposed the eXpose neural network, which could extract features and make classification through character-level embeddings simultaneously[8]. Shahzad et al presented a DGA classifier that leveraged a structure based on RNN (Recurrent Neural Network), which could detect DGA domain name without contextual information or manually created features[9].

By using different character features in word embedding, this paper took the comparison of two-classification models based on neural network for detecting DGA domain name.

2. TWO-CLASSIFICATION MODELS

Two-classification methods are shown in Figures 1-2. which consists of word embedding layer, neural network layer, and

* panrui@caict.ac.cn

Third International Conference on Computer Science and Communication Technology (ICCSCT 2022)
edited by Yingfa Lu, Changbo Cheng, Proc. of SPIE Vol. 12506, 125062N
 · 0277-786X · doi: 10.1117/12.2661832

two-classification layer.

First, character features of domain name are extracted and mapped into word vector. Then, Bi-LSTM or CNN (Convolutional Neural Network) is employed to make automatic feature extraction. Finally, a fully connected neural network is used to make two-classification[10].

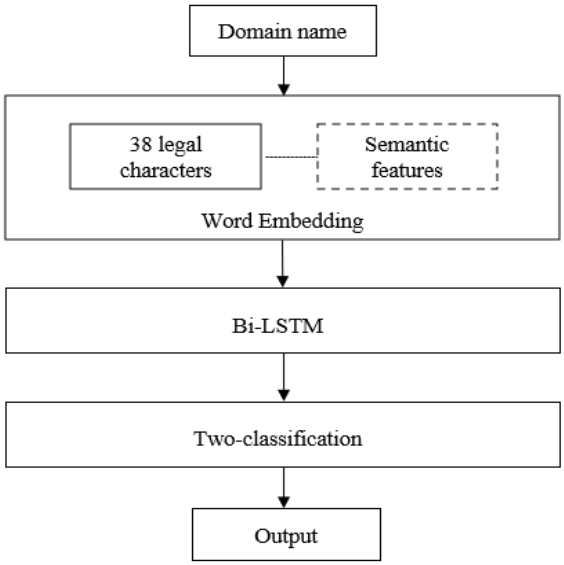

Figure 1. Bi-LSTM model.

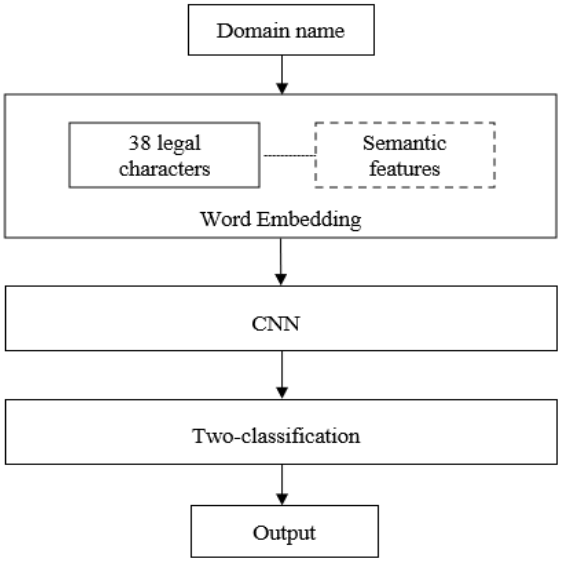

Figure 2. CNN model

2.1 Word embedding

As deep learning algorithm cannot directly process text, domain names need to be transformed into numeric data. Word embedding is a processing method in NLP (Natural Language Processing), and the model takes character-level word embedding.

Domain name is composed of 38 legal characters (a-z, 0-9, "-", "."), which could be used as character features in word embedding. After removing Top Level Domain (such as ".com", ".net" and ".org"), domain name could be divided into single character.

Domain name is also a kind of short text. And semantic features of domain name, which include bigram class and part of speech (such as noun class, verb class, adjective class and other part of speech), could be extracted to extend character features[6].

Through one-hot encoding, character features could be transformed into feature sequence, then mapped into word vector. The word vector is represented by a $n \times L$ dimensional matrix, where n is the length of feature sequence, and L is the dimension of word vector[11].

2.2 Bi-directional long short-term memory

Bi-LSTM is a variant of RNN models and has been widely used in text analysis. Bi-LSTM consists backward and forward hidden layers to access the preceding and succeeding context of sequence, which also improves the contexts available to the network[12].

There are two independent LSTM in Bi-LSTM. x_t is the input, and h_t is based on the outputs of forward LSTM and backward LSTM. The structure of forward LSTM and backward LSTM are the same, as shown in Figure 3.

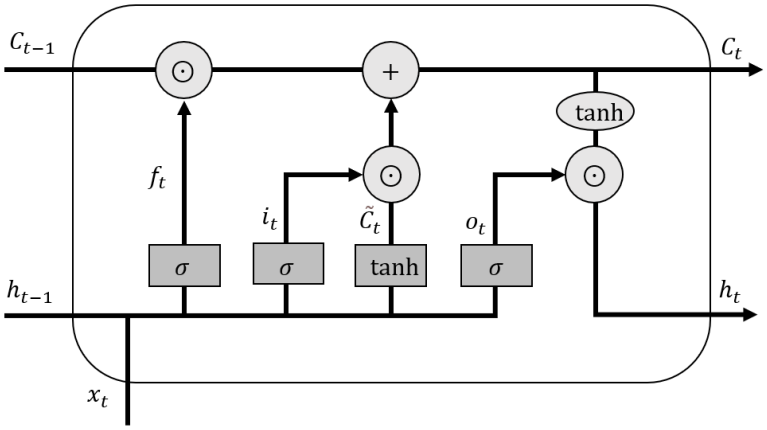

Figure 3. LSTM Structure.

When new information is added, some old information needs to be forgot through forget gate *f*. Output of f_t is between 0 and 1, where 0 means "completely discarding" and 1 means "completely keeping". i_t is input gate, which is used to decide what information needs to be updated. o_t is output gate, and a sigmoid function is employed to determine the output.

$\tilde{C}_t$ is the potential updated content, C_t and h_t could be updated as follows:

$$\tilde{C}_t = \tanh(W_C \cdot (h_{t-1}, x_t) + b_C) \tag{1}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{2}$$

$$h_t = o_t \odot \tanh(C_t) \tag{3}$$

2.3 Convolutional neural network

Through word embedding, domain names are mapped from a sparse, 1-of-V encoding (here V is the vocabulary size) to a lower dimensional vector space. In essence, word vector is feature extractors that encode character features of domain names in their dimensions[13].

CNN is widely used not only in image processing and speech recognition, but also in NLP. As shown in Figure 4, CNN consists of convolution layer, pooling layer and concatenation layer. In this paper, four parallel convolution layers are used to extract features.

CNN, which is also a kind of the standard neural network, could extract features and reduce the dimensions of the input. The convolution kernel *w* will perform a convolution operation with the input matrix each time to obtain local features.

$$z_i = f(w \cdot x_{i:i+k-1} + b), \quad i \in \{1, 2, \cdots, n-k+1\} \tag{4}$$

$$Z = [z_1, z_2, \cdots, z_{n-k+1}] \tag{5}$$

where *f* is the nonlinear activation function, x_i is word vector, *k* is the size of the convolution kernel, *b* is bias term and *n* is the length of feature sequence[14].

Maximum pooling is employed to capture the most obvious feature. Through pooling layer, feature invariance could be enhanced and the dimension of data could be reduced.

Finally, outputs of four pooling layers are concatenated into a vector in concatenation layer.

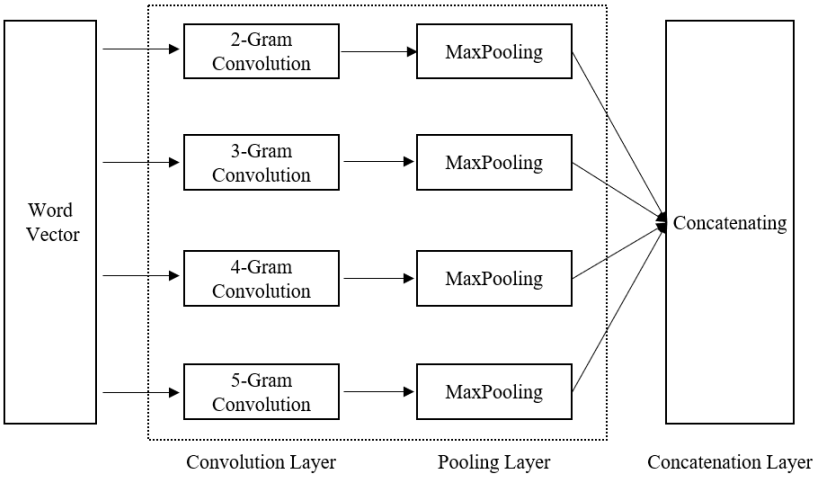

Figure 4. Parallel Convolutional Neural Network Structure.

4. EXPERIMENT AND ANALYSIS

4.1 Experiment design

Data used in the experiment came from publicly resources such as Alexa and 360 Netlab (Network Security Research Lab). There were one million normal domain names and one million DGA domain names. Dataset was divided into training set, test set and validation set by 6:2:2.

According to detection models in chapter 2, four experiment groups were designed to compare the performance between CNN and Bi-LSTM in detection of DGA domain name.

- Experiment group that using 38 legal characters as character features in Bi-LSTM was labeled by Bi-LSTM (CF).
- Experiment group that using character features including 38 legal characters and semantic features in Bi-LSTM was labeled by Bi-LSTM (SF).
- Experiment group that using 38 legal characters as character features in CNN was labeled by CNN (CF).
- Experiment group that using character features including 38 legal characters and semantic features in CNN was labeled by CNN (SF).

4.2 Experiment settings

According to parameters mentioned in chapter 2, settings for word embedding were as follows:

- n is the length of feature sequence and should be longer than the length of each domain name in dataset, and experiment set n to 100.
- L is the dimension of word vector, and experiment set L to 100.

In neural network. The number of neurons used in Bi-LSTM was 128, and settings for CNN were as follows:

- Convolutional layer. ReLU (Rectified Linear Unit) was used as the activation function. k, which is the size of the convolution kernel, were 2, 3, 4, 5, respectively. d, which is the same as L, was set to 100.
- Pooling layer. Slide size was 1.
- Concatenation layer. Outputs of four pooling layers could be concatenated into a vector with the length of 1024.

4.3 Experiment results and analysis

In two-classification, precision, recall and F1 score were evaluation indicators. Experiment results were given in Table 1, where 0 represented normal domain name and 1 represented DGA domain name.

CNN (SF) performed better than CNN (CF) and Bi-LSTM (SF) performed better than Bi-LSTM (CF), showing that with the use of semantic features, the performance of CNN and Bi-LSTM in two-classification was improved.

F1 score of Bi-LSTM (SF) and CNN (SF) were almost the same, which were 0.9932 and 0.9938, respectively. And precision of Bi-LSTM (SF) and CNN (SF) were almost the same also, which were 0.9947 and 0.9955, respectively. The results indicated that in two-classification, the detection effect of CNN model was similar to Bi-LSTM model.

Table 1. Precision, Recall and F1 score in two-classification.

Experiment Group	Classification	Precision	Recall	F1 score	Support
Bi-LSTM (SF)	0	0.9918	0.9948	0.9933	200,000
	1	0.9947	0.9917	0.9932	200,000
Bi-LSTM (CF)	0	0.9898	0.9942	0.9920	200,000
	1	0.9941	0.9897	0.9919	200,000
CNN (SF)	0	0.9921	0.9955	0.9938	200,000
	1	0.9955	0.9921	0.9938	200,000
CNN (CF)	0	0.9912	0.9952	0.9932	200,000
	1	0.9952	0.9912	0.9932	200,000

5. CONCLUSION

Detection of DGA domain name has become an important research in network security in recent years. As character-level word embedding could reduce the dependence on training data and deep learning algorithms have the advantage of automatic feature extraction. This paper used different character features in word embedding, and compared the performance between CNN and Bi-LSTM for detecting DGA domain name.

Four experiment groups were designed to compare the performance between CNN and Bi-LSTM in detection of DGA domain name. And this paper used normal domain names and DGA domain names from Alexa and 360 Netlab as datasets. Experiment on publicly datasets has shown that by using semantic features, F1 score of Bi-LSTM increased by 1.3%, and the gap of F1 score between CNN and Bi-LSTM had narrowed to 0.6%.

In summary, using character features including semantic features could improve the performance of neural network, especially for Bi-LSTM. And F1 score of CNN and Bi-LSTM are both nearly 100%, so there is little difference between CNN model and Bi-LSTM model in two-classification of DGA domain name.

ACKNOWLEDGMENTS

This work is supported by 2020 Industrial Internet Innovation and Development Project: Network Identifier Construction Project.

REFERENCES

[1] Ghosh, I., Kumar, S., Bhatia, A., et al., “Using auxiliary inputs in deep learning models for detecting DGA-based domain names,” 2021 International Conference on Information Networking, 391-396(2021).
[2] Sivaguru, R., Peck, J., Olumofin, F., et al., “Inline detection of DGA domains using side information,” IEEE Access, 8, 141910-141922(2020).
[3] Woodbridge, J., Anderson, H., Ahuja, A. et al., “Predicting domain generation algorithms with long short-term memory networks,” arXiv:1611.00791, (2016).
[4] Zhang, Y., “Automatic algorithmically generated domain detection with deep learning methods, “2020 IEEE 3rd International

Conference on Automation, Electronics and Electrical Engineering, 463-469(2020).
[5] Koh, J. and Rhodes, B., "Inline detection of domain generation algorithms with context-sensitive word embeddings," 2018 IEEE International Conference on Big Data, 2966-2971(2018).
[6] Pan, R., Chen, J., Ma, H. et al., "Using extended character feature in Bi-LSTM for DGA domain name detection," IEEE/ACIS 22nd International Conference on Computer and Information, 115-118(2022).
[7] Yu, B., Pan, J., Gray, D., et al., "Weakly supervised deep learning for the detection of domain generation algorithms," IEEE Access, 7, 51542-51556(2019).
[8] Saxe, J. and Berlin, K., "eXpose: A character-level convolutional neural network with embeddings for detecting malicious urls, file paths and registry keys," arXiv:1702.08568, (2017).
[9] Shahzad, H., Satta, A. and Skandaraniyam, J., "DGA domain detection using deep learning," 2021 IEEE 5th International Conference on Cryptography, Security and Privacy, 139-143(2021).
[10] Wang, Z., Li, S., Chi, Y., et al., "Deep learning based detection of DGA domain names," Computer Engineering and Design, 42, 601-606(2021). (in Chinese)
[11] Du, P. and Ding, F., "A DGA domain name detection method based on deep learning models with mixed word embedding," Journal of Computer Research and Development, 57, 433-446(2020). (in Chinese)
[12] Tam, S., Said, B. R. and Tanriöver, Ö., "A ConvBiLSTM deep learning model-based approach for twitter sentiment classification," IEEE Access, 9, 41283-41293(2021).
[13] Kim, Y., "Convolutional neural networks for sentence classification," arXiv:1408.5882, (2014).
[14] Yang, K., Huang, Z., Wang, X. and Li, X., "A blind spectrum sensing method based on deep learning," Sensors, 19(20), 2270(2019).