MALICIOUS URL RECOGNITION AND DETECTION USING ATTENTION-BASED CNN-LSTM

Yongfang Peng[1], Shengwei Tian[1*], Long Yu[2], Yalong Lv[3], Ruijin Wang[4]
[1]School of Software, Xinjiang University
Urumqi, 830008, China,
[e-mail: m13999412597@163.com]
[2]Network Center, Xinjiang University
Urumqi, 830046, China
[e-mail: yul_xju@163.com]
[3]College of Information Science and Engineering, Xinjiang University
Urumqi, 830008, China,
[e-mail: 2029854022@qq.com]
[4]School of Computer Science and Engineering, University of Electronic Science and Technology of China
Chengdu, 611731, China
[e-mail: 438947064@qq.com]
*Corresponding Author: Shengwei Tian

Received December 23, 2018; revised February 10, 2019; revised March 8, 2019; accepted May 15, 2019; published November 30, 2019

Abstract

A malicious Uniform Resource Locator (URL) recognition and detection method based on the combination of Attention mechanism with Convolutional Neural Network and Long Short-Term Memory Network (Attention-Based CNN-LSTM), is proposed. Firstly, the WHOIS check method is used to extract and filter features, including the URL texture information, the URL string statistical information of attributes and the WHOIS information, and the features are subsequently encoded and pre-processed followed by inputting them to the constructed Convolutional Neural Network (CNN) convolution layer to extract local features. Secondly, in accordance with the weights from the Attention mechanism, the generated local features are input into the Long-Short Term Memory (LSTM) model, and subsequently pooled to calculate the global features of the URLs. Finally, the URLs are detected and classified by the SoftMax function using global features. The results demonstrate that compared with the existing methods, the Attention-based CNN-LSTM mechanism has higher accuracy for malicious URL detection.

Keywords: Malicious URL; Recognition and Detection; Attention-Based CNN-LSTM; Deep Learning

1. Introduction

With the rapid development of Internet technology, maintaining the Network security becomes more and more important. Network attackers make use of phishing websites, hacker attacks, malicious attacks, exploits and other new technologies [1]-[5]to attack and deceive users,which makes it more difficult to detect malicious Uniform Resource Locators (URL). For example, rogue websites leak user's sensitive information, resulting in property loss or personal information being stolen, or even install malware in the users' system to implement financial fraud, causing tremendous property losses to the users and leading to great confusion to the state management. Therefore, how to use the existing technology of feature extraction and deep learning to effectively identify malicious URL has become a research hotspot.

For malicious URL recognition and detection issues, therehave been a lot of studies. There was a method based on the blacklist [6], which is a list containing malicious URLs already marked out, IP addresses or keywords. Sahoo D et al.[7] used the method to identify and detect malicious URLs, mainly by looking up the URL blacklist to judge whether a URL to be detected is malicious. Provided that it is marked in the list, the URL is malicious, otherwise it is benign. Thanks to the blacklist technology, individuals can simply and accurately detect malicious URLs that have been identified, thereby lowering the error rate. Nevertheless, the method can merely spot malicious URLs that have been found and it is not suitable for others, consequently it can easily cause mistakes. In order to improve the situation of missed judgments, Prakash Pawan et al.[8] proposed a method named Phishnet for blacklisting technology.

As the blacklist mechanism have disadvantages of false judgments, the researchers designed and implemented a method for malicious URL recognition and detection based on heuristic rules, detailedly, on the correlation among malicious URLs. In 2007, Zhang Yue et al. [1] analyzed the relevance of URLs by using the classical algorithm named Term Frequency–Inverse Document Frequency (TF-IDF) and got results and other statistical information. The method does not need to know the malicious URL and other information in advance, it can spot some unrecognized malicious URLs according to the existing rules. Therefore, the fuzzy matching technology based on heuristic rules greatly lowers the mis-judgment rate. However, heuristic rules are obtained by statistical analysis of existing malicious URLs or manual summarization and these rules depend on the knowledge of corresponding fields, so it is difficult to update.

Due to the high rate of false judgement of the blacklist method and the difficulty of updating the heuristic rules, the researchers further proposed a more systematic method based on machine learning for identifying and detecting malicious URLs[9]-[13]. Machine learning can be broadly divided into supervised learning, unsupervised learning and semi-supervised learning, using a lot of tagged URL samples as the trained set to acquire the ability of prediction. In 2010, Liu Gang et al.[14]used the Hyperlinks and the sorting relationship of keywords as statistical features to identify malicious URL attacks using Density-Based Spatial Clustering Application with Noise (DBSCAN), one of the machine learning algorithms. Ma et al.[15]-[17] used machine learning algorithms to classify and detect malicious URLs based on DNS information, WHOIS information, and URL grammar features. Although machine learning methods generalize well, one potential drawback of these methods for malicious URL detection is their instensive resourcesespecially when extracting non-trivial and computationally expensive features.

In view of the current exponential growth trend of the importance of cybersecurity to human beings, it is imperative to propose a more accurate method for detecting malicious URLs. By mining the texture information, the string statistical information of attributes (SSIA) and WHOIS information hid in the malicious URLs, this article uses Attention-Based CNN-LSTM to effectively identify malicious URLs and improve the accuracy rate.

2. The algorithm model

2.1 Attention build

Attention is a mechanism that can flexibly select context information and use it as a reference [18]-[19]. It was first proposed in machine translation [20] to solve the encoder-decoder problem, which implies all necessary information should be compressed as the encoding vector with the length being X. In essence, attention is a measure of similarity. If the accuracy in identifying a target depends on the input greatly, the weights will be larger and the output depends on the current input largely.

In this paper, the attention mechanism is introduced in the expression of the malicious URL feature, which can highlight the importance of key features to detect malicious URLs. Assume that x^i represents an URL feature vector, $x^i \in R^{n\times d}$ (n represents the amount of the URL data, d represents the dimension of the feature vector), a_i represents the importance of different features in the URL. The vector a_i is weighted average of the n pieces of data, and the larger a_i is, the more important this feature is in detecting malicious URLs. The feature vector x^i is input into the attention mechanism and the output value y^i is obtained by automatical weight a_i:

$$y^i = \sum_i^n a_i h_i . \tag{1}$$

where,

$$a_i = \frac{\exp[G(h_i, h_k)]}{\sum_{i=0}^{n} \exp[G(h_j, h_k)]} . \tag{2}$$

where, G is a function to calculate the importance of x^i.

$$G(h_i, h_k) = A^T \cdot \tanh(V_\alpha \cdot h_i + V_\beta \cdot h_k + b) . \tag{3}$$

where, V_α and V_β represent the parameter matrix, b represents the offset matrix, A represents the parameter vector,and A^T represents the transposition of the parameter vector.

Through the attention model, the URL characteristic valuesare obtained. The features which have greater impact on the URL classification are selected, and then are granted with larger weightto participate in the next step of calculation. Therefore, we can obtain more critical URL features as input for malicious URL detection. The illustration of the attention framework is shown in **Fig. 1**:

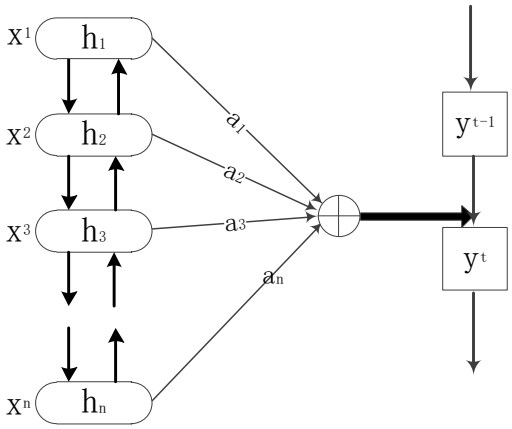


Fig. 1. Attention framework diagram

2.2 Attention-Based CNN-LSTM construction

In this paper, the attentional weight can be used to select the malicious URL features with high correlation. Therefore, the attention-based CNN-LSTM model is composed of the attention mechanism and the CNN-LSTM model.The model is imported the results of local features extracted by CNN into LSTM model, and maximizes pooling processing [21]-[22] to obtain global features. Last, the Softmax classifier is used. The specific network structure of the model is shown in **Fig. 2**.

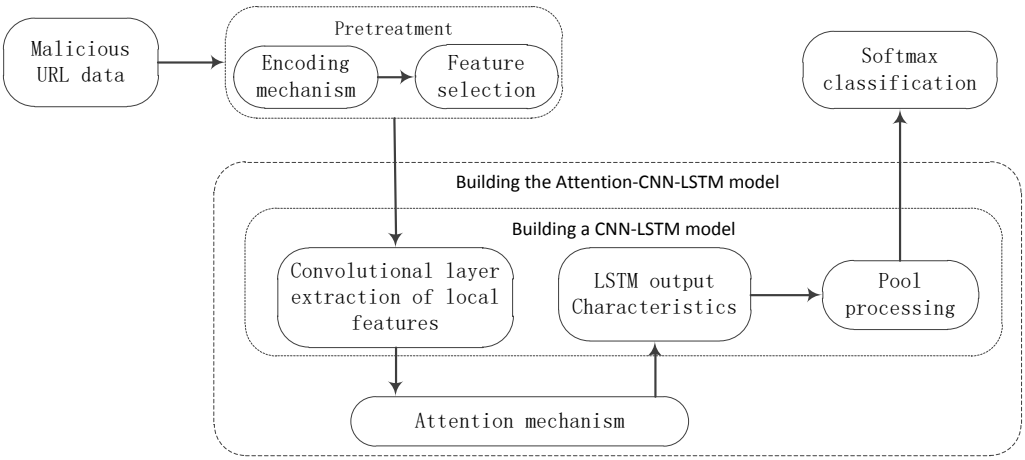


Fig. 2. Network structure of the Attention-Based CNN-LSTM model

The URL dataset is used to obtain n-dimensional URL features by WHOIS check. First, CNN is used to extract local features. Assuming that h_t is the t-th convolutional layer, then

$$h_t = g(h_{t-1} \otimes \omega_t + e_t). \tag{4}$$

Where ω_t represents the weight vector of convolution kernel of the t-th layer; the operation symbol "$\otimes$" represents the convolution operation of the convolution kernel and the

output of the convolution is added to the offset vector e_t of the t-th layer. Finally, through the nonlinear activation function $g(x)$, h_t is obtained.

Assume that the matrix corresponding to the i-th URL and its features can be represented as $R^{x^i \times x^i_j}$, $|x^i|$ represents the i-th URL, $|x^i_j|$ represents the j-th feature of i-th URL, the i-th row in the matrix represents the characteristic value corresponding to the i-th URL. After processing the features through CNN, the input sequences $H = \{h_1, h_2, h_3, \cdots, h_n\}$ are obtained, the weight a_i averaged using the Attention mechanism and the matrix calculated by the formula (1) is added into the LSTM thereby, an output feature is obtained. Among them, I_i, F_i, O_i represent the three mechanisms of Input, Forget, and Output in LSTM respectively. The specific addition method is described in the following formulas:

$$I_i = f(w_{aI} \cdot y_i + w_{bI} \cdot h_{i-1} + b_I). \tag{5}$$

$$F_i = f(w_{aF} \cdot y_i + w_{bF} \cdot h_{i-1} + b_F). \tag{6}$$

$$O_i = f(w_{aO} \cdot y_i + w_{bO} \cdot h_{i-1} + b_O). \tag{7}$$

$$P_i = \tanh(w_{ac} \cdot y_i + w_{bc} \cdot h_{i-1} + b_c). \tag{8}$$

Where P represents a sigmoid activation function, and the input transform:

In summary, the specific algorithm steps of the Attention-Based CNN-LSTM model are as follows:

① Use the WHOIS check method to extract URL features;

②Obtain the feature vector x^i through the encoding mechanism;

③ Drop out the feature vector x^i and inputting it into the model (Attention-Based CNN-LSTM) to extract local features according to Eq. (4);

④Calculate the weight a_i according to Eq. (2) and calculating the output characteristics of the LSTM model by using the Eq. (5), (6), (7), and (8);

⑤ Pool the output features of the model and calculating the global features of the URL waited to be identified;

⑥ The classification result is obtained by activating the function of softmax, the result is equal to 1 for a malicious URL and the result is equal to 0 for a benign one.

3. Feature extraction

3.1 URL overview

A URL represents a unified address of a network resource. The URL has two main components:

(I) Protocol identifier, which indicates the protocol to use. The most common mode is the Hypertext Transfer Protocol (HTTP), which can be available to the network. Other agreements are shown in **Table 1**:

Table 1. Protocol names and their meanings

Protocol Name	Protocol Meaning
http	Hypertext transfer protocol resourcesPlanning
https	Hypertext Transfer Protocol with Secure Sockets Layer Transport
ftp	File Transfer Protocol
mailto	Email address
ldap	Lightweight Directory Access Protocol Search
file	Local computers or files shared online
news	Usenet News Group
gopher	Gopher protocol
telnet	Telnet protocol

(II) Resource name refers to the name or IP address of the server, the latter being the path to the file and the name of the file itself. The server's name or IP address is sometimes followed by a colon and a port number. It can also include the user name and password of the contacted server.

3.2 Feature selection

In the actual application of machine learning, the number of features is always large, among which may exist always irrelevant features or relevant ones, so the number of features easily leads to the following phenomenon:

(I)The larger the number of features is, the longer time it takes to analyze features and train the model.

(II)The larger the number of features is, the more likely it is to cause ‘dimensionaldisasters’, the more complex the model will be and its ability to popularizewill decline.Feature selection can eliminate irrelevant or redundant features, thereby reducing the number of features, improving the accuracy rate of the model and declining the running time. On the other hand, the selection of truly relevant features can simplify the model and make it easier for researchers to understand the process of data generation. Therefore, the feature selection can improve the accuracy of detecting malicious URLs to a certain extent.

3.3 Feature extraction

The most important part of malicious URL recognition and detection is the extraction of malicious URL features. The selection of features is directly related to the accuracy of classification.

3.3.1 Texture feature extraction

For the same type of malicious URLs, there are certain similarities in the texture. Each byte in the .csv file expressed in hexadecimal is in the middle of 00-FF, which can correspond to the gray value 0-255. Malicious URL classification strategy based on the machine learning displayed the texture features in grayscale [23]-[24].The analysis of malicious URL texture features is shown in **Fig. 3**:

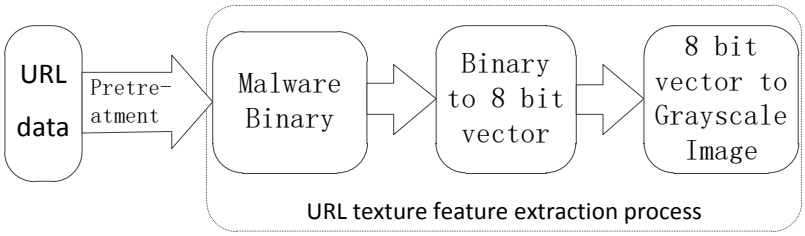


Fig. 3. Analysis of malicious URL texture features

3.3.2 URL string statistics information of attributes

The URL contains a lot of information, some of which can be used as the basis for malicious URL classification. The purpose of URL analysis is to find out the features that are useful for classification. The information contained in the URL includes the length of URL and whether the URL contains some character strings, so this information is called the URL string statistical information of attributes. The basic data of this experiment is obtained in advance from the open source website PhishTank and through web crawlers. It obtains relevant information coming from URL analysis and related research, including the total length of the URL, the number of URL letters, the number of URL digits, DNS, URL Date and so on. According to the URL standard specification [25] and the observation and statistics done by Lin Hailun et al.[26], it can be found that a malicious URL has the following characteristics:

Table 2. URL SSIA extraction rules

URL SSIA	Meaning
FT1	The number of "." in the URL
FT2	The total length of the URL
FT3	The number of uppercase English letters in the URL
FT4	The number of lowercase English letters in the URL
FT5	The number of Arabic numerals in the URL
FT6	The number of special characters ("#", "@", "_", "&", etc.) in the URL
FT7	The proportion of uppercase English letters in the URL to the total length of the URL
FT8	The proportion of lowercase English letters in the URL to the total length of the URL
FT9	The proportion of Arabic digits in the URL to the total length of the URL
FT10	The proportion of special characters in the URL to the total length of the URL
FT11	Does it contain an IP address
FT12	Whether the top-level domain is the top five domains (com,cn, net, org, cc)

In addition to the main string statistical information of attributes listed in **Table 2**, there are such things as the number of the separator "/", the maximum length of the character between the separator "/", the largest length of continuous number, the largest length of consecutive letter, the conversion frequency of number and letter, the conversion frequency of uppercase and lowercase letters and the significant coefficients in the primary domain name. The string statistical information of attributes obtained by extracting the URL, which can be used as one of the bases for detecting a malicious URL.

3.3.3 WHOIS information features

WHOIS is a database which helps determine whether a domain name has already been registered and contains the relevant details such as domain name owner, domain name registrar, domain registration date and expiration date, etc. if it has. It is available to obtain

WHOIS information and extract key features such as registrant's name, phone number, and email by probing WHOIS basis of the domain name. Through the WHOIS check, it is possible to know how many domain name and sites the registrar owns, the registration information of the domain name usually includes the domain name, domain name registrant, domain name registrar, telephone number, and email. **Table 3** provides a brief description of the characteristics and meanings of URLsin the WHOIS check.

Table 3. Features and meanings obtained from WHOIS check

Feature name	Meaning
Registrar	A business entity or organization.
Registrants	Registered domain names for individual objects.
Email	The mailbox used to register the domain name.
Domain Name Server(DNS)	Used to translate domain names into computer-readable IP addresses.
Registration time	The earliest possible starting time for registering a domain name.
Expire date	The time when the registered domain name can be used up to

The above key information extracted by WHOIS informationcan be used as the features for malicious URL detection.

4. Experimental evaluation

4.1 Experimental environment and experimental data sources

The experimental hardware andsoftware environment is shown in **Table 4** and **Table 5**:

Table 4. Hardware environment

Name	Value
Operating System	Microsoft Windows 7 Ultimate (64-bit / Service Pack 1)
CPU	AMD E2-3800 APU withRadeon(TM) HD Graphics
CPU frequency	1.30GHz
CPU cores	4 Nuclear
Memory (RAM)	4.00GB
Storage	500 GB

Table 5. Software environment

Name	Function	Version
Anaconda	Scientific computing package (theano, pandas, TensorFlow, sklearn, numpy, etc.)	3-5.1.0
PyCharm	Building deep learning models environment	2017.3.3

In this experiment, the dataset is from PhishTank, a well-known open source website. It contains 16055 malicious URLs. In addition, we crawlednormal URLsfrom some popular websites and obtained 12091 items after pre-processing.

4.2 Experimental results and analysis

This experiment analyzes the influence of the variable on the experimental results from two aspects: model parameters and different models. Randomly select 17,000 experimental data to determine variable parameters (number of features, number of iterations, etc.) through multiple sets of experiments, randomly select 80% of the samples as training data, 20% as test

ones, and adopt a 10 fold cross-validation method, then use uniform performance indicators to predict accuracy (Accuracy) Eq. (9) andloss rate (Loss) Eq. (10) in order to evaluate the performance of the model.

$$Accurary = \frac{Q_{positive}}{Q_{total}}. \tag{9}$$

In equation (9), $Q_{positive}$ indicates the number of samples correctly classified by the model, and Q_{total} indicates the total number of experimental samples.

$$Loss = \frac{1}{n}\sum_{i=1}^{n} l_i \log \hat{l}_i + (1-l_i)\log\left(1-\hat{l}_i\right). \tag{10}$$

In equation (9), n indicates the number of samples, l_i indicates i-th sample actual category, and $\hat{l}_i$ indicates i-th sample predicted category.

4.2.1 The effect of model parameters on experimental results

The setting of experimental parameters has important significance to the overall performance of the model. The irrationality of setting parameters will affect the effective use of features, consequently, affect the recognition and detection of malicious URLs. Therefore, in order to optimize the experimental results, on the same dataset, the variable parameters are tested and the optimal parameters are determined according to the prediction accuracy, including the setting of variable parameters such as the type of features, vector dimensions and the number of iterations.

(1) Influence of feature types on experimental results

Table 6. Experimental result of different characteristic types

Feature	SSIA	Texture	Time	DNS1	DNS2	Registrar	Registrant	Email
Accuracy (%)	96.74	80.85	78.00	80.97	80.35	80.12	76.09	76.00
Loss (%)	10.46	42.56	50.59	42.84	42.79	42.36	49.21	51.73

(2) The effect of vector dimensions on experimental results

Table 7. Experimental results of vector dimensions

vector dimensions	224	217	187	157	127	97	67	66	55	21
Accuracy (%)	97.59	97.79	97.88	97.79	98.00	98.18	97.62	97.68	93.82	91.71
Loss (%)	6.97	6.59	6.97	6.65	6.81	6.16	7.01	7.63	15.75	19.86

In order to test the accuracy of data feature for malicious URLs detection, this section experiment on the condition of the same dataset (17,000 data is selected, in which the number of malicious URL data and benign URL data is 8500 respectively), verify the impact on the experimental results from feature types of the data characteristics and vector dimensions.

As is shown in **Table 6**, because the URL statistical information of attributes is the most basic feature of the URL, it has the highest accuracy for detecting malicious URLs. In addition to the URL statistical information of attributes, the DNS feature embraces obvious effect for detecting malicious URLs. Therefore, selecting a sample type that has a higher accuracy rate for detecting a malicious URL further verifies the effect of the URL feature on the experimental result.

It can be seen from **Table 7** that when the number of URL features is 97, the accuracy of the experimental results reaches a maximum of 98.18% and when the number of URL features increases or decreases, the accuracy of the experimental results is sligtly lower than that. This experiment indicates that the phenomenon of over-fitting and dimensional catastrophe occurs when the number of URL features is too large, and this is also verified by the observation of the experimental process.

As is shown in **Table 7**, when the vector dimension is gradually increased, the classification effect is improved. It can be seen that the model fail to fully learn the higher dimensional feature informations of the URL during the training process. When the vector dimension is 97, the accuracy of the experimental results maximally reaches 98.18%, achieving optimal classification effect. However, with the gradual further increasement of the vector dimension, the classification effect falls behind. This demonstrates that when the vector dimensions hits a certain threshold, it will cause characteristic information that cannot fully express the URL and over-fitting phenomenon, resulting in fluctuation of the classification results. Reasonable selection of vector dimensions has a significant impact on the results, so in this paper the number of 97 is selected as the dimension of URL detection.

(3) The effect that the number of iteration has on experimental results

Table 8. Experimental results of the number of iteration

Iterations	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**
Accuracy (%)	97.71	97.82	97.97	97.59	97.76	98.18	97.82	97.65	98.00	97.79
Loss (%)	6.98	6.24	6.18	6.67	7.07	6.16	6.45	6.96	7.36	7.84

It can be seen from **Table 8** that when the number of iterations of the experiment is 18, the accuracy of the experimental results reaches a maximum of 98.18%; It also reached 98% for 12 and 24, and there are troughs between the number of iterations 12 and 18, 18 and 24, but the lowest accuracy of the experimental results also reached 97.59%. When the experimental model learns the number of iterations overtraining, the noise in the training data and the non-representative features in the training examples are fitted. Therefore, according to the accuracy and loss rate of the experimental results in this section, experimental model's the number of iteration is set to 18 as the optimal experimental parameter.

4.2.2 Comparison with other models

In order to further verify that the proposed method has better recognition and detection effect than the traditional methods, the model detection results proposed in this paper are compared with the results of shallow Random Forest (RF), Gaussian Bayesian (GaussianNBO), deep Long Short-Term Memory Neural Network (LSTM) and Convolutional Neural Networks (CNN), while the other neural networks have the same network topology as the Attention-Based CNN-LSTM model. The different models are compared with each other under the precondition of same variables, including the setting of hyperparameters such as the number of experimental sample, the ratio of training samples to test samples, the ratio of benign sample numbers to malicious sample numbers, the type of features, the votor dimensions and the number of iterations. The specific experimental results are shown in **Table 9**:

Table 9. Comparison of experimental data for every feature of different models

Feature/ Accuracy (%)	Attribute-Information	Texture	DNS1	Registrar	Time	URL Overall-Features
CNN	96.38	79.68	77.29	79.68	77.47	96.74
LSTM	96.06	79.18	74.09	78.56	77.18	96.71
Attention-Based CNN-LSTM	96.74	80.85	80.97	80.12	78.00	98.18
RF	94.19	69.88	76.06	79.37	70.40	93.35
GaussianNBO	96.38	62.91	64.09	67.16	66.15	94.91

The accuracy of the recognition and detection of the URL string statistical information of attributes by each model is close to the accuracy of the recognition and detection of the overall features, indicating that the URL string information feature contributes the most to the malicious URL recognition and detection and the second is the texture information feature. For the texture information feature, because the deep neural network can map similar URLs to the nearest neighbor distance of the texture feature vector and establish texture similarity matching deeply, the effect of deep neural network on malicious URL recognition and detection is obviously better than that the shallow makes.

The deep neural network model has better effect on the recognition and detection of the overall features than the individual feature detection. For the shallow neural network, the use of URL string statistical information of attributes to identify and detect malicious URLs is better than the overall feature recognition and detection. It shows that when the vector dimension increases, the deep neural network can deal with the feature processing of higher dimensions while the shallow neural network is prone to over-fitting.

Under the same characteristics, because of the outstanding feature learning ability of the deep neural network model, deeper hidden features can be excavated and the feature representation ability is enhanced. Therefore, the effect of the recognition and detection of the deep neural network model is better than the shallow and [27]-[28]; in the deep neural network model, the Attention-Based CNN-LSTM introduces attention mechanism. Because the accuracy of the current URL feature input and target detection in the attention mechanism is higher, the input of the current feature will be assigned with larger weights. Attention-Based CNN-LSTM will prioritize the characteristics that have larger weights to identify and detect the target. Therefore, the Attention-Based CNN-LSTM recognition and detection is generally better than CNN and LSTM.

5. Conclusion

This paper proposes an Attention-based CNN-LSTM model to achieve malicious URL recognition and detection. Experiments have shown that the accuracy of this method in detecting malicious URLs is significantly higher than those of shallow neural networks and single deep neural networks. The key of this model lies in extracting and filtering the texture information of the URL to be detected, the statistical attribute information of the URL string and the WHOIS information and then encoding those characteristics so that there is good linear separability of the features. Attention mechanism is then combined with the CNN-LSTM model to highlight key features and maximize their pooling for global features. Although the texture information feature is considered in the feature extraction process, and the Attention-Based CNN-LSTM model is used for classification

detection, there are still some areas need to be improved in identifying and detecting malicious URLs, such as the number of feature extractions and limitation of the effectiveness. How to make full use of the deep learning model and neural network model to better detect malicious URLs will be the focus of the next step.

Acknowledgement

This papers is supported by the Project of Cernet Next Generation Internet Technology Innovation Project (NGII20170420), Research Innovation Project of Graduate Student in Xinjiang Uygur Autonomous Region (XJGRI2017007).
Ms Code: TP309.2

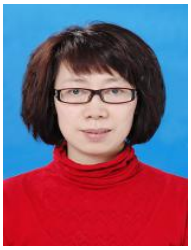
References

[1] Yue Zhang, Jason Hong, Lorrie Cranor, "Cantina: A Content-Based Approach to Detecting Phishing WebSites," in *Proc. of International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May. DBLP*, 639-648, 2007. Article (CrossRef Link).
[2] Mahmoud Khonji, Youssef Iraqi, Andrew Jones, "Phishing Detection: A Literature Survey," *IEEE Communications Surveys & Tutorials*, 15(4), 2091-2121, 2013. Article (CrossRef Link).
[3] Lance Spitzner, *Honeypots: tracking hackers*, Hacker, Boston, MA, USA, 2003. Article (CrossRef Link).
[4] Jiuxin Cao, Bo Mao, Junzhou Luo, Bo Liu, "A Phishing web Pages Detection Algorithm Based on Nested Structure of Earth Mover's Distance," *Chinese Journal of Computers*, 32(5), 922-929, 2009. Article (CrossRef Link).
[5] Shouxu Jiang, Jianzhong Li, "A Reputation-based Trust Mechanism for P2P E-commerce Systems," *Journal of Software*, 2007, 18(10), 2551-2563, 2007.
[6] Hongzhou Sha, Qingyun Liu, Tingwen Liu, Zhou Zhou, Li Guo, Binxing Fang, "Survey on Malicious Webpage Detection Research," *Chinese Journal of Computers*, 39(3), 529-542, 2016. Article (CrossRef Link).
[7] Sahoo D, Liu C, Hoi S C H, "Malicious URL Detection using Machine Learning: A Survey," 2017. Article (CrossRef Link).
[8] Pawan Prakash, Manish Kumar, Ramana Kompella, Minaxi Gupta, "Phishnet: predictive blacklisting to detect phishing attacks," in *Proc. of 2010 Proceedings IEEE INFOCOM*, 1-5, 2010. Article (CrossRef Link).
[9] Dharmaraj R Patil, Jayantrao Patil, "Survey on Malicious Web Pages Detection Techniques," *International Journal of u- and e- Service, Science and Technology*, vol. 8, no. 5, pp. 195–206, 2015. Article (CrossRef Link).
[10] Sujata Garera, Niels Provos, Monica Chew, Aviel D. Rubin, "A framework for detection and measurement of phishing attacks," in *Proc. of the 2007 ACM workshop on Recurring malcode. ACM*, pp. 1–8, 2007. Article (CrossRef Link).
[11] Mahmoud Khonji, Youssef Iraqi, Andy Jones, "Phishing Detection: A Literature Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013. Article (CrossRef Link).
[12] Raj Nepali, Yong Wang, "You Look Suspicious!!: Leveraging Visible Attributes to Classify Malicious Short URLs on Twitter," in *Proc. of 2016 49thHawaii International Conference on System Sciences (HICSS). IEEE*, pp. 2648–2655, 2016. Article (CrossRef Link).
[13] Masahiro Kuyama, Yoshio Kakizaki, Ryoichi Sasaki, "Method for Detecting a Malicious Domain by Using WHOIS and DNS Features," in *Proc. of The Third International Conference on Digital Security and Forensics (Digital Sec2016)*, pp. 74-80, 2016. Article (CrossRef Link).
[14] Liu G, Qiu B, Liu W, "Automatic Detection of Phishing Target from Phishing Webpage," in *Proc. of International Conference on Pattern Recognition. IEEE Computer Society*, 4153-4156, 2010. Article (CrossRef Link).

[15] Ma J, Saul LK, Savage S, GM Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious URLs," in *Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July. DBLP*, 1245-1254, 2009. Article (CrossRef Link).
[16] Ma J, Saul L K, Savage S, GM Voelker, "Identifying suspicious URLs: an application of large-scale online learning," in *Proc. of International Conference on Machine Learning. ACM*, 681-688, 2009. Article (CrossRef Link).
[17] Ma J, Saul L K, Savage S, GM Voelker, "Learning to detect malicious URLs," *Acm Transactions on Intelligent Systems & Technology*, 2(3), 1-24, 2011. Article (CrossRef Link).
[18] Xuejian Wang, Lantao Yu, Kan Ren, Guanyu Tao, Weinan Zhang, Yong Yu, Jun Wang, "Dynamic Attention Deep Model for Article Recommendation by Learning Human Editors' Demonstration," in *Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM*, 2051-2059, 2017. Article (CrossRef Link).
[19] Mnih V, Heess N, GravesA, K Kavukcuoglu, "Recurrent models of visual attention," in *Proc. of NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2204-2212, 2014. Article (CrossRef Link).
[20] Ming Sun, Anirudh Raju, George Tucker, Sankaran Panchapagesan, Gengshen Fu, "Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting," in *Proc. of Spoken Language Technology Workshop. IEEE*, 474-480, 2017. Article (CrossRef Link).
[21] Bahdanau D, Cho K, Bengio Y, "Neural Machine Translation by Jointly Learning to Align and Translate," *Computer Science*, 2014. Article (CrossRef Link).
[22] Bulò S R, Neuhold G, Kontschieder P, "Loss Max-Pooling for Semantic Image Segmentation," in *Proc. of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. Article (CrossRef Link).
[23] Xiaoguang Han, Wu Qu, Xuanxia Yao, Changyou Guo, Fang Zhou, "Research on malicious code variantsdetection based on texture fingerprint," *Journal on Communications*, 35(8), 16-136, 2014. Article (CrossRef Link).
[24] Shiqi Luo, Shengwei Tian, Long Yu, Jiong Yu, Hua Sun, "Detection on Android malware analysis based on Malware Image Fingerprint and Malware Activity Embedding in Vector Space," *Journal of Computer Applications*, 38(4), 1058-1063, 2018. Article (CrossRef Link).
[25] https://url.spec.whatwg.org/[EB/OL], 2018
[26] Hailun Lin, Wei Li, Weiping Wang, Yinliang Yue, Zheng Lin, "Efficient segment pattern based method for malicious URL detection," *Journal on Communications*, 36(s1),141-148, 2015. Article (CrossRef Link).
[27] Cao J, Li Q, Ji Y, et al., "Detection of Forwarding-Based Malicious URLs in Online Social Networks," *International Journal of Parallel Programming*, 44(1), 163-180, 2016. Article (CrossRef Link).
[28] Shi Y, Chen G, Li J, "Malicious Domain Name Detection Based on Extreme Machine Learning," *Neural Processing Letters*, vol. 48, no. 3, pp. 1347-1357, 2018. Article (CrossRef Link).

Yongfang PENG, born in 1994. Postgraduate student in the School of Software, Xinjiang University. Her main research interests include Information Security.

Shengwei TIAN, born in 1973. Ph.D. Professor in the School of Software, Xinjiang University. His main research interests include Intelligence Computing, Information Security, Image processing.



long YU, born in 1974. Professor in the Xinjiang University. She received the M.S. degree in Xinjiang university. Her research interests include Intelligence Technology, Information Security.

Yalong LV, born in 1992. Postgraduate student in the School of Information Science and Engineering, Xinjiang University. His main research interests include Information Security.

Ruijin WANG, born in 1980. Ph.D. lecturer in the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include information system security, quantum communication security, cloud security.