Phishing Detection Research Based on LSTM Recurrent Neural Network

Wenwu Chen[1(✉)], Wei Zhang[2], and Yang Su[1,2]

[1] Key Laboratory for Network and Information Security of Chinese Armed Police Force, Engineering University of Chinese Armed Police Force, Xi'an, Shaanxi, China
Chenwen5abc@163.com
[2] Department of Electronic Technology, Engineering University of the Chinese Armed Police Force, Xi'an, Shaanxi, China

Abstract. In order to effectively detect phishing attacks, this paper designed a new detection system for phishing websites using LSTM Recurrent neural networks. LSTM has the advantage of capturing data timing and long-term dependencies. LSTM has strong learning ability, has strong potential in the face of complex high-dimensional massive data. Experimental results show that this model approach the accuracy of 99.1%, is higher than that of other neural network algorithms.

Keywords: Phishing detection · LSTM · RNN · Deep learning
Cyberspace security

1 Introduction

Phishing attacks are growing threats to cyber security in worldwide. According to the Phishing Activity Trends Report (the first half year of 2017 and the third quarter of 2017) [1] released by the Anti-Phishing Working Group (APWG), from the first quarter of 2017 to the third quarter of 2017 with an increase of 65%, targeting a month more than 420 brands. This is the most frequent attack found since phishing was started in 2004 to track and report (Fig. 1).

In order to obtain the user name, password, ID number, bank card number and other private information, the attackers attract unknown victims to click the fake websites and deceptive E-mails [2]. These criminals are usually profitable using phishing, so their goal usually is online banking, online payment platform, and mobile commerce applications. Researchers firstly developed the blacklist technology to combat phishing attacks [3]. Although URL blacklists have been somewhat effective, the attacker can bypass the blacklist system by slightly modifying the characters in the URL string, and the time of blacklist suspicious sites is relatively delayed and cannot effectively identify new phishing websites.

To make up for the shortcomings of blacklist technology, researchers have tried heuristic detection methods, such as CANTINA [4] and CANTINA+ [5], and the visual similarity test [6]. Recently, the use of machine learning algorithms to identify phishing links becomes the mainstream of current research [7–9]. Long Short-Term Memory

Q. Zhou et al. (Eds.): ICPCSEE 2018, CCIS 901, pp. 638–645, 2018.
https://doi.org/10.1007/978-981-13-2203-7_52

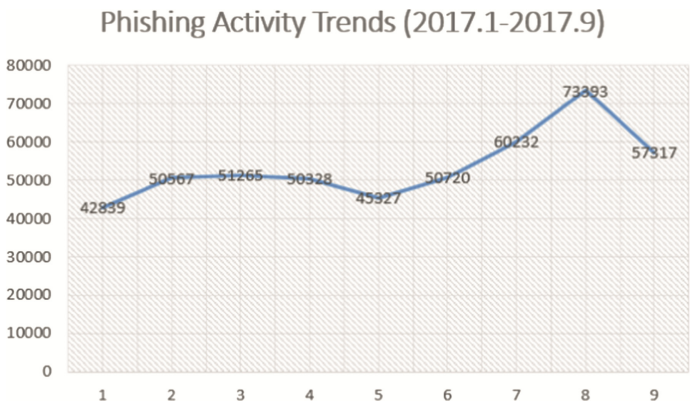

Fig. 1. Phishing activity trends (2017.1–2017.9)

(LSTM) is an architecture proposed by Hochreiter and Schmidhuber [10]. LSTM is a recurrent neural network (RNN), but it differs from the RNN mainly in that it incorporates an LSTM cell that determines the usefulness of information in the algorithm. LSTM has already had many applications in the field of science and technology. The LSTM-based system can learn the tasks of translating languages, controlling robots, image analysis, document summaries, speech recognition image recognition, handwriting recognition, controlling chatbots, predicting diseases, click-through rates and stocks, and synthesizing music.

2 URL Feature Extraction and Analysis

2.1 Uniform Resource Locator Standard Format

The Uniform Resource Locator is a standard resource address on the Internet, and the entrance to a website. Uniform Resource Locator confusing is very common to phishing, to lure users to click on the URL to visit their phishing website is an important part of phishing. To increase the likelihood of users visiting phishing sites, phishing attackers often use deceptive URLs that are visually similar to the fake ones. The format of a standard URL is as follows:

Protocol://hostname[:port]/path/[;parameters][?query]#fragment

The common way to confuse URLs is to construct a phishing URL by partially modifying and replacing the host name part and the path part based on the target URL in order to confuse the user.

For example, the attackers using "www.amaz0n.com" as a fake Amazon website (the real URL is "www.amazon.com"), or using the "www.interface-transport.com/www.paypal.com/" as a fake PayPal website (the real URL is "www.paypal.com") and so on.

2.2 Extract the Features of the Uniform Resource Locator

The purpose of the attacker's phishing URL is to convince the user that this is a legitimate website. In this way, the cybercriminals can get the user's personal and leaked financial information [12]. In order to achieve this goal, attackers use some common methods to camouflage phishing links. Through the research on the common means of attacker, we have identified a set of features that can be used to detect if the URL is a phishing link:

Domain names exist in the Alexa ranking: Alexa ranking is a list of domain names ordered by the Internet. Most phishing sites are hacked into the legitimate sites or new domains. If the phishing attack is made on a hijacked website, then it is unlikely that the domain name will be a part of the TLD because the top-ranked domain names tend to have better security. If the phishing website is located in a newly registered domain name, the domain name will not appear in the Alexa rankings.

Subdomain length: The length of the URL subdomain. Phishing sites attempt to use their domain as their subdomain to mimic the URL of a legitimate website. Legitimate websites tend to have a short subdomain name.

URL length: Phishing URLs tend to be longer than legitimate URLs. Long URLs increase the likelihood of confusing users by hiding the suspicious part of the URL, which may redirect user-submitted information or redirect uploaded web pages to suspicious domain names.

Prefixes and suffixes in URLs: Phishers trick users by remodeling URLs that look like legitimate URLs.

Length ratio: Calculate the ratio between the length of the URL and the length of the path; phishing sites often have a higher proportion of legitimate URLs.

The "@" and "-" counts: The numbers of "@" and "-" in the URL. In the URL, the symbol "@" causes the browser to ignore inputs of previous and later redirects the users to the typed links.

Punctuation counts: The number of "! # $% &" in the URL. Phishing URLs usually have more punctuation.

Other TLDs: The number of TLDs displayed in the URL path. Phishing web links emulate legitimate URLs by using domain names and TLDs in the path.

IP address: The host name - part of the URL uses an IP address instead of a domain name.

Port Number: If a port number exists in the URL, verify that the port is included in a list of known HTTP ports, such as 21, 70, 80, 443, 1080 and 8080. If the port number is not in the list, mark it as a possible phishing URL.

URL Entropy: Calculate URL Entropy. The higher the entropy of the URL, the more complicated it is. Because phishing URLs tend to have random text, so we can try to find them by their entropy.

3 Algorithm Model

3.1 Long Short-Term Memory Cell

Long-term short-term memory (LSTM) is a neural network architecture proposed by Hochreiter and Schmidhuber [10] in 1997. It differs from the RNN mainly in that it incorporates an LSTM unit that determines the usefulness of information in the algorithm. Figure 2 shows a single LSTM cell.

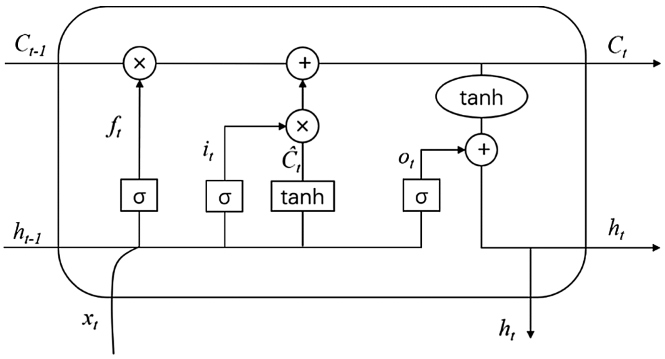

Fig. 2. LSTM cell

At time t, the components of the LSTM cell are updated as follows:

(1) Forgotten information from the cell state, determined by the Sigmoid layer of the Forgotten Gate, with the input xt of the current layer and the output ht − 1 of the previous layer as input, and the cell state output at the time t − 1 is formula (1)

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

(2) Store information in the cell state, consisting mainly of two parts:

(a) Results of the Sigmoid layer is i_t entering the gate as information to be updated;

(b) Vector C_t newly created by the tanh layer, to be added in the cell state. The old cell state C_{t-1} is multiplied by f_t to forget the information, and the new candidate information $i_t * \hat{C}_t$ is summed to generate an update of the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\hat{C}_t = \tan h(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \tag{4}$$

(c) Output information, determined by the output gate. First use the Sigmoid layer to determine the part of the information to output the cell state, and then use tanh to

process the cell state. The product of the two parts of the information yields the output value.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tan h(C_t) \tag{6}$$

Among them σ is the sigmoid function, h_{t-1} represents the hidden state of the $t-1$ moment, b represents the bias of each gate, i_t, f_t, o_t and C_t are the input gate, forget gate, output gate, and unit status, respectively. W_f, W_i and W_o are represented as a weight matrix for the connection. In LSTM cells, the three gates determine the status of the LSTM cell by controlling the flow of information. With LSTM, the gradient vanishing problem can be effectively solved.

3.2 Performance Evaluation

The purpose of phishing websites detection is to detect phishing instances from the test data set that contains phishing websites and legal websites, which is essentially a binary classification essence. In binary classification, a total of four kinds of classification, used to measure the accuracy of classification confusion matrix (Fig. 3).

		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

Fig. 3. Confusion matrix

Each URL falls into one of the four possible categories: true positive (TP, correctly classified phishing URL), true negative (TN, correctly classified as non- Phishing URL), false positives (FP, non-phishing URLs are incorrectly classified as phishing) and false negatives (FN, phishing URLs are incorrectly classified as non-phishing). Standard measures, such as accuracy, precision, recall, false negative rate, were determined using the following equation:

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN} \tag{7}$$

$$Precision = \frac{TP}{FP + TP} \tag{8}$$

$$Recall = \frac{TP}{FN + TP} \tag{9}$$

$$FNR = \frac{FN}{TP + FN} \tag{10}$$

4 Experimental Methods

The experiment uses the Python programming language. The LSTM model is implemented by a deep learning class such as Keras. It contains 5 LSTM layers with 128 nodes each. The model uses a stochastic gradient descent (SGD) optimization method with an initial learning rate of one thousandth and a batch size of 128. The objective function of the least-squares fit is a quadratic polynomial function. The dataset used consisted of 2000 legitimate websites collected from Yahoo Directory (http://dir.yahoo.com/) and 2,000 phishing websites collected from Phishtank (http://www.phishtank.com/). Collected data sets carry label values, "legal" and "phishing". In this data set randomly selected 70% for training, 30% for the test. The training dataset is used to train the neural network and adjust the weight of the neurons in the network, while the test dataset remains unchanged and used to evaluate the performance of the neural network. After training, run the test data set on the optimized neural network.

Predict phishing websites using LSTM Recurrent neural network. The above ten features are taken as input, that is, the number of input layer nodes in the LSTM network is 10 and the number of output layer nodes is one. Training network to choose a strong adaptability of the three-layer LSTM network, incentive function is sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{11}$$

LSTM neural network for classifying phishing URLs based on LSTM units. When entering a URL into an RNN, one-hot encoding is performed on each URL first. Since the characters composing the URL are all contained in ASCII characters (128 characters in total), each URL becomes one-hot encoded. An input vector with a dimension of (len_of_URL)*128 and then brings the input vector into the RNN. So each input character is translated by an 128-dimension embedding. The translated URL is fed into a LSTM layer as a 100-step sequence. Finally, the classification is performed using an output sigmoid neuron. The learning rate of LSTM neural network is 0.1.

In order to better illustrate the accuracy of the algorithm in this paper, an ordinary CNN is used to test the experimental data set. By experimenting with the selected data set, the results show that LSTM network are better than normal CNN, and their prediction accuracy is higher than that of CNN (Table 1).

Table 1. Evaluations of LSTM RNN and CNN

Method	Accuracy	Precision	Recall	FNR
CNN	0.9742	0.9648	0.9723	0.0591
LSTM	0.9914	0.9874	0.9891	0.0212

5 Conclusion and Discussion

In the phishing site testing process, many factors affect the test results, with a certain degree of non-linearity, this paper implements a LSTM-based phishing detection method, which solves the problem that it is difficult for other machine learning methods to extract valid features from the data. It is proved that the prediction method is effective in practice and can solve the problems that traditional methods are difficult to solve. At the same time, this paper adopts LSTM deep learning method and optimizes the training method of the model in combination with the characteristics of RNN. The training time of deep learning model is generally possible from hours to days, and the optimization convergence time is strict on the timeliness of power dispatching and other issues. It is of great significance.

References

1. Phishing Activity Trends Report: Phishing Activity Trends Report 1st Half. Methodology (2017)
2. PHISHTANK: Free community site for anti-phishing service. http://www.phishtank.com/
3. Sinha, S., Bailey, M., Jahanian, F.: Shades of grey: on the effectiveness of reputation-based "blacklists". In: International Conference on Malicious and Unwanted Software, pp. 57–64. IEEE (2008)
4. Zhang, Y, Hong, J.I., Cranor, L.F.: Cantina: a content-based approach to detecting phishing web sites. In: International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May, pp. 639–648. DBLP (2007)
5. Xiang, G., Hong, J., Rose, C.P., et al.: CANTINA+: a feature-rich machine learning framework for detecting phishing web sites. ACM Trans. Inf. Syst. Secur. **14**(2), 21 (2011)
6. Wenyin, L., Huang, G., Xiaoyue, L., et al.: Detection of phishing webpages based on visual similarity. In: Special Interest Tracks and Posters of the 14th International Conference on World Wide Web, pp. 1060–1061 (2005)
7. Ma, J., Saul, L.K., Savage. S., et al.: Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–July, pp. 1245–1254. DBLP (2009)
8. Choi, H., Zhu, B.B., Lee, H.: Detecting malicious web links and identifying their attack types. In: Usenix Conference on Web Application Development, p. 11 (2011)
9. Ma, J., Saul, L.K., Savage, S., et al.: Identifying suspicious URLs: an application of large-scale online learning. In: International Conference on Machine Learning, pp. 681–688. ACM (2009)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
11. Sadeghi, B.H.M.: A BP-neural network predictor model for plastic injection molding process. J. Mater. Process. Technol. **103**(3), 411–416 (2000)
12. Ma, J., Saul, L.K., Savage, S., et al.: Learning to detect malicious URLs. ACM Trans. Intell. Syst. Technol. **2**(3), 1–24 (2011)
13. Sahoo, D., Liu, C., Hoi, S.C.H.: Malicious URL detection using machine learning: a survey (2017)
14. Kim, D., Achan, C., Baek, J., et al.: Implementation of framework to identify potential phishing websites, p. 268. IEEE (2013)

15. Garera, S., Provos, N., Chew, M., et al.: A framework for detection and measurement of phishing attacks. In: ACM Workshop on Recurring Malcode, pp. 1–8. ACM (2007)
16. Olivo, C.K., Santin, A.O., Oliveira, L.S.: Obtaining the threat model for e-mail phishing. Appl. Soft Comput. J. **13**(12), 4841–4848 (2013)
17. Herzberg, A., Jbara, A.: Security and identification indicators for browsers against spoofing and phishing attacks. ACM Trans. Internet Technol. **8**(4), 1–36 (2008)
18. Pan, Y., Ding, X.: Anomaly based web phishing page detection. In: 2006 Computer Security Applications Conference, ACSAC 2006, pp. 381–392. IEEE (2006)
19. Fu, A.Y., Liu, W., Deng, X.: Detecting phishing web pages with visual similarity assessment based on earth mover's distance (EMD). IEEE Trans. Dependable Secur. Comput. **3**(4), 301–311 (2006)