Advances in Computer and Communication, 2020, 1(1), 1-6
http://www.hillpublisher.com/journals/acc/
ISSN Online: ****-****
ISSN Print: ****-****

URL Security Detection on the Basis of a New Optimization Algorithm

B. Wang[1,2,*], B. F. Zhang[1], X. W. Liu[1], S. M. Zhong[2]

[1]School of Electrical Engineering and Electronic Information, Xihua University, Chengdu, 610039, China.
[2]School of Applied Mathematics, University Electronic Science and Technology of China, Chengdu 610054, China.

How to cite this paper: B. Wang, B. F. Zhang, X. W. Liu, S. M. Zhong. (2020) URL Security Detection on the Basis of a New Optimization Algorithm. *Advances in Computer and Communication*, 1(1), 1-6.
DOI: 10.26855/acc.2020.12.001

Received: August 31, 2020
Accepted: September 23, 2020
Published: September 28, 2020

***Corresponding author:** B. Wang, School of Electrical Engineering and Electronic Information, Xihua University, Chengdu, 610039, China; School of Applied Mathematics, University Electronic Science and Technology of China, Chengdu 610054, China.
Email: 63569368@qq.com

Abstract

In this paper, the issue on URL security detection is investigated. Considering the problems about local optimization and speed, chaotic mapping is introduced into PSO to design the optimization algorithm for BP neural network to achieve URL security detection with better performance. Some typical experimental examples are included and corresponding results display the advantage and effectiveness of the optimization algorithm proposed.

Keywords

URL Security Detection, Optimization, Neural Network

1. Introduction

URL security detection has always been the highlight in the research on Web security protection. At present, malicious URL is mainly detected by black and white list-based URL detection method and machine learning-based URL detection method. Forblack and white list-based URL detection method, website cannot be visited until confirming the URL is not within the blacklist database through checking the blacklist. Featured by simplicity and efficiency, it has been widely used in many mainstream browsers, such as IE8, Mozilla Firefox 2.0, Safari, and Chrome, etc. However, the method requires regular blacklist maintenance, which results in the high cost and may lead to the problem of judgment omission. For machine learning-based URL detection method, some characteristics, such as URL characteristics, domain name characteristics, host characteristics and so on, will be utilized to improve the ability to identify malicious URL, and related researches have received more and more attention recently. For instance, in Sahingoz et al. [1], some typical classification approaches and NLP-based features are adopted to detect phishing URL; in Li et al. [2], a linear learning approach of two-stage distance metric and nonlinear Nyström method of kernel approximation are adopted to improve the malicious URLs detection. In Y. Li et al. [3], lightweight URL and HTML features are introduced to design the detection method of phishing webpage with better performance; in Wei et al. [4], a way to detect malicious URL addresses using convolutional neural networks is proposed and can be applied in mobile devices without significantly affecting the performance; in Wang et al. [5], a multi-view neural network is introduced and URL feature mining technique is adopted to achieve malware detection.

However, most of these research outcomes are mainly about the extraction and analysis of URL features, and hardly lay emphasis on machine learning to improve URL detection performance. As a commonly used machine

learning method for URL security detection, BP neural network (BPNN) was proposed by Rumelhart and Hinton in 1986. As a three layers feedforward network, BP neural network can earn the relationship between input and output without the need of knowing its mathematical expressions in advance. However, "localoptimization" and "speed" are the problems that PB neural network always concerns, and have not been well solved yet.

Characterized with ergodicity, nonperiodicity and random, chaos can be able to provide an approach to solve the problem [6-10]. In this paper, chaotic mapping will be introduced into PSO to design the optimization algorithm of BP neural network to achieve URL security detection with better performance. However to our knowledge, the related work has seldom been carried out at present, and these motivate our investigation.

2. Preliminaries

Term Frequency-Inverse Document Frequency (TF-IDF) is a kind of the commonly used weighting technique of information exploration.

TF:

$$tf_{ij} = \frac{n_{ij}}{\sum_k n_{kj}}$$

where, tf_{ij} is the frequency of the term t_i appearing in the text, n_{ij} is the times of the term t_i appearing in the file d_j, and $\sum_k n_{kj}$ is the total times of all terms appearing in the file d_j.

IDF:

$$idf_{ij} = \log \frac{|\mathrm{D}|}{|\{j : t_i \in \mathrm{d}_j\}|}$$

where, idf_{ij} is the measure of the importance of the term t_i in the whole file set; $|\mathrm{D}|$ is the total number of files in the corpus; $|\{j : t_i \in \mathrm{d}_j\}|$ is the number of files containing the term t_i.

TF-IDF:

$$tfidf_{ij} = tf_{ij} \times idf_{ij}$$

Remark 1: It can be seen that the bigger value of $tfidf_{ij}$ means the greater importance the term has to the text. Therefore, TF-IDF tends to filter out common terms and retain important ones.

N-grams refer to N words appearing consecutively in a text. N-grams model is a probabilistic language model based on $N-1$ Markov chain. It infers the structure of sentences by the probability under which N words appear. N-grams text is widely used in text mining and natural language processing tasks.

For instance, URL:

www.foo.com/1

When N = 3, one can get N-gram:

'ww.','w.f','.fo','foo','oo.','o.c','.co','com','om/','m/1'

The Framework to produce the sparse matrixes of URLs is shown in Figure 1.

3. Main results

The main results of this paper are expounded below with typical examples.

Normal URLs: 1,265,974, from http://secrepo.com

Malicious URLs: 44,532, from https://github.com/foospidy/payloads

The total number of URLs is 1,310,506.

In view of the large total characteristic matrix, set "batch size" of BP neural network as 2,000. Each time 1,000 formal requests and 1,000 malicious requests are input, and formed into a matrix $X_{train} \in R^{S_0 \times 2000}$ through TF-IDF and n-grams techniques, S_0 represents the number of sample attributes, and the number of input neurons is set as S_0, $Y_{label} \in R^{2\times 2000}$ represents the classification result of the sample, 1 means that it belongs to this class, and 0 means that it does not belong to this class. An example of Y_{label} is shown in Table 1.

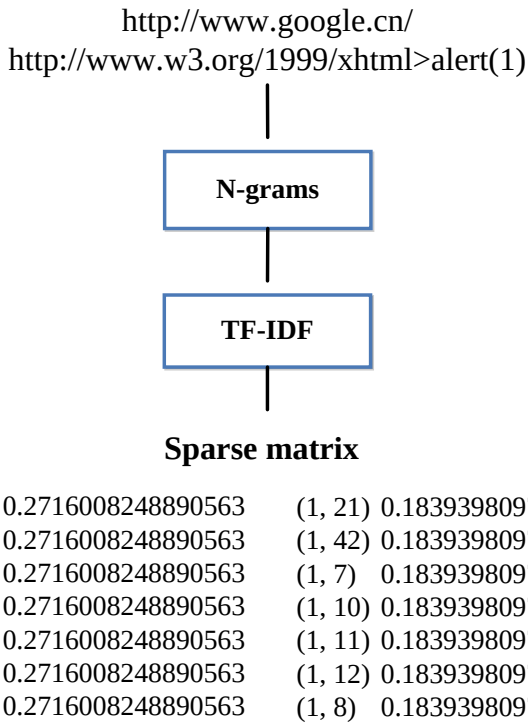

Figure 1. Framework to produce the sparse matrix of URLs.

Table 1. An example of Y_{label}

Y_{label}	Sample 1	Sample 2	Sample 3	Sample 4
Class I	1	0	0	1
Class II	0	1	1	0

Remark 2: From Table 1, it can be seen that this Y_{label} contains 4 samples, Samples 1 and 4 belonged to Class I, and Samples 2 and 3 belonged to Class II.

BP neural network: $epoch$ is Training period, lr is learning rate, S_0 and S_2 are neurons number of input and output layer, S_1 is neurons number of hidden layer, such that

$$S_1 = \sqrt{S_0 + S_2} + a$$

where $a \in [1,10]$

In this paper, the fitness function is defined as follow:

$$Fitness = \frac{1}{SE}$$

where

$$SE = \text{sumsqr}(X_{test} - A_2)$$
$$A_2 = \text{purelin}(W_2 A_1, b_2)$$
$$A_1 = \text{tansig}(W_1 X_{train}, b_1)$$
$$W_1 = X_{out}(1, \ldots, S_0 S_1)$$
$$W_2 = X_{out}(S_0 S_1 + 1, \ldots, S_0 S_1 + S_1 S_2)$$
$$b_1 = X_{out}(S_0 S_1 + S_1 S_2 + 1, \ldots, S_0 S_1 + S_1 S_2 + S_1)$$
$$b_2 = X_{out}(S_0 S_1 + S_1 S_2 + S_1 + 1, \ldots, S_0 S_1 + S_2 S_1 + S_1 + S_2)$$
$$S = RS_1 + S_2 S_1 + S_1 + S_2$$

where sumsqr denotes sum of squares operator, purelin denotes linear transformation, tansig denotes sigmoid transformation, X_{train} denotes training data, X_{test} denotes testing data.

Adjustable parameter: W_1 denotes the first weight matrix, W_2 denotes the second weight matrix, b_1 denotes the first offset value, b_2 denotes the second offset value.

PSO:

$$v_{id} = wv_{id} + c_1 \cdot r_1 \cdot \left(p_{id} - x_{id} \right) + c_2 \cdot r_2 \cdot \left(p_{gd} - x_{id} \right)$$
$$x_{id} = x_{id} + v_{id}$$

where $x_{id}, 1 \le i \le M, 1 \le d \le D$, M denotes size of swarm, D denotes dimension of particle, c_1, c_2 are learning factors, w is inertia weight, and r_1, r_2 are random numbers between 0 and 1.

Then a novel optimization algorithm for URL security detection based on PSO is designed as follow:

Step 1: Initializing particle swarm A

Step 2: Updating particle velocity v_{ir} and particle position x_{ir}

Step 3: Evaluating fitness of A

Step 4: Determining local optimum p_{ir} and global optimum p_{gr} of A

Step 5: Obtaining chaotic sequence

Let

$$x_0 = \frac{p_{gr} - p_{min}}{p_{max} - p_{min}}$$

Chaotic mapping

$$x_{n+1} = \mu \cdot x_n \cdot (1 - x_n)$$

where $\mu = 4$

After M iterations, one can get

$$P = [p_1, p_2, \text{L}, p_M]$$

where

$$p_i = x_i (p_{\max} - p_{min}) + p_{min}$$

Step 6: Calculating fitness of P

Step 7: Determining the best particle fitness p_{gr}^{best} of P

Step 8: Obtaining p_{gr}

$p_{gr} \leftarrow p_{gr}^{best}$, if p_{gr}^{best} is superior to p_{gr};

Otherwise, p_{gr} remains the same.

Step 9: Operation judgment

p_{gr} is output as the optimal value when $t \geq T_{max}$;

Otherwise, jump to Step 2.

4. Experiments

Let $lr = 0.01$, $epoch = 1000$, $S_0 = 25621$, $S_2 = 2$, $a = 3$, $S_1 = 163$, $T_{max} = 100$, $w = 0.8$, $c_1 = 2$, $c_2 = 2$, and sampling time is 0.001s, the experiment results are displayed in Figure 2.

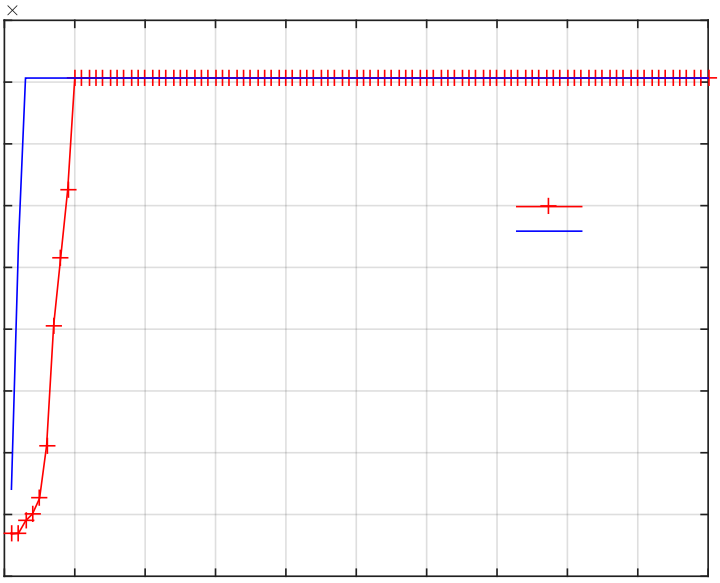

Figure 2. Convergence curve of fitness of algorithms for URL security detection.

Remark 3: It can be seen that BP algorithm-based fitness curve converges to optimal value after 10 iterations, proposed CPSO-BP algorithm-based fitness curve converges to optimal value after 2 iterations, which means that the proposed algorithm has the better searching ability for URL security detection.

To better show the effectiveness of the method in this paper, 10 URLs with known security features are selected as test data. The experimental results are shown in Figure 3.

```
{'url(1)': 'www.foo.com/id=1&amp;lt;script&amp;gt;alert(1)&amp;lt;/script&amp;gt'}result: bad request
{'url(2)': 'www.foo.com/name=admin&amp;#x27; or 1=1'}result: bad request
{'url(3)': 'abc.com/admin.php'}result: good request
{'url(4)': '&amp;quot;&amp;gt;&amp;lt;svg onload=confirm(1)&amp;gt;'}result: bad request
{'url(5)': 'test/q=&amp;lt;a href=&amp;quot;javascript:confirm(1)&amp;gt;'}result: bad request
{'url(6)': 'q=../etc/passwd', 'res': 'bad request'}result: bad request
{'url(7)': '/stylesheet.php?version=1331749579'}result: good request
{'url(8)': '/&amp;lt;script&amp;gt;cross_site_scripting.nasl&amp;lt;/script&amp;gt;.idc'}result: bad request
{'url(9)': '&amp;lt;img 9src=x onerror=&amp;quot;javascript:alert(1)&amp;quot;&amp;gt;'}result: bad request
{'url(10)': '/jhot.php?rev=2 |less /etc/passwd'}result: bad request

Process finished with exit code 0
```

Figure 3. Run result of experiment of URL security detection.

Remark 4: From Figure 3, it can be seen that the normal URL is terms 3 and 7, and the malicious URL is terms 1, 2, 4, 5, 6, 8, 9, 10. The predicted results are consistent with the security of the actual URL.

5. Conclusion

In the view of problems about local optimization and speed, chaotic mapping has been introduced into PSO to design the optimization algorithm for BP neural network to achieve URL security detection with better performance, and some experiments have been carried out and corresponding results have shown the advantage and effectiveness of the given optimization algorithm.

Acknowledgments

This work is partially supported by Sichuan Science and Technology Program under grant No. 2018GZDZX0008, Sichuan Science and Technology Program under grant No. 2019YFG0509, and Sichuan Science and Technology Program under grant No. 2019YFG0508.

References

[1] Sahingoz, O., Buber, E., and Demir, O. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications, 1171, 345-357.

[2] Li, T., Kou, G., and Peng,Y. (2020). Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation methods. Information Systems, 91, 101494.

[3] Li, Y., Yang, Z., and Chen, X. (2019). A stacking model using URL and HTML features for phishing webpage detection. Future Generation Computer Systems, 94, 27-39.

[4] Wei, W., Ke, Q., and Nowak, J. (2020). Accurate and fast URL phishing detector: A convolutional neural network approach. Computer Networks, 1784, 107275.

[5] Wang, S., Chen, Z., and Yan, Q. (2020). Deep and broad URL feature mining for android malware detection. Information Sciences, 513, 600-613.

[6] Rösslor, O. E. (1976). An equation for continuous chaos. Physics Letters A, 57, 397-398.

[7] Dong, E. and Yuan, M. (2019). A new class of Hamiltonian conservative chaotic systems with multistability and design of pseudo-random number generator. Applied Mathematical Modelling, 73, 40-71.

[8] Shabestari, P., Panahi, S., and Hatef, B. (2018). A new chaotic model for glucose-insulin regulatory system. Chaos, Solitons & Fractals, 112, 44-51.

[9] Chen, G. R. and Ueta, T. (1999). Yet another chaotic Attractor. International Journal of Bifurcation and Chaos, 9, 1465-1466.

[10] Balootaki, M., Rahmani, H., and Moeinkhah, H. (2020). On the Synchronization and Stabilization of fractional-order chaotic systems: Recent advances and future perspectives. Physica A: Statistical Mechanics and its Applications, 551, 124203.