2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications

Effective phishing website detection based on improved BP neural network and dual feature evaluation

Erzhou Zhu
School of Computer Science and Technology, Anhui University
Hefei, P.R. China
ezzhu@ahu.edu.cn

Dong Liu
School of Computer Science and Technology, Anhui University
Hefei, P.R. China
dongliucn723@gmail.com

Chengcheng Ye
School of Computer Science and Technology, Anhui University
Hefei, P.R. China
ccyeahu@gmail.com

Feng Liu
School of Computer Science and Technology, Anhui University
Hefei, P.R. China
fengliu@ahu.edu.cn

Xuejun Li
School of Computer Science and Technology, Anhui University
Hefei, P.R. China
xjli@ahu.edu.cn

Hui Sun*
School of Computer Science and Technology, Anhui University
Hefei, P.R. China
sunhui@ahu.edu.cn

***Abstract*—Nowadays, phishing poses a big threat to people's daily network environment. By phishing, attackers obtain the network users private information by inducing them to open illegal websites. Due to the active learning ability and preferable classifying ability for many datasets, BP neural network is an important heuristic machine learning method in phishing websites detection and prevention. However, improper selection of initial parameters, such as the initial weight and threshold, will induce the BP neural network into local minimum and slow learning convergence. Aiming at these problems, this paper proposes DF.GWO-BPNN, an effective phishing website detection model based on the improved BP neural network and dual feature evaluation mechanism. Under this model, the grey wolf algorithm is firstly used to optimize the BP neural network to reasonably select initial parameters. Then, the dual feature mechanism is used to evaluate the results of the improved BP neural network. By the dual feature evaluation mechanism, the accuracy of phishing website recognition is improved. Meanwhile, the black and white list is used to improve the efficiency of the proposed model. The DF.GWO-BPNN model is compared with some existing phishing website detection models. The experimental results have demonstrated that our model is accurate and strong adaptability.**

***Index Terms*—Phishing website detection, BP neural network, Grey wolf algorithm**

I. Introduction

Phishing is a kind of cybercrime behavior in which attackers obtain the network users private information by inducing them to open illegal websites. On available of the stolen private information, phishing attackers can get money and other benefits from network users. With the progress of network technology, phishing attackers can use a variety of techniques to make the phishing websites look legitimate. Phishing websites are becoming more and more capable of avoiding detection [1]. Nowadays, phishing websites are widely flooded in the daily PC and mobile environments. Meanwhile, the number of phishing websites is growing rapidly, which poses a big threat to the people's online life. The distribution and harm of phishing websites are crossing the national borders and becoming a global problem [2]. It is urgently needed effective techniques to prevent and detect phishing websites.

In order to predict and detect phishing websites, many techniques have been proposed. The black and white list is a direct method. However, on facing of increasing number and continuing change of phishing websites, this method alone cannot effectively detect phishing websites. The Google's PageRank based method [3] can detect the phishing websites at the page level. However, due to low rank of new established legitimate websites, the PageRank based method cannot deal with them properly. By extracting and analyzing the source code of webpage, some malicious behaviors can be detected [4]. However, this method takes too much time in processing source code. Meanwhile, this method may cause misjudgments duo to improper selection of phishing features.

Nowadays, machine learning is a reliable method in predicting and detecting phishing websites. In this method, machine learning algorithms are used to construct phishing website prediction models. Commonly used machine learning models are Bayesian model, SVM (Support Vector Machine) model and neural network model. The Bayesian based method is efficient, but it is sensitive to phishing features [5]. The improper selection of features will result in imprecise and unstable of phishing websites detection. The SVM based method take effect in common cases. However, on facing of large scale datasets, due to the difficulty in finding feasible

*Corresponding Author : Hui Sun

This paper is supported by the Nation Natural Science Foundation of Education Department of Anhui province (China) [Grant No. KJ2018A0022].

978-1-7281-1141-4/18/$31.00 ©2018 IEEE
DOI 10.1109/BDCloud.2018.00114

kernel functions, the accuracy of this method is relatively low [6]. Due to the active learning ability and preferable classifying ability for many datasets, neural network is commonly used in phishing websites detection and prevention [7].

As an important and most widely used neural network method, BP neural network uses the error back propagation algorithm to train global sample sets [8], [9]. Under this algorithm, the output of each node in the neural network is calculated at first. If the result cannot reach desired accuracy, the feedback function is used to inversely calculate the error generated by each neuron node. Finally, these errors are then used to adjust the threshold of the node throughout the network and the weight among nodes of a layer. However, due to the converge along the direction of gradient descent of mean square error, the BP neural network is very sensitive to the initial values of threshold and weight. Improper selection of initial parameters, such as the initial weight and threshold, will induce the BP neural network into local minimum and slow learning convergence. This problem greatly limits the application range of BP neural networks in classification.

In this paper, we propose DF.GWO-BPNN, an effective phishing website detection model based on the improved BP neural network and dual feature evaluation mechanism. In the proposed model, the GWO (Grey Wolf Optimizer) algorithm [10], [11] is used to overcome the shortages of traditional BP neural network. By optimizing with the GWO, the local minimum and slow coverage problems of BP neural network are avoided. The new the dual feature mechanism is used to evaluate the results of the improved BP neural network. By the dual feature evaluation mechanism, the accuracy of phishing website recognition is improved. Meanwhile, for the purpose of efficiency, the black and white list is used to cache websites that have already been processed. Experimental results on testing many commonly used samples have demonstrated that our proposed DF.GWO-BPNN model is accurate and strong adaptability.

The remainder of this paper is organized as follows: Section 2 overviews the workflow of the proposed DF.GWO-BPNN model. Section 3 gives it implantation. Section 4 evaluates the performance of this model.

II. Overview of the Proposed DF.GWO-BPNN Model

Generally, as showing in Fig. 1, the DF.GWO-BPNN is composed of four modules, the *Dynamic Hash Library*, the *Feature Extraction and Classification* module, the *DF.GWO-BPNN Classifier* (the BP neural network optimized by GWO) and the *Comprehensive Evaluation* module.

Specifically, the *Dynamic Hash Library* caches the black and white lists. For the purpose of improving efficiency, this module is used to cache websites that have already been processed. The *Feature Extraction and Classification* module extracts and classifies features from the URL composition. By this module, URL features are divided into two categories, dominant features and the recessive features. URLs with recessive features are feeded to the *DF.GWO-BPNN Classifier* for further procession. As the core component of the proposed

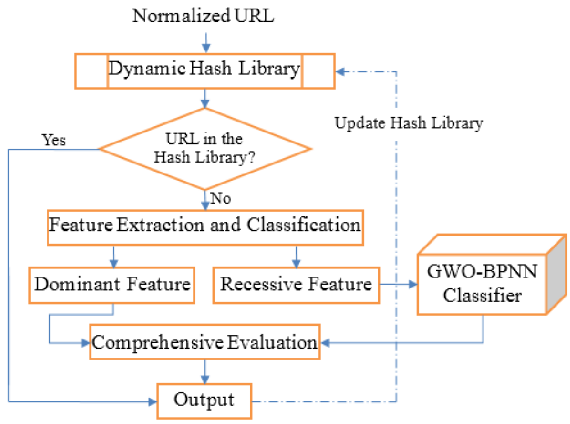

Fig. 1. Overall workflow of DF.GWO-BPNN phishing detection model

model, the *DF.GWO-BPNN Classifier* is used to classify the phishing websites. Due to the global search ability, the GWO is used to optimize the BP neural network to reasonably select initial parameters. By utilizing the GWO, problems of local minimum and slow learning convergence of traditional BP neural network are avoided. The *Comprehensive Evaluation* module is built on the dual feature evaluation mechanism. By comprehensively evaluating the dominant features and the result of the *DF.GWO-BPNN Classifier*, this module gives the final results of our proposed DF.GWO-BPNN model.

III. Implementation

A. Dynamic hash library

For the purposes of saving the network bandwidth and the processing time, the preprocessing procedure is utilized to normalize the input URLs. Firstly, "GET" requests used for applying Internet resources are captured. Then, captured URLs are parsed for eliminating duplicates (also call de-reprocessed [12]). By doing this, we can guarantee that there are no two or more identical URLs in the library. Finally, these URLs are normalized and passed to the dynamic hash library.

In the dynamic hash library, normalized URLs are filtered by the black and white list. The black list library is a collection of URL hash values that have been identified as phishing websites. The white list library caches the full domain name hash values of the trusted websites. Under this circumstance, URLs fall into the white list are marked as the trusted ones; URLs fall into the black list are marked as phishing websites. URLs with no corresponding record in the white list or black list will be passed to the next component of the model for further processing.

In order to improve the speed of URLs retrieving, records in the hash library are encrypted by MD5 algorithm. Since the content of the black list in the hash library is one of the important indicator of phishing recognition ability, the continuously update of the black list is an important mean to improve the efficiency of a model. So, in the proposed DF.GWO-BPNN model, we regularly check and re-evaluate

the black list to improve the accuracy of phishing website identification. Meanwhile, URLs of the phishing websites detected by our proposed model are also added to the black list.

B. URL feature extraction and classification

An attacker often disguises an illegal URL as a legitimate one to deceive users to obtain the private information [13]. Compared with legitimate URLs, some illegal URLs have obvious identifiable features (dominant features). However, there are also many illegal URLs cannot be detected by directly processing their URL features (recessive feature). In our model, the *Feature Extraction and Classification* module extracts and classifies features from the URL composition. By this module, URL features are classified into two categories, dominant features and the recessive features. URLs with dominant features will be directly feeded to the Comprehensive Evaluation module. URLs with recessive features need to be further processed by the *DF.GWO-BPNN Classifier* before being feeded to the *Comprehensive Evaluation* module.

a) Dominant features: In this section, vector $D(D_1, D_2, D_3, D_4)$ is used to record dominant features of a URL.

- *Whether a URL is in the form of IP*. Usually, phishing attackers try to deceive users by indicating the IP address as their website's URL domain name. Based on this observation, we choose this feature as the first dominant feature. In our model, D_1 ($D_1 \in \{0, 1\}$) is used to specify this feature. D_1=1 means the corresponding URL is in the form of IP. D_1=0 means the corresponding URL with no IP.
- *The length of a URL*. By increasing the length of URLs, phishing attackers conceal their real domain names. By doing this, common users cannot identify whether they are phishing URLs. In order to detect this kind of phishing websites, we set the length of URLs as the second dominant feature. D_2 ($D_2 \in \{0, 0.5, 1\}$) is used to specify this feature. D_2 means the number of characters of the corresponding URL is less than 54; D_2=0.5 means the length of the corresponding URL is in the interval of (54, 75]; D_2=0 means length of the URL is more than 75.
- *Whether submit user information to email*. Legitimate websites usually do not use the *mailto()* or *mail()* functions to send users information. So, if the *mailto()* or *mail()* functions are evoked in a website, we will have a suspect that whether it is a fishing website. In model, D_3 ($D_3 \in \{0, 1\}$) is used to specify this feature. D_3=1 means the corresponding website has used the above two functions to send users information; while D_3=0 means not.
- *Domain age and effective time*. Due to update frequently, domain names of phishing websites usually have short lifecycles. Consequently, a website with short domain name and short effective time will be suspected as a phishing website. We use D_4 ($D_4 \in \{0, 1\}$) to specify this feature.D_4=1 means the domain age is less than six months or the domain effective time is less than one year. D_4=0 specifies the other situations.

b) Recessive features: Vector $R(R_1,R_2,R_3,R_4,R_5)$ is used to record recessive features of a URL. The recessive features vector R will be passed to the *DF.GWO-BPNN Classifier* for further processing. By predicting of the *DF.GWO-BPNN Classifier*, a result R_s ($R_s \in 0, 1$) will be generated for the subsequent component of our model. Where, R_s=0 means the output value of the classifier is no less than 0.5; R_s=1 means the output value of the classifier is less than 0.5.

- *PrimaryDomain*. The value of PrimaryDomain is marked as R_1. Let V_1 as Levenshtein distance between the PrimaryDomain of a URL and the corresponding result of Google search engine spelling suggestion. By this, if V_1=0, R_1 is set as R_1=1 (specifies the legal website); otherwise, R_1 is set as R_1=1-(1/V_1).
- *SubDomain*. The value of SubDomain is marked as R_2. Let V_2 as Levenshtein distance between the SubDomain of a URL and the corresponding result of Google search engine spelling suggestion. By this, if V_2=0, R_2 is set as R_2=1 (specifies the legal website); otherwise, R_2 is set as R_2=1-(1/V_2).
- *PathDomain*. The value of PathDomain is marked as R_3. Let V_3 as Levenshtein distance between the PathDomain of a URL and the corresponding result of Google search engine spelling suggestion. By this, if V_3=0, R_3 is set as R_3=1 (specifies the legal website); otherwise, R_3 is set as R_3=1-(1/V_3).
- *PageRank*. The value of PageRank is marked as R_4 ($R_4 \in [0, 10]$). Let V_4 as domain value of the tested URL obtained from Google PageRank. By this, if V_4=0, R_4 is set as R_4=1 (specifies the corresponding website may be a phishing website); otherwise, R_4 is set as R_4=1-(1/V_4).
- *AlexaRank*. The value of AlexaRank is marked as R_5. Let V_5 as domain value of the tested URL obtained from AlexaRank. By this, R_5 is set as R_5=1-(1/V_5).

C. DF.GWO-BPNN Classifier

As an important heuristic machine learning method, BP neural network has been widely used in classification. However, traditional BP neural networks are very sensitive to the initial values of threshold and weight. Improper selection of initial parameters will induce BP neural networks into local minimum and slow learning convergence. This problem greatly limits the application range of BP neural networks in classification. Due to characteristics of simple structure, few parameters, easy implementation, fast convergence and good global performance, in this paper, the GWO algorithm is used to optimize the BP neural network [14].

a) Principles of GWO algorithm: GWO is a new group intelligent optimization algorithm that imitates the leadership hierarchy and hunting mechanism of the grey wolf group in nature [15]. According to fitness values, the GWO algorithm divides the wolves of a group into four categories: α, β, δ and ω. Among them, α is the leader of gray wolves; β is the suboptimal gray wolf; δ is the third gray wolf to help decision;

and the other gray wolves are in the category of ω. During the process of capturing their prey, grey wolves α, β, and δ directly pursue the prey. Following the first three gray wolves, the remaining gray wolves ω track and encircle the prey. In the GWO algorithm, the position of the prey is corresponding to the solution to a specific problem.

The mathematical model of GWO is as follows:

$$D = |C \cdot X_p(t) - X(t)| \tag{1}$$

Where, $X(t+1)=X_p(t) - A \cdot D$;$A=2a \cdot r_2 - a$; $C=2r_1$; $a=2 - t/max$; D is the distance vector between an individual(any single grew wolf in the group) and the prey; t specifies the current iteration number; X_p is the position of the prey; X is the position vector of all grew wolves in the group; A represents the convergence factor ($|A| > 1$ (corresponding to global search) means enlarging the search scope, $|A| < 1$ (corresponding to the local search) means narrowing the circle of encirclement); as the number of iterations increases, parameter a linearly decreases from 2 to 0; max is the maximum number of iterations; r_1 and r_2 are random numbers between [0, 1].

When the position of the prey is locked, the head grew wolf α leads β, δ and other grew wolves to surround the prey. Since α, β and δ are closest to the prey, the approximate position of the prey is judged from the position of the three wolves:

$$D_\alpha = |C_1 \cdot X_\alpha(t) - X(t)| \tag{2}$$

$$D_\beta = |C_2 \cdot X_\beta(t) - X(t)| \tag{3}$$

$$D_\delta = |C_3 \cdot X_\delta(t) - X(t)| \tag{4}$$

Where, X_α, X_β, X_δ represent the current positions of the gray wolves α, β, and δ respectively; C_1, C_2 and C_3 are the random vector coefficients, $X(t)$ represents the current position vector of the gray wolves. According to (2) - (4), the forward step size and the forward direction of other grew wolves (ω) are determined:

$$X_1 = X_\alpha - A_1 \cdot D_\alpha \tag{5}$$

$$X_2 = X_\beta - A_2 \cdot D_\beta \tag{6}$$

$$X_3 = X_\delta - A_3 \cdot D_\delta \tag{7}$$

$$X(t+1) = (X_1 + X_2 + X_3)/3 \tag{8}$$

According to the above equations (specify the forward step size and the forward direction), grew wolves gradually approach and capture the prey. Many researchers have proved that the GWO algorithm is superior to the particle swarm optimization algorithm, genetic algorithm and other intelligent optimization algorithms in forming the global optimal solution to a problem [16]. So, in this paper, we use GWO to optimize the weight and threshold of BP neural network.

b) GWO optimized BP neural network: Due to the ability of efficient finding global optimal solutions for many problems, the GWO algorithm is used to optimize the BP neural network. The basic idea of utilizing the GWO algorithm is to specify the range of initial parameters (the values of weight and threshold) of the BP neural network. By optimized with GWO, the convergence speed and accuracy of the BP neural network are improved. Specifically, by setting the weight and threshold of the BP neural network as the positions of gray wolves, positions of the gray wolves on the prey are determined. Then, Operations of continuously updateing of weight and threshold of the neural network are converted to operations of changing positions of gray wolves. By doing this, global optimal solutions about the neural network are obtained. The specific steps of the GWO optimized BP neural network are as follows:

① Initialize the parameters. According to the structure of BP neural network, calculate the dimensions of the gray wolf individual position information, the gray wolves population size, the maximum number of iterations, the upper and lower bounds of the gray wolves dimensions. Then, randomly initialize positions of grew wolves.

② Determine the fitness function and the excitation function (for hidden layer nodes and output nodes) of the BP neural network. In this paper, the root mean square error (RMSE) is selected as the fitness function; the *Sigmoid()* function is selected as the excitation function of the hidden layer and the output layer.

③ Calculate fitness values.
(a) Select gray wolves α, β and δ;
(b) update the position information of the remaining gray wolves according to formula (2) - (8);
(c) update the values of parameters A, C and a.

④ Record the error between the training samples and the testing samples. Meanwhile, the corresponding position of the optimal gray wolf is also recorded.

⑤ In each dimension, determine whether the position of an individual gray wolf is out of bound. If so, set the value of the cross-border to the upper or lower bound of the corresponding dimension.

⑥ Repeat step③ - step⑤, until the number of iterations reaches the maximum value or the error value reaches the preset one.

⑦ The final results are the final position of the gray wolf α, the positions of the gray wolf α in iterations of the training process, the minimum error of the location of the gray wolf and the error between the training samples and the test samples.

By optimized with the GWO algorithm, feasible initial weight and threshold of BP neural work are generated, and consequently, problems of local minimal and slow coverage are properly resolved.

c) Configurations of the DF.GWO-BPNN Classifier: When the BP neural network is used to solve practical problems, the main concerns are the complexity of problems to be

resolved, the number of nodes on hidden layers and the number of hidden layers. Hecht-Nielsen [17] has proved that a three-layer BP neural network can perform arbitrary mapping from a m-dimensional space to a n-dimensional space. Therefore, the number of hidden layers in the BP neural network is chosen as one. By repeatedly performing many experiments, we found that we can get high efficiency and the smallest error when the number of hidden layer nodes in the BP neural network is set as 4. Therefore, the BP neural network structure in this paper is 5-4-1. That is, in our BP neural network, there are 5 nodes (number of variables) in the input layer, 4 nodes in the hidden layer, and 1 output node.

As for the optimization parameters of the GWO algorithm in this paper, we set the number of wolves in the wolves group as 100; the number of maximum iterations as 80. Meanwhile, we use the *sigmoid()* function to activate the BP neural network. The output values of the output node of the BP neural network are in the interval of (0, 1). By doing this, the URL of a website with output node value closes to 0 is recognized as a phishing URL; the URL of a website with output node value closes to 1 is recognized as a legal URL. Finally, for URL with recessive features, if the value of the output node of *DF.GWO-BPNN Classifier* (R_s) is less than 0.5, it will be marked as a phishing URL; Otherwise, R_s equal or greater than 0.5, it will be marked as a legal URL.

D. Comprehensive Evaluation

In this section, the dual feature evaluation mechanism is used to comprehensively evaluate the dominant features and the result of recessive features processed by the *DF.GWO-BPNN Classifier*. The result of the dual feature evaluation mechanism is the final result of our proposed DF.GWO-BPNN model. For a given URL, the vector (D, R_s) is used to represent its signature. Where, D=(D_1, D_2, D_3, D_4) records the extracted dominant features of this URL; R_s is the result of recessive features processed by the DF.GWO-BPNN Classifier. Specifically, the value of vector (D, R_s) is calculated as follows:

$$D_1 = \begin{cases} 1 & f = true \\ 0 & f = flase \end{cases} \tag{9}$$

$$D_2 = \begin{cases} 1 & n \leq 54 \\ 0.5 & 54 < n \leq 75 \\ 0 & n > 75 \end{cases} \tag{10}$$

$$D_3 = \begin{cases} 1 & s = true \\ 0 & s = false \end{cases} \tag{11}$$

$$D_4 = \begin{cases} 1 & da < 0.5 \ or \ dv < 1 \\ 0 & other \end{cases} \tag{12}$$

$$R_s = \begin{cases} 1 & ov < 0.5 \\ 0 & ov \geq 0.5 \end{cases} \tag{13}$$

Where f indicates that the URL is in IP form; n represents the number of characters in the URL; s represents the website of the URL using *mailto()* or *mail()* to send user information; *da* indicates the domain name age, *dv* indicates the domain name validity period (year); *ov* indicates the value of the *GWO-BPNN Classifier* output node.

In the dual feature evaluation mechanism, we use percentage system to give the final evaluation of a URL. In the percentage system, dominant features and recessive features are distributed with the weight of 50 respectively. Since there are 4 dominant features, each dominant feature is given the weight of 12.5. We use W to record the weight. By this, the weight of dominant features are W_D=50 (W_{D_1}=W_{D_2}=W_{D_3}=W_{D_4}=12.5);the weight of recessive features processed by the *DF.GWO-BPNN Classifier* is W_{R_S}=50. Based on the above weight distribution, the final evaluation of a URL can be calculated as (14):

$$Z = \sum (D_i \cdot W_{D_i}) + R_S \cdot W_{R_s} (i = 1, 2, 3, 4) \tag{14}$$

Finally, if the comprehensive evaluation value of Z is greater than 60, the corresponding website of the URL is marked as a phishing website.

IV. Experimental results

This section presents the experimental results on testing the proposed DF.GWO-BPNN model. The test environment of this section consists of an Intel Core CPU (i7 at 3.6 GHz), 16GB RAM and Windows 10 OS. Meanwhile,the matlab (7.0) is used to write our experimental programs.

A. Descriptions of dataset and Evaluation indices

The experimental dataset in this paper is selected from Phishtank and the International Anti-Phishing Organization (APWG) and Security Alliance [18] with 4500 forward samples (phishing URLs) and 4300 reverse samples (legal URLs). Furthermore, this experimental dataset is divided into two parts, the training dataset and the testing dataset. The training dataset (with 2500 forward samples and 2300 reverse samples) is used to train the *GWO-BPNN Classifier*. The testing dataset (with 2000 forward samples and 2000 reverse samples) is used to verify the performance of the *GWO-BPNN Classifier* and the performance of the entire DF.GWO-BPNN model. As lists in Table I, the forward samples in the testing dataset are divided into 4 groups with proportion of 10% , 20%, 30% and 40% respectively; the reverse samples in this dataset are evenly divided into 4 groups with 500 samples in each group.In Table I, *PS* represents a forward sample and *RS* represents a reverse sample.

TABLE I
COMPOSITION OF THE TESTING DATASET

PS ratio	*PS* number	*RS* ratio	*RS* number
10%	200	25%	500
20%	400	25%	500
30%	600	25%	500
40%	800	25%	500

Experiments will generate 4 possible results (*TP*, *TN*, *FP* and *FN*). *TP* represents the number of phishing URLs (in forward samples) been correctly predicted as phishing ones; *TN*

represents the number of legitimate URLs (in reverse samples) been correctly predicted as legitimate ones; *FP* represents the number of phishing URLs (in forward samples) been predicted as legitimate ones (false negative rate); *FN* represents the number of legitimate URLs (in reverse samples) been predicted as phishing ones (false alarm rate). *W* is the number of overall URLs be recognized. By this, we can derive that *W* is the sum of *TP*, *TN*, *FP* and *FN* (i.e. *W=TP+TN+FP+FN*). As defined by (15), the overall recognition accuracy rate (abbreviated by Accuracy) is defined as the ratio of the number of correctly categorized URLs (*TP+TN*) to the number of all URLs (*W*).

$$Accuracy = (TP + TN)/W \tag{15}$$

The forward sample recognition rate (*P.Acc*) is defined as (16):

$$P.Acc = TP/(TP + FP) \tag{16}$$

The total false negative rate (*MR*) is defined as (17):

$$MR = FP/W \tag{17}$$

The root mean square error (*RMSE*) is defined as (18):

$$RMSE = \sqrt{\frac{\sum(A_i - D_i)^2}{N}} \tag{18}$$

Where A_i is the number of all samples to be detected; D_i is the number of samples been detected; N is the number of tested datasets.

B. Performance of GWO-BPNN Classifier

In this experiment, training dataset with 2500 forward samples and 2300 reverse samples is firstly used to train the *GWO-BPNN Classifier*.Then, the testing dataset (as listed in Table I) with 2000 forward samples and 2000 reverse samples is used to evaluate the performance of the *GWO-BPNN Classifier*. Table II lists the experimental results, Where *TNS* represents the total number of samples.

TABLE II
EXPERIMENTAL RESULTS OF THE *GWO-BPNN Classifier*

TNS	*PS* number	Accuracy	*P.Acc*	*MR*	*RMSE* of *PS*
700	200	97.71%	98.00%	0.57%	9.14
900	400	98.33%	98.25%	0.78%	
1100	600	97.55%	98.33%	0.91%	
1300	800	97.92%	98.38%	1.00%	

From this table, we can see that the overall recognition rate and the overall false negative rate of the GWO-BPNN model are relatively stable with the change of the sample number; the overall recognition level is higher; and the number of underreported samples is less; the forward sample recognition rate is high. At the overall level, the GWO-BPNN model is optimal for forward sample identification.

C. Performance of the entire DF.GWO-BPNN model

In order to improve the recognition rate of the model, the dual-feature evaluation mechanism is used to comprehensively evaluate dominant features and recessive features of input URLs. Table III gives the final results of performance of our proposed DF.GWO-BPNN model. The configurations of this experiment are the same as the ones in Section 4.2.

TABLE III
EXPERIMENTAL RESULTS OF THE DF.GWO-BPNNM MODEL

TNS	*PS* number	*Accuracy*	*P.Acc*	*MR*	*RMSE* of *PS*
700	200	98.29%	99.00%	0.29%	7.31
900	400	98.56%	98.75%	0.56%	
1100	600	97.91%	98.67%	0.73%	
1300	800	98.08%	98.63%	0.85%	

From this table, we can see that the recognition accuracy of the DF.GWO-BPNN model is significantly increased when it is compared with the *GWO-BPNN Classifier*. Meanwhile, the RMSE of Positive samples of the DF.GWO-BPNN model is sharply decreased when it is compared with the ones of the *GWO-BPNN Classifier*.

D. Performance comparisons among different models

In order to further verify the efficiency of our proposed model, in this subsection, 6000 test samples with 3000 phishing sites are tested. Meanwhile, we compare the performance of DF.GWO-BPNN with some popularly used classification models. They are the SVM classification model [19], the PSO-BPNN model (particle group optimized BPNN model) [20] and the traditional BPNN model [7]. Table IV lists the final experimental results. In this table, we also list the results of the *GWO-BPNN Classifier* (the third line of this table).

TABLE IV
EXPERIMENTAL RESULTS OF DIFFERENT MODELS

models	*Accuracy*	*P.Acc*	*MR*	*RMSE*
DF.GWO-BPNN	98.78%	98.70%	0.65%	36.59
GWO-BPNN	98.53%	98.43%	0.78%	44.10
PSO-BPNN	98.08%	98.03%	0.98%	57.52
BPNN	96.80%	96.57%	1.72%	96.25
SVM	96.93%	96.97%	1.52%	92.01

From this table, we can see that our proposed DF.GWO-BPNN model is outperformed the other models for all testing indicators. For better illustrating the overall recognition rate (*Accuracy*) and forward sample recognition rate (*P.Acc*) of the DF.GWO-BPNN model, we use Fig. 2 to compare the values of the two indicators among different classification models.

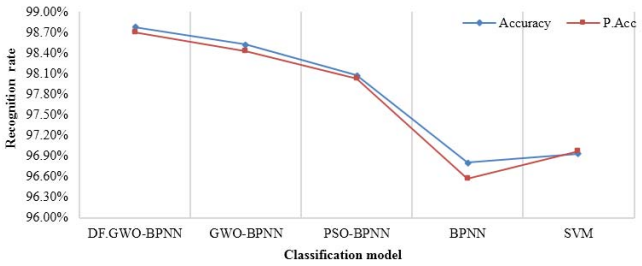

Fig. 2. Experimental results of Accuracy and *P.Acc* among different classification models

V. Conclusion and Future Works

In this paper, we proposed DF.GWO-BPNN, an effective phishing website detection model based on improved BP neural network and dual feature evaluation. On constructing this model, we used the GWO algorithm to overcome the shortages of local minimum and slow learning convergence of the BP neural network. In order to elevate the phishing websites recognition rate, we used the dual feature evaluation mechanism to comprehensively evaluate the extracted features from URLs. Meanwhile, the black and white list was used to improve the efficiency of the proposed model. The DF.GWO-BPNN model was compared with some existing phishing website detection models. The experimental results demonstrated that our proposed model was accurate and strong adaptability for detecting phishing websites. Although the GWO optimization algorithm in this model can get the global optimal, but the convergence speed is reduced in the later stages. We have already noticed that extracting more explicit features from URLs and enriching the black and white list can alleviate this problem. So, in the later of our work, more features will be extracted from URLs and more reasonable black and white list will be construct.

References

[1] Javier Vargas, Alejandro Correa Bahnsen, Sergio Villegas, and Daniel Ingevaldson. Knowing your enemies: leveraging data analysis to expose phishing patterns against a major us financial institution. In *Electronic Crime Research*, pages 1–10, 2016.

[2] J. A. Chaudhry, S. A. Chaudhry, and R. G. Rittenhouse. Phishing attacks and defenses. *International Journal of Security and Its Applications*, 10(1):247–256, 2016.

[3] A. Naga Venkata Sunil and Anjali Sardana. A pagerank based detection technique for phishing web sites. In *Computers Informatics*, pages 58–63, 2012.

[4] Mona Ghotaish Alkhozae and Omar Abdullah Batarfi. Phishing websites detection based on phishing characteristics in the webpage source code. *International Journal of Information Communication Technology Research*, 1(6), 2011.

[5] Pawe Mazurek and Roman Z. Morawski. Application of naive bayes classifier in fall detection systems based on infrared depth sensors. In *IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pages 717–722, 2015.

[6] Guang Xiang, Jason Hong, Carolyn P. Rose, and Lorrie Cranor. Cantina+:a feature-rich machine learning framework for detecting phishing web sites. *Acm Transactions on Information System Security*, 14(2):1–28, 2011.

[7] Luong Anh Tuan Nguyen, Lam To Ba, Huu Khuong Nguyen, and Minh Hoang Nguyen. An efficient approach for phishing detection using single-layer neural network. *Advanced Technologies for Communications*, pages 435–440, 2014.

[8] Wen Jin, Zhao Jia Li, Luo Si Wei, and Han Zhen. The improvements of bp neural network learning algorithm. In *International Conference on Signal Processing Proceedings, 2000. Wccc-Icsp*, pages 1647–1649 vol.3, 2002.

[9] Simon Haykin. *Neural Networks: A Comprehensive Foundation (3rd Edition)*. Macmillan, 1998.

[10] Shahrzad Saremi, Seyedeh Zahra Mirjalili, and Seyed Mohammad Mirjalili. Evolutionary population dynamics and grey wolf optimizer. *Neural Computing Applications*, 26(5):1257–1263, 2015.

[11] Seyedali Mirjalili. *How effective is the Grey Wolf optimizer in training multi-layer perceptrons*. Kluwer Academic Publishers, 2015.

[12] Li Zhu Zhou and Ling Lin. Survey on the research of focused crawling technique. *Computer Applications*, 2005.

[13] Doaa Hassan. On determining the most effective subset of features for detecting phishing websites. *International Journal of Computer Applications*, 122(20):1–7, 2015.

[14] G. M. Komaki and Vahid Kayvanfar. Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time. *Journal of Computational Science*, 8:109–120, 2015.

[15] Hossam M. Emary, E.and Zawbaa, Crina Grosan, and Abul Ella Hassenian. *Feature Subset Selection Approach by Gray-Wolf Optimization*. Springer International Publishing, 2015.

[16] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 69(3):46–61, 2014.

[17] HechtNielsen. Theory of the backpropagation neural network. *Neural Networks*, 1(1):445–445, 1988.

[18] Peter Cassidy. Statement of support anti-phishing working group (apwg). *Journal of Digital Forensic Practice*, 1:75–76, 2006.

[19] Huajun Huang, Liang Qian, and Yaojun Wang. A svm-based technique to detect phishing urls. *Information Technology Journal*, 11(7):921–925, 2012.

[20] Wenwu Chen, Xu An Wang, Wei Zhang, and Chunfen Xu. Phishing detection research based on pso-bp neural network. *International Conference on Emerging Internetworking*, pages 990–998, 2018.