Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

NEURAL NETWORK WITH PSO BASED TRAINING TO DETECT SPOOFING WEBSITES

[1]SOZAN ABDULLAH MAHMOOD, [2]NOOR GHAZI M. JAMEEL, [3]SHAIDA JUMA SAYDA

[1]Assistant Professor. Department of Computer Science, College of Science, University of Sulaimani, Kurdistan Region, Iraq

[2]Lecturer. Department of Computer Networks, Technical College of Informatics, Sulaimani Polytechnic University, Kurdistan Region, Iraq

[3]MSc. Student, University of Sulaimani, Kurdistan Region, Iraq

E-mail: [1]sozan.mahmood@univsul.edu.iq, [2]noor.ghazi@spu.edu.iq, [3]shaida.shaida@yahoo.com

ABSTRACT

With the advent of Internet, various online attacks were increased among them and the most well-known is a spoofing attack. Web spoofing is a type of spoofing in which fake and spoofing websites made by means of fraudsters to duplicate real websites. Spoofing websites represent legitimate websites which attract users into visiting fake websites to steal users sensitive, personal information or install malwares in their devices. The scammers will use the stolen information for illegal purposes. The specific intention of this paper is to build a new intelligent system that detects and recognize between trusted and spoofing websites which try to mimic the trusted sites because it is very difficult to visually recognize whether they are spoofing or legitimate. This paper deals with the detection of spoofing websites using Neural Network (NN) trained with Particle Swarm Optimization (PSO) algorithm. An Information gain algorithm is used for feature selection, which was a useful step to remove the unnecessary features and reduce time. The Information gain seems to improve the classification accuracy via reducing the number of extracted features and used as an input for training the NN using PSO. Training neural network using PSO provides less training time and high accuracy which achieved 99.18% compared to NN trained with back propagation algorithm which takes more time for training and less accuracy which was 98.20%. The proposed technique is evaluated with a dataset of 2500 spoofing sites and 2500 legitimate sites. The results show that the technique can detect over 99.18% spoofing sites with NN trained using PSO.

Keywords: *Web Spoofing, Information Gain, Neural Network, Particle Swarm Optimization*

1. INTRODUCTION

The world wide web is a global information network. This network consists of a collection of web sites that users can access through the Internet [1]. Web site spoofing is the act of replacing a world wide web site with a forged, probably altered, copy on a different computer. The attacker's web server to sit between the victim and the rest of the web. This kind of arrangement is called a 'man in the middle attack [2]. Spoofing sites are imitations of real commercial sites, intended to deceive the authentic sites. The objective of spoofing site is identity theft, capturing users' account information by having them log in to a fake site. Commonly spoofed web sites include eBay, PayPal, and various banking and escrow service providers. The intention of these sites is online identity theft: deceiving customers of the authentic sites into providing their information to the fraudster. These spoofing sites are used to attack millions of internet users [3]. Spoofing web site is a copy of any legitimate website. Access to the imitated web site is done through the attacker's machine, in which the victim's activities, including passwords or account numbers that the victim enters are monitored by the attacker. The attacker can also cause false or misleading data to be sent to web servers in the victim's name, or to the victim in the name of any web server. Cyber criminals also use spoofed websites to deploy malware into the visitor's PC thus making it as a part of their botnet.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In spoofing, an attacker gains unauthorized access to a computer or a network by making it appears that a malicious message has come from a trusted machine by "spoofing" the address of that machine [4]. This work presents a new, intelligent and fast approach to classify a web site as a spoofing web site or not by neural network (NN) trained with particle swarm optimization (PSO). The proposed system uses minimum number of features in short training time with high accuracy. The proposed approach is used to classify the websites depending on (21) features selected from 36 features using Information Gain feature selection algorithm. Particle swarm optimization algorithm is used to train the neural network to get the optimal set of weights for the NN and apply Feed forward NN for web spoofing detection. This paper is organized as follows: In section 2 Literature review is explained. Section 3 methodology and background for information gain, NN and PSO are explained. Section 4 presents the proposed model design, implementation and extracting features. In section 5 a set of tests have been performed to evaluate the system performance and accuracy. The results of some experimental tests are listed and discussed. More over the effects of the involved system parameters are illustrated. Finally, in section 6 provides concluding remarks and recommendation of future works

2. LITERATURE REVIEW

Over the past few years' number of researchers presented many related works in web spoofing attack detection and classification. Some of these papers work on the detection of web spoofing in general and others work on web spoofing as part of one of the most dangerous type of attack nowadays which is called phishing attack.
Nguyen et al [5] proposed an efficient approach which used single-layer neural network for detecting phishing web sites, the system was evaluated using a dataset of 11,660 phishing sites and 10,000 legitimate sites. The results show that the proposed system can detect over 98% phishing sites. Ramanathan and Wechsle [6] proposed an algorithm, which incorporates the power of natural language processing and machine learning techniques and used Probabilistic Latent Semantic Analysis to build a topic model. Topics are used as features to build the classifier using Adaboost and co-training. The experimental results show that the proposed system achieved high performance. The work by Garera et al [7] used logistic regression and 18 features to classify phishing URLs. The features include the existence of certain keywords in the URL, some features based on Google's PageRank and web page quality guidelines. the pre-computed page based features from Google's proprietary infrastructure were used, that they call Crawl Database. The results show that the classification accuracy was 97.3% over a set of 2500 URLs. Zhang et al [8] presented CANTINA. It is a content-based approach to detect phishing websites, this approach used TF-IDF information retrieval algorithm and the Robust Hyperlinks algorithm with 8 features (4 content-related, 3 lexical, and 1 WHOIS-related). The results show that they system can detect approximately 95% of phishing websites. Pradeepthi et al [9] the dataset for the proposed system was collected from public repository DMOS, which has a large collection of genuine URLs from different domains, the phishing URLs were collected from the phishtank, which is a collection of phishing URLs. A total of 10,000 URLs were collected, of which 6000 were genuine and 4000 are fake. There were a total of 27 features which belong to various categorized, like lexical, domain based (collected from DNS server), network based and URL feature based. Binary Particle Swarm Optimization (BPSO) technique used for the detection of phishing URLs, a dataset of 10,000 URLs was constituted and an accuracy of 98.7% was achieved by using this method. Mao et al. [10] used extracted features from the Cascading Style Sheet (CSS) of web pages, and effective feature sets were selected for similarity rating. The approach was used in the Google Chrome browser for analyzing suspicious web pages. The results show the effectiveness of the algorithm and low performance overhead. Nivaashini [11] used deep Boltzmann machine learning as an automatic unknown URL is either a phishing URL or benign URL. DNN binary classifier performed well with True Positive (TP) rate of 97.62% and False Positive (FP) rate of 5.27%. Alejandro et al [12] used URLs as input for machine learning models applied for phishing site prediction. feature-engineering approach followed by a random forest (RF) classifier is compared against a novel method based on recurrent neural networks. The recurrent neural network approach provides an accuracy rate of 98.7% even without the need of manual feature creation, beating by 5% the random forest. Both models showed great statistical results. On one hand the RF had an F1-Score of 0.93 and an accuracy of 93.5%, while the Long Short Term Memory network (LSTM) had F1-Score of 0.98 and an accuracy of 98.7%. The training time for RF

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

less than 3 minutes, while LSTM required 238 minutes.

Most of the related works used machine learning algorithms with different number of features to detect spoofing websites. In this work, an optimized machine learning algorithm is used for detection purpose in order to reduce training time and achieve high performance.

3. METHODOLOGY

In this section, the theoretical aspects of methods used in this paper are explained.

3.1 Feature Selection

Feature selection, sometimes called variable selection, attribute selection or feature subset selection. It is the process of selecting relevant features in terms of a target learning problem. The purpose of feature selection is to remove redundant and irrelevant features. Irrelevant features are features that provide no useful information about the data, and redundant features are features that provide no more information than the currently selected features. In other words, redundant features do provide useful information about the data set, but the information has been provided by the currently selected features [13].

Feature selection algorithms can be classified into three categories: embedded approaches, wrapper approaches, and filter approaches. The filter model relies on the general characteristics of data and evaluates features without involving any learning algorithm. The wrapper model requires having a predetermined learning algorithm and uses its performance as evaluation criterion to select features. The embedded model incorporates variable selection as a part of the training process, and feature relevance is obtained analytically from the objective of the learning model [13]. In This paper, information gain algorithm used as a filter approach for feature selection step, to select the best feature subset for the learning algorithm.

3.1.1 Information Gain

Information Gain is supervised univariate feature selection algorithm of the filter model which is a measure of dependence between the feature and the class label. It is one of the most powerful, easy to compute and simple to interpret feature selection technique. Information Gain (IG) of a feature X and the class labels Y is calculated as in equation (1).

$$IG(X,Y)=H(X) - H(X|Y) \quad (1)$$

Entropy (H) is a measure of the uncertainty associated with a random variable. H(X) and H(X/Y) is the entropy of X and the entropy of X after observing Y, respectively. Entropy (H) is calculated as in equation (2).

$$H(X) = - \sum P(X\ i) \log2 (P(X\ i)) \quad (2)$$

The maximum value of information gain is 1. A feature with a high information gain is relevant. Information gain is evaluated independently for each feature and the features with the top-k values are selected as the relevant features [14].

3.2 Artificial Neural Network

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. Traditionally neural network was used to refer as network or circuit of biological neurons, but modern usage of the term often refers to ANN [15].

There are three essential components on which ANN is built, input, activation functions of the unit, network architecture and the weight of each input connection. Given that the first two aspects are fixed; the behavior of the ANN is defined by the current values of the weights. The weights of the net, to be trained, are initially set to random values, and then instances of the training set are repeatedly exposed to the net. The values for the input of an instance are placed on the input units and the output of the net is compared with the desired output for this instance. Then, all the weights in the net are adjusted slightly in Activation Functions the direction that would bring the output values of the net closer to the values for the desired output [16].

3.3 Particle Swarm Optimization

Particle swarm optimization (PSO) is a swarm intelligence algorithm inspired by the social behavior of birds flocking or fish schooling. In PSO, the problem is represented as a particle, which is represented by a vector or an array. Particles move in the search space to search for the optimal solutions. During the movement, each particle can remember its best experience. The whole swarm searches for the optimal solution by updating the position of each particle based on the best experience of its own and its neighboring particles. PSO is a simple but powerful search

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

technique, which has been successfully used and applied to solve problems in a variety of areas.

In PSO, each candidate solution of the problem is encoded as a particle moving in the search space. The whole swarm searches for the optimal solution by updating the position of each particle based on the experience of its own and its neighboring particles. Generally, a vector xi = (xi1; xi2; :::; xiD) is used in PSO to represent the position of particle i, where D is the dimensionality of the search space and a vector vi = (vi1; vi2; :::; viD) represents the velocity of particle i. During the search process, the best previous position of each particle is recorded as the personal best called pbest and the best position obtained by the swarm thus far is called gbest. The swarm is initialized with a population of random solutions and searches for the best solution by updating the velocity and the position of each particle according to the equations (3) and (4).

$$X\,id = X\,id + Vid \quad (3)$$

$$Vid = W * Vid + C1 * r1i * (Pid - X\,id) + C2 * r2i * (Pgd - X\,id) \quad (4)$$

where t denotes the t iteration in the search process. d ЄD denotes the dth dimension in the search space. c1 and c2 are acceleration constants. r1i and r2i are random values uniformly distributed in [0, 1]. pid and pgd represent the elements of pbest and gbest in the dth dimension, respectively. w is inertia weight. The velocity vtid is limited by a predefined maximum velocity, vmax and vtid Є [-vmax, vmax] [17], [18].

The elements used in PSO are the followings [19]:

- Particle-- the particle is defined as Pi Є[a,b] where i=1,2,3 .... D and a, b Є R. Here D is for dimension and R is for real numbers.
- Fitness Function---Fitness Function is the function used to find the optimal solution. Usually it is an objective function.
- Local Best---It is the best position of the particle among its all positions visited so for.
- Global Best---The position where the best fitness is achieved among all the particles visited so Velocity Update---Velocity is a vector to determine the speed and direction of the particle. Velocity is updated by the equation (4).
- Position Update---All the particles try to move toward the best position for optimal fitness. Each particle in PSO updates their positions to find the global optima. Position is updated by equation (3).

4. PROPOSED SYSTEM

In this paper, an approach for website spoofing detection is introduced. Neural Network using PSO, as a training algorithm, Feed forward Neural Network has been used for detecting spoofing websites. To detect spoofing websites using neural network the two phases (training and testing) need to be done. The steps used for detecting websites using a feed forward neural network.

In this work, the system parameters which were involved in the training and testing phases of neural network are as the followings:

- The parameters used in training the NN using PSO.
  The parameters utilized for training the NN using PSO comprise: the input nodes, maximum iterations, number of particles and hidden nodes. The features extracted from URL, HTML source code, WHOIS and ranker values, were fed into the NN as input data. These considered parameters generated optimal weights which involved in the testing phase.
- The parameters used in training the NN using backpropagation algorithm.
  The parameters used in the training phase of NN using backpropagation are: input nodes, learning rate and hidden nodes. The features extracted from the URL, HTML source code, WHOIS and ranker of websites, were fed into the NN as input data. These considered parameters generated weights which involved in the testing phase.

The proposed system model as shown in Figure 1 consists of three stages, namely, pre-processing, features selection using information gain algorithm and classification stage for spoofing websites detection in this approach, 36 features are implemented as a binary value (0 or 1); with a value 1 indicating this feature appeared on the tested website and 0 for the non-appearance case. Figure 1 shows the proposed system for spoofing website detection.

4.1 Features Used in Website Classification

Spoofing detection techniques are based on identifying a set of features are usually involving the URL features, HTML source code features, WHOIS features and page ranking. In this work a list of 36 features are extracted; they are binary features. All of these features are extracted using Visual C# programming language. These features are briefly described in table 1.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

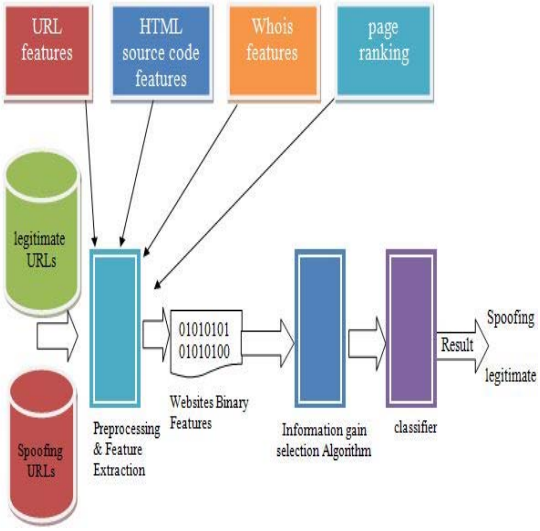

Figure 1: Proposed System for Spoofing Websites Detection

Table1: Extracted Features of Spoofing and Legitimate Website Dataset

No.	Feature Description	No.	Feature Description
1	Port number	19	Domain token count
2	Length of URL	20	Path token count
3	'.' in path	21	'?' in URL
4	'/' in URL	22	Google page rank
5	'=' within URL	23	Alexarank
6	'@' in URL:	24	Disable right click
7	'%' in URL	25	page rank
8	(-) symbol to Domain	26	LDigit [0-9] in Host
9	',' in path	27	Keyword-based URL
10	';' in path	28	IP Based Host
11	'.' In host	29	Hex Based Host
12	Length of host	30	Redirect page
13	Length of path	31	Request URL
14	Suspicious "//" in URL Path	32	Script
15	Length of sub domain	33	Mouse over
16	Dash within hostname	34	DNS record
17	Sub domain	35	Protocol
18	Age of domain	36	Number of Domains in the URL

4.2 Pre-Processing

The preprocessing step consists of three modules:

4.2.1 Dataset preparation

In this paper, websites are used as a dataset to work on. The dataset consists of various websites such as banking sites, online shopping sites, reservation sites, etc. These websites classified to two main datasets:

- Legitimate Dataset: Legitimate dataset was built using DMOZ URL Classifier. DMOZ is the largest, most comprehensive human-edited directory of the Web. It was known as the Open Directory Project (ODP). It contains a categorized list of Web URLs. Their listings are updated on monthly basis and published in RDF files. DMOZ provides the means for the Internet to organize itself.
- Spoofing Dataset: Spoofing URLs was downloaded from PhishTank site (https://www.phishtank.com). The first collected spoofing URLs from January 1 to December 31 of 2015. Spoofing dataset was downloaded as .csv file then the csv file of the dataset is converted and data is stored in xls format.

4.2.2 Data cleaning and source code retrieval

In this module, Redundant URLs and the URLs which their source code or WHOIS database values could not be retrieved, and URLs which are blocked or expired were removed. In the dataset of spoofing URLs, there are many URLs, which have been expired or blocked so extracting features from these links will be impossible.

4.2.3 Feature extraction

In this module, both spoofing and legitimate datasets that consists of 5000 websites were used to extract the 36 features. The features are extracted from the URL, HTML source code, WHOIS database and using Google Rank and Alexa Rank. Then the extracted features are stored as 0 or 1 according to the conditions and rules of the features. The rules are presented in [20], [21], [22], [23]. Each value represented with binary values, which means that the feature or attribute is present or not. Then the extracted 36 features for 5000 URLs are saved. figure 2 and figure 3 show 36 features extracted for legitimate and spoofing websites, respectively.

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

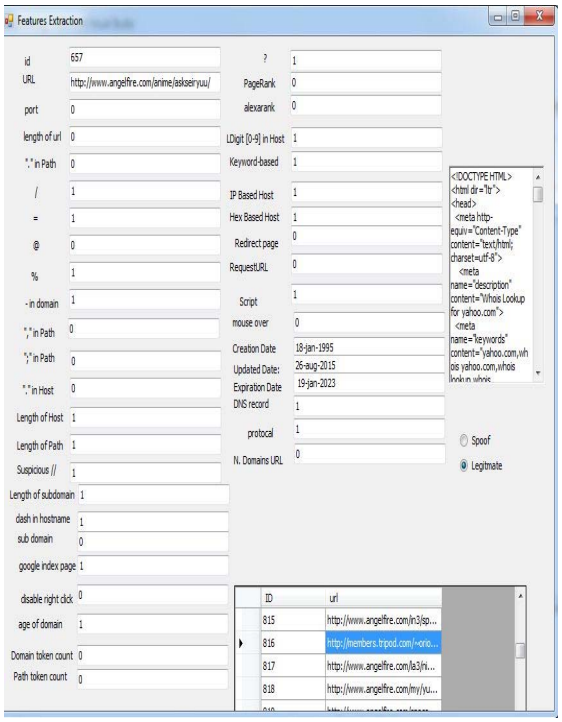

Figure 2: Extraction of a legitimate website Features

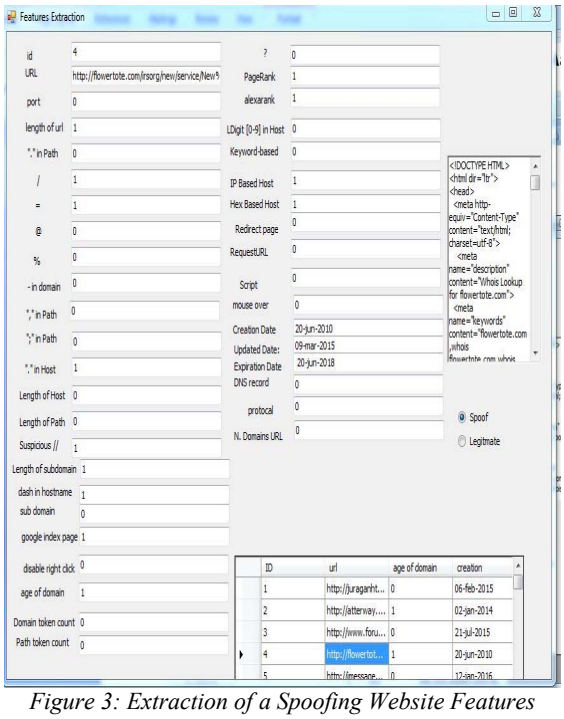

Figure 3: Extraction of a Spoofing Website Features

4.3 Features Selection Using Information Gain Algorithm

In this step, a set of features were selected to be used as an input to the classifier. The method called feature subset selection which was applied to reduce the data size. The goal of feature subset selection is to find a minimum set of features. In this work, information Gain algorithm was used as feature selection method. For each feature, the algorithm produced a value between 0 and 1. 0 means the feature is a weak feature and 1 means the feature is strong feature. In this proposed system, information gain value was calculated for the 36 features based on 5000 legitimate and spoofing websites and just 21 features with the high information gain values were selected for the classification purpose, since the system achieved high performance using this set of features.

4.4 Classification of Spoofing and Legitimate Web Sites

Classification of websites into spoofing and legitimate consists of two phases:

- Phase 1: Training NN using PSO.
- Phase 2: Test Phase.

4.4.1 Training the neural network using PSO

In PSO algorithm, each particle had a virtual position that represents a possible solution to some minimization problem. In the case of a neural network, a particle's position represents the values for the network's weights and biases. The goal was to find position/weights so that the network generates computed outputs that match with the outputs of the training data. In each iteration, every particle moves to a new position. A particle's movement is based on the particle's current speed and direction (velocity), the best position discovered by the particle at any time and the best position found by any of the other particles in the swarm. Algorithm 1. Illustrates the steps of training the NN using PSO.

The PSO algorithm is vastly various than any of the traditional methods of training. PSO does not just train one network, but rather training networks. PSO builds a set number of ANN and initializes all network weights to random values and starts training each one. On each pass through a dataset, PSO compared each network's fitness. The network with the highest fitness was considered the global best. Each neuron contains a position and velocity. The position corresponds to the weight of a neuron. The velocity is used to update the weight. If a neuron is further away, then it will adjust its weight more than a neuron that is closer to the global best.

The number of samples prepared to train and test the system was 5000 samples, %60 of these samples had been used to train the neural network using PSO, and %40 of the samples had been used to test the system using neural network.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

For training phase, 3000 samples are used to train a NN using PSO, as a result the optimal set of weights is stored in a text file.

Algorithm 1: Training NN with PSO

Begin:
- Read Data
- Initialize the value of parameters for neural network (No.input nodes ,No. hidden nodes ,No.output nodes, Epoch value, exiterror value)
- itration=0
- while iteration < epoch
  Initialize each particle to random state (position, velocity, error, best-position, best-error)
  save best position of any particle (global-best)

loop until done
  for each particle in swarm
    compute new particle velocity Eq.3
    use new velocity to compute new position Eq.4
    compute error of new position
    if new error better than best-error
    best-position = new position
    if new error better than global-best
    global-best = new position
  end for
end loop
  save global-best position
  send global-best position as a weight in NN
  calculate error
If error < exiterror then exit from while
  Increment iteration
End while
End

4.4.2 Test phase

In the test phase, the test URLs were represented in terms of binary feature vector consists of 21 features which selected by information gain algorithm. The binary feature vector used as input to the feed forward neural network with the optimal set of weights which were calculated from training phase by using PSO algorithm, to classify the URL into spoofing or legitimate. The number of samples prepared to test the system is 2000 URLs.

5. RESULTS

The system is based on the features extracted from legitimate and spoofing websites. The features were evaluated and selected using information gain algorithm so as to select the best features which the system perform high accuracy and less training and testing time. Table 2 shows the information gain value for 36 features. 21 features with higher information gain values were used as input to the NN. The experiment results were based on the effect of features, the training and testing phases used for the classifier with their parameters, the results of some conducted experimental tests and various cases were studied to choose the most suitable models and assessment of the performance of the proposed system, is described.

Table 2: Information Gain Value for 36 Features

Feature	Information Gain Value
PageRank	0.997453861
Alexarank	0.997453861
Number_of_Domains_in_the_URL	0.379503879
length_of_url	0.11886803
Length_of_host	0.095602383
age_of_domain	0.091160523
LDigit_0_9_in_Host	0.07995388
dash in url	0.068914304
Keyword_based	0.030460365
'?' in URL	0.016392436
dot_in_path	0.015267851
dot_in_Host	0.013456551
'/' in URL	0.013327373
Domain_token_count	0.010478842
'@' in URL	0.009667125
_;_in_Path	0.009261623
'=' within URL	0.008856358
(-) symbol to Domain	0.007439796
Length_of_subdomain	0.007238
Suspicious_//	0.005824
% in URL	0.005421
Script	0.003007
Path_token_count	0.00153
',' in path	0.0006
Redirect_page	0.0006
Length_of_path	0.0004
Protocol	0.0002
Port	0.000167
mouse_over	0
RequestURL	0
DNS_record	0
google_index_page	0
sub_domain	0
Hex_Based_Host	0
disable_right_click	0
IP_Based_Host	0

The dataset was divided into two parts, namely training and testing datasets. The training

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

dataset is utilized for learning the system and redirecting the system for making decisions in the testing phase, whereas the testing dataset was used to evaluate the performance of the proposed system. A dataset with 3000 URLs which consists of 1500 legitimate URLs and 1500 spoofing URLs was used for training Neural Network (NN) with Particle Swarm Optimization (PSO). The same dataset was used for training the NN with backpropagation algorithm for a comparison test.

In each subsystem, number of experiments were performed to determine the best result by using different values of each parameter: the input nodes, maximum iterations, number of particles and hidden nodes. Best result was determined by calculating percentage of accuracy for the testing samples using table 3.

Table 3: Performance Calculation Formula

Performance Measure		Description
Percentage % Classification	Accuracy	$\frac{TP+TN}{TP+TN+FP+FN} \times 100$
	True Negative Rate (TNR)	$\frac{TN}{TN+FP} \times 100$
	Recall/True Positive Rate (TPR)	$\frac{TP}{TP+FN} \times 100$
Error Percentage (%)	False Positive Rate(FPR)	$\frac{FP}{FP+TN} \times 100$
	False Negative Rate(FNR)	$\frac{FN}{FN+TP} \times 100$

Where:

TP: represents the number of website correctly classified as legitimate.

TN: represents the number of websites classified correctly as spoofing website.

FP: represents the number of legitimate web sites classified as spoofing website.

FN: represents the number of websites classified as legitimate websites when they were actually spoofing websites.

In the training Phase, PSO was used for to get the optimal weights and biases which provide a minimum error for feed forward neural network. Legitimate and spoofing datasets with 21 features and a class with (1, 0) values was used for recommending for promotion or not. The NN was trained with different number of neurons in the hidden layer and different number of particles, 9 neurons in the hidden layer and 10 particles achieved the better training accuracy as shown in table 4 and table 5. Figure 4 shows the effect of different number of particles on training accuracy.

Table 4: Training NN using PSO with 36 Features and 21 Features

No. Hidden Node	Training Accuracy 36 feature	Training Accuracy 21 Feature
8	0.97301 %	98.14%
9	0.96555 %	99.18%
10	0.95995 %	96.97%
11	0.97423%	97.61%
12	0.97651 %	97.85%
13	0.97543 %	97.73%
14	0.97725 %	97.88%

Table 5: The Effect of Different Number of Particles with 9 Nodes in Hidden Layer

Number of Particles	Training Accuracy
10	99.18%
12	98.23%
14	97.89%
20	97.13%

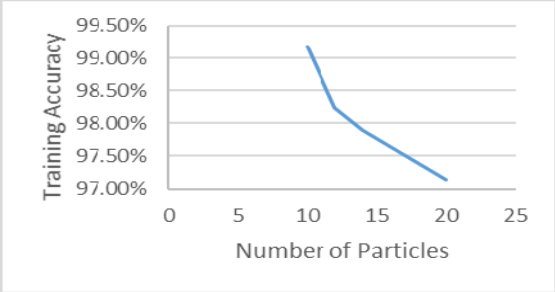

Figure 4: The Effect of Different Number of Particles

For the testing phase, once the neural network was properly trained, it was tested over the test website data set (which is not used in the training phase). The system was tested using feed forward neural network. The confusion matrix for the testing phase is presented in table 6. The table presented the result of the testing phase with testing datasets which consists of 2000 samples. According to the table, it is clear that there are a number of correctly classified instances, with a number of cases that are classified incorrectly in both cases. This makes the system to be fair to the error more than other systems with less incorrectly classified cases, which lead the system to misclassification during test phases. The system achieved higher accuracy with 9 neurons in the hidden layer. Figure 5 shows the test accuracy when NN trained with PSO.

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 6: Confusion Matrix of Testing Phase

No. of Nodes in Hidden Layer	TN Rate	FN Rate	TP Rate	FP Rate	Test Accuracy
8	0.9852	0.0369	0.9631	0.0148	98.14%
9	0.9966	0.0105	0.9895	0.0034	99.18%
10	0.9638	0.0389	0.9613	0.0362	96.97%
11	0.9700	0.0373	0.9627	0.0300	97.61%
12	0.9343	0.0082	0.9918	0.0657	97.89%

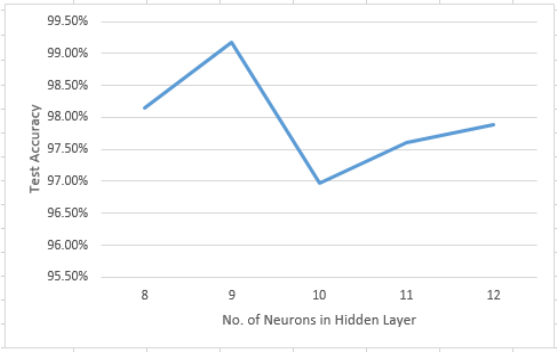

Figure 5: Test Accuracy Using Feed Forward Neural Network

Table 7 illustrate a comparison between the training time for training NN with PSO and training time for NN with backpropagation algorithm in seconds with 21 features as input, different number of nodes in the hidden layer, 1000 iterations and learning rate 0.5. As shown figure 6 the training time with PSO is less than training time with backpropagation algorithm.

Table 7: Training Time Comparison

No. of Nodes in the hidden layer	Training time (NN trained with PSO) In seconds	Training time (NN trained with backpropagation) In seconds
8	6	34
9	9	36
10	11	40
11	14	45
12	16	47

For testing phase, dataset with 2000 instances which consists of 1000 legitimate URLs and 1000 spoofing URLs was used. 21 input features were used to test the system with 9 nodes in the hidden layer. Neural network with PSO achieved 99.18% test accuracy, which was the best performance compared to the neural network NN with backpropagation in which the test accuracy was only 98.20% as shown in table 8.

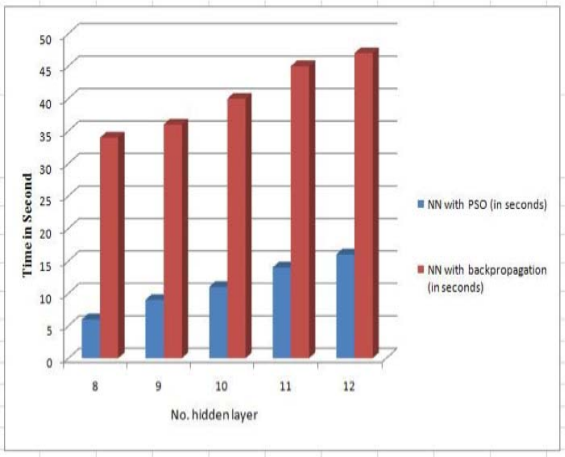

Figure 6: Training Time Comparison

Table 8: Test Accuracy Comparison between NN Trained with PSO and NN Trained with Backpropagation

Classifier	Test Accuracy
Neural Network Trained with PSO	99.18 %
Neural Network Trained with backpropagation	98.20 %

6. CONCLUSION

From the experiments and the results some conclusions have been deduced. Information gain for feature selection is a useful step to eliminate the unnecessary features. The results of the conducted tests indicated that a significant reduction in the network size could be done. One of the most important conclusions for this paper is reducing the number of features extracted and used for training the NN using PSO help to reduce the time needed by the NN using PSO to make the classification and ranking. The second most important conclusion of training neural network using PSO provides good accuracy for testing %99.18 compare to test accuracy for NN trained with backpropagation which achieved %98.20. It showed almost 1% better accuracy than Neural Networks with the same datasets. The time for training is less than the time required for training the system based on neural network using PSO with 9 seconds over back propagation with 36 seconds. In future work suggested using more data set and comparing various machine learning algorithm such as Naïve Bayes, Regression trees, and Deep learning, we can also apply to our methods a toolbar for web spoofing detection, or a desktop application, so that it can run as a background process to be used as an independent spoofing detection tool.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFRENCES:

[1] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs", *World Wide Web Internet Web Inf. Syst.*, Paris, France, June 28–July 1, 2009, pp. 1245–1253.

[2] Peter Stephenson, "Investigating Computer - Related Crime a Handbook for Corporate Investigators", ISBN 0-8493-2218-9, United States, 2000.

[3] Ahmed Abbasi, Jay F. Nunamaker, Jr, Zhu Zhang, "Detecting Fake Websites: The Contribution of Statistical Learning Theory ", *MIS Quarterly,* Vol. 34, No. 3, 2010.

[4] Edward W. Felten, Dirk Balfanz, Drew Dean, and Dan S. Wallach," Web Spoofing: An Internet Con Game", *Department of Computer Science, Princeton University,* 1997.

[5] Nguyen, L.A.T., To, B.L., Nguyen, H.K., and Nguyen, M.H, "An Efficient Approach for Phishing Detection Using Single-Layer Neural Networks", *International Conference on Advanced Technologies for Communications (ATC 2014) IEEE,* Hanoi, Vietnam, October 5-17, 2014, pp.435-440.

[6] Venkatesh Ramanathan, and Harry Wechsler, "PhishGILLNET- Phishing Detection Methodology Using Probabilistic Latent Semantic Analysis, AdaBoost, and Co-training", *EURASIP Journal on Information Security*, 2012, pp.1-22.

[7] S. Garera, N. Provos, M. Chew, and A.D. Rubin, "A framework for Detection and Measurement of Phishing Attacks", 5*th ACM Workshop on Recurring Malcode, WORM"07, ACM*, New York, NY, USA, 2007, pp.1-8.

[8] Yue Zhang, Jason Hong, and Lorrie Cranor, "CANTINA: A Content –Based Approach to Detecting Phishing Web Sites", *International World Wide Web Conference Committee (IW3C2)*, Canada, May 8–12, 2007.

[9] Pradeepthi K.V., and Kannan A., "Detecting Phishing URLs Using Particle Swarm", *Australian Journal of Basic and Applied Sciences*, Vol 10, No. 2, 2016.

[10] Jian Mao, Wenqian Tian, Pei Li, Tao Wei, and Zhenkai Liang, "Phishing Website Detection Based on Effective CSS Features of Web Pages", *International Conference on Wireless Algorithms, Systems, and Applications WASA*, China, June 19-21, 2017, pp. 804-815.

[11] Nivaashini. M, "Deep Boltzmann Machine Based Detection of Phishing URLS", *International Journal of Advances in Electronics and Computer Science*, Vol. 4, No. 9, 2017, pp. 6-11.

[12] Alejandro Correa Bahnsen, Eduardo Contreras Bohorquez, Sergio Villegas†, Javier Vargas† and Fabio A. Gonz´alez, "Classifying Phishing URLs Using Recurrent Neural Networks", *Electronic Crime Research (eCrime), IEEE,* USA, April 25-27, 2017.

[13] Bangsheng Sui, "Information Gain Feature Selection Based On Feature Interactions", MSc. Thesis, *University of Houston,* 2013.

[14] R. Porkodi, "Comparison of Filter Based Feature Selection Algorithms: An Overview", *International Journal of Innovative Research in Science & Technology*, Vol.2, No. 2, 2014, pp. 108-113.

[15] Sonali, B. Maind, and PriyankaWankar, "Research Paper on Basic of Artificial Neural Network", *International Journal on Recent and Innovation Trends in Computing and Communication*, Vol. 2, Issue 1, 2014, pp. 96 – 100.

[16] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques", *Informatica*, Vol. 31, 2007, pp. 249-268.

[17] Bing Xue, Mengjie Zhang, and Will N. Browne, "New Fitness Functions in Binary Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach", *IEEE TRANSACTIONS ON CYBERNETIC,* 2012.

[18] Liam Cervante, Bing Xue, Mengjie Zhang, and Lin Shang, "Binary Particle Swarm Optimisation for Feature Selection: A Filter Based Approach", *IEEE World Congress on Computational Intelligence,* June 2012.

[19] Muhammad Imran, Rathiah Hashim, and Noor Elaiza Abd Khalid, "An Overview of Particle Swarm Optimization Variants", *Procedia Engineering*, Vol. 53, 2013, pp. 491-496.

[20] Banet, R. B. Sung A.H., and Liu Q., "Rule Based Phishing Attack Detection", I*nternational Conference on Security and Management (SAM2011)*, Las Vegas, NV, 2011.

[21] P. Singh, Y. P. S. Maravi, and S. Sharma, "Phishing Websites Detection Through Supervised Learning Networks", *International Conference on Computing and Communications Technologies (ICCCT), IEEE*, 2015, pp. 61-65.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[22] R. Basnet, A. Sung, and Q. Liu, "Learning to Detect Phishing URLs", *International Journal of Research in Engineering and Technology*, Vol. 3, No. 6, 2014, pp. 11–24.

[23] R. Basnet, S. Mukkamala, and A. Sung, "Detection of Phishing Attacks: A Machine Learning Approach", *Soft Computing Applications in Industry, Studies in Fuzziness and Soft Computing*, vol. 226, 2008, pp. 373–383.