Predicting Phishing Websites using Neural Network trained with Back-Propagation

Rami M. Mohammad[1], **Fadi Thabtah**[2], **and Lee McCluskey**[3]
[1]Computing and Engineering, University of Huddersfield, Huddersfield, UK, rami.mohammad@hud.ac.uk
[2]E-Business Department, CUD, Dubai, UAE, fadi@cud.ac.ae
[3]Computing and Engineering, University of Huddersfield, Huddersfield, UK, t.l.mccluskey@hud.ac.uk

Abstract — *Phishing is increasing dramatically with the development of modern technologies and the global worldwide computer networks. This results in the loss of customer's confidence in e-commerce and online banking, financial damages, and identity theft. Phishing is fraudulent effort aims to acquire sensitive information from users such as credit card credentials, and social security number. In this article, we propose a model for predicting phishing attacks based on Artificial Neural Network (ANN). A Feed Forward Neural Network trained by Back Propagation algorithm is developed to classify websites as phishing or legitimate. The suggested model shows high acceptance ability for noisy data, fault tolerance and high prediction accuracy with respect to false positive and false negative rates.*

Keywords: *Web Threat*, *Phishing, Information Security, Neural Network, Data Mining.*

1. INTRODUCTION

Internet facilitates reaching customers all over the globe without any market place restrictions and with effective use of e-commerce. As a result, the number of customers who rely on the Internet to perform procurements is increasing dramatically. Hundreds of millions of dollars are transferred through the internet every day. This number was tempting the fraudsters to carry out their fraudulent operations. Thus, internet-users were vulnerable to different types of web-threats. Hence, the suitability of the internet for commercial transactions becomes doubtful. Phishing is a form of web-threats that is defined as the art of mimicking a website of an authentic enterprise aiming to acquire private information [1]. Presumably, these websites have high visual similarities to the legitimate ones in an attempt to defraud the honest people. Social engineering and technical tricks are commonly combined together in order to start a phishing attack. Typically, a phishing attack starts by sending an e-mail that seems authentic to potential victims urging them to update or validate their information by following a URL link within the e-mail. Predicting and stopping phishing attack is a critical step toward protecting online transactions. Several approaches were proposed to mitigate these attacks. Anti-phishing measures may take several forms including legal, education and technical solutions. Technical solution is the subject of our interest, particularly, heuristic-based approach. The most popular techniques in designing technical anti-phishing solutions include:

- *Blacklist Approach*: In which the requested URL is compared with those in that list. The downside of this approach is that the blacklist usually cannot cover all phishing websites since a newly created fraudulent website takes considerable time before it is being added to the list. The gap between launching and adding the suspicious website to the list may be enough for the phishers to achieve their goals.
- *Heuristic Approach*: Where several features related to website are collected to classify it as either phishy or legitimate. In contrast to the blacklist method, a heuristic-based solution can recognize freshly created fake websites in real-time.

The accuracy of the heuristic-based solution depends mainly on a set of discriminative criteria's picked out from the website. Hence, the way in which those features are processed plays an extensive role in classifying websites correctly. Therefore, an effective and fast retrieval method of information is essential for taking a good decision. Data mining is one of the techniques that can make use of the features extracted from the websites to find patterns as well as relationships among them [2]. Although plenty of applications offered for combating phishing websites, few of them make use of data mining techniques in distinguishing phishing websites from legitimate ones. Moreover, most of these suggested solutions are inapplicable, inaccurate and produce an unacceptable level of false positives rates, which means classifying legitimate website as phishy. Phishing detection is a type of classification tasks in data mining, which have been applied successfully in different domains, i.e. classification, clustering, etc. each instance in the testing dataset is assigned to one of predefined classes. Phishing is considered a binary classification problem because the target class has two possible values "Phishy" or "Legitimate". Neural Network (NN), which is the subject of our interest, is a computerized model of the human brains and nervous system. NN composed of interconnected processing units called (neurons) [3]. The links that connect the neurons to each other hold values that signify the relative importance of each input to a neuron and it

is called connections weights [3]. Connection weights are the crucial elements in any NN model. Connection weights are adjusted repeatedly during the training phase until reaching an acceptable solution. A trained neural network is considered as an expert in the field of information to which it is applied.

In this article, we try to answer the following research questions:

1- How NN can be trained to achieve an acceptable predictive performance.
2- What is the best NN architecture in predicting phishing websites?

This article structured as follows: Section II discuses related works and highlights different phishing detection methods presented in the literature. Section III describes the features used in our model. Section IV introduces traditional NN modelling techniques. In Sections V, VI, several experiments conducted. We conclude in Section VI.

2. Related Work

Although a wide-range of anti-phishing solutions are offered, most of them are not talented to make a decision perfectly thus the false positive decisions rose intensely. In this section, we review current anti-phishing approaches as well as techniques utilized in developing solutions for phishing problem. One approach employed in [4]. is based on fuzzy data mining. The model works on multilayered approach i.e. each layer should have its own rules; however, it was not clear if the rules were established based on human experience, or extracted using an automated tool. Moreover, the authors classify the website as very-legitimate, legitimate, suspicious, phishy or very-phishy, but they did not clarify what is the fine line that separate one class from one another. Another method proposed in [5] suggested a new way to detect phishing websites by capturing abnormal behaviours demonstrated by these websites. Structured website consists of "W3C DOM" features. The authors have selected six structural features: (Abnormal URL, abnormal DNS record, abnormal anchors, Server form handler, abnormal cookies and abnormal certificate in SSL). Support-Vector-Machine classifier (Vapnik) is used to determine whether the website is phishy or not. The classification accuracy in this method was 84%, which is relatively considered low. However, this method snubs important features that can play a key role in determining the legitimacy of the website, which explains the low detection rate. One solution to improve this method could be by using security related features.

The method proposed in [6], suggested utilising "CANTINA" which is content-based technique to detect phishing websites using the term-frequency-inverse-document-frequency (TF-IDF) information retrieval measures [7]. TF-IDF often produces weights that assess the word importance to a document by counting its frequency. CANTINA works as follow:

1. Calculate the TF-IDF for a given webpage.
2. Take the five highest TF-IDF terms and add them to the URL to find the lexical signature.
3. The lexical signature is fed into a search engine.

If the N tops searching results having the current webpage, it is considered a legitimate webpage. If not, it is a phishy webpage. N was set to 30 in the experiments. If the search engine returns zero result, the website is labelled as phishy. However, a limitation of this method is that some legitimate websites consist of images and so extracting the TF-IDF terms may not be accurate in this case. Moreover, this method is delayed in querying through a search engine and thus the user may have started in the disclosure of his personal information. Lastly, this approach does not deal with hidden texts, which might be effective in detecting the type of the webpage. In 2010, a survey presented in [8] evaluated the performance of machine-learning-based-detection-methods including: "AdaBoost, Bagging, SVM, Classification and Regression Trees (CART), Logistic Regression (LR), Random Forests (RF), NN, Naive Bayes and Bayesian Additive Regression Trees (BART)". Results showed that 7 out of 9 of machine-learning-based-detection-methods outperformed CANTINA [9] in predicting phishing websites, those are: AdaBoost, Bagging, (LR), (RF), (NN), Naive Bayes and (BART)". Another study in [10] compared the predictive accuracy of several machine-learning strategies (LR), (CART), (BART), (SVM), (RF), and (NN) for predicting phishing emails. A dataset consist of 1171 phishing emails and 1718 legitimate emails are used within the comparative study. A set of 43 features were used to train and test the classifiers. The experiments showed that (RF) has the lowest error rate of 7.72%, followed by CART 08.13%.

3. Phishing Websites Features

There are several features that distinguish phishing websites from legitimate ones. In our study, we used 18 features descried briefly hereunder:

1. IP address: Using IP address in the hostname part of the URL address means user can almost be sure someone is trying to steal his personal information.

2. Long URL: Phishers resort to hide the suspicious part of the URL, which may redirect the information submitted by the users or redirect the uploaded page to a suspicious domain.

3. URLs having "@" symbol: The "@" symbol leads the browser to ignore everything prior it and redirects the user to the link typed after it.

4. Prefix and Suffix in URLs: Phishers deceive users by reshaping the URL to look like legitimate ones. A technique used to do so is by adding prefix or suffix to the legitimate URL so users might not notice any difference.

5. Sub-domain(s) in URL: Another technique used by the phishers to deceive the users is by adding sub-domain(s) to the URL thus the users may believe that they are dealing with a credited website.

6. Misuse of HTTPs protocol: The existence of the HTTPs protocol every time sensitive information is being transferred reveals that the user certainly connected with an honest website. However, phishers may use a fake HTTPs protocol so that users might be deceived. In [11] a recommendation to

check whether the HTTPs protocol is offered by a trusted issuer such as "GeoTrust, GoDaddy".

7. Request URL: A webpage usually consists of a text and some objects such as images and videos. Typically, these objects are loaded to the webpage from the same domain where the webpage exists. If the objects are loaded from a domain different from the domain typed in the URL address then the webpage is potentially suspicious.

8. URL of Anchor: Similar to "Request URL" but for this feature the links within the webpage might refer to a domain different from the domain typed on the URL address bar. This feature is treated exactly as "Request URL".

9. Server Form Handler "SFH": Once the user submits his information, that information will be transferred to a server to be processed. Normally, the information is processed from the same domain where the webpage is being loaded. Phishers resort to make the server form handler either empty or the submitted information are transferred to different domains.

10. Abnormal URL: If the website identity does not match its record shown in the WHOIS database (http://who.is/) the website is classified as "Phishy". This feature is a binary feature.

11. Redirect Page: This feature is commonly used by phishers by hiding the real link which asking users to submit their information to a suspicious website.

12. Using Pop-up Window: It is unusual to find a legitimate website that asks users to submit their credentials through a popup window.

13. Hiding the Suspicious Links: Phishers resort to hide the suspicious link by showing a fake link on the status bar of the browser or by hiding the status bar itself.

14. DNS Record: If the DNS record is empty or not found the website is classified as "Phishy", otherwise it is classified as "Legitimate".

15. Website Traffic: Legitimate websites are of high traffic since they are visited regularly. Phishing websites often a short life thus their web traffic is either not exists ranked is below the limit that gives it the legitimate status.

16. Age of Domain: the website is considered "Legitimate" if the domain aged more than 2 years [11].

17. Disabling Right Click: Phishers use JavaScript to disable the right click function, so that users cannot view and save the source code.

18. Port number: We examine if there is a port number in the URL and check if the port belongs to the list of well-known HTTP ports such as 80, 8080, 21, 443, 70, and 1080. If the port number does not belong to the list, we flag it as a possibly phishing URL.

4. MODELLING NEURAL NETWORK

An Artificial Neural Network (ANN) is an information-processing model that is stimulated by how biological nervous systems process information. The key element of this model is the unique structure of the information-processing scheme. NN consist of a large number of highly interconnected processing elements "neurones", working in harmony to solve problems. ANNs, like human, learn by example. NNs, with their amazing ability to derive meaningful data from complex dataset, can be used to mine patterns that are too difficult to be noticed by humans [2]. A trained NN can be thought of as an "expert" in the domain it has been applied and can be used to predict class of new cases. Other advantages include [3]:

- Nonlinearity: NN is very effective technique in modelling classification problems where the output values are not directly related to its input.
- Adaptive: Neural network has the ability to adjust the weights based on the changes of its surrounding environments.
- Generalisation: NN is able to find the suitable output for the inputs that does not exist in the training data.
- Fault-tolerance: NN performance is not significantly affected under difficult circumstances such as losing connection between some neurons, noisy or missing data.
- Identical designing steps: The same principles, scheme and methodological steps are employed in designing ANN in all domains.

In our study, we used MATLAB to train our model. The NN performance is assessed by means of Mean Square Error (MSE). We show how NN can be structure to classify websites. MATLAB is a numerical computing environment and a programming language as well. The NN Toolbox is used to design, implement, visualize and simulate our NNs. MATLAB provides wide-ranging support for several NN paradigms, and graphical user interfaces (GUIs) supported by MATLAB enables the user to design NN in a very simple way. We developed Multi Layer Perceptron (MLP) model and calculated the resulting NN model performance by means of MSE. Fig. 1 shows the steps required to create an NN model. The MLP program starts by reading the training, validation and testing datasets, each dataset is stored in an Excel file. To read the datasets we used "xlsread" built-in function. Then, after loading the datasets, the training examples are randomized using the function "randperm.

Next, the input variables website features (Using_IP address, Long URL, URL having @ symbol … etc.) and output variable (website_class) are stated for both training and validation datasets.

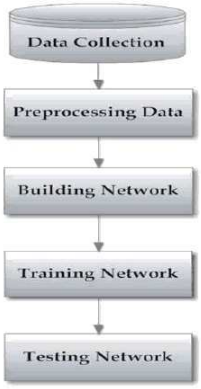

Figure 1 Steps to create an ANN models

MATLAB facilitate creating the MLP model using the "newff" built-in function, which creates a feed-forward back-propagation network. By using this function, we were able to specify the number of hidden-layers, the number of neurons in each layer, the transfer function, the training function, the weight/bias learning function and the performance function.
Once NN training is fully structured, the network performance has to be tested. Therefore, unseen dataset will be presented to the model to show its performance.

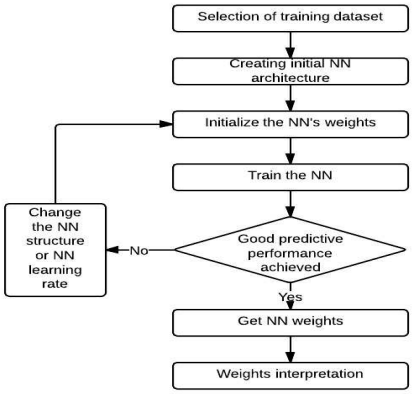

Figure 2 Phishing detection model

The phishing detection model is shown in Fig 2. The model starts by loading the training dataset, then we create the initial NN structure by means of number of layers, number of neurons in each layer and the learning parameters i.e. learning rate, momentum value and number of epochs. Once the NN structure is determined, the weights are initialized to small non zero values. The model is then trained until the maximum number of epochs or the desired error rate is achieved. The model is then tested on the testing dataset which is never being seen once before. If the predictive performance is acceptable then the NN is generated and the weights are produced. Otherwise, the NN structure is improved by changing the number of neurons in the hidden layer or by updating the network parameters i.e. learning rate and momentum value. In our model we adopted the pruning approach to specify the number of neurons in the hidden layer, since we started with a large number of neurons, and the progressively one or more neurons removed during training until the desired performance is met.

5. EXPERIMENTAL METHODOLOGY

A dataset that consists of 1828 websites were used to extract the 18 features using our own tool. The dataset is composed of 859-legitimate website collected from yahoo directory (http://dir.yahoo.com/) and starting point directory (http://www.stpt.com/directory/), and 969-phishing website collected from Phishtank (http://www.phishtank.com/) and Millersmiles archives (http://www.millersmiles.co.uk/). The collected dataset holds categorical values i.e. "Legitimate", "Suspicious" and "Phishy". These values should be transformed to numerical values so that the neural network can perform its calculations thus we replaced the values 1,0 and -1 instead of "Legitimate", "Suspicious" and "Phishy" respectively. We are interested in obtaining a model with a good generalisation performance. However, most models are susceptible to overfitting, which means, while the error rate on the training dataset decreases during the training phase, the error rate on the unseen dataset (testing dataset) increases at some point. To overcome this problem, we used the "Hold-Out" validation technique, by dividing our dataset into training, validation and testing datasets. The examples in each dataset were selected randomly. We split our dataset to 15% for validation, 15% for testing and 70% for training. Training dataset is used to train the network and to adjust the weights of the network, while the testing dataset remains unseen and it is used to assess the predictive performance of the model. After training, we ran the network on the testing dataset. The error value on the testing dataset offers an unbiased approximation of the generalization error. There are several methods for supervised training of NNs. The backpropagation algorithm [3]is the most frequently used training method for ANNs. Backpropagation is usually implemented along with feed-forward NNs that have no feedback. The main idea in feed-forward NNs is to propagate the error through the hidden layers to update the weights of NN. The back-propagation algorithm is described as the following pseudo code:

```
Initialize the weights vector
S = the training set fed to the network
Repeat
        For each "input-output" pair denoted by P in S
            In = input pattern in P
            Out = desired output
            Compute network output (netout)
            network error = Out – netout
        end For
    Find weight change for weights connecting hidden to output
    Find weight change for weights connecting input to hidden
    Update weights
Until reaching (a satisfactory network error value OR maximum iteration)
```

6. TRAINING TECHNIQUES

Determining the network architecture is one of the difficult tasks in constructing a NN model but one of the most essential steps. The NN architecture employed in this study is feed-forward with one hidden layer, which sometimes called multi-

layered perceptron. Problems that need more than one hidden layer are infrequently encountered. Determining the number of hidden layers is only a small problem. We must also decide the number of neurons in each hidden layer. Too few neurons in the hidden layers will cause under-fitting, and too many neurons can result in overfitting. Therefore, the number of hidden layers and the number of neurons in each hidden layers must be carefully determined. Sigmoid transferring function is used in our network. Table 1 summarises the predictive performance achieved in our experiments. The results showed that the best predictive performance was achieved when the number of neurons in the hidden layer was set to "2" and the learning rate was set to 0.7. Moreover, using more than one hidden layer does not improve the predictive performance on the model, thus a single hidden layer is enough to achieve a good predictive performance.

Table 1 Experimental results

Exp #	Number of layers	Number of neurons	Learning rate	Momentum	MSE
1	1	8	0.2	0.7	0.005200
2	1	8	0.6	0.7	0.002950
3	1	5	0.2	0.7	0.005490
4	1	4	0.2	0.7	0.005698
5	1	4	0.6	0.7	0.003956
6	1	4	0.4	0.5	0.005160
7	1	3	0.2	0.7	0.005902
8	1	3	0.4	0.4	0.005695
9	1	3	0.6	0.7	0.004658
10	1	2	0.2	0.7	0.005863
11	1	2	0.7	0.7	0.002234
12	2	3, 2	0.2	0.7	0.005599

7. CONCLUSION

The main goal of this paper was to develop an ANN model to classify websites as either "Phishy" or "Legitimate". Several NNs structures were studied to determine the NN parameters i.e. "number of hidden layers, number of hidden neurons, learning rate and momentum value"; that provide the best predictive accuracy. The selection of a suitable number of hidden neurons during constructing the NN showed to be crucial. One hidden layer was enough for the training and it achieved good performance. We created different networks aiming to lower the error. We assumed that the hidden neurons are 8, 5, 4, 3 and 2. The experimental results showed that the best performance is obtained when the number of hidden neurons was set to 2. Furthermore, the results indicated that the success of NN is impacted when the network parameters are changed i.e. "learning rate and momentum value".
Overall, we were able to show that NN is a good technique in predicting phishing websites. In near future we think of automating the process of building a NN in order to reduce the training time.

8. REFERENCES

[1] APWG, G. Aaron and R. Manning, "APWG Phishing Reports," APWG, 1 February 2013. [Online]. Available: http://www.antiphishing.org/resources/apwg-reports/. [Accessed 8 February 2013].

[2] I. H. Witten and E. Frank, "Data mining: practical machine learning tools and techniques with Java implementations," ACM, New York, NY, USA, March 2002.

[3] B. Widrow, M. and A. Lehr, "30 years of adaptive neural networks," *IEEE press,* vol. 78, no. 6, pp. 1415-1442, 1990.

[4] M. Aburrous, M. A. Hossain, K. Dahal and F. Thabtah, "Intelligent phishing detection system for e-banking using fuzzy data mining," *Expert Systems with Applications: An International Journal,* pp. 7913-7921, December 2010.

[5] Y. Pan and X. Ding, "Anomaly Based Web Phishing Page Detection," in *In ACSAC '06: Proceedings of the 22nd Annual Computer Security Applications Conference.*, Washington, DC, Dec. 2006.

[6] Y. Zhang, J. Hong and L. Cranor, "CANTINA: A Content-Based Approach to Detect Phishing Web Sites," in *Proceedings of the 16th World Wide Web Conference*, Banff, Alberta, Canada, 2007.

[7] C. . D. Manning, P. Raghavan and H. Schütze , Introduction to Information Retrieval, Cambridge University Press, 2008.

[8] D. Miyamoto, H. Hazeyama and Y. Kadobayashi, "An Evaluation of Machine Learning-based Methods for Detection of Phishing Sites," *Australian Journal of Intelligent Information Processing Systems,* pp. 54-63, 2 10 2008.

[9] X. Guang, o. Jason, R. Carolyn P and C. Lorrie, "CANTINA+: A Feature-rich Machine Learning Framework for Detecting Phishing Web Sites," *ACM Transactions on Information and System Security,* pp. 1-28, 09 2011.

[10] S. Abu-Nimeh, D. Nappa, X. Wang and S. Nair, "A Comparison of Machine Learning Techniques for Phishing Detection," in *Proceeding eCrime '07 Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit* , New York, NY, USA , 2007.

[11] R. M. Mohammad, F. Thabtah and L. McCluskey, "An Assessment of Features Related to Phishing Websites using an Automated Technique," in *The 7th International Conference for Internet Technology and Secured Transactions (ICITST-2012)*, London, 2012.