Detection of Phishing Emails using Feed Forward Neural Network

Noor Ghazi M. Jameel
Computer Science Institute
Sulaimani Polytechnic University
Kurdistan Region, Iraq

Loay E. George, Ph.D
Assistant Professor
College of Science
University of Baghdad, Iraq

ABSTRACT

Phishing emails are messages designed to fool the recipient into handing over personal information, such as login names, passwords, credit card numbers, account credentials, social security numbers etc. Fraudulent emails harm their victims through loss of funds and identity theft. They also hurt Internet business, because people lose their trust in Internet transactions for fear that they will become victims of fraud. This paper deals with the phishing detection problem and how to detect phishing emails. The proposed phishing detection model is based on the extracted email features to detect phishing emails, these features appeared in the header and HTML body of email using feed forward neural network to classify the tested email into phish or ham email. The results of the conducted tests indicated good identification rate (98.72%) with short required processing time (0.00067 msec.).

General Terms

Phishing Attack, Phishing Email, Fraud, Identity Theft..

1. INTRODUCTION

Phishing is one of the most challenging security problems which is based on people's behaviour more than on device or application vulnerabilities. The term phishing has come into use to describe techniques for tricking individuals into disclosing confidential information, such as account numbers, Social Security numbers, or financial data with personal information; criminals masquerade as the victim and withdraw money from bank accounts, sell investments, and transfer funds. Another troubling and increasing related problem is identity theft [1]. Phishing attacks use email messages and web sites designed to look as if they come from a known and legitimate organization, in order to deceive users into disclosing personal, financial, or computer account information. The attacker can then use this information for criminal purposes [2]. This paper presents an approach to quickly detect phishing emails using feed forward neural network. This approach is based on some characteristics that are present in phishing emails. A set of 18 features are extracted from tested email for phishing detection purpose. Then, a multilayer feed forward neural network is used to classify the tested email into phish or ham email. This model has accuracy of 98.72% with 18 features, 5 hidden neurons in the hidden layer and learning rate 0.01 with high True Negative (TN) and True Positive (TP) and low False Positive (FP) and False Negative (FN). Where TN denotes ham emails correctly identified as ham and TP represents phish emails correctly identified as phish. Where FP denotes ham email marked as phishing and FN represents the phish email is incorrectly identified as ham. The email samples consist of 9100 phishing and ham emails. The samples of phishing emails (4550 emails) have been collected from publicly available phishing Corpus: http://www.monkey.org/~jose/wiki/doku.php?id =PhishingCorpus; they belong to the time period from November 2004 to August 2007. The samples of ham emails (4550 emails) have been also collected from the ham corpora of the SpamAssassin project; they belong to the time period 2002 and 2003, and contains easy and hard, non-phishing and non-spam emails). In the Training Phase, 6000 emails (3000 phish emails and 3000 ham emails) were used. In the Test Phase, the remaining 3100 emails (i.e., 1550 phish emails and 1550 ham emails) have been used.

2. RELATED WORKS

As phishing emails represents the main gateway of phishing websites, by reviewed a set of papers discussing the various phishing emails methodologies used. One of the main approaches in phishing email detection and classification is the machine learning technique that depends on supervised or unsupervised learning techniques. Chandrasekaran and et al [3] proposed a technique to classify phishing based on structural properties of phishing emails. They used 25 features mixed between style markers (e.g. the words suspended, account, and security) and structural attributes, such as: the structure of the subject line of the email and the structure of the greeting in the body. They tested 200 emails (100 phishing and 100 legitimate). They applied simulated annealing as an algorithm for feature selection. After a feature set was chosen, they used information gain (IG) to rank these features based on their relevance. They applied one-class Support Vector Machine (SVM) to classify phishing emails based on the selected features. Their results claim a detection rate of 95% of phishing emails with a low false positive rate. Abu-Nimeh and et al [4] compared six classifiers related with the machine learning technique for phishing detection and used 43 features to train and test the six classifiers. The results indicated that there is no standard classifier for phishing email detection; for example, some classifiers have low FP levels but have high FN levels such as the Logistic Regression classifier, which has good FP results but has high FN score. Fette and et al [5] proposed an approach called PILFER; it is a machine-learning based approach for classification. PILFER, worked on ten features and used a random forest as a classifier. Random forests create a number of decision trees and each decision tree is made by randomly choosing an attribute to split on at each level, and then pruning the tree. In this approach, the achieved detection rate was 99.5%, when it is used in cooperation with an anti-Spam tool. Despite of the high classification rate, this technique needs 10 features, anti-Spam tool and querying external sources (the WHOIS service) to discover the "age of a domain" of the e-mail sender or some

URL in the e-mail body. Gansterer and PÖlz [6] introduced a ternary classification approach for distinguishing three groups of e-mail messages in an incoming stream (ham, spam, and phishing). The classification is based on a partly new designed set of features to be extracted from each incoming message. Various classifiers have been tested and the results compared to assign them into one of the three groups. Over all three groups, a classification accuracy of 97% was achieved, which is better than solving the ternary classification problem by a sequence of two binary classifiers. AL-Momani and et al [7] proposed a novel concept that adapts the Evolving Clustering Method for classification (ECMC) to build new model called the Phishing Evolving Clustering Method (PECM). PECM functions are based on the level of similarity between two groups of features of phishing emails. PECM model proved highly effective in terms of classifying emails into phishing emails and ham emails in online mode. This introduced method is fast because it is one-pass algorithm. Also, the tests proved PECM capability to classify email by decreasing the level of false positive and false negative rates while increasing the level of accuracy to 99.7%.

3. PROPOSED MODEL

In this paper an approach for email phish detection is introduced. A multilayer feed forward artificial neural network with back propagation, as a training algorithm, has been used for detecting phishing emails. To detect phishing emails using neural network the two phases (training and testing) need to be done. The steps used for detecting phishing emails using feed forward neural network is shown in Figure 1. The model consists of three stages, namely, pre-processing, neural network training and application of phish detection using feed forward neural network. In this approach, 18 features are implemented as a binary value (0 or 1); with a value 1 indicating this feature appeared in the tested email and 0 for non-appearance case.

3.1 Features used in Email Classification

Phishing detection techniques are based on identifying a set of features are usually involving the e-mail header and body. In this work a list of 18 features are extracted; they are binary features. All of these features are extracted using Visual Basic.Net programming language. These features are briefly described in Table 1.

Table 1 Features used in Email Classification

Features	Description
Feature 1 (F_1)	This is a binary feature take a value 1 if there is HTML code embedded within the email and 0 otherwise.
Feature 2 (F_2)	This feature takes a value 1 if the number of pictures used as link is more than 2 otherwise it takes 0 value [7].
Feature 3 (F_3)	This feature takes a value of 1 if the number of different domains in the email is more than 3 and 0 otherwise [7].
Feature 4 (F_4)	This feature takes a value 1 if the number of embedded links in the email is more than 3 otherwise, its value set 0 [7].
Feature 5 (F_5)	This feature takes a value 1 if the message has HTML code included <form > tag otherwise, its value set to 0.
Feature 6 (F_6)	This feature takes a value 1 if “From” domain is not equal to “ReplyTo” domain otherwise, its value set to 0.
Feature 7 (F_7)	This feature takes a value 1 if the message size less than 25 KB otherwise, its value set to 0 [7].
Feature 8 (F_8)	This feature takes a value 1 if the message has java script code otherwise, its value set to 0.
Feature 9 (F_9)	This feature takes a value 1 if non-matching between target and appeared text of URLs in the email otherwise, it sets to 0.
Feature 10 (F_{10})	This feature takes a value 1 if email message has a link like the IP address otherwise, its value set to 0.
Feature 11 (F_{11})	If the message has one of the words “click here” , “click” or “here” or ”login” in text part of links then its value is set 1 otherwise, it set 0.
Feature 12 (F_{12})	This feature takes a value 1 if the number of dots in the domain is more than 3 otherwise, it sets to 0 [7].
Feature 13 (F_{13})	This feature takes a value 1 if the message has @ symbol in URL otherwise, it sets to 0.
Feature 14 (F_{14})	This feature takes a value 1 if the URL in the message has a port value other than 80 or 443 otherwise, its value set to 0.
Feature 15 (F_{15})	This feature takes a value 1 if the domain of any embedded links in the HTML body is not equal to the sender’s domain otherwise, its value set to 0.
Feature 16 (F_{16})	This feature takes a value 1 if https:// is used instead of http://, to lure the user that is a legitimate URL supported with Secure Socket Layer (SSL), otherwise, the value is set to 0.
Feature 17 (F_{17})	This feature takes a value 1, if there is a URL in the email with hexadecimal numeric representation otherwise, the value is set to 0.
Feature 18 (F_{18})	This feature takes a value 1 if the email is classified as spam by SpamAssassin3.2.3.5 Win32; otherwise it takes a value 0.

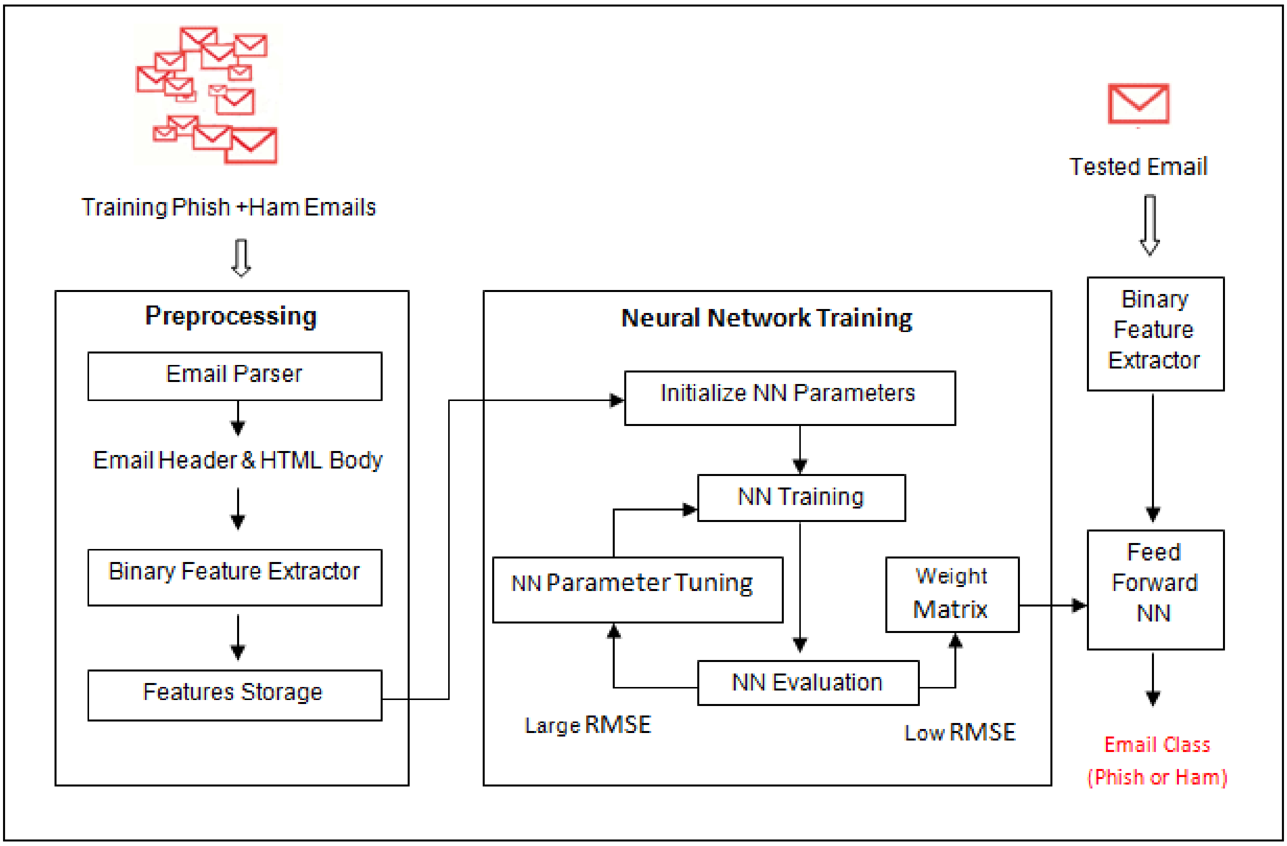

Fig 1: Feature based System using Feed Forward Neural Network (NN)

3.2 Pre-processing Stage

This stage consists of three main modules:

1) Email Parser: In this step, the emails are loaded and parsed into Header and Body. The header of the email is divided into: "From" part, "Reply To" part and X-Spam Status. The body of the email is divided into: "Text" Part of the email and the "HTML" part. The header and HTML parts of the email will be used to extract the necessary discriminating binary features for each email.

2) Binary Feature Extractor: A set of 18 features are extracted. These features are binary coded, with a value 1 refer to the feature existence (i.e., found) in the email and 0, for not found case. For each email in the email data set, a feature vector is extracted.

3) Features Storage: In this step, two binary files are created, one for ham emails and the other for phish emails which store the binary features vectors for ham and phish emails to be used in the neural network training phase.

3.3 Training the Neural Network Stage

This stage works on the binary files that are created in the pre-processing stage which consists of the features vectors of the emails. The input is the set of the 18 extracted features from the header and HTML body of the email. A single hidden layer was used; the number of nodes in the hidden layer was varied to find out the best number of nodes which leads to minimum root mean square error (RMSE) for detecting phishing emails. The number of output nodes was one, to binary represent the email class; such that the output value 1 indicates a phish email and 0 indicates ham email.

The taken sigmoid function is:

$$output = \frac{1}{1+ e^{-Input}}$$

In this project, the neural network was trained with different number of hidden nodes. Also, the value of the learning rate has been varied between (0.01 and 0.1). The number of email samples used to train and test the neural network is 9100 phish and ham emails. 6000 emails were used to train the neural network and 3100 emails were used to test the system performance.

3.4 Testing the Neural Network

In the test phase of neural network, the test emails are represented in terms of the binary feature vector of (i.e., consists of 18 features). The binary feature vector is entered to the feed forward neural network that has the best found artificial neural network weight coefficients set, which is computed in the training phase, to classify the email into phish or ham email.

4. RESULTS

This method is based on the features extracted from the header and the HTML body of the email. Eighteen common features have been extracted from each email in the training data set. The neural network consists of two phases:

1. Training Phase: In this phase the neural network was trained using 6000 phish and ham emails. The input to the neural network was 18 features extracted from each email with 1 hidden layer and 1 neuron in the output layer which

is either (1 or 0). The output 1 indicates that the email is phish and 0 indicates that the email is ham.

2. Testing phase. Once the neural network was properly trained, it was tested over the test email data set (which is not used in the training phase), and also the neural network was tested using training data set.

To evaluate the performance of the system various values of the involved system parameters have been tested. The considered parameters are the number of hidden nodes and the learning rate value. Table 2 shows the results of TN, TP, FN, FP, Accuracy, the test time for a single email and training time for different numbers of neurons in hidden layer with learning rate 0.01 which led to best identification accuracy.

From the results listed in table 2, the case of 5 neurons in the hidden was selected to build the neural network because it gave us the best accuracy (of 98.72%). Figure 2 shows the values of TN, TP, FN and FP using feed forward neural network with different number of neurons in hidden layer and with learning rate value 0.01. Figure 3 shows the relation between the number of neurons in hidden layer and the training time for the neural network. Figure 4 shows the relation between the number of neurons in hidden layer and test time required for a single email. The conclusion from figures 3 and 4, the training and test time will increase when the number of neurons in the hidden layer increases.

Table 2 the Results of using Feed forward Neural Network for Different Number of Neurons in Hidden Layer (learning rate is set 0.01)

No. of neurons in the hidden Layer	TN	FN	TP	FP	Accuracy	Training Time 100 iteration in (msec.)	Test Time for single email in (msec.)
1	0.9947	0.0253	0.9747	0.0053	98.47%	64.18	0.00054
2	0.9879	0.0202	0.9798	0.0121	98.38%	91.84	0.00057
3	0.9879	0.0202	0.9798	0.0121	98.38%	120.93	0.00058
4	0.9875	0.02	0.98	0.0125	98.38%	139.15	0.00062
5	0.9947	0.0202	0.9798	0.0053	98.72%	173.55	0.00069
6	0.9879	0.0198	0.9802	0.0121	98.4%	190.84	0.00073
7	0.987	0.0176	0.9824	0.013	98.47%	215.00	0.00076
8	0.9884	0.0191	0.9809	0.0116	98.46%	244.80	0.0008
9	0.9888	0.018	0.982	0.0112	98.54%	271.08	0.00083
10	0.9884	0.0191	0.9809	0.0116	98.46%	295.34	0.00087
11	0.9879	0.0178	0.9822	0.0121	98.5%	326.34	0.00091
12	0.9884	0.0191	0.9809	0.0116	98.46%	346.77	0.00096
13	0.9879	0.0189	0.9811	0.0121	98.45%	377.51	0.00100
14	0.9875	0.0193	0.9807	0.0125	98.41%	406.25	0.00103
15	0.9879	0.0189	0.9811	0.0121	98.45%	452.91	0.00107
16	0.9879	0.0193	0.9807	0.0121	98.43%	451.74	0.00110
17	0.987	0.0174	0.9826	0.013	98.48%	478.22	0.00114
18	0.9875	0.0191	0.9809	0.0125	98.42%	500.84	0.00118

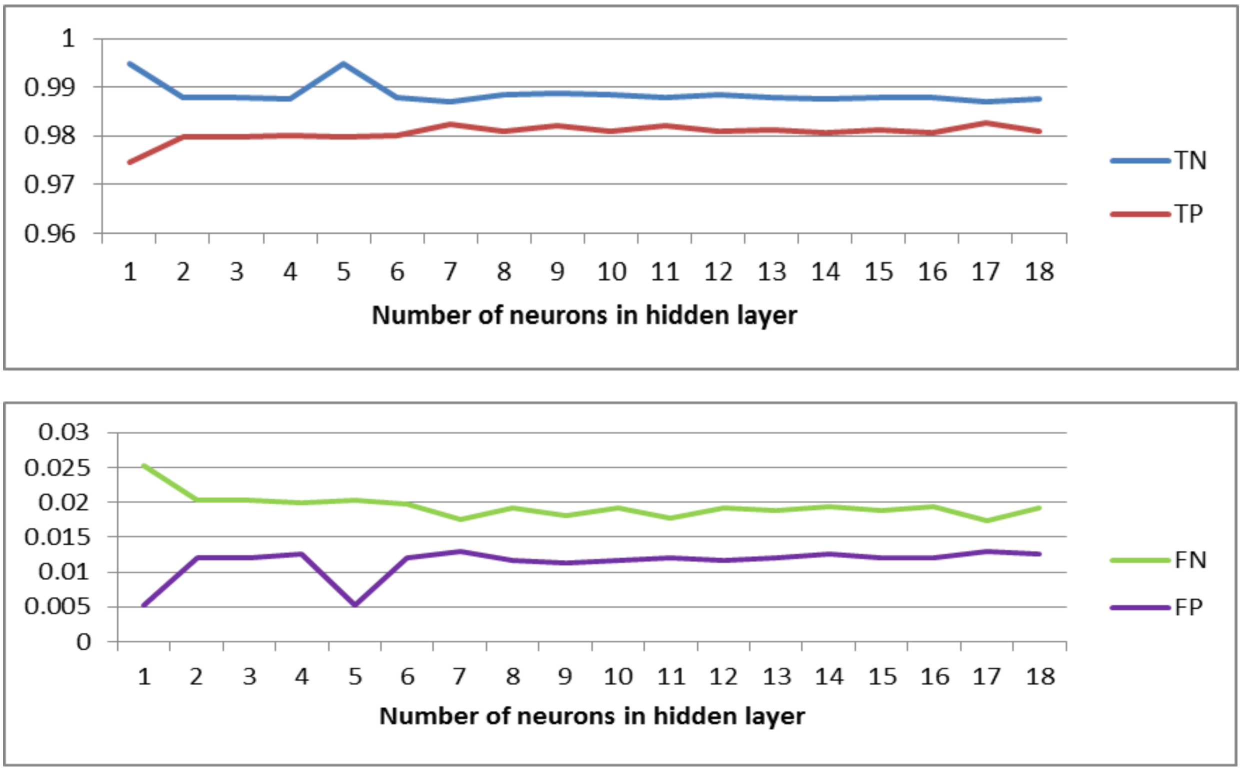

Fig 2: the Values of TN, TP, FN and FP using Feed Forward Neural Network with Different Number of Neurons in Hidden Layer

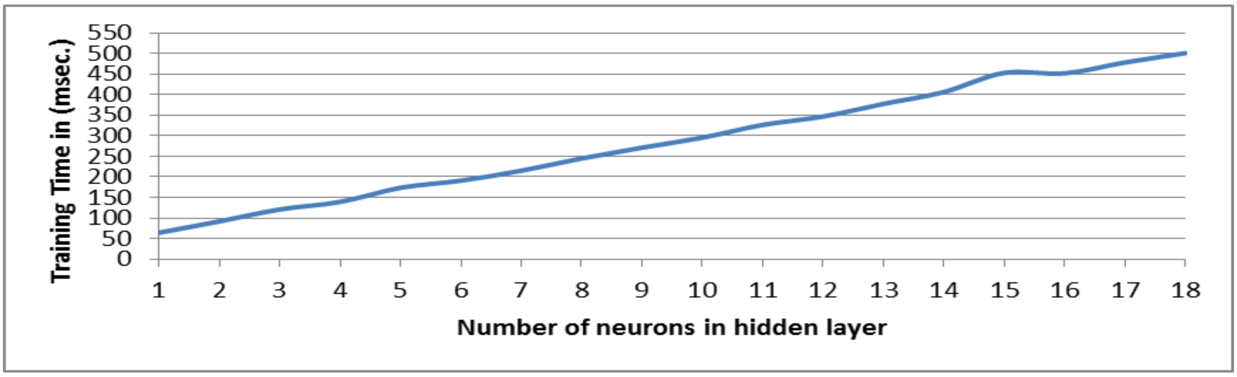

Fig 3: the Relation between the Number of Neurons in Hidden Layer and Neural Network Training Time in (msec.)

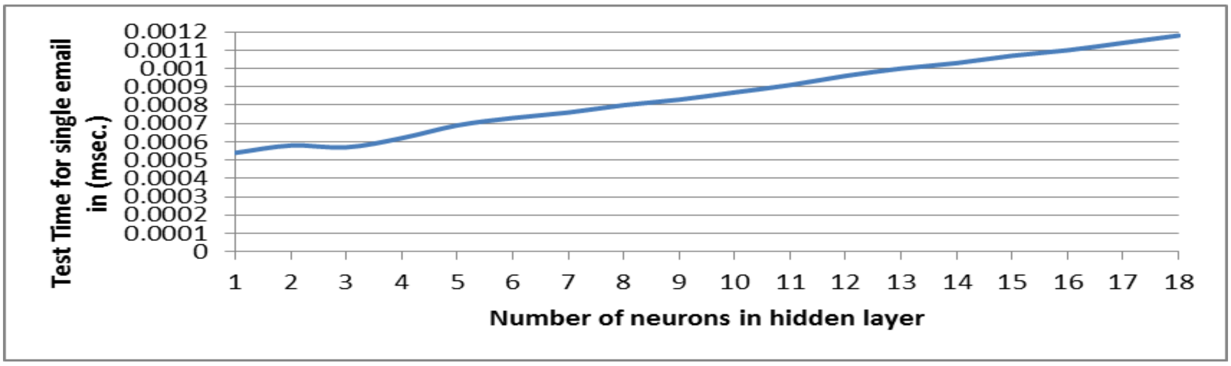

Fig 4: the Relation between the Number of Neurons in Hidden Layer and Neural Network Test Time for Single Email in (msec.)

5. CONCLUSIONS

This algorithm achieved accuracy 98.72% using all the 18 email features with hidden neurons equal to 5 and learning rate 0.01. This algorithm requires a training phase. The time required for training was 173.55 msec. The time for testing a single email is 0.00069 msec. which is very low test time. The training and test will increase when the number of neurons increases.

6. REFERENCES

[1] Sullivan D. 2005 The Definitive Guide to Controlling Malware, Spyware, Phishing, and Spam. Realtime Publishers.

[2] Geeta 2011 Phishing, Security Research. [Online]. Available:http://securityresearch.in/index.php/projects/malware_lab/phishing-2.

[3] Chandrasekaran M., Narayanan K. and Upadhyaya S. 2006 Phishing E-mail Detection Based on Structural Properties, first annual Symposium on Information Assurance: Intrusion Detection and Prevention, New York, pp. 2-8

[4] Abu-Nimeh S., Nappa D., Wang X., and Nair S. 2007 A Comparison of Machine Learning Techniques for Phishing Detection. In Proceedings of the Anti-phishing Working Groups 2nd Annual eCrime Researchers Summit, ACM New York, NY, USA, pp. 60-69.

[5] Fette I., Sadeh N. and Tomasic A. 2007 Learning to Detect Phishing Emails. International World Wide Web Conference Committee (IW3C2) pp. 649-656.

[6] Gansterer W. N. and PÖlz D. 2009 E-Mail Classification for Phishing Defense. 31th European Conference on IR Research on Advances in Information Retrieval, Springer-Verlag Berlin, Heidelberg, pp. 449 – 460.

[7] AL Momani A. A. D., Wan T., Al-Saedi K., Altahr A., Ramadass S., Manasrah A., Melhiml L. B. and Anbar M. 2001 An Online Model on Evolving Phishing E-mail De Detection and Classification Method. Journal of Applied Sciences, vol. 11, Issue 18, pp. 3301-3307.