Volume 3, Issue 7, July 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Detection Phishing Emails Using Features Decisive Values

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

Loay E. George
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 auto 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. The developed model introduces statistical based parameters called features existence weight to decide whether the tested email is phishing or not. The results of the conducted testes indicated good identification rate (97.79%) with short required processing time (0.0004 msec.).**

***Keywords*— Phishing Attack, phishing Email, Fraud, Identity Theft**

I. INTRODUCTION

As people increasingly rely on the Internet for business, personal finance and investment, Internet fraud becomes a greater and greater threat. One interesting species of Internet fraud is phishing [1]. Phishing is an online identity theft technique used to lure consumers into disclosing their personally identifiable information including Social Security numbers (SSN), account names and passwords, credit card information and any other personal information [2]. In recent years, phishing has become an enormous problem and threat for all big internet based commercial operations. The term covers various criminal activities which try to fraudulently acquire sensitive data or financial account credentials from internet users. Phishing attacks use both social engineering and technical means in order to get access to such data [3]. Phishing attack begins with a spoofed email masquerading as trustworthy electronic correspondence that contains hijacked brand names of banks, credit card companies, Social networking sites, or ecommerce sites. The persuasive inflammatory language of the email combined with a legitimate looking Web site is used to convince recipients to disclose sensitive information [4]. In the end, consumers are lured in by these seemingly legitimate communications into providing sensitive information, often resulting in credit card fraud; identify theft, and even financial loss [2].

This paper presents a new approach to quickly detect phishing emails 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, the proposed algorithm is used to classify each email depending on existences flags of the adopted 18 features and their corresponding weights. Many studies have introduced several e-mail features (properties) to detect phishing, but in general there is no distinct evaluation about the degree on relevancy of each feature and the degree of classification accuracy enforcement when they combined together, so we take these points into consideration and evaluate the features relevancy degrees and the existing cross coupling when they combined together. This model has accuracy of 97.79% with 7 features only 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 where 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).

II. 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 [5] 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 [6] 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 [7] 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 [3] 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 [8] 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%.

III. PROPOSED MODEL

In this paper a new approach for email phish detection is introduced. A criteria based on the relative probability of occurrence of the adopted18 features was calculated for both the phish and ham training and test set of emails. This relative probability is called Feature Decisive Value. The mechanism of this approach is shown in Fig 1. The model consists of three stages, namely, pre-processing, feature analysis and application of phish detection using Feature Existence and Feature Decisive Value criteria (FEFDV). In this approach, the appearances of the 18 adopted discriminating features have been analysed and a subset consists of the most effective and common features have been used only. In the previous adopted approach all of these features have been used, and they selected because most of them are usually used in many earlier studies. The 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. In this method of detection, we evaluate and compute the weight of each feature, and then we use the most effective features for classifying the emails. Up to this point, we have found that a high accuracy can be attained with use of only 7 features among the 18 features. This makes the decision boundaries less complex, and therefore both more intuitive and faster to evaluate.

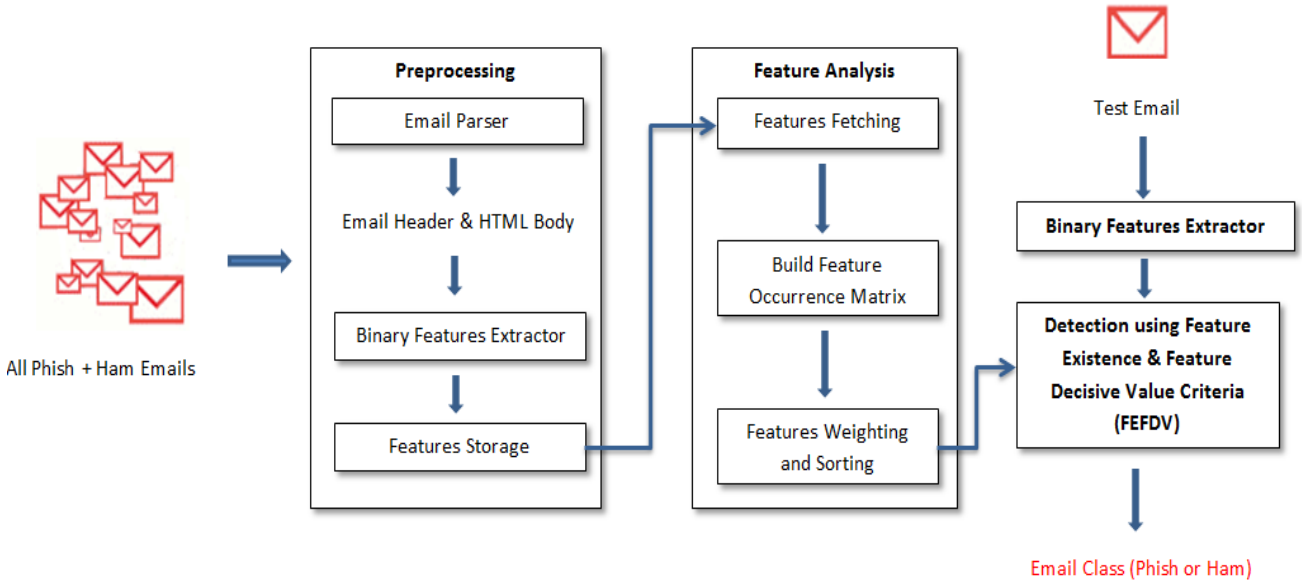

Fig 1 Feature based System using FEFDV Criteria

A. 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 [8].
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 [8].
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 [8].
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 [8].
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 [8].
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.

B. 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 the ham emails and the other for the phish emails which store the binary features vectors for ham and phish emails to be used in the features analysis stage.

C. Feature Analysis 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. After this stage the decisive value of each feature (or weight) is computed. The decisive feature value will be in the range [1,-1]. If the value is 1, it means this feature only occur in the phish emails. If the value is -1, it means this feature only occur in ham emails. If the value is 0, it means it is a (i.e. non-discriminating) weak feature. The decisive values are only used in the email detection stage. This stage consists of three steps.

1) *Features Fetching:* In this step the extracted features vector of each email will be fetched from each binary file which was created in the pre-processing stage, and it will be stored in a two dimensional array. The two dimensional array consists of 4500 row and 18 columns.
2) *Build Occurrence Matrix:* In this step, two occurrence (i.e. Histogram) symmetric matrices of size (18×18) will be created from the fetched matrices which were created from the previous step. One of the occurrence matrices is for all phish emails and the other is for all ham emails. Each occurrence matrix consists of the frequency of occurrence each feature alone and the frequency of occurrence each feature with other 17 features in the phish emails and ham emails.
3) *Features Weighting and Storing*: In this step, the feature decisive values are computed for each one of the 18 features according to the following suggested equation:

$$Di = \frac{FiPhish - FiHam}{FiPhish + FiHam} \quad \text{.....(1)}$$

Where i= {1, 2, 3… 18},
Di: The Decisive value of feature i.
FiPhish: The Frequency of occurrence of Feature i in Phish email.
FiHam: The Frequency of occurrence of Feature I in Ham emails.

D. Detection using Feature Existence and Feature Decisive Value (FEFDV)Criteria

In this introduced approach the Feature Decisive Values (FDV) (i.e., features weights) are used to classify the emails as a phish or ham email. After the determination of FDV values for each of the 18 features, it is found that the FDV values of the features: F9, F10, F12, F14, F16, F17 and F18 are the highest ones. So, the use of these features for Phish/Ham discrimination can lead to the better classification accuracy in comparison with other cases of using combination of features or all the features together. If many of these features (or sometimes one of them) has the value 1, then the applied algorithm will classify the tested email as a Phish Email. If no one of these features has the value 1, the email will be checked by two different DV criteria, one of them is used to compute how much this email is close to be a phish email , this criterion is called Phish Decisive Value criteria (PDV). The other criterion is to compute how much the email is close to be a Ham email; this criterion is called Ham Decisive Value criteria (HDV). The email is classified according to the largest FDV value. Algorithm 1 shows the steps of computing the HDV and Algorithm 2 illustrates the corresponding steps for computing the PDV.

Algorithm 1 Ham Decisive Value (HDV) Criteria

```
Input: Email_Vector (P) as byte // feature vector
Output: D_Ham as Double

Step1: initialize the Weight() with the P weights
       Define D as Double=0.0
       Define F as Byte
Step2:  For I= 0 to P-1
            If Weight (I) <0 then
                 If Email_Vector(I) =1 then  F=1
                 If Email_Vector(I)=0 then   F=0
            Else
               If Weight (I)>0 then
                 If Email_Vector(I)=1 then   F=0
                 If Email_Vector(I)=0 then   F=1
               End If
            End If
         D= D+ (F* Math.Abs(Weight(I)))
       Next
Return (D)
```


Algorithm 2 Phish Decisive Value (PDV) Criteria

```
Input: Email_Vector (P) as Byte // feature vector
Output: D_Phish as Double

Step1: initialize the Weight() with the P weights
       Define D as Double= 0.0
       Define F as Byte
Step2:  For I= 0 to P-1
            If Weight (I) <0 then
                 If Email_Vector(I) =1 then  F=0
                 If Email_Vector(I)=0 then   F=1
            Else
              If Weight (I)>0 then
                   If Email_Vector(I)=1 then   F=1
                   If Email_Vector(I)=0 then    F=0
                End if
            End If
            D= D+ (F* Math.Abs(Weight(I)))
         Next
Return (D)
```

IV. 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 email data set (which consists of 4550 phish emails and 4550 ham emails) in order to compute the frequency of occurrence (i.e., co-occurrence) of each feature alone and the frequency of occurrence each feature with the other 17 features in the phish and ham emails. Then the decisive value was computed for each feature using equation 1, which was described previously. The features decisive values are shown in Table 2 and sorted in descending order from the higher feature decisive value to the lower feature decisive value. Finally Table 3 shows the results of TN, TP, FN, FP, Accuracy, and the test time when Feature Existence and Feature Decisive value (FEFDV) Algorithm with different combinations of features is used.

TABLE 2

THE SORTED FEATURES DECEIVE VALUES

Features	**F_{10}**	**F_{12}**	**F_{14}**	**F_{16}**	**F_{18}**	**F_9**	**F_{17}**	**F_1**	**F_{15}**
Decisive Value	1	1	1	1	0.99	0.96	0.92	0.85	0.83

Features	**F_{11}**	**F_4**	**F_8**	**F_5**	**F_7**	**F_2**	**F_3**	**F_{13}**	**F_6**
Decisive Value	0.78	0.61	0.27	0.24	0.02	-0.05	-0.05	-0.55	-0.68

TABLE 3

THE RESULTS OF USING FEFDV ALGORITHM FOR DIFFERENT COMBINATION OF FEATURES

Feature Group	**Features**	**TN**	**FN**	**TP**	**FP**	**Accuracy**	**Time Required for single email in (msec.)**
1	F_{10}, F_{12}, F_{14}, F_{16}	0.9996	0.4064	0.5936	0.0004	79.66%	0.0007
2	F_{10}, F_{12}, F_{14}, F_{16}, F_{18}	0.9965	0.0369	0.9631	0.0035	97.98%	0.0004
3	F_9, F_{10}, F_{12}, F_{14}, F_{16}, F_{18}	0.9855	0.0299	0.9701	0.0145	97.78%	0.0004
4	F_9, F_{10}, F_{12}, F_{14}, F_{16}, F_{17}, F_{18}	0.9842	0.0284	0.9716	0.0158	97.79%	0.0004
5	F_1, F_9, F_{10}, F_{12}, F_{14}, F_{16}, F_{17}, F_{18}	0.922	0.0132	0.9868	0.078	95.44%	0.0004
6	F_1, F_9, F_{10}, F_{11}, F_{12}, F_{14}, F_{16}, F_{17}, F_{18}	0.922	0.0132	0.9868	0.078	95.44%	0.0004
7	F_1, F_9, F_{10}, F_{11}, F_{12}, F_{14}, F_{15}, F_{16}, F_{17}, F_{18}	0.922	0.0132	0.9868	0.078	95.44%	0.0004
8	F_1, F_4, F_9, F_{10}, F_{11}, F_{12}, F_{14}, F_{15}, F_{16}, F_{17}, F_{18}	0.922	0.0132	0.9868	0.078	95.44%	0.0004
9	F_1, F_4, F_5, F_9, F_{10}, F_{11}, F_{12}, F_{14}, F_{15}, F_{16}, F_{17}, F_{18}	0.922	0.0132	0.9868	0.078	95.44%	0.0004
10	F_1, F_4, F_5, F_8, F_9, F_{10}, F_{11}, F_{12}, F_{14}, F_{15}, F_{16}, F_{17}, F_{18}	0.9187	0.013	0.987	0.0813	95.28%	0.0004
11	F_1, F_4, F_5, F_7, F_8, F_9, F_{10}, F_{11}, F_{12}, F_{14}, F_{15}, F_{16}, F_{17}, F_{18}	0.0046	0.002	0.998	0.9954	50.13%	0.0001

Figure 2 presents the values of TP, TN, FP and FN using FEFDV algorithm for different combinations of features. The figure shows the intersection point between TP with TN and FP with FN which was 4. It means the forth feature group {F_9, F_{10}, F_{12}, F_{14}, F_{16}, F_{17}, F_{18}} from Table 3 gave best compromised values for TN, FN, TP, FP with identification accuracy 97.79%. Figure 3 presents the relation between the accuracy and the features group which consists of specific combination of features. It shows that, increasing the number of features in the feature group, it doesn't mean that the accuracy will increase.

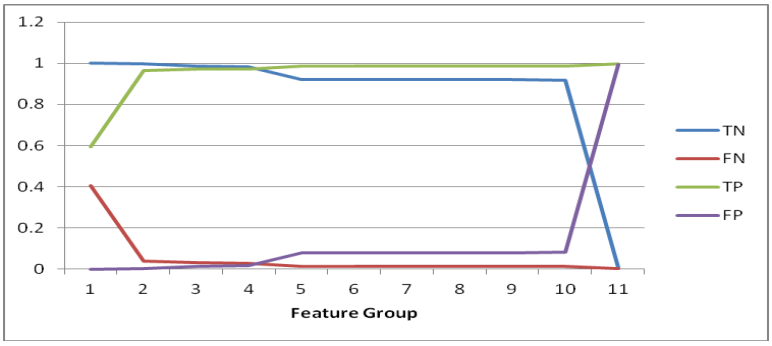

Fig 2 The Values of TP, TN, FP and FN using FEFDV Algorithm for Different Combinations of Features

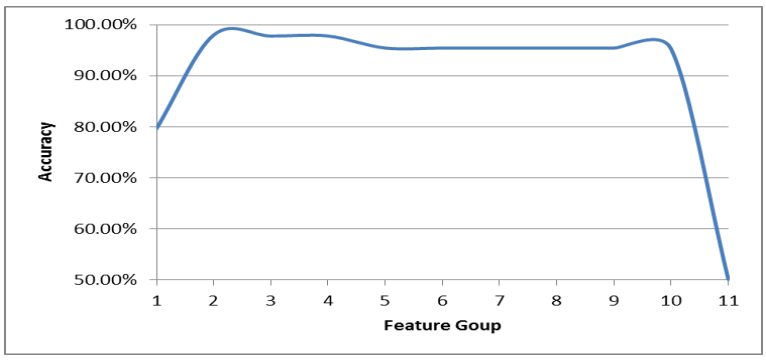

Fig 3 The Relation between the Accuracy and the Specific Group of Features using FEFDV Criteria

V. CONCLUSIONS

This new algorithm is used to classify emails into phish or ham email based on the existence and the weight of features appeared in the email using a new equation to compute the features weight. This new algorithm achieved accuracy 97.79% using only 7 email features from the total 18 features. The time required to test a single email was 0.0004 msec. which is very low test time.

REFERENCES

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

[2] (2005) "How Not to Look like a Phish" Truste organization site [Online]. Available: http://www.slideshare.net/*TRUSTeprivacyseal/truste-white-paper-how-not-to-look-like-a-phish.*

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

[4] Barnes D. S., "A Defense-In-Depth Approach to Phishing", M.Sc. Thesis, Naval Postgraduate School, 2006.

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

[6] Abu-Nimeh S., Nappa D., Wang X., and Nair S. "A Comparison of Machine Learning Techniques for Phishing Detection", *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*,2007, ACM New York, NY, USA, pp. 60-69.

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

[8] AL Momani A. A. D., Wan T., Al-Saedi K., Altahr A., Ramadass S., Manasrah A., Melhiml L. B. and Anbar M, "An Online Model on Evolving Phishing E-mail Detection and Classification Method", Journal of Applied Sciences, vol. 11, Issue 18, pp. 3301-3307, 2011.

