Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 11 (18): 3301-3307, 2011
ISSN 1812-5654 / DOI: 10.3923/jas.2011.3301.3307

An Online Model on Evolving Phishing E-mail Detection and Classification Method

[1]Ammar Ali Deeb Al-Momani, [1,2]Tat-Chee Wan, [1]Karim Al-Saedi, [1]Altyeb Altaher, [1]Sureswaran Ramadass, [1]Ahmad Manasrah, [1]Loai Bani Melhiml and [1]Mohammed Anbar
[1]National Advanced IPv6 Centre of Excellence (NAV6), Universiti Sains Malaysia, 11800 USM, Penang, Malaysia
[2]School of Computer Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia

Abstract: Phishing e-mails pose a serious threat to electronic commerce as they are used broadly to defraud both individuals and financial organizations on the Internet. Criminals lure online users into revealing their passwords or account numbers by sending e-mails as if they come legitimately from financial organizations; these users in turn update and/or provide their account and billing information. In the current study, propose 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 e-mails. PECM model proved highly effective in terms of classifying e-mails into phishing e-mails or ham e-mails in online mode, speed and use of a one-pass algorithm. PECM also proved its capability to classify e-mail by decreasing the level of false positive and false negative rates while increasing the level of accuracy to 99.7%. The model was built to work in online mode and can learn continuously without consuming too much memory because it works on a one-pass algorithm. Therefore, data are accessed one time from the memory and then the rule is created depending on the evolution of the profile if the characteristics of the phishing e-mail have been changed. PECM is a clustering-based learning model that adaptive Evolving Clustering Method to distinguish between phishing e-mails and ham e-mails in online mode.

Key words: Phishing email, clustering, evolving connectionist system, evolving clustering method, classification

INTRODUCTION

For the past several years, the volume of e-mail traffic has indicated a considerable increase in the number of phishing e-mails (Subramanian and Ramaraj, 2007). Phishing e-mails depend on social engineering in order to try and steal the financial account credentials or passwords of users. Phishing e-mails are designed to appear from legitimate businesses and agencies and have an embedded link which will redirect users to a fake Web site. This will then trick them into providing vital information by requiring them to fill out a form purportedly intended to update their account or billing information; the phisher then uses the data obtained to steal from online customers (Zhen *et al*., 2008; APWG, 2010).

Gartner (2007) found that approximately 3.6 million computer users in the United States have experienced phishing attacks, with the total losses amounting to approximately US$ 3.2 billion. In fact, the number of user victims increased from 2.3 million in 2006 to 3.6 million in 2007, representing an increase rate of 56.5%.

One of the most serious problems brought about by phishing is to harm electronic commerce as it causes users to lose their trust in electronic business (Folorunso *et al*., 2006; Lin, 2010). There is a wide range of phishing e-mails ranging from the very simple to the very complex messages. Moreover, phishing attacks are capable of deceiving even the cleverest users, making it difficult to have a fixed rule in solving the problem. Therefore, a system that evolves and can interact with any new type of phishing e-mail is required. Toward this end, a number of studies have attempted various approaches to address the issue of phishing.

The present study has proposed a novel idea related to a model based on the evolving connectionist system (Kasabov, 2003). PECM model is called the Phishing Evolving Clustering Method (PECM) which is designed to work in online mode and has a high speed of classification method and one-pass memory; it was adapted from the Evolving Clustering Method for Classification (ECMC) (Song and Kasabov, 2003). In the current study, PECM was shown to have a high level of accuracy with low False Positive (FP) and False Negative

Corresponding Author: Ammar Ali Deeb Almomani, National Advanced IPv6 Centre of Excellence (NAV6) 6th Floor, School of Computer and Mathematical Sciences Building, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia Tel: 0060142457833

(FN) rates compared with other approaches. Where FP denotes non-phishing e-mails marked as phishing where as FN represents the misidentification of a phishing e-mail.

Although, a large number of studies have focused broadly on phishing or Web phishing, comparatively few have examined phishing e-mail detection, particularly phishing e-mail detection and classification in an online model. As phishing e-mails represent the main gateway of phishing Web sites, by reviewed a set of papers discussing the various phishing e-mail methodologies used.

One of the main approaches in phishing e-mail detection and classification is the machine learning technique that depends on supervised or unsupervised learning techniques (Pugazhenthi and Rajagopalan, 2007; Yang *et al*., 2009). The machine learning technique depends on classifiers that attempt to create a map between the inputs to the desired output depending on a specific function. The main rule of the classifier is based on learning of several inputs or features to predict a desired output (Christy and Thambidurai, 2006). Abu-Nimeh *et al*. (2007) 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 the there are no standard classifiers for phishing e-mail 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 a high FN score.

Saberi *et al*. (2007) proposed a mechanism based on three learning algorithms to detect phishing e-mails which depends on the binary classification of a scam or non-scam e-mail. He built combinations between three algorithms, namely, the K nearest neighbor, the Poisson theory and the Bayesian theory and then used the merged results to propose a new mechanism capable of enhancing accuracy. His proposed method had an accuracy of up to 94.4%, with the FP reaching up to 0.08%. Bergholz *et al*. (2008) proposed arithmetic filtering capable of detecting new phishing messages with different contents depending on the trained characteristic features of the e-mails and author-trained e-mail features, which was conducted using Dynamic Markov Chains algorithm. These approaches can reduce the memory consumed by the system while reducing the number of FPs and FNs.

The approach by Islam *et al*. (2009) was related to machine learning and depends on a built system consisting of a three-tier classification to detect phishing e-mails. The methodology involved in this technique depends on feature extraction and classification in sequential style, after which the output is sent to the decision process. Therefore, if the e-mail is misclassified by any tier (classifier), the last decision will be from the final tier. This technique proved that sequential order classifiers including Support Vector Machines (SVM), boosting (ada boost) and Bayesian (naïve bayes) have the highest levels of accuracy reaching up to 97%.

Yearwood *et al*. (2010) suggested a new technique based on the profiling of phishing e-mail and emphasized embedded hyperlink information only by extracting 12 features signifying phishing e-mail; the data were then divided into two classes. The extraction features are based on a binary value which is 1 if the e-mail has these features and 0 otherwise. Classifier algorithms such as SVM and adaboost are then used to classify e-mails.

Only a few approaches have used clustering for the classification of phishing e-mails. Dazeley *et al*. (2010) proposed a method that depended on the combined method between unsupervised and supervised classification algorithms. In this method, independent clustering was first used to randomize input data and then Consensus Clustering was developed by combining it with independent clustering. Next, Consensus Clustering was used for training and classifying the entire data set. This technique increased the speed of classification and had better accuracy compared with the k-means algorithm (Dehuri *et al*., 2006; Ranjan and Khalil, 2007). In this study, which proposed a new model able to distinguish between phishing email and ham email in online mode with high accuracy, it can learn continuously without consuming too much memory because it works on a one-pass algorithm. It has the capability to use a shorter period of e-mail data capturing to change its profile if the characteristics of phishing e-mail features have changed.

PROPOSED MODEL

PECM proposed model has order steps as shown in Fig. 1. These order steps explain how the classification of e-mails in online mode into two classes, phishing e-mails and ham e-mails, is done. The model consists of three stages, namely, pre-processing, e-mail object similarity and application of the clustering technique Phishing Evolving Clustering Method (PECM). All of these stages work after the features of phishing e-mail utilized in our model have been determined.

Phishing e-mail features used in PECM: The PECM collects e-mails separately and then filters them consecutively. The method depends on 16 features representing he ost effective features of phishing e-mail. All of these features have been proposed in earlier studies (Drake *et al*., 2004; Fette *et al*., 2007), with some

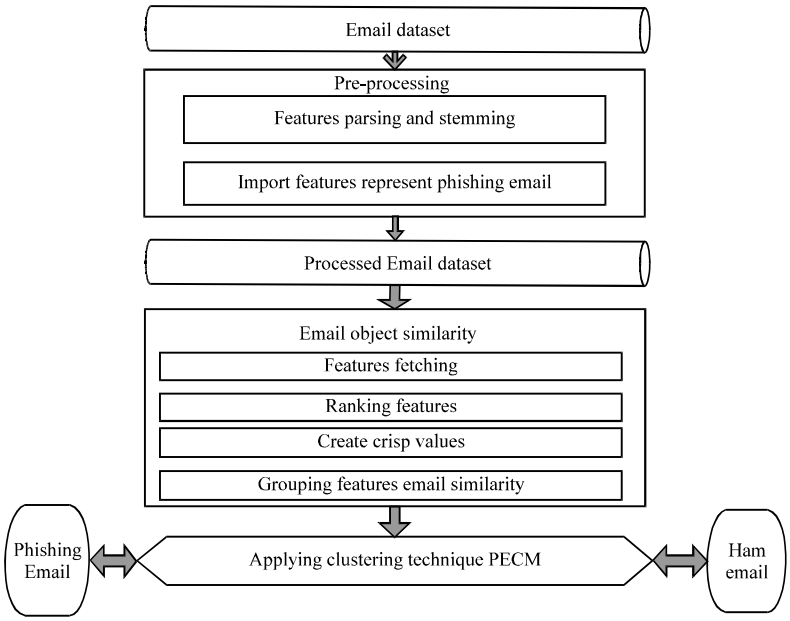

Fig. 1: The overall phishing Email clustering approach-PECM

Table 1: Phishing E-mail features used in PECM

Features	Explanation
Non matching between target and text of URLs (Tardif link)	If they have different host a value 1 and 0 otherwise
Using IP address (ip address)	If email message has a link like IP address a value 1 and 0
The difference between sender's domain with the domain of embedded links (diffsenlindom)	When the embedded links of HTML not equal the sender's domain, a value 1 and 0 otherwise
Number of links (nulinks)	We suggest This feature takes a value 1 if the number of embedded links is more than 3 and 0 otherwise
Number of different domains	We suggest This feature takes a value of 1 if the number of different domains is more than 3 and 0 (nudiffdomain). otherwise.
Number of dots in a domain (nodot)	We suggest This feature takes a value of 1 if the number of dots in the domain is more than 3 and 0 otherwise
Click Here (clickhere)	If the message has one of the three words click here, click or here in text part of links then take a value of 1 and 0 otherwise
Pictures number used as link (NoPicLinks)	We suggest this feature take a value of 1 if there the number of pictures used as link is more than 2 and 0 otherwise
HTML- Email (htmlemail)	This is binary feature take a value 1 if there is HTML code embedded with this message and 0 otherwise
Use of JavaScript (jascript).	This is binary feature take a value 1 if the message has java scrip code and 0 otherwise
Non-standard port in the URL (nonstport)	This is binary features take a value 1 if the message port use other than 80 or 443 ports and 0 otherwise"
URL contains hexadecimal characters or @ symbol (hexorat).	This is a binary features take a value 1 if the message has % or @ symbol in URL and 0 otherwise
Message Size (messize)	We suggest This is binary feature takes the value 1 if the message size less than 25 KB and 0 otherwise
Fake a Secure Connection (facksecon).	IF https://Instead of using http:// to lure the user that is legitimate URL supported with Secure Sockets Layer (SSL)
HTML-form (htmlform).	This is binary feature take a value 1 if the message has HTML code included <form tag> and 0 otherwise
Spam features (spamfeatures)	we used Spamassassin version 3. 2. 3. 5 (Fette *et al*. 2007) with the default rule, and default threshold. This is binary features take a value 1 if the message classified as spam and 0 otherwise

enhancements on the feature extraction technique. The 16 features were implemented as a binary value (0 or 1), with a value of 1 flagging this feature as a phishing e-mail and 0 otherwise. The 16 features are explained in Table 1.

Pre-processing: Pre-processing has two stages. The first stage is parsing and stemming of e-mails. Parsing is used to extract the features of phishing e-mails, whereas stemming is used to clean the text data integrated with the features of a phishing e-mail. The second stage involves importing features that represent phishing e-mails only, then translating the data to binary values (1, 0), with a value of 1 indicating a phishing e-mail feature and 0 otherwise. The processed e-mail data set to E-mail Object Similarity is then taken. In this phase, the extraction

features based on a series of short code scripts are used to extract and analyze phishing e-mail features as explained in the preceding section.

E-mail object similarity: This phase involves three processes as explained below.

Feature ranking and classification: The process of ranking features is used to determine the most effective feature in the system. Our model used Information Gain Ratio method (IGR) algorithms which work based on the extraction of similarities between sets of e-mails and then give the highest weight to the most effective features based on the class of phishing and ham e-mails belonging to IGR (Mori, 2002), as explained in the following equations.

$$\text{Gain\_r}(X, C) = \frac{\text{gain}(X,C)}{\text{split\_info}(C)} \quad (1)$$

where, gain_r (X,C) represents the gain ratio of the feature X frequency in class C.

$$\text{Split\_info}(C) = -\sum_i \left(\frac{|C_i|}{|C|}\right) \log |C_i|/|C| \quad (2)$$

where, C_i and |Ci| denote the frequency of features X in class C, the i-th sub-class of C and the number of features in Ci, respectively.

Table 2 provides a complete ranking of all features selected for the proposed model. The more the information gain is, the more helpful a feature will be. By examination, html e-mail is the feature with the best quality, whereas diffsenlindom is the least helpful and possibly causes noise in the classifier.

Creation of crisp value: The creation of crisp value is used to convert the binary values (0, 1) of all e-mail data sets into crisp values by dividing all features on a 100 score based on this algorithm:

$$X_i = (100^* \text{IGR}_i/\text{sum}(\text{IGR}_i)) \quad (3)$$

where, x is Crisp value, i is the feature number and GR is the information gain ratio. Then for each feature in the data set, each binary value is multiplied and the product is the crisp value, as shown in Table 3.

Grouping features e-mail similarity: This phase is intended to make the data set simpler and faster in the classification processes. In our method, there are two groups of phishing e-mail features, with each group consisting of summation eight features. The first group is the body features and the second is the URL features. The 16 features are classified into two and are stored on a database to be used in the PECM (Table 3).

Table 2: Phishing e-mail features with IGR

Ranking	Features (privations)	Information gain ratio (IGR)
1	htmlemail	0.59503
2	facksecon	0.31765
3	Tardiflink	0.31685
4	NoPicLink	0.25802
5	ipaddress	0.23858
6	clickhere	0.22802
7	nudiffdomain	0.22105
8	nodot	0.13793
9	hexorat	0.10658
10	nulinks	0.07158
11	htmlform	0.03486
12	spamfeatures	0.03215
13	messize	0.02702
14	nonstport	0.02132
15	jascript	0.01760
16	diffsenlindom	0.00493
sum		2.629178

Table 3: Phishing e-mail groups features with crisp values

Group names	Group features	Crisp value
Bodyemail features	htmlemail	22.63
	NoPicLink	12.09
	nudiffdomain	12.05
	nulinks	9.81
	htmlform	9.07
	spamfeatures	8.67
	mesize	8.41
	jascript	5.25
URL features		
	facksecon	4.05
	Tardiflink	2.72
	ipaddress	1.33
	clickhere	1.22
	Nodot	1.03
	Hexorat	0.81
	nonstport	0.67
Sum = 100	diffsenlindom htmlform	0.19

Applying the clustering technique PECM: The PECM algorithm adapts ECMC (Kasabov and Song, 2002; Song and Kasabov, 2003) and classifies e-mails into two classes, ham e-mails or phishing e-mails, in the n-dimensional input space based on evolving rule nodes. The rule created from each node is connected with a class by a constant. These rule nodes are sometimes related to the same class. There are two stages of PECM.

The first is the learning stage which involves the application of the ECM algorithm on the data pairs (x, y), where x is the input vector value (features group values in our model) and y is the output of input vectors. The input

vectors are dealt sequentially with a known class label in the learning stage. The order steps of the learning phase are as follows:

Step 1: IF the system is learning all input vectors, then finish the learning phase; ELSE enter a new input vector from the data source

Step 2: For any class, find all existing rule nodes related with this class as the class of the input vector.

Step 3: IF this is the first time of the system, Then create a new rule but the position of a new rule should be the same as the current input vector in input space and the radius = Min-radius parameter; otherwise, Go to Step 1

Step 4: FOR each rule created before, IF the input vector does not belong within the related field, increase this field if it has potential based on the field having a radius of R^0; the shortest distance between the rule node and the input vector is d. The increased radius is $R^{new} = (R^0 + d) / 2$

Step 5: IF the increase on the field which depends on whether a new field does not contain any input vectors from the data set, is successful, Then the rule node changes its location and the field increases; otherwise, the system will not change both its rule node and field

Step 6: IF the input vector belongs to a related field, go back to step 1; ELSE, a new node is created. Go back to Step 1

Step 7: End of the learning process. The procedure of learning takes one iteration only but all input vectors may change their position many times in input space

The second stage is the classification of new input vectors and involves two steps.

A new input vector (phishing e-mail features values) is entered, then the distance between input vectors with all rule nodes is calculated. IF the input vector is contained by a field of one or more rule nodes related with one class then the input vector will belong to this class. IF the input vector does not belong to any field, then the input vector will go to the nearest rule node.

IMPLEMENTATION AND TEST RESULTS

Data set: The data set used for the assessment of the proposed model was taken from two sources. A sample set of 4,550 phishing messages received from November 2004 to August 2007 was provided by the monkey Web site (http://monkey.org/%7Ejose/wiki/doku.php?id=PhishingCorpus). The second source is the ham corpora from the SpamAssassin project (http://spamassassin.apache.org/publiccorpus/), which consists of both easy and difficult messages comprising 4150 ham e-mails. The ham corpus was collected from 2002 and 2003 and the most recent update was in January 2006.

In the proposed method, the newest 2000 e-mails were selected from each ham and phishing e-mail class. Through the extraction of hyperlinks, e-mails that did not include any hyperlink information were removed. The final data set of the two classes comprised 4000 e-mails.

Experimental analysis: We used MATLAB version 7.10, in PECM for the computation and analysis. In our test, we m ixed the two classes, with one class consisting of 2,000 phishing e-mails and the second class consisting of 2,000 ham e-mails. The mix was ordered serially and the data were divided into two parts: the first with 3,000 samples input for the learning data set and the second with 1,000 samples input for the testing data set. The parameters for PECM were maxfield = 0.1, minfield = 0.01 and one epoch, where maxfield is the maximum radius of the cluster and minfield is the initial radius when a new cluster is created (new rule). The accuracy of the learning phase was 100%, whereas the accuracy of the testing phase was 99.7%. The model used rule extraction in the learning phase to obtain the results in the testing phase. Ours has a few rules reaching up to 24 rules and is capable of classifying a large numbers of e-mails at a high speed without consuming much time in online mode. Some of the rules are shown below.

An example of rules was extracted by PECM:

- IF Centre is [0.53, 0.24] and Radius is 0.10 Then Class is [1] 223 Samples in Cluster
- IF Centre is [0.70, 0.16] and Radius is 0.10 Then Class is [1] 45 Samples in Cluster
- IF Centre is [0.05, 0.08] and Radius is 0.05 Then Class is [2] 1097 Samples in Cluster
- IF Centre is [0.51, 0.05] and Radius is 0.03 Then Class is [2] 403 Samples in Cluster

Table 4: A comparison between our models with other Approaches for phishing email detection

Methods	False positive rate FP (%)	False negative rate FN (%)	Accuracy (%)
PECM	0.01	0.01	99.7
Learning and Ensemble (Saberi *et al.*, 2007)	0.08	5.60	94.4
Multi-tier phishing email classification (Islam *et al.*, 2009)	0.03	0.03	97.0
Pilfer (Fette *et al.*, 2007)	4.00	0.20	99.5

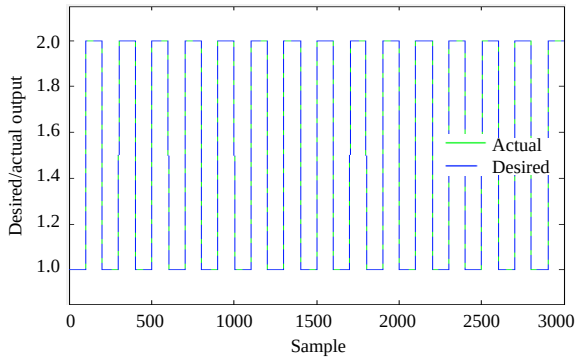

Fig. 2: PECM (on-line, one-pass) model-Learning samples

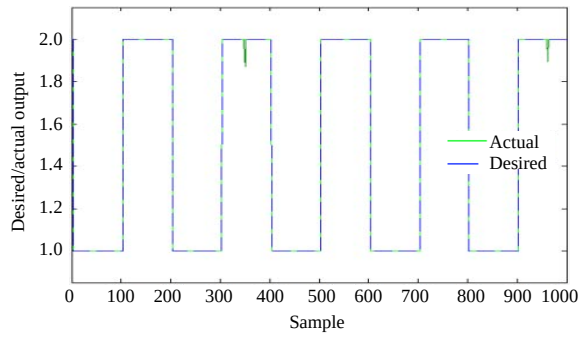

Fig. 3: PECM (on-line, one-pass) model-testing samples Testing samples

We also compared the results of our model with those of other approaches to prove that our model has the best results in terms of FPs, FNs and accuracy. Notably, FP denotes non-phishing e-mails marked as phishing, whereas Fns represent missing a phishing e-mail as shown in Table 4. Figure 2-5 show some of the figures denoting the accuracy of the training and testing phases in PECM in 2d space.

In Fig. 2, PECM shows the level of accuracy between the actual and desired results in training 3000 samples of data has two values (phishing email or ham email) in On-line mode by 2D space.

In Fig. 3, PECM shows the level of accuracy between the actual and desired results in testing 1000 samples of (phishing email and ham email) in Online mode by 2d space.

In Fig. 5, PECM shows the level of general accuracy, by two columns represent two class (number 1 represent phishing email and number 2 as ham email) the level of accuracy from 0% up to 100%.

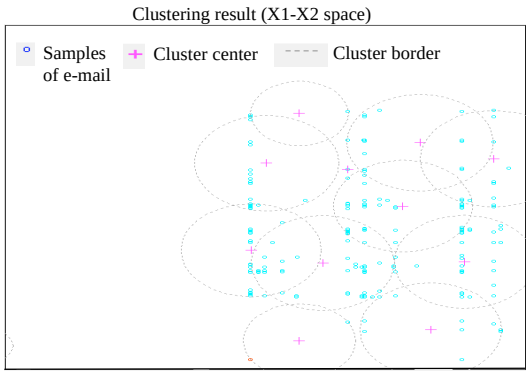

Fig. 4: The groups of phishing email and ham email by adaptive PECM

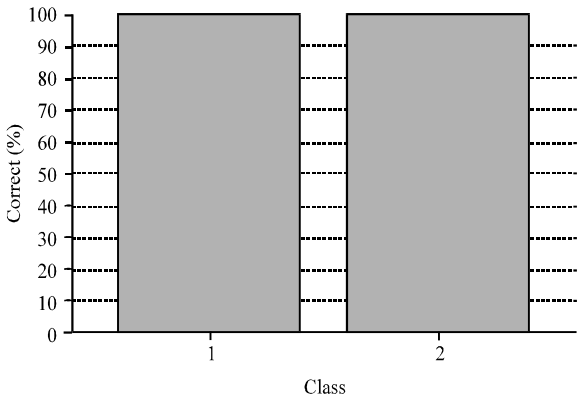

Fig. 5: PECM class accuracy where class 1= phishing email samples and class 2 = ham email samples

CONCLUSIONS AND FUTURE WORK

PECM is a clustering-based learning algorithm that uses clusters of different shapes to distinguish between phishing and ham e-mails in online mode. We used a new technique for extracting features depending on the suggestion that all features have a binary value of either 0 or 1. The approach uses a new incremental clustering algorithm adapted for this purpose and depends on the (MaxDist) between the input vector and the cluster center for classification and building a new rule. The experiments proved that our model has good performance and high accuracy compared with other learning algorithms. PECM works in online mode, making it potentially useful in real time. For future works we suggest the use of online and offline clustering methods to build a system that can work in real time in order to ensure higher accuracy and better performance.

ACKNOWLEDGMENTS

This research is supported by National Advanced IPv6 Centre of Excellence (NAV6) Universiti Sains Malaysia (USM).

REFERENCES

APWG, 2010. Phishing activity trends report. http://www.antiphishing.org/reports/apwg_report_q2_2010.pdf

Abu-Nimeh, S., D. Nappa, X. Wang and S. Nair, 2007. A comparison of machine learning techniques for phishing detection. Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit, (eCrime '07), ACM New York, pp: 60-69.

Bergholz, A., J.H. Chang, G. Paaβ, F. Reichartz and S. Strobel, 2008. Improved phishing detection using model-based features. http:// www.ceas.cc /2008/papers/ceas2008-paper-44.pdf

Christy, A. and P. Thambidurai, 2006. Efficient information extraction using machine learning and classification using genetic and C4.8 algorithms. Inform. Technol. J., 5: 1023-1027.

Dazeley, R., J.L. Yearwood, B.H. Kang and A.V. Kelarev, 2010. Consensus Clustering and Supervised Classi Cation for Pro Ling Phishing Emails in Internet Commerce Security. In: Knowledge Management and Acquisition for Smart Systems and Services, Kang, B.H. and D. Richards (Eds.), Springer, Berlin Heidelberg, New York, pp: 235-246.

Dehuri, S., C. Mohapatra, A. Ghosh and R. Mall, 2006. A comparative study of clustering algorithms. Inform. Technol. J., 5: 551-559.

Drake, C.E., J.J. Oliver and E.J. Koontz, 2004. Anatomy of a phishing email. http://www.mailfrontier.com /docs/MF_Phish_Anatomy.pdf

Fette, I., N. Sadeh and A. Tomasic, 2007. Learning to detect phishing emails. Proceedings of the 16th International World Wide Web Conference, May 8-12, ACM Press, Banff, Alberta, Canada, pp: 649-656.

Folorunso, O., S.K. Sharma, H.O.D. Longe and K. Lasaki, 2006. An agent-based model for agriculture e-commerce system. Inform. Technol. J., 5: 230-234.

Gartner, 2007. Gartner survey shows phishing attacks escalated in 2007: More than $3 billion lost to these attacks. http://www.gartner.com/it/page.jsp?id=565125

Islam, M.R., J. Abawajy and M. Warren, 2009. Multi-tier phishing email classification with an impact of classifier rescheduling. Proceeding of the International Symposium on Pervasive Systems, Algorithms and Networks, Dec. 14-16, IEEE, Kaohsiung, Taiwan, pp: 789-793.

Kasabov, N. and Q. Song, 2002. Denfis: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction. Fuzzy Syst., 10: 144-154.

Kasabov, N.K., 2003. Evolving Connectionist Systems: Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines. Springer Verlag Heidelberg, New York, London, Pages: 307.

Lin, J., 2010. Information systems for enhancing customer relationships. Inform. Technol. J., 9: 1306-1316.

Mori, T., 2002. Information gain ratio as term weight: The case of summarization of IR results. Proceedings of the 19th International Conference on Computational Linguistics, (COLING '02), Association for Computational Linguistics, pp: 1-7.

Pugazhenthi, D. and S.P. Rajagopalan, 2007. Machine learning technique approaches in drug discovery, design and development. Inform. Technol. J., 6: 718-724.

Ranjan, J. and S. Khalil, 2007. Clustering methods for statistical analysis of genome databases. Inform. Technol. J., 6: 1217-1223.

Saberi, A., M. Vahidi and B.M. Bidgoli, 2007. Learn to detect phishing scams using learning and ensemble: Methods. Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, (WI-IATW '07), IEEE Computer Society Washington, DC., pp: 311-314.

Song, Q. and N. Kasabov, 2003. Weighted data normalizations and feature selection for evolving connectionist systems proceedings. Proceedings of the 8th Australian and New Zealand Intelligence Information Systems, (IIS'03), Queensland, Sydney, Australia, pp: 285-290.

Subramanian, S.K. and N. Ramaraj, 2007. Automated classification of customer emails via association rule mining. Inform. Technol. J., 6: 567-572.

Yang, L.Y., J.Y. Zhang and W.J. Wang, 2009. Selecting and combining classifiers simultaneously with particle swarm optimization. Inform. Technol. J., 8: 241-245.

Yearwood, J., M. Mammadov and A. Banerjee, 2010. Profiling phishing emails based on hyperlink information. Proceedings of the International Conference on Advances in Social Networks Analysis and Mining, (ASONAM '10), IEEE Computer Society Washington, DC., pp: 120-127.

Zhen, L., T. Liang and Z. Ming-Tian, 2008. Research on spam classifier based on features of spammer's behaviours. Inform. Technol. J., 7: 165-169.