British Journal of Mathematics & Computer Science
7(1): 58-67, 2015, Article no.BJMCS.2015.102
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

Efficient Harmful Email Identification Using Neural Network

Thangairulappan Kathirvalavakumar[1*], Krishnasamy Kavitha[1] and Rathinasamy Palaniappan[1]

[1]Research Centre in Computer Science, V.H.N.S.N College, Virudhunagar–626001, Tamilnadu, India.

Article Information

DOI: 10.9734/BJMCS/2015/15279
Editor(s):
(1) Xiaodi Li, School of Mathematical Sciences, Shandong Normal University Ji'nan, 250014, Shandong, P.R. China.
Reviewers:
(1) Ravi Narayan, Computer Science & Engineering Department, Thapar University, India.
(2) K. Gnana Sheela, Anna University, Regional Centre, Coimbatore, India.
(3) Anonymous, Jordan.
(4) Manu Pratap Singh, Department of Computer science, Dr. B. R. Ambedkar University, Agra, India.
Complete Peer review History: http://www.sciencedomain.org/review-history.php?iid=932&id=6&aid=7977

Original Research Article

Received: 17 November 2014
Accepted: 14 January 2014
Published: 31 January 2015

Abstract

Phishing is a form of online fraud that aims to steal a user's sensitive information such as online banking passwords or credit card numbers. In this paper, we present a technique to quickly detect suspicious email using Neural Network Pruning approach. The goal is to determine whether the email is suspicious or legitimate. A Multilayer feedforward neural network with Pruning Strategy is used for Feature Extraction and extracted features are used for identifying email as phishing email. Pruning Strategy extracts important features which are playing a key role in identifying phishing mail which looks similar to a legitimate one. To verify the feasibility of the proposed approach experimental evaluation has been performed using a dataset composed of phishing emails along with legitimate emails. The experimental results are satisfactory in terms of false positives and false negatives. The results of conducted test indicated good identification rate with very short processing time.

Keywords: Feedforward neural network, feature selection, pruning algorithm, phishing email, ham email.

1 Introduction

Phishing is one of the most challenging security problems which are designed to steal valuable personal data such as credit card numbers, password and account data. The impact of phishing is

**Corresponding author: kathirvalavakumar@yahoo.com;*

quite dramatic when it involves the threat of identity theft and financial loses. Phishing has increased enormously over the last years and is serious threat to global security and economy. Phishing email contains harmful content which looks like the email comes from legitimate company or bank. When the link embedded in the phishing email is clicked it redirects the user to fake website. Phishing email causes a serious risk because they are used mostly to exploit both individuals and financial organizations on the internet.

A number of recent papers have evaluated various machine learning techniques in detecting phishing emails, URL's and Webpage. Fette, et al. [1] have proposed an approach called PILFER which is based on machine learning for classifications. This is worked on 10 features and used random forest as a classifier. Random forest creates a number of decision trees and each decision tree is made randomly by choosing an attribute to split on at each level and then Pruning the tree. ZHANG, et al. [2] have developed a content based approach CANTINA i.e. Carnegie Mellon Ant phishing and Network Analysis Tool for antiphishing by employing the idea of robust hyperlinks. Aburrous, et al. [3] have developed a resilient model by using fuzzy logic to quantify and qualify the website phishing characteristics at different layers on the Phishing website rate. Bergholz [4] has presented a source of filtering information based on the content of the email. He has proposed dynamic markov chain and class topic model which extracts totally 77 features from the email. Blanzieri, et al. [5] have proposed various spam filtering techniques to detect suspicious email and elaborates the machine learning applications for spam filters. Chandrasekaran, et al. [6] have introduced a classification method based on structural characteristics of phishing emails which employed Information Gain (IG) for feature selection and SVM for Phishing classification. Gansterer [7] has introduced ternary classification approach for differentiating three groups of email messages in an incoming stream which is based on the features. Almomani, et al. [8] have proposed a novel concept that adopts the evolving clustering method for classification to build a new model called Phishing Evolving Clustering Method (PECM). PECM functions are based on the level of similarity between two groups of features of phishing email. Bergholz, et al. [9] have identified large number of new graphical features such as hidden salting detection, image distortion and logo detection for phishing email classification. Ma, et al. [10] have proposed a robust classifier model which detects phishing emails by hybrid features based on Information Gain (IG) algorithm and decision tree algorithms. Basnet, et al. [11] have used random forest and SVM as a classifier and introduced 10 different features including WHOIS query. Almomani, et al. [12] have proposed a framework to detect zero day phishing email which is based on adoptive Evolving Fuzzy neural Network (EFUNN) to predict dynamically the zero day phishing emails. Jameel, et al. [13] have used Feature Existence and Feature Decisive Value Criteria (FEFDV) and identified statistical based features to detect Phishing email. Almomani, et al. [14] have presented adoptive algorithms of ECM, ECMC, DENFIS, DYNFIS from evolving connectionist system to detect and predict dynamically the zero day phishing email. Almomani, et al. [15] have developed the method to detect phishing email based on Bayesian Additive Regression Trees Algorithm. Ram Basnet, et al. [16] have presented a fuzzy technique based on the features to detect the phishing email with limited prior knowledge. Abu-Nimeh, et al. [17] have compared 6 different machine learning techniques for phishing detection. The techniques considered for comparison are Logistic Regression, Classification and Regression Trees, Bayesian Additive Regression Trees, Support Vector Machines, Random Forests and Neural Networks for predicting phishing emails. They have concluded from the result that Random Forests outperformed well than others.Noor Ghazi, et al. [19] have proposed a framework to classify phishing email based on structural properties. They have used Feedforward neural network to classify the tested email into phish or ham. Gethsiyal Augasta and Kathirvalakumar [20] have proposed a new pruning algorithm PIHNS. The algorithm relies on reverse engineering technique to prune the insignificant input neurons and to discover the technological network in classification. Daisuke Miyamoto et al. [21] have presented a HumanBoost approach by using past trust decisions of web users to detect phishing sites. This past trust decision is used as new heuristic and incorporate this with the eight existing heuristics of AdaBoost and proved that this improve the detection accuracy for web user. Daisue Miayamoto et al. [22] have analyzed users' rust decision patterns for detecting phishing

sites. This work identify the type of users whose past trust decision is useful for detecting phishing sites. Authorship aways provides a means to glean information about the author of a document originating from the internet or elsewhere. William Deitrick [23] has used stylometric and word count features in conjunction with the modified balanced winno neural network to predict the author gender of an email. Elsagheer Mohamed [24] has proposed an offensive approach by attacking the spammers by building software to collect links from the spam and junk folders of the users to visit the links periodically and actively. It is acted as a storm of distributed denial of service attack on the spammer's servers and their bandwidth will be completely consumed by the act and their site will be unavailable.

In this paper, an efficient approach is presented to quickly detect phishing emails using Feedforward Neural Network. Phishing emails are identified based on 18 features appeared in the email which are extracted and captured the content and structural properties of the email [19]. Pruning algorithm namely Weight Elimination Algorithm has been used to identify unavoidable features which are used to detect phishing emails. The unpruned features are used to identify phishing emails.

The Sections are organized as follows. Section 2 presents the procedure, neural network training, backpropagation and pruning algorithms. Section 3 discusses the implementation and evaluation results.

2 Procedures

The procedure of detecting phishing email includes 3 stages namely,

- Preprocessing
- Neural network training
- Feature selection using pruning

2.1 Preprocessing

This stage includes 2 phases.

- Email parser
- Binary feature extractor

2.1.1 Email parser

Parsing is a process used to extract the Email features. In this phase, the emails are divided into header part and body part. The header part is again divided into From part, Reply To part and X-Spam status. The body of the email is divided into Text part and HTML part. The header and HTML parts of the emails are used to extract the necessary binary features for each mail [19].

2.1.2 Binary feature extractor

The traditional goal of feature extractor is to characterize an object to be recognized by measurements whose values are very similar for objects in some category, and very different for objects in different categories. This leads to the idea of seeking distinguishing features that are invariant to irrelevant transformations of the input [18]. Totally 18 Features specified in section 2.1.3 are extracted from the email. These features are converted into binary with a value 1 if the feature is existing in the email otherwise it takes the binary value 0.

2.1.3 Features used

Feature 1: If HTML code is used in the email, it takes the binary value 1 otherwise it takes the value 0.
Feature 2: If the number of pictures in the email which act as a link is more than 2 then it is considered as the binary value 1 otherwise 0.
Feature 3: If the number of domains in the body part of the email is more than 3 then binary value 1 is considered otherwise value 0 is considered.
Feature 4: If the number of links appeared in the email is more than 3, it takes binary value 1 otherwise 0.
Feature 5: The binary value 1 is considered if the HTML code appeared in the email contains <form> tag otherwise value 0 is considered.

Feature 6: The binary value 1 is taken, if from domain is not equal to Reply To domain, otherwise it takes the value 0.
Feature 7: If the size of the email is lesser than 25 KB, it takes value 1 otherwise 0.
Feature 8: The binary value 1 is considered, if JavaScript code is embedded in the email otherwise it takes binary value 0.
Feature 9: If the domain consists of more than 3 dots, the value is taken as 1 otherwise 0.
Feature 10: The binary value 1 is considered when the email contains the IP address as link otherwise the value 0 is considered.
Feature 11: If the text part of the email has the words like click here, click, here or login then its value is taken as 1, otherwise 0.
Feature 12: If the domains in the header part of the email is more than 3, its value is set to 1 otherwise 0.
Feature 13: If the email contains @ symbol in URL, it takes value 1 otherwise 0.
Feature 14: If the port value in the URL of the email has other than 80 or 443, it takes the binary value 1 otherwise 0.
Feature 15: If the domain of the link appeared in the email is not redirected to the sender's domain then this value is taken as 1 otherwise 0.
Feature 16: If the URL in the header part of the email contains https:// instead of http://, to make the user to believe the email as legitimate, then binary value 1 is considered otherwise value 0 is considered.
Feature 17: If the URL in the header part of the email has hexadecimal representation then it takes the value 1 otherwise 0.
Feature 18: If the email is classified by Spam Assassin 3.2.3.5 Win 32 then its value is set to 1 otherwise 0.

2.2 Neural Network Training

A single hidden layer feedforward neural network as in Fig.1 is used for training. Let X= (x_i) be the input vector, Y= (y_j) be the output vector, W = (w_{ij}) be the weight matrix between input layer and hidden layer, and V= (v_{ij}) be the weight matrix between hidden layer and output layer. The weighted sum for neurons in hidden layer and output layer can be calculated by,

$$\mathrm{net}_j^t = \sum_{i=1}^{n} w_{ij}^t \cdot y_i^{t-1} \quad (1)$$

Where t represents layer, n represents number of neurons. The outputs of the hidden and output layers are obtained by propagating the training patterns through the network. The output for hidden and output layer is calculated by using sigmoid function as,

$$f(net) = \frac{1}{1 + e^{-net}} \quad (2)$$

Network is learned by minimizing the Root Mean Square Error (RMSE) of the network. RMSE is defined as

$$RMSE = \sqrt{\frac{1}{p}} \sum_{p=1}^{p} \sum_{j=1}^{n} (d_j - o_j)^2 \quad (3)$$

Where 'p' represents patterns, 'j' represents j^{th} neuron of the output layer, 'd' represents desired value and 'o' represents obtained value.

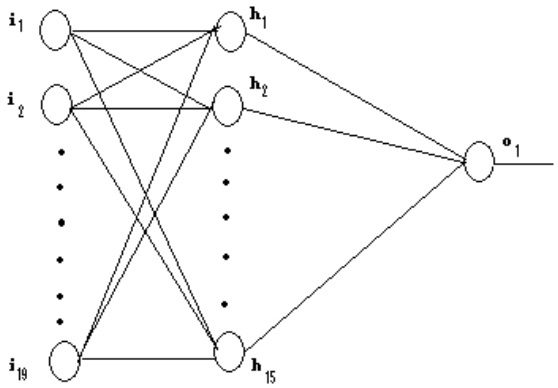

Fig. 1. Multilayer feedforward neural network

2.3 Backpropagation Algorithm

Backpropagation algorithm is used in layered feed forward neural network. The backpropagation algorithm uses supervised learning which means that we provide the algorithm with examples of the inputs and outputs we want the network to compute. Train the network based on the root mean squared error by the weight adjustment formula. The idea of backpropagation is to reduce the error, until the neural network learns the training data. The training begins with random weights and the goal is to adjust them so that the network error is minimal. The sum of squared error of the network is,

$$E_p = \frac{1}{2} \sum_{j=1}^{n} (e_{ij})^2 \quad (4)$$

Where the non linear signal e_{ik} is $e_{ik} = d_k - y_k$. d_k and y_k represent desired and obtained outputs for j^{th} unit in the output layer. The weight update rule for the hidden layer is,

$$\Delta w_{ij}^{h} = \frac{\mu \partial E_p}{\partial w_{ij}} \quad (5)$$

$$= \mu x_{ij} f^{1}(net_i^h)\, e_{2i} \quad (6)$$

where x_i represents the output of i^{th} neuron of the hl^{th} layer. μ is the learning rate, h represents hidden layer.

$$e_{2i} = \sum_{j=1}^{n} e_{ij}\ f'(net_j^o) \quad (7)$$

Then the hidden layer weight will be updated. The weight update rule for output layer is,

$$\Delta w_{jk}^{o} = \frac{\mu \partial E_p}{\partial w_{jk}} \tag{8}$$

$$= \mu h_{jk} f'\left(net_j^o\right) e_{2i} \tag{9}$$

Then the output layer weight will be updated.

2.4 Feature Selection Methods

In practice, machine learning algorithms tend to degrade in performance when faced with many features that are not necessary for predicting the legitimate emails [17]. The problem of selecting subset of relevant features while ignoring the rest is a challenge that all learning schemes are faced with. We apply pruning algorithm to select necessary features to detect phishing emails.

2.4.1 Weight elimination algorithm (WEA)

The weights of the corresponding neurons from the input to hidden and hidden to output are eliminated by using the following steps.

1. Train the neural network using backpropagation algorithm.
2. Let η1 and η2 be positive scalars such that η1 + η2 < 0.5 (η1 is the learning rate, η2 is a threshold that determines whether a weight can be removed), where η1 Є (0, 0.5).
3. To remove the weights W_m from input to hidden layer, the product of the weight value W_m and V_m is calculated and checked whether the product value is less than the η2 value which is multiplied by 4. If it is then remove W_m i.e.

$$V_m * W_m <= 4\eta 2 \tag{10}$$

4. To remove the weights V_m from hidden to output layer, the weight belonging to the hidden to output layer V_m is checked as whether the value is lesser than the η2 value multiplied by 4. If it is then remove V_m i.e.

$$V_m <= 4\eta 2 \tag{11}$$

5. If none of the weight satisfies the step 3 and step 4 then remove W_m with the smallest product of $W_m * V_m$.
6. If classification rate of the network falls below an acceptable level, then stop otherwise go to step 1.

3 Experimental Results

3.1 Experiments

In our experimental analysis, phishing email detection is focused and is based mainly on retrieving information by extracting the features from the email. Finally trained pruned neural network is used to detect the email into phish or ham.

The implementation has been achieved by using Net Beans IDE 6.5 on the system i5 with 2 GB RAM and 3GHZ speed.

3.1.1 Dataset

The samples of 2000 phishing 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 Nov. 2004 to Aug. 2007. The samples of 2000 ham emails have been collected from the ham corpora of the Spam Assassin project. They belong to the time period 2002 and 2003 which contains easy and hard non phishing/non spam emails. The above two datasets are combined into single by randomly mixing both emails. Among these first 3000 emails have been considered for training and the remaining 1000 emails have been considered for testing.

3.1.2 Evaluation criteria

Nh denotes the total number of ham emails, (nh → H) is the number of ham emails classified as ham, (nh → P) is the number of ham emails misclassified as Phishing, Np denotes the total number of phishing emails, (np → P) is the number of phishing emails classified as phishing emails and (nh → H) is the number of phishing emails misclassified as ham. Performance of phishing email detection system is evaluated by the following manner:

True Positive (TP): The number of phishing emails correctly classified as phishing.

$$TP = \frac{np \rightarrow p}{\mathrm{Np}} \tag{12}$$

True Negative (TN): The number of ham emails correctly classified as ham.

$$TN = \frac{nh \rightarrow h}{N_h} \tag{13}$$

False Positive (FP): The number of ham emails wrongly classified as phishing.

$$FP = \frac{nh \rightarrow p}{N_h} \tag{14}$$

False Negative (FN): The number of phishing emails wrongly classified as ham.

$$FN = \frac{np \rightarrow h}{N_p} \tag{15}$$

The accuracy of the classifier performance is computed, by the formula,

$$Accuracy = \frac{TN+TP}{TN+FP+TP+FN} \tag{16}$$

3.1.3 Results

The features are extracted from the selected database as per the section 2.1.3 and make those as patterns. The data from the resultant patterns are selected sequentially to train and test the network.

Single hidden layer feedforward neural network with 19 input neurons including bias, 15 hidden neurons and 1 output neurons is used here for classifying email as phishing or not. The network is trained with generalized delta learning back propagation algorithm. After different trails, it has been observed that the network is converged fast when the learning parameter λ is assumed as 0.1 and

momentum as 0.1. When applying weight elimination algorithm on the trained network, the network has been pruned to be 9-3-1 architecture for the parameter value η2=0.3. The values of FP=0, FN =0.002 and TP=1.002, TN=1 after the pruned network is trained. The result of network training is tabulated in Table 1. The trained pruned network is tested with the test data set and it has been observed that the network classify the emails with 99.9% accuracy for the selected database. The experiment has been carried out for 20 different times. The results on epoch at training, epoch after pruning and accuracy on test data obtained are tabulated in Table 2. On every trail the hidden and input neurons are pruned to same number every time. It has been observed that, in average 23463.5 ms and 470.15 ms time is needed to train the network and to train the pruned network respectively and is shown in Table 3. The number of epochs needed to train the pruned network for different learning parameter is shown in Table 3.

Table 1. Classification results using Neural network with backpropagation algorithm

Initial architecture	Pruned architecture	Accuracy obtained from pruned network	RMSE	ג
9-15-1	9-3-1	99.9%	0.001	0.1

Table 2. Experimental results for different trials

S. No.	Epoch	Neural network training time (ms)	Training time after pruning (ms)	Test accuracy
1.	467	23281	234	99.9%
2.	472	23563	218	99.9%
3.	462	23031	219	99.9%
4.	462	23125	219	99.9%
5.	472	23562	219	99.9%
6.	472	23562	235	99.9%
7.	486	24234	234	99.9%
8.	487	24266	234	99.9%
9.	462	23047	219	99.9%
10.	455	22657	234	99.9%
11.	473	23625	219	99.9%
12.	454	22735	218	99.9%
13.	471	23484	235	99.9%
14.	463	23109	219	99.9%
15.	508	25328	234	99.9%
16.	488	24391	219	99.9%
17.	453	22641	218	99.9%
18.	462	23031	235	99.9%
19.	472	23567	219	99.9%
20.	462	23031	235	99.9%

Table 3. Number of epochs for different learning parameters

Initial architecture	Pruned architecture	Epoch before pruning	Epoch after pruning	λ	Accuracy
19-15-1	9-3-1	1481	3	0.01	99.9%
19-15-1	9-3-1	470.15	1	0.1	99.9%

4 Conclusions

An effective approach is presented to detect phishing emails by extracting significant features in the email using weight elimination pruning technique in a neural network. 18 features [19] have

been considered and performed experimental evaluation of the proposed technique to assess its effectiveness in detecting phishing email. Dataset consisting of 4000 emails including phish and ham have been used. The results in terms of false positives and false negatives are satisfactory. Usage of pruning algorithm reduces the input features used for identifying phishing email into minimal number which leads to minimum computation for classifying the email. The experiment has been carried out for the dataset belonging to the period up to 2007. The emails corresponding to the current period involves new features. Those can be classified accurately with the proposed technique after identifying those new features and incorporate them into input domain for training.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Fette I, Sadeh N, Tomasi A. Learning to detect phishing emails, in Proc. 16th International World Wide Web Conference (WWW 2007), ACM Press, Banff, Alberta, Canada. 2007;649-656.

[2] Zhang Y, Hong JI, Cranor LF. CANTINA: A content based approach to detecting phishing websites. Proceedings of the 16th International Conference on World Wide Web (WWW '07), Banff, Alberta, Ca; 2007.

[3] Aburrous M, Hossain MA, Thabatah F, Dahal K. Intelligent Phishing website detection system using Fuzzy Techniques. IEEE conf, Damascus, Syria. 2008;37-44.

[4] Bergholz A, Paab G, Reichartz F, Strobel S. Improved phishing detection using model-based features, in Proc. Conference on Email and Anti-Spam (CEAS). Mountain View Conf, CA; 2008.

[5] Blanzieri E, Bryl A. A survey of learning-based techniques ofemail spam filtering. Artificial Intelligence Review. 2008;29:63-92.

[6] Chandrasekaran M, Chinchani R, Upadhyaya S. Phoney: Mimicking user response to detect phishing attacks," in In: Symposium on World of Wireless, Mobile and Multimedia Networks, IEEE Computer Society. 2006;668-672.

[7] Gansterer WN, David Polz. E-Mail classification for phishing defense, presented at the Proc. 31st European Conference on IR Research on Advances in Information Retrieval, Springer Conf, Toulouse, France. 2009;449-460.

[8] Almomani A, Wan T, Al Saedi K, Altaher A, Ramadass S, Manasrah A, Melhiml L, Anbar M. An online model on evolving phishing e-mail detection and classification method. Journal of Applied Science. 2011;11(18):3301-3307.

[9] Bergholz A, Jan De Beer, Glahn S, Moens M, Paab G, Strobel S. New filtering approaches for phishing email. Journal of Computer Security. 2010;18:7-35.

[10] Ma L, Ofoghi B, Watters P, Brown S. Detecting phishing emails using hybrid features. IEEE Conf. 2009;493-497.

[11] Basnet B, Sung H. Learning to detect phishing emails, in Proc. 16th International World Wide Web Conference (WWW 2007), ACM Press, New York, NY, USA. 2007;649-656.

[12] Almomani A, Wan T, Altaher A, Manasrah A, ALmomani E, Anbar M, ALomari E, Ramadass S. Evolving Fuzzy neural network for phishing emails detection. Journal of Computer Science. 2012;8:1099-1107.

[13] Jameel M, George E. E-Mail classification for phishing defense. Presented at the Proc. 31th European Conference on IR Research on Advances in Information Retrieval, Springer Conf, and Toulouse, France. 2009;449-460.

[14] ALmomani A, Wan T, Manasrah A, Altaher A, Baklizi M, Ramadass S. An enhanced online phishing e-mail detection framework based on evolving connectionist system. International Journal of Innovative Computing, Information and Control (IJICIC). 2013;9(3).

[15] Abu-Nimeh S, Nappa D, Wang X, Nair S. Distributed phishing detection by applying variable selection using Bayesian additive regression trees, in IEEE International Conference on Communications. 2009;1:1-5.

[16] Ram Basnet SM, Andrew H. Sung. Detection of phishing attacks: A machine learning approach" studies in fuzziness and soft computing. Springer. 2008;226:373-383.

[17] Abu-Nimeh S, Nappa D, Wang X, Nair S. A comparison of machine learning techniques for phishing detection, in Proc. eCrime Researchers Summit, Pittsburgh, ACM Conf, Pittsburgh, PA. 2007;60-69.

[18] Richard O. Duda, Peter E. Hart, David G. Stprk. Pattern classification, second edition. A Wiley - Interscience Publication; 2001.

[19] Noor Ghazi M. Jameel, Loay E. George. Detection of phishing emails using feed forward neural network. International Journal of Computer Applications. 2013;77:0975–8887.

[20] Gethsiyal Augasta M, Kathirvalakumar T. Reverse engineering the neural networks for rule extraction in classification problems. Neural Processing Letters. 2012;35:131-150.

[21] Daisuke Miyamoto, Hiroaki Hazeyama, Youki Kadobayashi. HumanBoost: Utilization of Users' Past Trust Decision for Identifying Fraudulent Websites. 2010;2:190-199.

[22] Daisuke Miyamoto, Hiroaki Hazeyama, Youki Kadobayashi, Takeshi Takahashi. Behind HumanBoost: Analysis of users' trust decision patterns for identifying fraudulent websites. Journal of Intelligent Learning Systems and Applications. 2012;4:319-329.

[23] William Deitrick, Zachary Miller, Benjamin Valyou, Brian Dickinson, Timothy Munson, Wei Hu. Author gender prediction in an email stream using neural networks. Journal of Intelligent Learning Systems and Applications. 2012;4:169-175.

[24] Samir A. Elsagheer Mohamed. A solution for fighting spammer's resources and minimizing the impact of spam. Int. Journal of Communication, Network and System Sciences. 2012;5:416-422.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=932&id=6&aid=7977