Journal of Theoretical and Applied Information Technology
20th January 2016. Vol.83. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MULTI STAGE PHISHING EMAIL CLASSIFICATION

[1]AMMAR YAHYA DAEEF, [2]R. BADLISHAH AHMAD, [3]YASMIN YACOB, [4]NAIMAH YAAKOB, [5]KU NURUL FAZIRA KU AZIR

[1,2,3,4,5]School of Computer and Communication Engineering, Universiti Malaysia Perlis (UniMAP), Perlis, Malaysia

[1]Middle Technical University, Baghdad, Iraq

E-mail: [1]ammaryahyadaeef@gmail.com, [2] badli@unimap.edu.my, [3]yasmin.yacob@unimap.edu.my, [4]naimahyaakob@unimap.edu.my, [5]fazira@unimap.edu.my

ABSTRACT

Phishing emails risk increases progressively, which pose a real threat to users of computers, organizations and lead to significant financial losses. Fighting zero day phishing emails using content based server side classifiers is considered as the best method to detect such attacks. This technique which is based on machine learning algorithms is trained by the set of phishing email features and the statistical classifier is used on stream of email to detect the class of fresh email received. The false positive rate (FPR) and false negative rate (FNR) are critical factors for these classifiers and should be as small as possible to increase the overall accuracy of the classifiers. Using the ham and phishing data sets available, this paper focuses on reduction of false positive rate (FPR), false negative rate (FNR), and increase the overall accuracy of the proposed classification system. The multi stage phishing email detection system (MSPEDS) shows very promising results compared with previous works in term of FPR, FNR, and accuracy.

Keywords: *Phishing, Emails, Email Features, Machine Learning, Classifiers*

1. INTRODUCTION

A phishing email is a fake email claiming originating from legal company or bank. Then, the phisher employing an embedded link in the email to redirect victims to forged website in order to get their sensitive information such as numbers of credit card, passwords, usernames, or other personal [1]. Anti-Phishing Working Group (APWG) phishing trend report shows the 2nd Quarter 2014 phishing activity registered the second highest number of phishing since 2012 [2]. Also, according to a Gartner survey [3] 109 million people in the USA have got phishing e-mails with approximately 1,244 USD dollar loss per victim.

The message of phishing email can be simple or very complicated and can deceive even the professional users of the internet and these attacks are destroying the electronic commercial trading through the internet world which leads to the significant loss of users and trust of the internet [1]. This serious problem has been targeted by many researchers [4-10] which led to the development of many techniques to detect the phishing emails in order to mitigate the effects of such attacks. In the same context, phishing email detection solutions side classifiers, tool in client side, authentication, protection at network level, and user education.

Zero day attack [1] is a challenge problem in email systems because such attacks are not detected by current filters (i.e. blacklists or machine learning classifiers trained by old data). Several phishing emails filtering techniques for server side have been developed based on machine learning (ML) to detect the phishing emails using content filtering and these approaches regarded as the most effective option to confront zero day attacks problem. One of the most important problems in such detection systems is how to reduce the false positive rate and increase the overall accuracy.

Although there are many good results in literature, decreases the false positive rate, false negative rate, and improve the accuracy of ML filters still remain among the open issues in the research community. This paper tries to tackle these problems by exploiting the power of integrated ML classifiers to achieve such improvements.

Multi stage phishing email detection system (MSPEDS) is presented in this paper which employs the best known classifiers. Our system reduces the rates of FP and FN in comparison with the existing work.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The remainder of this paper is structured as follows: Section 2 presents the works related to our research. Section 3 gives an overview of research framework. We present the selected features in section 4. In section 5, we show the datasets used in the research. We explain the evaluation metrics in section 6. Section 7 presents the experimental results. Finally, we conclude and put direction for future works in section 8.

2. RELATED WORK

Recently many researches have been targeting the problem of phishing email detection to confront the growing phishing attacks. These solutions comprise server side and client side techniques. As reflected by its name, the server side solutions are implemented on the server side such as the Internet Service Provider (ISP). In contrast, client side solutions targeted the end users such as email analysis and plug-ins in browsers. Filters on server side generally depend on approaches use in content filtering and these solutions are the most important option to confront the problem of zero day attacks. Hence, the majority of research efforts try to tackle this problem from server side. The solutions in server side depend on features extraction from the phishing email and by employing machine-learning algorithms to classify labeled emails as phishing and legitimate email. These algorithms can be used to classify new received emails from a stream of emails [1].

In this section, machine learning based techniques for phishing email detection on server side are discussed. Research in [11] compares the accuracy of six machine learning algorithms including Classification and Regression Trees (CART), Bayesian Additive Regression Trees (BART), Logistic Regression (LR), Random Forests (RF), Support Neural Networks (NNet), and Vector Machines (SVM). For training and testing the classifiers, 43 features are extracted from phishing emails. The results showed that there is a trade off in term of false positive (FP) and false negative (FN) where LR classifier provided the best precision of 95.11% with 04.89% 17.04% FP and FN respectively. However, the results showed some classifiers provide low false positive rate but have high rate of false negative this make no standard classifiers to tackle the problem of phishing detection. Ram Basnet [12] used sixteen features to detect phishing emails. Several machine learning algorithms are tested to discover phishing and legitimate emails. Biased Support Vector Machine (SVM) provides the best results with an accuracy of 97.99%. Scam and non-scam binary classification has proposed in [13], this technique depends on three different machine learning methods and then employed the ensemble methods to enhance the classification accuracy. The results showed an accuracy of 94.4% with false positive of 0.08%. The study in [14] focused on employing the lexical URL analysis to enhance the accuracy of phishing email classifiers. The work used Random Forest classifier and the results show the classifier provides an accuracy of around 99.38 % with FP rate of 0.25%. However, the FPR still need more improvement by employing more features and testing other classifiers.

Authors of [15] used 30 features, 15 features proposed by previous papers while the rest 15 are completely new features proposed by the authors. In addition, this paper compares the binary classification (not spam and spam) with the ternary classification (phish, ham, or spam). The new features provided accuracy of 97% using SVM with the ternary classification approach. However, the main limitation in this study is the online features which depend on internet connection, where extraction of several online features will affects the filtering system performance in large email servers. New approach for detecting phishing email is proposed in [16]. This approach employed the ontology concept with training and testing data sets in order to help Naive Bayes algorithm. The heuristics proposed in this paper offers a word as an attribute and it values the frequency of this word. Small size data set has been used with 200 phishing email and provided 94.87% accuracy. The main limitation in this study is the small size of data set which insufficiently characterizes the proposed concept.

In addition, using ontology approach introduces extra overhead on the classifiers which make it not suitable for online environments. Phishing emails detection using PILFERS method has been proposed in [5]. One feature which represent the age of linked-to-domain names has been extracted using WHOIS query, while 9 features has been extracted from the email. The data set represented by 860 phishing and 6,950 ham emails. The best results show 0.12% FPR with 7.35% FNR which mean that the accuracy is not well enough. Research in [17] classified the emails as phishing and ham by employing statistical classification. New features are generated using Dynamic Markov Chains and Class-Topic Models. Twenty seven features used in this paper where the model provided reduction in memory consumption in comparison with other papers with better results

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

than PILFER method on the same data set. In addition, this method tested in an online environment at a commercial internet server provider in [18]. However, this method is time consuming as it employed many algorithms for classification. Hybrid features approach is proposed by [19]. The hybrid features consist of orthographic and derived, content, and method for feature selection. Information gain algorithms are used for features selection, 7 features are used as the best features. Decision tree algorithm showed the best results with an accuracy of 99.8%. However, this approach is time consuming as the decision required five stages and the data sets used are not standard which make the results are not benchmarkable. FRALEC system proposed by [20] is three stage system to classify emails to ham and phishing. The three stages are Bayesian Classifier, Rule Based Classifier, and Emulator-Based Classifier. The data sets used by authors consist of 10 legitimate emails and 1028 phishing emails. The system provided best results with 96% precision. However, from the used data sets are not sufficient to give us clear results with time consuming as the system depend on many layers to give the result. Islam et al [21, 22] proposed multi-tier classification system. The system used 3 classifiers where the email features are extracted and classified in sequential and the outputs are sent to decision classifier process. The results showed c2 AdaBoost, c3 Naive Bayes, and c1 SVM provided the best results with accuracy of 97%. However, the system is complex and time consuming due to the many stages. PDENFF [23] is a novel proposal to dynamically detect and predict the zero day email fishing attack. The framework used evolving connectionist system and provides 3 % to 13% improvement in comparison with previous techniques. However, the system needs continuous feeding and time consuming. PhishStorm proposed by [24] is an automated phishing detection system which based on analysis of URL lexical in real time environment. The system implemented as central detection unit places in front of the email server. 12 features from a URL are extracted using the searching engines followed by supervised classification step. The PhishStorm provided 94.91%, 1.44% accuracy and false positive respectively. However, the system depend on searching engines which added overhead processing and time consuming and depend only on URL lexical features.

Although there are several solutions for server side email phishing detection, the designed solutions still need to reduce the FPR and FNR. Our system used the most effective features of email existed in the literature to increase the overall accuracy of phishing detection systems.

3. RESEARCH FRAMEWORK

Three parts generally represent the email message, namely envelope, header, and body. Figure 1 shows all email data parts.

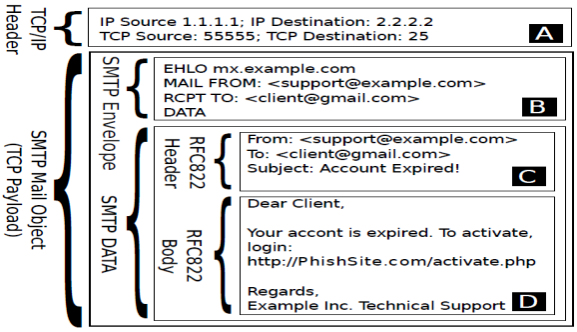

Figure 1: Email data parts [25]

In this research, the most effective features found in literature extracted from part D and C. The study comprises three phases implemented respectively. The first phase is to prepare the sufficient dataset for learning the algorithms. The second phase comprises the used algorithms and making learning process to bring out the classification system to detect phishing email. Finally, the third phase aims to provide the required analysis for the evaluation of the final results.

3.1 Phase 1: Dataset Preparation and Feature Extraction

The preparation processing of the datasets was implemented to make them suitable for the study requirements where these processing comprise the extraction and normalization of the features. These steps are essential to make the classifiers understand the data and be able to classify it to the specified class.

3.2 Phase 2: Machine learning algorithms

Three classifiers commonly used for phishing detection are used for this phase, namely Naïve Bayes, Random Forest and Tree.J48.

3.3 Phase 2: Machine learning algorithms

The performance of MSPEDS will be evaluated in terms of evaluation metrics as will explain in the next sections.

Journal of Theoretical and Applied Information Technology
20th January 2016. Vol.83. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. THE PROPOSED SYSTEM

MSPEDS has a number of steps as shown in Figure 2. These steps show how the emails classified into two classes phishing and ham. The proposed model mainly consists of three classifiers C1 (Naïve Bayes), C2 (Random Forest) and the C3 (Tree.J48), the sequence of system work can be summarized as follows:

1) Classifier 1 and classifier 2 trained based on the training dataset.

2) The output of the two classifiers has to be tested using decision maker, where the work of this decision maker done for each incoming instance (email) if both classifiers (C1 and C2) classified this instance as phish (C1p and C2p) then the decision will be phishing by decision maker, if both classifiers says ham (C1h and C2h) the decision maker will say Ham, in other cases that mean if classifier 1 say phish and classifier 2 say ham or the reverse, the decision maker will classify it as unknown.

3) The correctly classified instances by the classifier 1 and classifier 2 will be used as training dataset for classifier 3 and the unknown instances will be input to classifier 3 then this classifier will classify all unknown instance as phishing and ham. The main purpose of this training process is to make one of the classifiers always perform the training using fresh datasets to overcome the drift concept in order to achieve high accuracy.

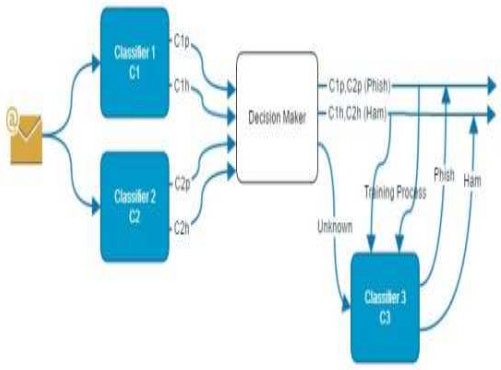

Figure 2: The proposed MSPEDS

5. SELECTED FEATURES

The 48 features collected by the authors in [9] have been used by most previous studies. In our study, we collected some features from [9] and combine with the ones in [26, 14] which make the total number of the selected features to become 48.

5.1 features of email body

The total number of features extracted from part D of email is 11 features explained as follows:

5.1.1 body_html: If HTML content exists in email message return 1 and -1 otherwise.

5.1.2 body_forms: The binary 1 represent that email massage contains HTML forms and -1 otherwise.

5.1.3 body_dear_word: this feature represents the presence of a dear word in email body 1 or -1 if not.

5.1.4 body_multi_part: The value of this feature is 1 if the message contains a multipart MIME type and -1 otherwise.

5.1.5 body_no_words: This feature counts the total number of words existing in the email body.

5.1.6 body_no_characters: This feature count the total number of characters existing in the email body.

5.1.7 body_richness: This feature returns the value of division the total number of words by the total number of characters existing in email body.

5.1.8 body_nodistinctwords: This feature counts the total number of distinct words existing in the email body.

5.1.9 body_suspension: The value of this feature is 1 if the suspension word exists in email body and -1 otherwise.

5.1.10 body_verifyyouraccount: The value of this feature is 1 if the verify your account phrase exist in email body and -1 otherwise.

5.1.11 body_nofunctionwords: This feature counts the total number of function words existing in the email body these words comprises: bank; access; click; password; identity; inconvenience; log; minutes; security; recently; limited; social; suspended; service; credit; information; risk and account.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5.2 features of email header

The features have been extracted from part C of email. Totally 14 features are used as listed below:

5.2.1 subject_debit: This feature return 1 if debit word occurs in the subject of an email and -1 otherwise.

5.2.2 subject_verify: This feature return 1 if verify word occurs in the subject of an email and -1 otherwise.

5.2.3 subject_bank: This feature return 1 if bank word occurs in the subject of an email and -1 otherwise.

5.2.4 subject_forward: This binary feature returns 1 if the email is forwarded from another account to the recipient and -1 otherwise.

5.2.5 subject_reply: This feature return 1 if Re word occurs in the subject of an email and -1 otherwise.

5.2.6 subject_nowords: This feature provide the total number of words occurs in the subject of an email.

5.2.7 subject _nocharacters: This feature provide the total number of characters occurs in the subject of an email.

5.2.8 subject_richness: This feature returns the value of division the total number of words by the total number of characters existing in the subject of an email.

5.2.9 send_noword: This feature records the total number of words existing in the sender field of an email.

5.2.10 send_unmodaldomain: If the address of sender uses an unmodal domain name the value will be 1 and -1 otherwise.

5.2.11 send_differentreply: If the difference between sender and reply to email addresses exist the value will be 1 and -1 otherwise.

5.2.12 unique_sender: This binary feature return 1 if the sender sends emails from more than a single domain and -1 otherwise**.**

5.2.13 unique_domain: This binary feature return 1 if the domain names are used by more than one sender domain email and -1 otherwise.

5.2.14 DMID validity: This binary feature return 1 if the message ID field has been forged by the phisher and -1 otherwise.

5.3 Features of URL

The features have been extracted from part D of email. Totally 18 features are used as listed below:

5.3.1 url_portnumber: This binary feature return 1 if the message contains URL with port number and -1 otherwise.

5.3.2 url_atchar: This binary feature return 1 if the message contains URL with @ sign and -1 otherwise.

5.3.3 url_IP: This binary feature return 1 if the message contains URL with an IP and -1 otherwise.

5.3.4 url_bagword: This binary feature return 1 if the following words exist: “click”, “login”, “update”, and “here”. Otherwise is -1.

5.3.5 url_no.port: This feature provide the number of URLs with their authority section contains port number.

5.3.6 url_no.IP: This feature provides the number of URLs with their authority section contains IP address as opposed to a domain name.

5.3.7 url_no. link: This feature provides the number of links with email body.

5.3.8 url_no.periods: This feature provides the number of periods with email body.

5.3.9 url_no.imglink: This feature provides the number of image links with email body.

5.3.10 url_no.domains: This feature provides the number of domains with URLs in the email.

5.3.11 url_update: This binary feature return 1 if the link text contains the “update” word, and -1 otherwise.

5.3.12 url_click: This binary feature return 1 if the link text contains the “click” word and 1 otherwise.

5.3.13 url_login: This binary feature return 1 if the link text contains the “login” word and 1 otherwise.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5.3.14 url_here: This binary feature return 1 if the link text contains the "here" word and -1 otherwise.

5.3.15 url_no.externallink: This feature provides the number of external links exist in an email.

5.3.16 url_no.internallink: This feature provides the number of internal links exist in an email.

5.3.17 url_unmodal: This binary feature return 1 if the link text contains an unmodal link with the following words "click", "link", and "here"., and -1 otherwise.

5.3.18 url_twodomains: This binary feature return 1 if the URL has two domain names and -1 otherwise.

5.4 features of JavaScrip

The features have been extracted from part D of email. Totally 5 features are used as listed below:

5.4.1 unmodalload: This binary feature return 1 if the email body has JavaScript downloaded from un modal domain name external website, and -1 otherwise.

5.4.2 JavaScriptexist: This binary feature return 1 if the email body has JavaScript and -1 otherwise.

5.4.3 JavaScriptchange: This binary feature return 1 if the email body has JavaScript aims to change the status bar and -1 otherwise.

5.4.4 popupScript: This binary feature return 1 if the email body has JavaScript event "onclick", and -1 otherwise.

5.4.5 onclickScript: This binary feature return 1 if the email body has JavaScript aims to open pop-up windows and -1 otherwise.

6. DATASETS

The datasets used in our study are publicly available and used by most studies in related work. The phishing dataset downloaded from [27], where our phishing dataset consist of 4550 phishing emails. On the other hand, the ham emails downloaded from [28] with the 4400 legitimate emails. Figure 3 shows the ratio of phishing to ham emails in the dataset.

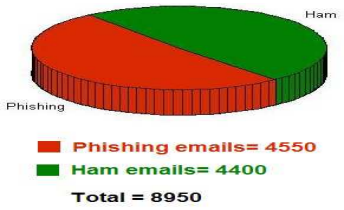

Figure3: The ratio of phishing and ham emails

7. EVALUATION METRIC

The machine learning classification performance evaluated commonly based on widely used metrics namely: False Positive Rate (FPR), False Negative Rate FNR, classification time and Accuracy.

- False Positive Rate (FPR): Defined as the ratio of ham class that incorrectly classified as a phishing class to the total number of ham class instances.

$$FPR = \frac{Nh \rightarrow p}{Nh \rightarrow h + Nh \rightarrow p} \quad (1)$$

- False Negative Rate: Defined as the ratio of phish class that incorrectly classified as a ham class to the total number of phish class instances.

$$FNR = \frac{Np \rightarrow h}{Np \rightarrow p + Np \rightarrow h} \quad (2)$$

- Accuracy: Defined as the percentage of correct classification over all attempts of classification.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (3)$$

- Classification Time: Defined as the total time for the classifier to build the general model, classifying the input instances and displaying the result on the interface screen.

$$Time = End\ time - Start\ time \quad (4)$$

Journal of Theoretical and Applied Information Technology
20th January 2016. Vol.83. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

8. EXPERIMENTAL RESULTS

Our experiment was tested on an Intel core i3 system with 4G RAM and windows operating system. The phishing and ham datasets combined into one dataset and converted into MySQL and the features extracted using PHP code and by employing phpmyadmin webserver [29] to facilitate features extraction process. The extracted features are in different ranges which need normalize process to get accurate result from the classifiers. Some features in the (-1, 1) range and some features have other value scales such as body_no_words and body_no_characters. Min-max normalization is used to perform a linear transformation on the extracted features. RapidMiner [30] is used to perform normalization process. In our study Java is used to implement the proposed system with 10 fold cross validation has been used for system evaluation. The output of classifiers monitored and compared with each other in term of accuracy, FP and FN. The accuracy of first stage results in Figure 4 presents the accuracies of each single classifier. RF classifier provides higher accuracy than Naïve Bayes with accuracy of 99.55%. The FPR and FNR are shown in Figure 5, where the lowest FPR and FNR provided by RF with 0.5% and 0.3% respectively. The decision maker tests the output of both classifiers and applies the rules explained in section 4. 99.40 % accuracy is achieved with 0.4% FPR which is less than both classifiers with trade off increase in FNR with 0.4% compared with RF FNR of 0.3%. Table 1 presents the results of decision maker stage.

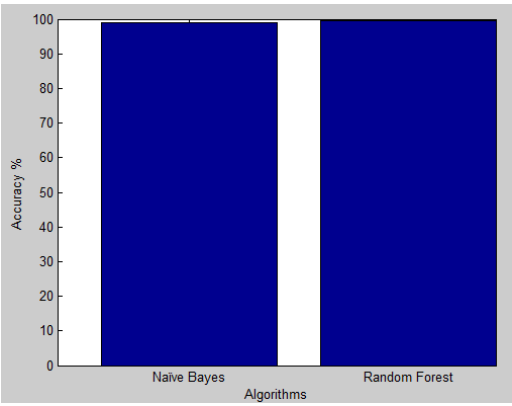

Figure4: The Ratio Of Accuracy

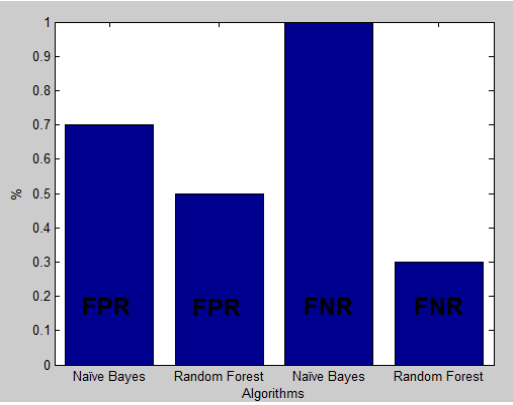

Figure5: FPR and FNR of first stage

Table 1: Decision Maker Results.

Accuracy	FPR	FNR
99.40 %	0.4%	0.7%

The correctly classified instances in first stage used to train C3. From our observations the number of correctly classified instances was 8897 and misclassified samples of 53 (20 ham, 33 phish) where C3 is used to classify these samples. C3 successfully classified 19 ham samples with just one classified as phish and 26 phish with 7 samples incorrectly classified as ham email. It is clear that our system misclassified just 1, 7 phishing and ham instances respectively. We compared the proposed MSPEDS with state of the art work as presented in Table2.

Table 2: Classifiers Comparison With MSPEDS.

Method	FPR	FNR	Accuracy
MSPEDS	0.02 %	0.15%	99.91%
A lexical URL approach [14]	0.52%	0.65%	99.38%
Pilfer [5]	4 %	0.20%	99.50%
Multi-tier [20]	0.03%	0.18%	97.00%
Ensemble [13]	0.08%	5.60%	94.4%

Server side solutions need high speed processing time, which is a very important factor to be considered and it must be as small as possible. As MSPEDS proposed for server side, it is benefit to evaluate the suitability of the system for the online environment. The time required to build the system and finish the classification is 1.5 minutes, which is regarded as long time and not suitable for server side environments. However, more work is required to reduce the classification time where one solution can employ the hardware accelerators to support online processing for phishing detection.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

9. CONCLUSION AND FUTURE WORK

MSPEDS is a multi-stage classification system which employs the best algorithms to distinguish between ham and phishing emails. The approach uses the best features exit in the literature. The experimental results proved that our proposed model has good performance and high accuracy compared with other systems. The processing time relatively high to be suitable for online processing. Therefore one of our future works is to improve the processing time using hardware accelerator technology.

REFRENCES:

[1] A.A. Almomani et al, "A Survey of Phishing Email Filtering Techniques", *Communications Surveys Tutorials (IEEE)*, Vol. 15, No. 4, 2013, pp. 2070-2090.

[2] APWG, "Phishing activity trends report", *Anti Phishing Working Group*, 2014.

[3] GARTNER, "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,* Accessed: 2015-01-22.

[4] M. Chandrasekaran et al, "Phishing email detection based on structural properties", *NYS Cyber Security Conference,* 2006, pp. 1-7.

[5] L. Fette, N. Sadeh , and A. y Tomasic , "Learning to detect phishing emails", *Proceedings of the 16th international conference on World Wide Web,* 2007, pp. 649--656.

[6] J. Zhang et al, "A behavior based detection approach to mass-mailing host", *2007 International Conference on Machine Learning and Cybernetics,* Vol. 4, 2007, pp. 2140-2144.

[7] A. Nimeh, et al. "A comparison of machine learning techniques for phishing detection", *Proceedings of the anti-phishing working groups 2nd annual ecrime researchers*, 2007, pp.60-69.

[8] A. Syed et al, "Feature selection for Spam and Phishing detection", *NIPS 2007 Workshop on Machine Learning in Adversarial Environments for Computer Security*, 2008.

[9] F. Toolan, and J. Carthy, "A Survey of Phishing Email Filtering Techniques", *eCrime Researchers Summit*, 2010, pp. 1-12.

[10] I. R. Ahamid, J. Abawajy, and T. H. Kim, "Using feature selection and classification scheme for automating phishing email detection", *Studies in Informatics and Control*, Vol. 1, No. 22, 2013, pp. 61-70.

[11] A. Nimeh et al, "A comparison of machine techniques for phishing detection", *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit,* 2007, pp. 60-69.

[12] B. Ram, S. Mukkamala, and A. H. Sung, "Detection of phishing attacks: A machine learning approach", *Soft Computing Applications in Industry*, 2008, pp. 373-383.

[13] A. Sabari M. Vahidi and B.M. Bidgoli, "Learn to detect phishing scams using learning and ensemble methods", *Proceedings of the IEEE/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2011, pp. 311-314.

[14] M. Khonji, Y. Iraqi and A. Jones, "Enhancing Phishing E-Mail Classifiers: A Lexical URL Analysis Approach", *International Journal for Information Security Research (IJISR),*Vol. 2.1, No. 2, 2012.

[15] W.N. Gansterer and D. Pölz, "E-mail classification for phishing defense", *In Advances in Information Retrieval*, 2009, pp. 449-460.

[16] M. Bazarganigilani, "Phishing E-Mail Detection Using Ontology Concept and Nave Bayes Algorithm", *International Journal of Research and Reviews in Computer Science*, Vol. 2, No. 2, 2011.

[17] A. Bergholz et al, "Improved Phishing Detection using Model-Based Features", *In CEAS.*2008.

[18] A. Bergholz et al, "A Real-Life Study in hishing Detection", *in Seventh annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference Redmond (CEAS)*, 2010.

[19] Ma, Liping, et al, "Detecting phishing emails using hybrid features", *Ubiquitous, Autonomic and Trusted Computing (UIC-ATC'09)*, 2009, pp. 493-497.

[20] D. Castillo et al, "An integrated approach to filtering phishing E-mails", *Computer Aided Systems Theory–EUROCAST 2007*, 2007, pp. 321-328.

[21] R. Islam and J. Abawajy, "A multi-tier phishing detection and filtering approach", *Journal of Network and Computer Applications*, Vol. 36, No. 1, 2013, pp. 324-335.

[22] M. R Islam, J. Abawajy, and M. Warren, "Multi-tier phishing email classification with an impact of classifier rescheduling", *Pervasive Systems, Algorithms, and Networks (ISPAN)*, 2009, pp. 789-793.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[23] A.A. Almomani et al, "Phishing Dynamic Evolving Neural Fuzzy Framework for Online Detection Zero-day Phishing Email", *arXiv preprint*, 2013.

[24] S. Marchal et al, "PhishStorm: Detecting Phishing With Streaming Analytics", *IEEE Transactions on Network and Service Management*, Vol. 11, No. 4, 2014, pp. 458-471.

[25] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: a literature survey", *Communications Surveys Tutorials IEEE,* Vol. 15, No. 4, 2013, pp. 2091-2121.

[26] I. R. Ahamid, J. Abawajy, and T. H. Kim, "Using feature selection and classification scheme for automating phishing email detection", *Studies in Informatics and Control,* Vol. 22, No. 1, 2013, pp. 61-70.

[27] J. Nazario, ""Phishingcorpus homepage", *http://monkey.org/%7Ejose/wiki/doku.php?id=*

PhishingCorpus, Vol. 22, No. 1, 2013, pp. 61-70, 2006.

[28] Apache Software Foundation, ""Spamassassin homepage",*http://spamassassin.apache.org/*.2006

[29] phpMyAdmin, "phpMyAdmin homepage", *http://www.phpmyadmin.net/home_page/index.php/*.2015

[30] rapidminer, "rapidminer homepage", *https://rapidminer.com/*.2015