PHISHING EMAIL CLASSIFIERS EVALUATION: EMAIL BODY AND HEADER APPROACH

[1]AMMAR YAHYA DAEEF, [2]R. BADLISHAH AHMAD, [3]YASMIN YACOB, [4]NAIMAH YAAKOB, [5]MOHD. NAZRI BIN MOHD. WARIP

[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]nazriwarip@unimap.edu.my

ABSTRACT

The internet is a great importance to millions of people in their social and financial activities every day. This not only limited to individual users of the Internet, but also to organizations for the purposes of trade and others. A huge number of financial activities occur every day with millions of dollars are transferred where this large amount of financial events open the appetite of fraudsters to implement fraudulent activities. Thus, users vulnerable to many threats, including the theft of private information, banking information, and many more. Recently, phishing is a serious threat which steals user's sensitive information and regarded as the most profitable cybercrime. Phishing mainly relies on email claiming originating from trusted source contains an embedded link to redirect victims to not benign website in order to get users financial data. As the risk of Phishing emails increases progressively, detecting and overriding this phenomenon has become very urgent, especially the zero day phishing campaigns which are new phishing emails not seen by anti-phishing tools. Although there are several solutions for phishing detection such as blacklists and heuristics, there is no clear discussion about the required processing time and the complexity of the designed solutions. This paper aims to make such dissection for server side solutions which proved to be the best choice to defeat zero day attacks.

Keywords: *Phishing, Emails, Body features, Header features, and Classifiers*

1. INTRODUCTION

A phishing email is a fake email claiming to originate 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 others [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 Gartner survey [3] 109 million people in USA have got phishing e-mail with approximately 1,244 USD dollars loss per victim.

The message of phishing email can be simple or very complicated and can deceive even the professional users of the internet. These attacks are destroying the electronic commercial trading through the internet world which lead to loss the use and trust of the internet [1]. Detection of phishing email has been studied by several researchers [4-10] and they provided many good techniques to mitigate phishing effects. In the same context, phishing email detection solutions mainly use blacklist or heuristic methods and can be positioned at different levels of attack flow [1], such as server 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). Server side approaches are regarded as the most effective option to confront zero day attacks problem and these solutions have been developed based on machine learning (ML) to detect the phishing emails using content filtering. Regardless of the fact that there are good results in the literature for such solutions in term of accuracy, there is lack discussion about the complexity and detection time of these solutions to answer the following question are the current solutions can work with online speed to support server side speed requirements?

As most solutions in server side depend on feature extraction and data mining algorithms, we interested to evaluate the most effective classifiers in previous works by using features extracted from body and header of email to answer the aforementioned question. The remainder of this paper is organized as follows: Section 2 presents the works related to our research. Section 3 gives an overview of the 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 WORKS

Recently, many researches targeted the problem of phishing email detection to confront the growing phishing attacks, where these solutions comprise server side and client side techniques. Server side solutions from the name 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 uses 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 on server side depend on features extraction from the phishing email and by employing machine learning algorithms to classify labeled emails as phishing and legitimate. These algorithms can be used to classify new received emails from a stream of email [1]. However, there are lack of researches on the processing requirements of classifiers in server side to answer the question of whether the current solutions provide the required processing speed for online mode? In this section, machine learning based techniques for phishing email detection on the server side is discussed.

Research in [11] compare 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). LR classifier provided the best precision of 95.11% and 04.89% 17.04% FP and FN respectively. However, there is no discussion about the classification time or the complexity of these classifiers and their suitability for the online classification environment. 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%. However, the authors used the same machine learning algorithms used by others with lack discussion of classification time and filters complexity. Authors of [13] used 30 features, 15 features proposed by previous papers while the rest 15 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 of this study is the online features which depend on internet connection, where extraction several online features will affect the filtering system performance in large email servers. Moreover, there is no discussion about the complexity and classification time of classifiers. New approach for detecting phishing email is proposed by [14]. This approach employed the ontology concept with training and testing data sets in order to help Naive Bayes algorithm. The proposed heuristics offer a word as an attribute and its value the frequency of this word. Small size data set used with 200 phishing email and provided 94.87%. The main limitation in this study is the small size of data set which insufficiently characterizes the proposed concept. In addition, using ontology approach puts overhead on the classifiers which make it not suitable for online environments. Phishing emails detection using PILFERS method proposed in [5]. One feature which represent the age of linked-to-domain names extracted using WHOIS query, while 9 features 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 means that the accuracy is not good enough. Research in [15] classified the emails as phishing and ham by employing statistical classification. New features generated using Dynamic Markov Chains and Class-Topic Models. 27 features used where the model provided a reduction in memory consumption in comparison with other papers with better results than PILFER method on the same data set. In addition, this method tested in online environment at commercial internet server provider [16]. However, this method is time consuming as it employed many algorithms for classification. Hybrid features approach is proposed by [17]. The hybrid features consist of orthographic and derived, content, and method for feature selection. Information gain algorithms used for features selection, 7 features 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 decision required five stages and the data sets used are not standard which make the results are not benchmarkable. FRALEC System proposed by [18] 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, 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 [19, 20] proposed multi-tier classification system. The system used 3 classifiers where the email features extracted and classified in sequence and the outputs are sent to the decision classifier process. The results showed c2 AdaBoost, c3 Naive Bayes, and c1 SVM provided the best results with an accuracy of 97%. However, the system is complex and time consuming due to the many stages. PDENFF [21] 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 [22] 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 email server. 12 features from a URL 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, there are not clear discussion about the required processing time and the complexity of the designed solutions. Our study differs from the previous work by making clear discussion about the complexity of the classifiers in term of the generated rules and the time required for classification where we used the most effective features of email body and header existed in the literature.

3. RESEARCH FRAMEWORK

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

In this research the most effective features found in literature are extracted from part D and C. Figure 2 clearly shows the steps of the framework used in this research. The study comprises three phases implemented respectively. The first phase, to prepare the dataset to get a sufficient data set to learning the algorithms. The second phase comprises the used algorithms and making the learning process to bring out the classifier to detect phishing email. Finally, the required analysis to evaluate the final results implemented in phase three.

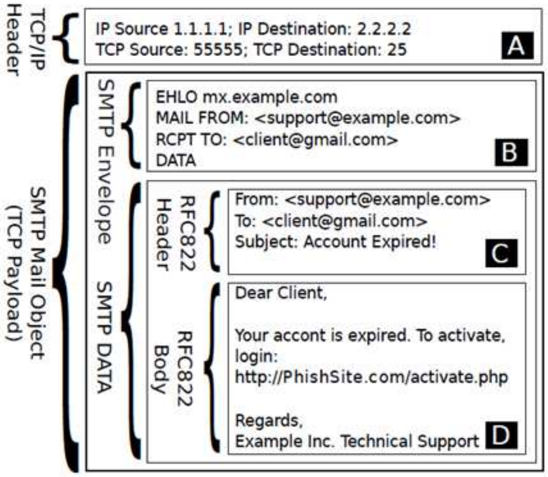
TCP/IP Header
IP Source 1.1.1.1; IP Destination: 2.2.2.2
TCP Source: 55555; TCP Destination: 25
A
SMTP Mail Object (TCP Payload)
SMTP Envelope
EHLO mx.example.com
MAIL FROM: <support@example.com>
RCPT TO: <client@gmail.com>
DATA
B
SMTP DATA
RFC822 Header
From: <support@example.com>
To: <client@gmail.com>
Subject: Account Expired!
C
RFC822 Body
Dear Client,
Your accont is expired. To activate, login:
http://PhishSite.com/activate.php
Regards,
Example Inc. Technical Support
D

Figure 1: Email data parts [23]

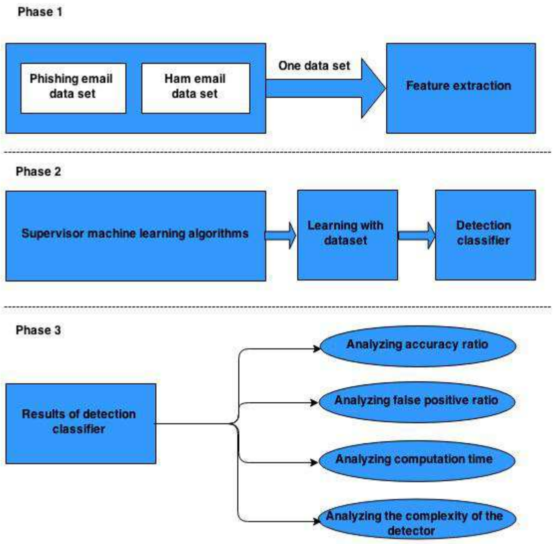
Phase 1
Phishing email data set
Ham email data set
One data set
Feature extraction
Phase 2
Supervisor machine learning algorithms
Learning with dataset
Detection classifier
Phase 3
Results of detection classifier
Analyzing accuracy ratio
Analyzing false positive ratio
Analyzing computation time
Analyzing the complexity of the detector

Figure 2: Research framework

3.1 Phase 1: Dataset Preparation and Feature Extraction

The preparation processing of the datasets implemented to make them suitable to the study requirements where these processing comprises extract and normalize 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 Rules.PART, Tree.J48, and Random Tree.

3.3 Phase 3: Evaluation

The performance of the detection classifiers will be evaluated in terms of evaluation metrics as will be explained in the next sections.

4. SELECTED FEATURES

The 40 features collected in [9] are used by most previous studies. In our study, we collected the features from [9] and [24] where the total number of the selected features is 25.

4.1 features of email body

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

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

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

4.1.3 body_dear_word: this feature represent the presence of dear word in email body 1 and -1 if not.

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

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

4.1.6 body_no_characters: This feature counts the total number of characters existing in the email body.

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

4.1.8 body_no_distinctwords: This feature counts the total number of distinct words existing in the email body.

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

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

4.1.11 body_no_functionwords: This feature count 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.

4.2 features of email header

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

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

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

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

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

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

4.2.6 subject_no_words: This feature provide the total number of words occur in the subject of an email.

4.2.7 Subject_no_characters: This feature provides the total number of characters occur in the subject of an email.

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

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

4.2.10 send_unmodal_domain: If the address of sender uses a unmodal domain name the value will be 1 and -1 otherwise.

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

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

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

4.2.14 DMID_validity: This binary feature return 1 if the message ID field is forged by the phisher and -1 otherwise.

5. DATASETS

The datasets used in our study are publicly available and used by most studies in related work. The phishing dataset downloaded from [25], where the phishing dataset consist of 4550 phishing emails. On the other hand, the ham emails downloaded from [26] with the 4400 legitimate emails. Figure 3 shows the ratio of phishing to ham emails in the dataset.

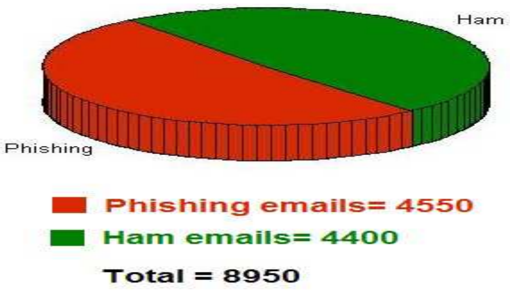

Figure3: The ratio of phishing and ham emails

6. EVALUATION METRICS

The machine learning classification performance evaluated commonly based on widely used metrics namely: False Positive Rate (FPR), True Positive Rate (TPR), Precision, Accuracy, F and measure. In addition, classification time and number of rules.

- 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.

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

- True Positive Rate (TPR): called as Recall which defined as the rate of phishing attacks correctly detected to total number of all phishing attacks.

$$TPR = \frac{Np \rightarrow p}{Np \rightarrow p + Np \rightarrow h} \qquad (2)$$

- Precision: Defined as the ratio of correctly detected phishing instances to all instances detected as a phishing attack.
- Accuracy: Defined as the percentage of total TP and TN that are correct.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \qquad (4)$$

- F-Measure: Defined as the mean of harmonic of precision and recall.

$$F\ Meature = \frac{2 * prcision * recall}{Precision + Recall} \quad (5)$$

- 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.

$$Time = End\ time - Start\ time\ (6)$$

- Number of rules: Defined as the total number of rules generated by the classifier to achieve the classification.

7. 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 [27] to facilitate features extraction process. The extracted features are in different ranges which need to 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 [28] is used to perform normalization process. After extracting all features and performs normalization process the dataset format converted into Attribute-Relation File Format (ARFF). In our study, Weka [29] (Waikato Environment for Knowledge Analysis) with 10 fold cross validation is used for classifiers evaluation. We used three classifiers namely: Rules.PART, Tree.J48, and Random Tree with their default parameters values.

The main objective to evaluate the accuracy of the selected classifiers and their suitability to work in online server side environment. Table 1 shows the classification accuracy of all classifiers. In all classifiers, the accuracy is acceptable with Rules.PART and Tree.J48 very close to each other.

As presented in table 2, the FPR for all classifiers are too high with the worst value provided by Tree.J48. As a result, more works are needed to reduce the rate of FP by using more effective features. However, the best Recall value is provided by Random Tree classifier.

Table 1: Classifiers Accuracy.

Algorithm	Accuracy
Rules.PART	98.0765 %
Tree.J48	98.0644 %
Random Tree	97.895 %

Table 2: Performance evaluation results of Classifiers.

Algorithm	FP Rate	Precision	Recall	F-Measure
Rules.PART	2%	98.2%	98%	98.1%
Tree.J48	2.1%	98.2%	97.9%	98.1%
Random Tree	1.8%	97.6%	98.2	97.9%

One of the most important aims of this research is to evaluate the suitability of the classifiers for the online environment. Therefore, the time of classification and the number of rules generated are very important factors to be considered and they must be as small as possible. Table 3 and Figure 4 show the classification time of each algorithm. The minimum time is provided by Random Tree (0.1 second) and Rules.PART has the highest time (0.88 second).

Table 4 and Figure 5 show the number of rules generated by each classifier. Rules.PART provided the lowest number of rules (49), where the Random Tree has the largest rules value among the three classifiers.

From our results, the best algorithm which provides the best trade off in term of accuracy, classification time, and the rule number is Tree.J48 which makes it the most suitable classifier for the online environment. However, the 2.1% FPR is too high this make the next step in our research to reduce this value by using more features exist in literature or proposing new features.

Table 3: Classifiers Classification Time.

Algorithm	Classification time
Rules.PART	0.88
Tree.J48	0.45
Random Tree	0.1

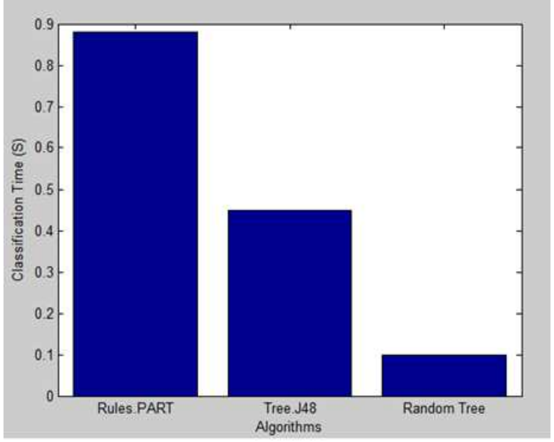

Figure 4: Classifiers Classification Time.

Table 4: Classifiers Number of Rules.

Algorithm	Number of classification rules
Rules.PART	49
Tree.J48	121
Random Tree	430

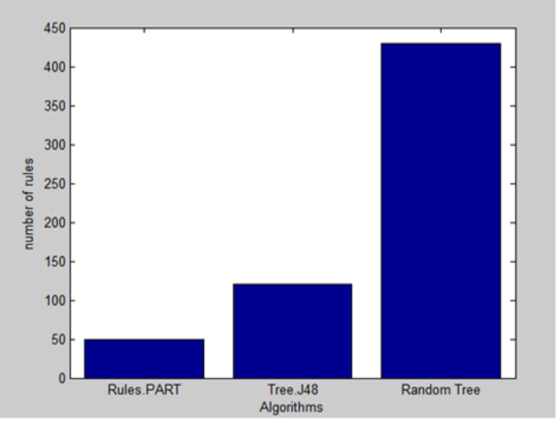

Figure 5: Classifiers Number of Rules.

8. CONCLUSION AND FUTURE WORK

In this paper, we presented a study on the appropriateness of existing classifiers on a server side for work in line speed. We studied the commonly used classifiers for phishing email detection with features extracted from the header and the body of the email. As evaluated using open source phishing and ham emails datasets, the Tree.J48 classifier provides the best tradeoff in term of accuracy, the number of rules, and classification time. However, more work is needed to reduce FPR of Tree.J48. Our future work includes the need to reduce the FPR by proposing new features. In addition, we are trying to collect fresh phishing and legitimate datasets to evaluate the effect of concept drift for the selected features to improve the phishing detection accuracy.

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

[14] 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.

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

[16] 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.

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

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

[19] 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.

[20] 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.

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

[22] 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.

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

[24] 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.

[25] J. Nazario, "Phishingcorpus homepage", *http://monkey.org/%7Ejose/wiki/doku.php?id=PhishingCorpus,* Vol. 22, No. 1, 2013, pp. 61-70, 2006.

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

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

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

[29] Weka 3: Data Mining Software in Java , "WEKA HOMEPAGE", *HTTP://WWW.CS.WAIKATO.AC.NZ/ML/WEKA/.*2015