Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 425731, 6 pages
http://dx.doi.org/10.1155/2014/425731

Research Article

Classification of Phishing Email Using Random Forest Machine Learning Technique

Andronicus A. Akinyelu and Aderemi O. Adewumi

School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Private Bag Box X54001, Durban 4000, South Africa

Correspondence should be addressed to Aderemi O. Adewumi; laremtj@gmail.com

Received 23 January 2014; Accepted 11 March 2014; Published 3 April 2014

Academic Editor: Olabisi Falowo

Phishing is one of the major challenges faced by the world of e-commerce today. Thanks to phishing attacks, billions of dollars have been lost by many companies and individuals. In 2012, an online report put the loss due to phishing attack at about $1.5 billion. This global impact of phishing attacks will continue to be on the increase and thus requires more efficient phishing detection techniques to curb the menace. This paper investigates and reports the use of random forest machine learning algorithm in classification of phishing attacks, with the major objective of developing an improved phishing email classifier with better prediction accuracy and fewer numbers of features. From a dataset consisting of 2000 phishing and ham emails, a set of prominent phishing email features (identified from the literature) were extracted and used by the machine learning algorithm with a resulting classification accuracy of 99.7% and low false negative (FN) and false positive (FP) rates.

1. Introduction

Phishing is one of the different (and lucrative) types of fraud committed today. In criminal law, fraud is defined as a deliberate deception made for the sole aim of personal gains or for smearing an individual's image. In general terms, fraud can be defined as an act of deceiving people into revealing their personal information, basically for the purpose of financial or personal gains.

Phishing is an act that attempts to electronically obtain delicate or confidential information from users (usually for the purpose of theft) by creating a replica website of a legitimate organization. Phishing is usually perpetrated with the aid of an electronic device (such as ipads and computer) and a computer network; they target the weaknesses existing in various detection systems caused by end-users (who are considered to be the weakest element in the security chain) [1, 2]. Phishing attackers usually perpetrate their evil by communicating well composed messages (known as social engineered messages) to users in order to persuade them to reveal their personal information which will be used by the fraudster to gain unauthorized access to the user's account. For example, a fraudulent email sent to a user might contain a malware (called man in the browser (MITB)), this malware could be in form of web browser ActiveX components, plugins, or email attachments; if this user ignorantly download this attachment to his pc, the malware will install itself on the user's pc and would in turn transfer money to the fraudster's bank account whenever the user (i.e., the legitimate owner of the bank account) tries to perform an online transaction [1].

Fraudulent activities is on the increase daily; individuals and companies who have been victims in the past now seek for ways to secure themselves from been attacked again. To achieve this, their defense mechanism has to be more secured to prevent them from falling prey again, which implies that the existing defense system (its designs and technology) needs to be greatly improved [3]. Behdad et al. [3] pointed out that improving the defense system is not enough to stop fraudsters as some of them could still penetrate; the system should also be able to identify fraudulent activities and prevent them from occurring.

Several traditional approaches used by various email filters today are static in nature; they are not robust enough to handle new and emerging phishing patterns; they only have

Table 1: Data used for testing.

Total Samples	**2000**
Total Phishing Emails	200
Total legitimate email	1800

the ability to handle existing phishing patterns, thus leaving email users prone to new phishing attacks. This is a loop hole because fraudsters are not static in their activities; they change their mode of operation as often as possible to stay undetected. This motivated many researchers into seeking for other effective techniques that can handle both known and emerging fraud, and this led to the discovery of machine learning algorithms.

Machine learning (ML) is a branch of artificial intelligence (AI) that employs the method of data mining to discover new or existing patterns (or features) from a dataset which is then used for the purpose of classification. In this work, we extracted a set of 15 prominent phishing features (identified from the literature) from a dataset consisting of 2000 emails; and after extraction, for each email, a vector representation of these features is formed, which is then used to train our classifier (see Table 1).

We present a detailed description of our machine learning method in this paper. In Section 2 we gave an overview of existing phishing detection techniques and also gave a brief description of our 15 features; in Section 3 we gave the details on our machine learning algorithm and also explained the result we obtained; finally we concluded the paper in Section 4.

2. Related Work

Prakash et al. [4] used a combination of blacklists and heuristics and they achieved a FP and FN rates of 5% and 3%, respectively. Cranor et al. [5] conducted an evaluation on some antiphishing toolbar and reported SpoofGuard (developed by Chou et al. [6]) to have a FP rate of 38% and a FN rate of 9%. Also, Yu et al. [7] developed a heuristics-based phishing detection system which achieved a FP and FN rates of 1% and 20%, respectively. Zhang et al. [8] also used heuristics and their method achieved a FP rate of 3% and FN rate of 11%. Fette et al. [9] used ML-based method and they achieved a FP rate of 0.0013% and a FN rate of 0.0036%. Bergholz et al. [10] combined the use of heuristics and ML-technique and their method achieved a FP rate of 0% and a FN rate of 1%.

All these proposed methods have relatively high FP rate and FN rate except for Fette et al. [9] and Bergholz et al. [10] whose methods achieved excellent results with very low FP and FN rates. However, Bergholz et al. [10] made use of model-based features involving the processing of images which in turn could lead to an increase in the run time and space. Fette et al. [9] also made use of a feature (the age of linked to domain names) that has to be obtained by sending of queries over the network, which could also lead to an increase in run time. In our method, all the features are extracted directly from each email itself; we do not need to send queries over the network or store large data, thereby reducing the run time and space complexities.

3. Problem Description

3.1. Email Filtering. Phishing attacks are prominently perpetrated via sending of emails. These emails usually contain social engineering messages (with specific phrases) that demand users to perform specific actions (such as clicking a URL). Therefore, the content of these emails are useful features for phishing detection.

Very few phishing email filters have been developed as opposed to many existing email filters that have been developed for spam emails. Many of them used several phishing detection techniques ranging from blacklists [4], visual similarity [11], heuristic [12], and machine learning [10]. Of all these techniques, ML-based technique (such as our own) achieved the best result.

Many approaches have been proposed to build email filters but many of them are only suitable for handling spam emails. For example, a popular method (known as "bag-of-words") extracts all the words present in an email, identifies the highest occurring words, and uses each of these words as the features for classification. This method (a.k.a. text classification method) works very well for filtering of spam emails but not for phishing emails, because phishing email contains some unique features that are only specific to phishing attacks, features such as presence of IP-based URLs and presence of nonmatching URLs. This indicates that spam filtering approaches cannot effectively handle phishing emails; therefore, a list of phishing-attack-specific features has to be defined and used to build an effective email filtering system. It is worth noting that some existing spam filtering approaches (such as SpamAssasin [13] and Spamato [14]) went beyond just "bag-of-word" methods; they designed a set of spam emails heuristics that could also successfully detect some existing phishing emails features (such as presence of IP-based URLs). These methods can be combined with our method to build a hybrid (phishing and spam) email filtering system with very low FPs and FNs.

3.2. Features Used in the Email Classification. The features we used for our email classification are described in this section. These features were identified from different literature; combination of these features together forms a feature set that effectively classified emails into phishing and nonphishing.

A group of 15 features frequently used by phishing attackers was identified from different literature and used in this paper. Although the features set are few (compared to some filters that used hundreds of features for detection), a high accuracy was still achieved. These features are described in the remaining part of this section.

3.2.1. URLs Containing IP Address. The URL for many legitimate websites usually contains the name of the website (e.g., http://www.yahoo.com/, which tells us that this URL can be used to connect to the website of yahoo). For the purpose of identity hiding, phishers usually mask their

website name by using URLs that contain IP address (e.g., "http://167.88.12.1/signin.ebay.com"); therefore the presence of IP-based URLs in an email is an indication that the email is a potential phishing email. This feature was used in [9].

3.2.2. Disparities between "href" Attribute and LINK Text. The HTML <a> tag defines an anchor that may be used to establish a link to another website. Linking to another website can be accomplished by defining a "href" attribute; this attribute describes the location of the website that is to be linked to. The links are usually rendered to the browser after the "Link text" has been clicked (e.g., Link Text). The link text could be a plain text (e.g., Click Here), a URL (yahoo.com), an image, or any other HTML element. If the link text is a URL (and it is a legitimate link), it should tally with the website location pointed to by the "href" attribute (e.g., yahoo.com); if there is a disparity between the href attribute and the link text (e.g., boguus.com), then the link is likely pointing to a phishing website. All the links (containing a URL-based link text) in an email are checked and if there is a disparity between the link text and the href attribute, then a positive Boolean feature is recorded. This feature was used in [9].

3.2.3. Presence of "Link," "Click," and "Here" in Link Text of a Link. The text of the links present in most phishing emails usually contain words like "Click," "Here," "Login," and "Update." For this feature, all the text of each link in an email is checked and a Boolean value is recorded based on the presence or absence of the words *Click*, *Here*, *Login*, *Update*, and *Link* in the Link text. Similar feature was used in [9, 10].

3.2.4. Number of Dots in Domain Name. The number of dots that should be contained in the domain name of a legitimate organization should not be more than three as proposed by Emigh [15]. A binary value of 1 is recorded if an email contains a URL whose number of dots is above three.

3.2.5. HTML Email. The email format for each email is defined by MIME standards. The MIME standard defines the type of content contained in each email. The content type (defined by the content-type attribute) could be plain text (indicated by "text/plain"), HTML (indicated by "text/html"). Fette et al. [9] proposed that an email is a potential phishing email if it contains a content-type with attribute "text/html"; they based their argument on the fact that it is almost impossible for phishing attacks to be launched without the use of HTML links.

3.2.6. Presence of Javascript. Javascript can either be embedded in the body of an email (using the script (<script>) tag) or in a link (using the anchor (<a>) tag). Some phishers use Javascript to hide information from users. Fette et al. [9] suggested that an email is a potential phishing email if the "javascript" string is contained in either the body of the email or in a link.

3.2.7. Number of Links. The total number of links embedded in an email is recorded and used as a feature for classification. Zhang and Yuan [16] explained that phishing emails usually contain multiple numbers of links to illegitimate websites.

3.2.8. Number of Linked To Domain. This feature (used in [9]) refers to all the URLs present in an email that are extracted, and a count is recorded for the number of distinct domain names present in each of the extracted URLs. The recorded value is used as a feature.

Take note that each domain name in an email is only counted once; subsequent occurrence (of an already counted domain name) is discarded not counted.

3.2.9. From_Body_MatchDomain Check. To extract this feature, all the domain names in an email are extracted and each of these domain names is matched with the sender's domain (i.e., the domain name referred to by the "From" field of the same email); If there is disparity between any of the comparisons, then ALmomani et al. [17] suggest that the email is likely a phishing email.

3.2.10. Word List Features. Some group of words that frequently appear in phishing emails were used as features. We grouped these words into six different groups and each of these groups is used as a single feature (making a total of six different features). For each group, presence of each word is counted and normalized. The groups of words include the following.

(1) Update; Confirm;

(2) User; Customer; Client;

(3) Suspend; Restrict; Hold;

(4) Verify; Account; Notif;

(5) Login; Username; Password; Click; Log;

(6) SSN; Social Security; Secur; Inconvinien.

This feature is similar to the one proposed by Basnet et al. [18]. Take note that some stemmed words (like secur and inconvinien were used).

4. Simulation Experiment

4.1. Data Used. For the implementation and testing of our machine learning algorithm, we used two publicly available datasets. We got our ham mails from the ham corpora provided by spam assassin project [13], and our phishing emails were gotten from the publicly available phishing corpus [19] provided by Nazario. We programmatically extracted the features described in Section 3.2 above using C#. All the emails coming from the ham corpora were labeled as ham emails and the emails coming from the phishing corpora was labeled as phishing email.

4.2. Machine Learning Implementation. When constructing our classifier, we first transformed each email into a format that will be suitable for our machine learning algorithm. Each

Begin RF Algorithm
Input: N: number of nodes
M: number of features
D: number of trees to be constructed
Output: V: the class with the highest vote
While stopping criteria is false **do**
Randomly draw a bootstrap sample A from the training data D
Use the steps below to construct tree T_i from the drawn bootstrapped sample A:
(I) Randomly select m features from M; where $m \ll M$
(II) For node d, calculate the best split point among the m features
(III) Split the node into two daughter nodes using the best split
(IV) Repeat I, II and III until n number of nodes has been reached
Build your forest by repeating steps I–IV for D number of times
End While
Output all the constructed trees $\{T_i\}_1^D$
Apply a new sample to each of the constructed trees starting from the root node
Assign the sample to the class corresponding to the leaf node.
Combine the decisions (or votes) of all the trees
Output V, that is, the class with the highest vote.
End RF Algorithm

Algorithm 1

of the emails is represented by a vector that contains a value (binary or continuous) for all the extracted features. For the purpose of testing our algorithm, we used random forest (RF) classifier [20]. More details on RF algorithm are provided below.

4.3. Random Forest: Overview. Random forest (RF) is an ensemble learning classification and regression method suitable for handling problems involving grouping of data into classes. The algorithm was developed by Breiman and Cutler [21]. In RF, prediction is achieved using decision trees. During the training phase, a number of decision trees are constructed (as defined by the programmer) which are then used for the class prediction; this is achieved by considering the voted classes of all the individual trees and the class with the highest vote is considered to be the output.

RF method has also been used to solve similar problem in the literature, such as in [9, 22, 23]. A summary of how a forest (i.e., collection of trees) is constructed is explained in Algorithm 1.

For more details about random forest, kindly refer to [20, 24].

In this work, we trained and tested our classifier using 10-fold cross validation. In 10-fold cross validation, the dataset is divided into 10 different parts; 9 of the 10 parts are used to train the classifier and the information gained from the training phase would be used to validate (or test) the 10th part; this is done 10 times, such that, at the end of the training and testing phase, each of the parts would have been used as both training and testing data. This method (i.e., cross validation method) ensures that the training data is different from the test data. In machine learning, this method is known to provide a very good estimate of the generalization error of a classifier.

4.4. Result and Discussion. Machine learning involves two major phases: the training phase and the testing phase. The predictive accuracy of the classifier solely depends on the information gained during the training process; if the information gained (IG) is low, the predictive accuracy is going to be low, but if the IG is high, then the classifier's accuracy will also be high.

As stated above, we used 10-fold cross validation. In our random forest classification, before the decision trees are constructed, the information gained for all the 15 features is calculated (using the IG method explained by Mitchell [24]) and the features with the best eight IG are selected and used for constructing the decision trees; the mode vote (from all the trees) is then calculated and used for the email prediction. Information gain is one of the feature ranking metric highly used in many text classification problems today. More details about our algorithm are described in the next section below.

We tested our method using varied dataset sizes (as shown in Table 2); this was done to know the performance of the algorithm on both small and large datasets. The full result is reported in Table 2. As shown in the table, the algorithm performed best when tested on the dataset that has the largest size (having an overall accuracy of 99.7%, FN rate of 2.50%, and FP rate of 0.06%); this implies that our method will work effectively if applied to real world dataset, which is usually large in size. Our method also achieved a higher prediction accuracy (99.7%) compared to an accuracy of 97% achieved by Fette et al. [9].

The computer used in running this test is a 32-bit desktop, having a processor speed of 2.20 GHz and a RAM size of 2.00 GB.

Table 3 and Figure 1, respectively, show a comparison between our method and another similar work in literature that also had a good result.

Table 2: 10-Fold cross validation Result.

S/N	Dataset Information			Performance Evaluation							
	Email Per Folder	Total Email	P : H Ratio (%)	PA (%)	SR	FP (%)	FN (%)	*R* (%)	Pr (%)	*F*-M (%)	*T* (s)
1	15	150	48 : 52	98.00	0.98	0.00	4.11	95.80	100	97.79	11.82
2	30	300	33 : 67	98.33	0.99	0.00	4.00	96.00	100	97.75	21.03
3	50	500	20 : 80	99.20	0.99	0.00	4.00	96.00	100	97.78	33.47
4	100	1000	10 : 90	99.60	0.99	0.00	4.00	96.00	100	97.78	65.46
5	200	2000	10 : 90	99.70	0.99	0.06	2.50	97.50	99.47	98.45	141.25

Key: PA: Prediction Accuracy, SR: Success Rate, FP: False Positive, FN: False Negative, *R*: Recall, Pr: Precision, *T*: Time, *F*-M: *F*-Measure, P : H: Phish : Ham.

Table 3: Classification Result for Random Forest ML on the best eight features.

Technique	FP-Rate	FN-Rate	Precision	Recall	*F*-Measure
Fette et al. [9]	0.13%	3.62%	98.92%	96.38%	97.64%
RF Result	0.06%	2.50%	99.47%	97.50%	98.45%

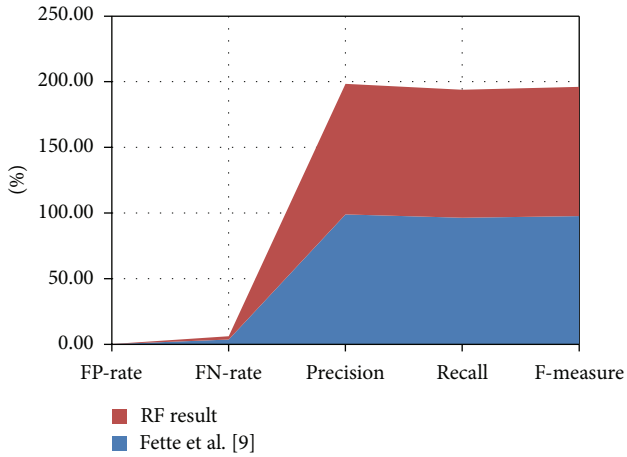

Figure 1: ROC curve showing the comparison between our work and Fette et al.'s [9].

5. Conclusion

Phishing has become a serious threat to global security and economy. The fast rate of emergence of new phishing websites and distributed phishing attacks has made it difficult to keep blacklists up to date. Therefore, in this paper, we have presented a content-based phishing detection approach which has bridged the current gap identified in the literature. This approach yielded high classification accuracy of 99.7% with negligible false positive rate of about 0.06%.

In the future, we plan on improving this work by combining this approach with a nature inspired (NI) technique. NI techniques (such as PSO or ACO) can be used to automatically and dynamically identify the best phishing features (from a feature space) that can be used to build a robust phishing email filter with very high classification accuracy. Using this technique will with no doubt enhance the predictive accuracy of a classifier since effective classification of emails depends on the phishing features identified during the learning stage of the classification.

Due to the rapid change in phishing attack patterns, current phishing detection techniques need to be greatly enhanced to effectively combat emerging phishing attacks. An online report noted that, in the future, phishers will shift their attention from syntactic attacks (i.e., attacks exploiting technical vulnerabilities) to semantic attacks (i.e., attacks exploiting social vulnerabilities). To handle some of these emerging phishing attacks, an online report recommended that companies should move from session-based security (based on a secure log-in), to message-based security (based on explicit authentication of individual transactions). Also, Fette et al. [9] suggested that using knowledge-based models built on federated identities and semantic based technologies will also help to combat carefully planned phishing attacks in the future.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: a literature survey," *IEEE Communications & Surveys Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013.

[2] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, "Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions," in *Proceedings of the 28th Annual SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*, pp. 373–382, Atlanta, Ga, USA, April 2010.

[3] M. Behdad, L. Barone, M. Bennamoun, and T. French, "Nature-inspired techniques in the context of fraud detection," *IEEE Transactions on Systems, Man, and Cybernetics C: Applications and Reviews*, vol. 42, no. 6, pp. 1273–1290, 2012.

[4] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: predictive blacklisting to detect phishing attacks," in *Proceedings of the IEEE Conference on Computer Communications (IEEE INFOCOM '10)*, pp. 1–5, IEEE, San Diego, Calif, USA, March 2010.

[5] L. F. Cranor, S. Egelman, J. I. Hong, and Y. Zhang, "Phinding phish: an evaluation of anti-phishing toolbars," in *Proceedings of the 14th Annual Network & Distributed System Security Symposium (NDSS '07)*, San Diego, Calif, USA, 2007.

[6] N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell, "Client-side defense against web-based identity theft," in *Proceedings of the 11th Annual Network & Distributed System Security Symposium (NDSS '04)*, San Diego, Calif, USA, February 2004.

[7] W. D. Yu, S. Nargundkar, and N. Tiruthani, "PhishCatch—a phishing detection tool," in *Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC '09)*, vol. 2, pp. 451–456, Seattle, Wash, USA, July 2009.

[8] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: a content-based approach to detecting phishing web sites," in *Proceedings of the 16th International World Wide Web Conference (WWW '07)*, pp. 639–648, ACM, Alberta, Canada, May 2007.

[9] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *Proceedings of the 16th International World Wide Web Conference (WWW '07)*, pp. 649–656, Alberta, Canada, May 2007.

[10] A. Bergholz, J. de Beer, S. Glahn, M. F. Moens, G. Paaß, and S. Strobel, "New filtering approaches for phishing email," *Journal of Computer Security*, vol. 18, no. 1, pp. 7–35, 2010.

[11] Y. Cao, W. Han, and Y. Le, "Anti-phishing based on automated individual white-list," in *Proceedings of the 4th ACM Workshop on Digital Identity Management (DIM '08)*, pp. 51–59, ACM, Alexandria, Va, USA, October 2008.

[12] L. Ma, B. Ofoghi, P. Watters, and S. Brown, "Detecting phishing emails using hybrid features," in *Proccedings of the Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing (UIC-ATC '09)*, pp. 493–497, IEEE, Brisbane, Australia, July 2009.

[13] Apache Software Foundation, "Spam assassin homepage," 2006, http://spamassassin.apache.org/.

[14] K. Albrecht, N. Burri, and R. Wattenhofer, "Spamato-an extendable spam filter system," in *Proceedings of the 2nd Conference on Email and Anti-Spam (CEAS '05)*, Stanford, Calif, USA, 2005.

[15] A. Emigh, "Phishing attacks: information flow and chokepoints," in *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*, M. Jakobsson and S. Myers, Eds., pp. 31–64, John Wiley & Sons, New York, NY, USA, 2007.

[16] N. Zhang and Y. Yuan, "Phishing detection using neural network," http://cs229.stanford.edu/proj2012/ZhangYuan-PhishingDetectionUsingNeuralNetwork.pdf.

[17] A. ALmomani, T.-C. Wan, A. Altaher et al., "Evolving fuzzy neural network for phishing emails detection," *Journal of Computer Science*, vol. 8, no. 7, pp. 1099–1107, 2012.

[18] R. Basnet, S. Mukkamala, and A. H. Sung, "Detection of phishing attacks: a machine learning approach," in *Soft Computing Applications in Industry*, pp. 373–383, Springer, Berlin, Germany, 2008.

[19] J. Nazario, "Phishingcorpus homepage," 2006, http://monkey.org/%7Ejose/wiki/doku.php?id=PhishingCorpus.

[20] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[21] L. Breiman and A. Cutler, "Random forests-classification description," Department of Statistics Homepage, 2007, http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm.

[22] I. Koprinska, J. Poon, J. Clark, and J. Chan, "Learning to classify e-mail," *Information Sciences*, vol. 177, no. 10, pp. 2167–2187, 2007.

[23] C. Whittaker, B. Ryner, and M. Nazif, "Large-scale automatic classification of phishing pages," in *Proceedings of the 17th Annual Network & Distributed System Security Symposium (NDSS '10)*, The Internet Society, San Diego, Calif, USA, 2010.

[24] T. M. Mitchell, *Machine Learning*, McGraw-Hill, New York, NY, USA, 1997.