Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing

Detecting Phishing Emails Using Hybrid Features

Liping Ma,Bahadorrezda Ofoghi, Paul Watters, Simon Brown
Internet Commercial Security Laboratory (ICSL)
Centre for Informatics and Applied Optimization
Graduate School of Information Technology and Mathematical Sciences
University of Ballarat, Australia
Email: (l.ma, b.ofoghi, p.watters, s.brown)@ballarat.edu.au

***Abstract*—Phishing emails have been used widely in fraud of financial organizations and customers. Phishing email detection has drawn a lot attention for many researchers and malicious detection devices are installed in email servers. However, phishing has become more and more complicated and sophisticated and attack can bypass the filter set by anti-phishing techniques. In this paper, we present a method to build a robust classifier to detect phishing emails using hybrid features and to select features using information gain. We experiment on 10 cross-validations to build an initial classifier which performs well. The experiment also analyses the quality of each feature using information gain and best feature set is selected after a recursive learning process. Experimental result shows the selected features perform as well as the original features. Finally, we test five machine learning algorithms and compare the performance of each. The result shows that decision tree builds the best classifier.**

***Index Terms*—Information Security, Text Classification, Feature Selection, Feature Elimination**

I. Introduction

phishing is the criminally fraudulent process of attempting to acquire sensitive information such as usernames, passwords and credit card details by masquerading as a legitimate trusted by customers in an electronic communication. Communications purporting to be from banks, online organizations, internet services providers, online retailers, insurance agencies and so on. popular social web sites (YouTube, Facebook, MySpace, Windows Live Messenger), auction sites (eBay), online banks (Wells Fargo, Bank of America, Chase), online payment processors (PayPal), or IT Administrators (Yahoo, ISPs, corporate) are commonly used to lure the users.

Phishing is typically carried out by email, and it often directs users to enter details at a fake website which is almost identical to the legitimate one. Even using server authentication, it still requires skill to detect that the website is malicious. Phishing is an example of social engineering techniques used to deceive users, and exploits the poor usability of current web security technologies. Attempts to deal with the growing number of reported phishing incidents include legislation, user training, public awareness, and technical security measures.

The email may look quite authentic, featuring corporate logos and formats similar to the ones used for legitimate messages. They often include official logos from real organizations and other identifying information taken directly from legitimate Web sites, but including a deceptive URL address linking to a scam web site. To make these phishing emails be like real, the phishers may place a link that appears to go to the legitimate web site, but it actually takes customers to a scam site.

Typically, phishing emails ask for verification of certain information, such as account numbers and passwords, allegedly for auditing purposes. And because these emails look very real, up to 20% of unsuspecting recipients may respond to them, resulting in financial losses, identity theft and other fraudulent activity against them.

Researchers at Harvard and Berkeley universities reported in The Register, reveal that 23% of users only look at the content of sites when deciding whether they are legitimate. A survey of Gartner [1] on phishing attacks shows that approximately 3.6 million computers in the United States suffered losses caused by phishing, totalling approximately US$3.2 billion. Especially, though the amount of each individual lose slightly decreased, the number of individual victims rose from 2.3 million in 2006 to 3.6 million in 2007, which is a 56.5% increase.

The damage caused by phishing ranges from loss of access to email to substantial financial loss. This style of identity theft is becoming more popular and important, because of the ease with which unsuspecting people often divulge personal information to phishers. There are also fears that identity thieves can obtain some such information simply by accessing public records.

However, phishing has become more and more complicated and sophisticated so that phishers can bypass the filter set by current anti-phishing techniques and cast their bait to customers and organizations. A possible solution is to create a robust classifier to enhance the phishing email detection and protect customers from getting such emails.

By analysing phishing emails, it is observed that phishing emails often include certain phrases, for example, “security”, “verify your account”, “if you don’t update your details within 2 days, your account will be closed”, “click here to access to your account” and so on. These phrases may appear in the “subject:” line in an email or email content. Therefore, most phishing emails are largely similar in wording, especially the most important terms, such as “security”, “expire”, “unauthorized”, “account”, “login”, etc. Such terms are useful to classify if an email is a phishing email. In addition, Phishing emails often alert customer to click links to other websites which the real link is not the same as it is shown in the pages.

978-0-7695-3737-5/09 $25.00 © 2009 IEEE
DOI 10.1109/UIC-ATC.2009.103

Such emails often alert customer to login using form, script and others.

According to above observations, this paper presents work on detecting phishing emails using hybrid features including features of link, key word, form, script, etc. The experiment in Section V shows that hybrid features are good discriminators in classifying phishing emails.

The rest of the paper is structured as follows: Section II provides the background of predicting phishing emails and places our work in the context of existing work in anti-phishing and text classification; Section III gives the details of feature selection and email representation; Section IV illustrates the phishing email detection process. Section V provides experimental results on the effectiveness of the classification and feature selection. Section VI concludes the work and directions for future work.

II. Related work

Various methodologies have recently been developed for document classification and representation to assist in predicting phishing [2], [3], [4], [5], [6], [7] using different machine learning approaches. [2] developed the system PILFER using a support vector machine (SVM), [4] employed a Markov model, while [5] used the decision tree ([8], [9], [10]) as their classifier for preferring the robustness that C4.5 provides. AntiPhish [3] is a browser extension which is used to protect inexperienced users against spoofed web site-based phishing attacks. AntiPhish is a plug-in tool which keeps track of users' sensitive information and prevents this information from being passed to a web site that is considered as untrusted.

A text classification algorithm is responsible for identifying whether a web site is a phishing site based on addresses used in a form. In detail, it compares a legitimate URL and IP address with URL the page actually locates. AntiPhish focuses more on tracking sensitive information provided by a user. While [7] identified a website as a suspect phishing site when the visual similarity value is above a pre-defined threshold.

Text classification[11] aims to automatically categorize text documents into pre-defined classes/types based on their contents. As documents cannot be directly interpreted by a classifier, a feature procedure is required that transfers a document into a compact representation suitable for a learning algorithm and the classification task. Deciding which features are relative or descriptive has always been a central problem in machine learning techniques. For example, [12] defined a "relevant feature" as one that is neither irrelevant nor redundant to the target concept, an "irrelevant feature" does not affect the target concept in any way, and a "redundant" feature does not add anything new to the target concept. Therefore, selecting features (SF) is often applied before classifier induction. SF aims to select the best of the vector space from original features to refined features.

Enhanced feature extraction methods either remove non-informative terms (feature selection) or combine and transform original terms to form new features (re-parameterisation).

Another widely-deployed technique is based on using a blacklist of phishing domains to force the browser to refuse to visit, such as PwdHash [13], [6] and SpoofGuard [6], [14] by Stanford University. However, it is currently unclear how effective such blacklisting approaches are in mitigating phishing attacks in reality.

As noted previously, phishing emails contain similar semantic and structure features. Making use of these particularities, we propose a new method to select "good features" without compromising the classification accuracy. First, features are collected based on observation; then a few machine machine learning models are implemented and a model is identified according to experimental results. Features are selected by implementing with the identified model. Finally, a classifier is built using both identified model and "optimized" feature set.

Since phishers use more and more sophisticated techniques, the existing filter in a system/server is not sufficient to detect new tricks. Most existing phishing prediction is based on content and URLs, there is very little work that considers structure and orthographic features. In fact, we have found that different types of features support each other, thus, we aim to develop a technique to build a robust and stable detector using a hybrid vector space.

1) Feature space: Phisher emails are largely similar in style. Therefore, we believe that not only the content is important, but also the structural feature and special features of phishing emails.
2) Feature elimination and selection: The quality of features used in phishing detection determines greatly the effectiveness of a classifier. We implement the classifier using a few machine learning methods and evaluates the accuracy across different methods and set of features. This algorithm reinforces the classifier by eliminating "noisy" features so that only good features are selected.
3) Classifier effectiveness: We develop a method to identify classfier by analysing a large number of figures which indicate the relationship among learning methods and accuracies. From the analysing, we are able to identify confident feature set and recommend a proper classifier which performs the best.

III. Document presentation

The standard document representation used in text classification is the vector space model (VSM)[11], [15] . In this model, a document d_j is represented as a vector of feature (term) weights $w_j = (w_{1j}, \ldots, w_{|\mathcal{F}|j})$, where $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$ is the set of features that occur at least once in at least one document of the training set, and $0 \leq w_{ij} \leq 1$, where w_{ij} is the weight (normalized when necessary) of term i in document j. The elements in each row correspond to the words involved. A collection of documents can then be represented as a set of document vectors or, alternatively, as a matrix $\mathcal{W}$, where w_{ij} is the corresponding feature value of term f_i in document d_j, and $\mathcal{X}$ is called Term-Document Matrix (TDM).

A. Features Defined in Emails

A phishing email usually contains multimedia information, including image and text, where the text information may contain plain text, HTML, URLs, scripts, styles, etc. However, the information cannot be recognized by a system directly, rather it needs to be characterized according to the needs of the system.

As discussed in Section I, phishing emails contain different types of features defined manually based on observation. Three types of features are defined as:

- **Content features** are domain-specific keywords that help to identify particular semantic contexts within the document. These contexts are used to assist in identifying if sensitive information exists, such as term in blacklist.
- **Orthographic features** are style characteristics that are used to convey the role of words or sentences, such as HTML features, size of document, the existence of url, forms, scripts or images, etc.
- **Derived features** are developed by the existing content or orthographic features. For example, whether in an email, the visible link is same as the hidden link; whether the content is readable (i.e. whether the colour contrast between background and font are enough for human's vision), etc.

In our preliminary implementation, we experimented with seven features belong to the above three types:

1) links: the total number of links in an emails.
2) nonv_links: total number of invisible links. This feature is calculated by an algorithm according to vision standard provided by W3C. In particular, if the colour deference between the background and font of link in an email is less than 500, the link is considered as a invisible link.
3) nonmatching_urls: a binary value to show whether the visible url is as the same as the hidden url.
4) forms: a binary value to show the existence of any forms in an email.
5) scripts: the existence or type of the scripts in an email. The value is 0 if there is no script in the email. The value will be from 1 to 6 for different script types, namely text/execmascripts, text/javascripts, application/ecmascripts, application/javascripts, text/vbscripts and other scripts.
6) body_BL_words: the total appearance of the words in the blacklist in the body of an email. The blacklist includes sensitive terms, such as account, update, confirm, verify, secur, notif, log, click, inconvenien, bank, urgent, alert, etc. [1].
7) subject_BL_words: total appearance of the words in the blacklist in the "subject: line in the heading of an email.

B. Email Presentation

After features are defined, we developed a set of methods to extract all seven possible useful features from each email.

[1]The blacklist sample shown here is after standard stemming

Let $D = \{d_1, d_2, \ldots, d_{|D|}\}$ denote all the documents and $V = \{v_1, v_2, \ldots, v_{|V|}\}$ be the feature vector space. Where $|D|$ and $|V|$ are the number of document and size of feature vector respectively. Let a_{ij} be the value of jth feature of ith document. Therefore, the presentation of each document is $A_i = (a_{i1}, a_{i2}, \ldots, a_{i|V|})$, and each document is $A = \{a_{ij}\}$ where $i = 1, 2, \ldots, |V|; j = 1, 2, \ldots, |D|$.

The values of all features are numerical but in a different range. For example, the body_BL_words could be hundreds words while the number of nonv_links may be under five. To treat all the original features as equally important, the value of each feature is normalized before the classification process. Feature values are normalized using the quotient of the actual value over the maximum value among the feature so that numerical values are limited to the range $[0, 1]$.

IV. Detecting Phishing Emails

Our approach to detecting the phishing emails is to extract feature vectors from the emails which effectively represents the the instances. For example, the "subject" of an email contains precisely the keywords that best characterise the email class. Also, the presence and absence of each features provides additional clues as to the email's class.

The architecture of our classification system consists of four components: **Feature Generator**, **Machine Learning Method Selection**, **Inductor** and **Feature Evaluation**. The system takes email instances as input and output selected feature vector space and well trained classifier. Figure 1 illustrates the system architecture.

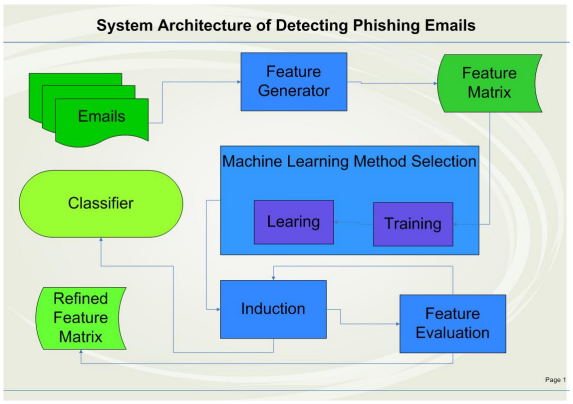

Fig. 1. System architecture

- The **Feature Generator** uses the content, orthographic and derived knowledge to produce a set of feature vectors, one per document. All the documents will be represented as a *Feature Matrix*.
- Giving a *Feature Matrix*, the Machine Learning Method Selection employs several machine learning algorithms (such as C4.5 [16], SVM [17], etc) to learn and train the classifiers. An algorithm is select according to the accuracies generated by different classifiers, the **Induction** is identified for future prediction.

- Information gain is generated by the **Induction**, and **Feature Evaluation** selects a smaller feature vector space, and evaluates accuracies before and after the removal of a feature. Again, the new training and testing are implemented and new information gain is generated. This **Induction** and **Feature Evaluation** are implemented repeatedly until the best feature vector is identified.
- Finally, the *Refined Feature Matrix* is identified which is the most optimized feature sets and a good **Classifier** is generated.

V. Experiment

We have implemented the system described in Section IV. The experiment is designed to illustrate the effectiveness of the detection and to show the potential of the work. The aim of our experiments is to provide some evidence on how effective the new method is and show the robustness of smaller vector space.

The data has been implemented using five learning algorithms. In this section, we will represent results of single classifier training, information gain and feature selection, compare accuracy between original data and refined data, and finally evaluate the five machine learning methods according to our experimental results.

Performance is measured in accuracy which is a percentage of correct answers over total number of instances.

A. Initial Implementing Using C4.5

The data used in our experiment are the live emails received by WestPac and their customer in 2007. We have used a total of 659,673 emails consisting of both phishing emails and legitimate emails, and those emails were semi-automatically classified. 613,048 emails are legitimate and 46,525 of the emails are phishing emails, which equates to 7% of the emails being phishing emails.

For each experiment, the data was partitioned into two disjoint sets (some documents formed the training set Tr contains both type emails, while the rest formed the testing set Te). The classifier was trained using Tr and then all of the documents in Te were classified using this classifier and the accuracy is measured. We used Cross-Validation[2] for the learning process. Te and Tr are randomly-generated combinations. The size of each training set is approximately 593,616, and each testing set is approximately 65,957.

We ran C4.5 over the generated ten training and testing sets. The results of the experiments are summarized in Table I that plots the accuracy over various sets of training documents.

The experimental results show that all the classification perform reasonably well without any feature selection done by machine, especially when the feature vector space is very small. The accuracy of training is all above 99.2% and most testing are above 99.5%.

[2]Given a sample size of n sets, a classifier is generated using $(n-1)$ sets and tested on the single remaining set. This experiment is repeated k (ie. 10 in our experiment) times.

TABLE I
Accuracy rate (%) of training and testing of the 10-fold cross validation.

Fold	Training Accuracy	Testing Accuracy
1	99.2%	99.8%
2	99.2%	99.8%
3	99.2%	99.8%
4	99.2%	99.8%
5	99.2%	99.8%
6	99.3%	99.6%
7	99.3%	99.6%
8	99.3%	99.3%
9	99.3%	99.1%
10	99.6%	96.1%

The initial experiment was carried on large training set and smaller testing set. We tested the classifier by swapping the training and testing data and discovered that the accuracy rate decreased on an average of 1%. A smaller training set does not perform well for classification, therefore the large training set is necessary to deal with such complicated emails.

B. Feature Selection

Feature collection provides a set of possible instances. However, not every feature is effective as a discriminator. Therefore, we need to select a relevant subset from the initial feature set upon which to focus our attention, while ignoring the rest. Under our approach, induction. is used for the feature selection. To discover the importance of each feature, the information gain (IG) of each features is calculated as shown as in Table II.

TABLE II
Information gain of each features

Ranking	Feature	IG	Average Merit
1	subject_BL_words	0.31773238	0.318
2	body_BL_words	0.2281349	0.229
3	links	0.22665916	0.227
4	nonv_links	0.01033348	0.01
5	nonmatching_urls	0.0011986	0.001
6	scripts	0.00031751	0
7	forms	0.00000374	0

Table II provides a comprehensive ranking of each features. The larger the information gain is, the more useful a feature will be. By observation, the "subject_BL_words" is the feature with best quality, while the "forms" does the least help and possibly brings noise to the classifier. The classifier is trained using smaller vector space feature. We took one "bad" feature away each time, from "forms", "scripts", "nonmatching", and "nonv_links". We discovered that the classifier performs the best when it is built based on the first fore features in table II. Table III shows that the shortened feature set generates the exactly same accuracy as the original data. A classifier can be built by using a certain number of instances without affecting the overall performance. Our solution can train a classifier much faster without reducing effectiveness because of the very low dimensionality.

TABLE III
COMPARISON BETWEEN ORIGINAL DATA VECTOR AND SHORTENED FEATURE VECTOR

	Accuracy of original data		Accuracy of short feature data	
Fold	Training	Testing	Training	Testing
1	99.2%	99.8%	99.2%	99.8%
2	99.2%	99.8%	99.2%	99.8%
3	99.2%	99.8%	99.2%	99.8%
4	99.2%	99.8%	99.2%	99.8%
5	99.2%	99.8%	99.2%	99.8%
6	99.3%	99.6%	99.3%	99.6%
7	99.3%	99.6%	99.3%	99.6%
8	99.3%	99.3%	99.3%	99.3%
9	99.3%	99.1%	99.3%	99.1%
10	99.6%	96.1%	99.6%	96.1%

C. Other Findings

Experiment were conducted with five machine learning methods to identify which machine learning method performs the best. We have implemented using decision tree, random forest([18]), multi-layer perceptron ([19]), naive bayers ([20]) and support vector machine (SVM) ([21]). The result comes that decision tree generated the highest accuracy which builds a good classifier. Comparing to decision tree methods, the accuracies of other learning algorithms are random forest (-0.02%), multi-layer perceptron (-0.72%), naive bayes (-0.94%) and support vector machine (-1.92%). This result recommends that decision tree works well in discrete and small vector space data which is agreed by [10].

VI. CONCLUSION

In this paper, we have presented an approach to detect phishing emails using hybrid features. The contribution of the work mainly consists of the usage of hybrid features namely content, orthographic and derived, and the feature selection method.

Most current information retrieval and classification systems focus on text features. Terms are largely similar in the same type of documents, therefore, they are useful discriminators. Because phishing has become more and more complicated and sophisticated, content only classification is not sufficient against the attack. Orthographic features reflect the author's styles and habit so that the features are also informative as discriminators. Derived features are mined and discovered from emails which also provide clues for classification. Experimental results carried out in this work show that the hybrid features performs well. This is because the hybrid features are extracted from different sources and view, they support and supplement each other.

We have conducted with five machine learning method (decision tree, random forest, multi-layer perceptron, naive bayers and support vector machine) and evaluated their performances. The result showed that the decision tree works the best in this instance.

We have developed a process to classify documents and remove redundant features at the same time. We utilized the decision tree algorithm recursively over different datasets and removed redundant features using information gain. Experimental results show that a simplified classifier is generated after the redundant features are removed. Experimental evaluation on a large number of computations demonstrates that the new classifier and feature selection techniques performs as well as with the complicated features. Lower dimensionality is an advantage of any classifier because it guarantees a fast process and less noise.

The work presented in this paper uses only part of potential features. We will explore the additional features to improve the classification and detection. Feature normalization was based on the values directly derived from instances where some value is unnecessarily large. We intend to improve the normalization process by identifying a threshold to ignore very noisy data. The results also motivate future work to build a stable and automatic filter of detecting phishing emails, which needs less supervision. We intend to develop an automated mechanism to discover new features from incoming phishing emails to update our classifier when necessary.

REFERENCES

[1] http://www.gartner.com/it/page.jsp?id=565125, "Agartner."
[2] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *Proceedings of the 16th International World Wide Web Conference (WWW 2007)*, May 2007.
[3] E. Kirda and C. Kruegel, "Protecting users against phishing attacks," *The Computer Journal*, 2005.
[4] C. Kruegel, G. Vigna, and W. Robertson, July 2005.
[5] C. Ludl, S. McAllister, E. Kirda, and C. Kruegel, "On the effectiveness of techniques to detect phishing sites," in *Proceedings of Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA) 2007*.
[6] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. Mitchell, "A browser plug-in solution to the unique password problem," http://crypto.stanford.edu/PwdHash/, 2005.
[7] L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, and X. Deng, "Detection of phishing webpages based on visual similarity."
[8] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.
[9] ——, "Improved use of continuous attributes in c4.5," *Artificial Intelligence Research*, vol. 4, pp. 77–90, 1996.
[10] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.
[11] T. M. Mitchell, "Machine learning," 1997.
[12] M. Dash and H. Liu, "Feature selection for classification," 1997.
[13] D. Boneh, "Spoofguard," http://crypto.stanford.edu/SpoofGuard/, Tech. Rep.
[14] B. Ross, C. Jackson, N. Miyake, J. Mitchell, and D. Boneh, "Stronger password authentication using browser extensions," in *14th Usenix Security Symposium*, Baltimore, MD, USA, 2005.
[15] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.
[16] J. R. Quinlan, "C4.5: Programs for machine learning," 1993.
[17] T. Joachims, "Transductive inference for text classification using support vector machine," in *Proceeding of ICML-99*, 1999.
[18] L. Breiman, "Random forests," vol. 45, October 2001.
[19] N. Loukeris, "Comparative evaluation of multi layer perceptrons, to hybrid multi layer perceptrons, with multicriteria hierarchical discrimination and logistic regression in corporate financial analysis source," in *Proceedings of the 11th WSEAS International Conference on Computers*, 2007, pp. 681–688.
[20] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *AAAI98 Workshop on Learning for Text Categorization*. Madison, Wisconsin: AAAI Press, July 1998.
[21] T. Joachims, "A statistical learning model of text classification with support vector machine," in *SIGIR'01, Proceedings of International Conference on Research and Development in Information Retrieval*. New Orleans, LA, USA: ACM, September 2001, pp. 128–136.