Semantic Feature Selection for Text with Application to Phishing Email Detection*

Rakesh Verma and Nabil Hossain

Department of Computer Science, University of Houston
4800 Calhoun Road, Houston, Texas, USA
rmverma@cs.uh.edu,nabilhossain@gmail.com

Abstract. In a phishing attack, an unsuspecting victim is lured, typically via an email, to a web site designed to steal sensitive information such as bank/credit card account numbers, login information for accounts, etc. Each year Internet users lose billions of dollars to this scourge. In this paper, we present a general semantic feature selection method for text problems based on the statistical t-test and WordNet, and we show its effectiveness on phishing email detection by designing classifiers that combine semantics and statistics in analyzing the text in the email. Our feature selection method is general and useful for other applications involving text-based analysis as well. Our email *body-text-only* classifier achieves more than 95% accuracy on detecting phishing emails with a false positive rate of 2.24%. Due to its use of semantics, our feature selection method is robust against adaptive attacks and avoids the problem of frequent retraining needed by machine learning classifiers.

Keywords: security, phishing, natural language detection, semantic classification, feature selection for text

1 Introduction

Phishing is an attack in which an unsuspecting victim is lured to a web site designed to steal sensitive information (e.g., login names, passwords and financial information). Every year money, time and productivity are lost by Internet users and businesses to this plague. Hence, phishing is a serious threat to societies and economies based on the Internet. Several communication channels are available to phishers for initiating attacks, but since email is the most popular medium, we focus on detecting phishing attacks launched through emails.

As [1] observed, detecting phishing email messages automatically is a non-trivial task since phishing emails are designed cleverly to look legitimate. Besides attachments, an email can be decomposed into three main components: a header, a text body, and links. While the header and links have been well studied by phishing detection methods previously, unsupervised natural language processing (NLP) techniques for text analysis of phishing emails have been tried by

* Research supported in part by NSF grants DUE 1241772 and CNS 1319212

only a few researchers. In [2], rudimentary analysis of the anchor text was used to enhance detection. In [3], hand-crafted patterns and some scoring functions for verbs in the email were designed using trial and error.

In contrast to the unsupervised techniques mentioned above, several machine learning approaches have been tried previously. The latest attempt to use machine learning techniques is [4], which uses probabilistic latent dirichlet allocation, ensemble methods such as Adaboost and cotraining. The latter two methods differ in the use of *unsupervised* NLP technique in [3] and the use of machine learning requiring labeled data in [4], which is somewhat alleviated by the cotraining idea. Besides the need for labeled training data, supervised machine learning methods have two additional disadvantages: the problem of overtraining and the need for retraining due to model mismatch with the new data over even short periods of time. For instance, some researchers retrain their logistic regression scheme for identifying phishing web-sites *every day*.

1.1 Feature Selection

A well-known problem in text classification is the extremely high dimensional feature space, which sometimes makes learning algorithms intractable [5]. A popular method, called feature selection, to deal with this intractability reduces the dimensionality of the feature space. Many feature selection methods have been studied earlier including document frequency, information gain, mutual information, chi-square test, Bi-Normal Separation, and weighted log-likelihood ratio [6, 7, 8]. The problem with comparing these methods is that they are typically based on different assumptions or measurements of the data sets. For example, mutual information and information gain are based on information theory, while chi-square is based on statistical independence assumption. Empirical studies that compare these methods are heavily affected by the datasets.

Moreover, as [5] states, in real applications, choosing an appropriate feature selection method remains hard for a new task because too many methods exist. In an early survey paper [9], eight methods are mentioned for dealing with different text classification tasks, but none is shown to be robust across different classification applications. Therefore, [5] proposed a framework for theoretical comparison of six popular feature selection methods for text classification.

In contrast to [5], we show a semantic feature selection method based on the statistical t-test that is simple, robust, and effective. The t-test is a well-known statistical hypothesis test. Using it, we determine whether a feature's variance between two corpora is of a *statistically significant* degree. We use a two-tailed, two samples of unequal variances t-test since usually the corpora are not even of the same size let alone same variance. Instead of a purely syntactic feature selection method, we use semantics based on WordNet. We apply our semantic feature selection technique to phishing email detection and show that our classifiers significantly outperform the best previous comparable method. As explained below, due to our use of semantics, our classifiers are robust against adaptive attacks, and they avoid the need for frequent retraining.

1.2 Our Contributions and Results

Our primary contributions include: a careful study of the t-test method for feature extraction, a comparison with the previous method of [3] on the *same*, public phishing email database, and more extensive testing of our approach on *public* good databases of 3,000 Enron inbox emails[1] and 4,000 Enron sent emails as opposed to the *private* database of 1,000 good emails used in [3]. Note that we keep the good databases separate in order to use the Enron sent emails only for testing purposes to evaluate how our classifier adapts to a different domain. This is the first detailed comparison of human-intuition based and statistical NLP based methods for phishing detection. We can also develop a comprehensive and effective NLP based phishing detection method by combining the link and header analysis as suggested in [3] and in [10].

Specifically, our statistical email classifier, achieves detection rates above 95% for phishing emails with an accuracy higher than 97% on the non-phishing emails. This is an increase in effectiveness of over 20% versus the best previous work [3] on phishing emails on the same dataset, and of over 10% on two larger non-phishing public email datasets.

Besides beating the text-based classification method of [3], our method, when combined with header and link analysis, gives comparable performance to the *best* machine learning methods in the literature, such as [4], *without the problem of retraining frequently*, since our methods are not purely statistical, but they use semantics as well. In addition, our semantic feature selection method has the advantage of robustness against adaptive attacks.

The adaptive phisher may try to defeat our method by varying the email's syntax, e.g. by using different words that have similar meanings to those used in previous attacks, or by varying sentence structures, while keeping the same sense of urgency in driving the recipient to precipitate action in clicking a link. However, since we combine semantics with statistics, this will be a very difficult exercise to carry out since it would require deep examination of WordNet and a thesaurus that is richer than WordNet to come up with such an email. The effort involved would rise significantly in the process and the payoff may be little since the resulting email will sound artificial, pedantic and stilted to the reader. The overall effect will be to reduce the return on the phisher's investment drastically since phishing websites typically last only a few days before they are shut down.

The rest of this paper is organized as follows: the next section introduces necessary natural language preliminaries. Section 3 outlines our hypotheses, goals, and gives a preview of our classifiers. Section 4 presents four different classifiers with varying use of semantics to give an idea of the performance gain due to semantics. The subsequent section presents our results and analysis. Section 6 presents relevant related work on phishing detection, and Section 7 concludes.

2 Natural Language Preliminaries

Some of our classifiers apply the following NLP techniques on the email text:

[1] `http://www.cs.cmu.edu/~enron/`

i) **lexical analysis:** to break the email text into sentences and to further split these sentences into words
ii) **part-of-speech tagging:** to tag each word with its part-of-speech using the Stanford POS tagger
iii) **named entity recognition:** to identify the named entities, which include proper nouns such as names of organizations, people, or locations
iv) normalization of words to lower case
v) **stemming:** to convert each word to its stem using the Porter Stemmer [11] (e.g. reducing the verb "watching" to "watch")
vi) **stopword removal:** to remove frequently occurring words such as 'a','an', etc. using a list of stopwords

As opposed to purely syntactic or statistical techniques based on feature counting, some of our classifiers also make use of semantic NLP techniques by incorporating the WordNet lexical database and word-sense disambiguation. The latter is used to derive the appropriate *sense* or meaning of a word based on the context in which the word occurs. For example, the word "bank" might exhibit the *sense* of a financial institution in one context and a shore in another.

2.1 WordNet

WordNet is a lexical database, which exhibits properties of both a dictionary and a thesaurus [12]. In WordNet, all the words that exhibit the same concept are grouped into a *synset*, in other words a set of synonyms. The synsets can be considered as WordNet's building blocks, forming the fundamental semantic relation in the lexical database through synonymy. The *hyponymy* relation between synsets is the semantic relation that inter-connects and organizes all the nouns into a hierarchy, building a graph of nouns. The hypernymy and hyponymy relations are viewed as the relations of subordination, in other words subsumption or class inclusion, defined as follows: A is a *hypernym* of B if the meaning of A encompasses the meaning of B, which is called the *hyponym* of A. For instance, "red" is a hyponym of "color" since red is a type of color. Here, "color" is the *hypernym* of "red", since "color" broadly captures or generalizes the meaning of "red". Nouns usually have a single hypernym [12], and the WordNet noun hierarchy graph is acyclic, having a tree-like structure. All WordNet nouns are encompassed by the word *entity*, which is the root of this tree. The more we proceed down this tree, the more specific the nouns become. For instance, the hyponym set of *entity* is {*physical entity*, *thing*, *abstract entity*}, and the hyponym set of *physical entity* include *object, substance, etc.* However, the hypernymy structure for verbs is not acyclic [13]. Although the hypernym relation for verbs is captured in a similar hierarchical structure, this structure is "forest-like". Note that it is not really a forest as it contains cycles.

As mentioned earlier, a word can exhibit different meanings or senses in different contexts. Because each synset is designed to capture a unique concept, the proper sense of a word must be used to obtain its appropriate synset from WordNet. Hence, we perform word sense disambiguation using SenseLearner [14] prior to invoking any WordNet inquiry.

3 Our Hypotheses, Goals and Preview of Classifiers

As mentioned in [3], NLP by computers is well-recognized to be a very challenging task because of the inherent ambiguity and rich structure of natural languages. This could explain why only a few researchers have used NLP techniques for phishing email detection. In this paper, we investigate two basic questions:

i) Can statistical techniques applied to the text of an email differentiate phishing emails from benign emails?
ii) Do NLP techniques such as part-of-speech (POS) tagging and the use of semantics help improve the statistical methods, and if so, by how much?

We explore methods for feature extraction based on statistical tests performed on the email's text with and without the use of semantics, and our results demonstrate that statistical methods based on semantics can achieve a somewhat surprisingly high degree of accuracy in detecting phishing emails. We show that NLP techniques such as part-of-speech tagging and the use of semantics through WordNet enhance the performance of the classifier, but there is not much room for improvement left after applying the statistical methods alone for these techniques to make a huge difference in performance. However, these methods are still important since they give our classifier a robustness against attacks, for instance, attacks by the active phisher mentioned earlier.

Our methods for statistical analysis focus on the key differences between a phishing and a legitimate email. First, a phishing email is designed to motivate the reader to take some action. The action typically requires the reader to visit a malicious site created with the goal of stealing personal sensitive information. Second, since phishing web sites are on the Internet for a week or two typically before they are discovered and either blacklisted or removed, the phisher must convey a sense of urgency or give a short deadline to the target in taking the action. The authors of [3] tried to take advantage of this combination of action and urgency to detect phishing emails. However, they used an intuition-based NLP component, which could not reach detection rates better than 70% with accuracy of no more than 80% on non-phishing emails. Our statistical methods significantly outperform this previous work, and our tests are conducted on two larger, public databases of non-phishing emails from Enron.

4 Phishing Classifiers

In this section, we discuss our dataset and describe four classifiers for phishing email detection, with particular emphasis on the feature selection criteria.

Dataset Our dataset comprised of 4,550 public phishing emails from [15] and 10,000 legitimate emails from the public Enron inbox email database. We randomly selected 70% of both the phishing and the legitimate emails for statistical analysis, hereafter called the analysis sets, and the remaining 30% for testing purposes. We also used a set of 4,000 non-phishing emails obtained from the "sent mails" section of the Enron email database as a different domain to test our classifiers. We now describe the four classifiers we designed.

4.1 Classifier 1: Pattern Matching (PM) only

This is the most basic classifier, which relies only on simple pattern matching. Here we design two subclassifiers: *Action-detector* and *Nonsensical-detector*.
Action-detector We analyzed random emails from our analysis sets and observed that phishing emails had a tendency to focus on the recipient. One observation was the frequent use of the possessive adjective "your" in phishing emails. In the analysis sets, 84.7% of the phishing emails had the word "your", as opposed to 34.7% of the legitimate emails. This trend occurred because in order to raise concern, phishers often talk about the breach in security of properties in the user's possession, such as a bank account owned by the user.

Next we performed a statistical analysis of the unigrams next to all the occurrences of "your". Our goal here was to detect those properties belonging to the recipient that the phisher often declares as compromised, e.g. "amazon" (indicating the amazon.com account of an online shopper). However, many of the unigrams were adjectives describing the property, defeating our purpose. Hence we chose to analyze bigrams following "your" instead. Bigrams allowed us to detect patterns such as "your credit card", where we are more interested in the word 'card', which indicates a secure property owned by the user.

Feature selection and justification: We constructed frequency data for all bigrams following "your" for both phishing and legitimate databases. Based on a 2-tailed t-test and an α value of 0.01 (the probability of a Type I error), we chose a bigram as a possible feature if the t-value for the bigram exceeded the critical value based on α and the degrees of freedom of the word. Then we calculated weights for each bigram b, denoted $w(b)$, using the formula:

$$\mathbf{w}(\mathrm{b}) = \frac{p_b - l_b}{p_b}, \text{ where}$$

- p_b = percentage of phishing emails that contain b
- l_b = percentage of legitimate emails that contain b

Features that had weights less than 0 were discarded as these features were significant for legitimate emails. Observe that the remaining features have weights in the interval [0,1], where features with higher weights allow better detection rate per phishing email encountered. Note that the denominator in the **weight formula** prioritizes a feature that is present in 20% phishing and 1% legitimate emails over a feature that is present in 80% phishing and 61% legitimate emails. Next, we computed a frequency distribution of the selected bigrams using their weights and then selected those bigrams that had weights greater than $m - s$, where m is the mean bigram weight, and s is the standard deviation of the distribution. We call this set *PROPERTY* as it lists the possible set of user's properties, which the phisher tends to declare as compromised. Note that from now on, the term *t-selection* will refer to the same statistical feature selection used to filter features for *PROPERTY*.

So far, we have designed a feature selection method for detecting the property which the phisher falsely claims to have been compromised. The next task is to detect the pattern that calls for an action to restore security of this property.

For this purpose, we checked the email for the presence of words that indicated the user to click on the links. First, we computed statistics of all the words in sentences having a hyperlink or any word from the set {"url","link", "website"}. Here we performed the same *t-selection*, as mentioned above, to choose the features. We call the resulting set of words *ACTION*, which represents the intent of the phisher to elicit an action from the user.

At this point, we are set to design the *Action-detector* subclassifier: for each email encountered, we mark the email as phishing if it has:

i) the word "your" followed by a bigram belonging to *PROPERTY* (e.g. "your paypal account"), and
ii) a word from *ACTION* in a sentence containing a hyperlink or any word from {"url","link", "website"} (e.g. "click the link"),

Nonsensical-detector If *Action-detector* fails to mark any email as phishing, control passes to the *Nonsensical-detector*. After analyzing some phishing emails incorrectly classified by *Action-detector*, we discovered that many of these phishing emails involved dumping words and links into the text, making the text totally irrelevant to the email's subject. This observation motivated the *Nonsensical-detector* subclassifier whose purpose is to detect emails where:

i) the body text is not *similar* to the subject, and
ii) the email has at least one link.

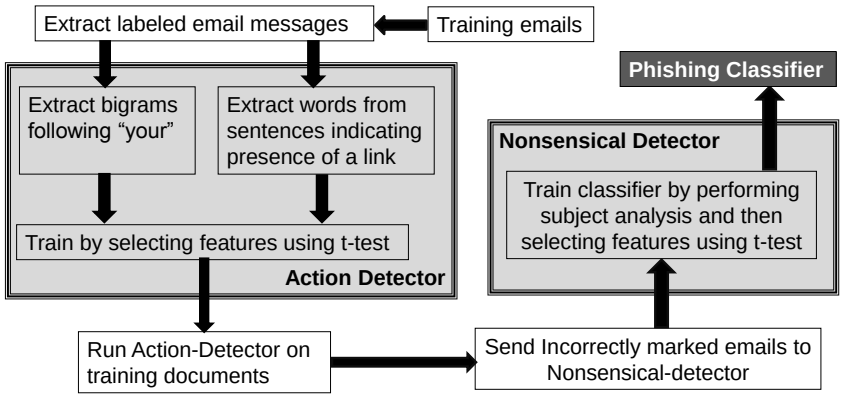
Definition 1. *An email body text is* **similar** *to its subject if all of the words in the subject (excluding stopwords) are present in the email's text.*

First, we removed stopwords from the subject and selected features from the subject using t-test on the remaining words. The goal here is to filter words that imply an awareness, action or urgency, which are common in subjects of phishing emails. We call this set *PH-SUB*. The *Nonsensical-detector* subclassifier is then designed as follows: for each email encountered, if the email subject has at least: a named-entity, or a word from *PH-SUB*, then we mark the email as phishing if:

i) it contains at least one link, and
ii) its text is **not similar** to the subject,

This detector requires a named-entity in the subject since the body of the email is completely tangential and irrelevant. Thus the phisher is relying on the subject of the email to scare the user into taking action with respect to some property of the user, which implies the presence of a named entity in the subject. In emails of this nature with irrelevant information in the email's body, we assume the named-entity in the subject to be the property of the user under threat (e.g. "KeyBank", when the subject reads: "KeyBank security"). A flowchart for building the classifier is shown in Fig. 1.

Fig. 1: Flowchart showing how the classifier is trained.



4.2 Classifier 2: PM + POS Tagging

This classifier builds on Classifier 1. Here we make use of part-of-speech tags in an attempt to reduce the error in classification that occurs when simple pattern matching techniques are used. When the bigrams following the word "your" are extracted, we perform the additional check to discard bigrams that do not contain a noun or a named-entity since the user's property, that the phisher tends to focus on, has to be a noun. When we perform statistical analysis on the words in sentences having a link, we discard words that are not verbs. Recall that the word we are looking for indicates the user to click on the link, and this word has to be a verb as it represents the action from the user's part. For the *Nonsensical-detector*, we impose the restriction of detecting named-entities, nouns, verbs, adverbs and adjectives only when selecting features for *PH-SUB*. Furthermore, for the similarity check, we select only named-entities and nouns from the subject and look for the absence of corresponding nouns and named entities in the email's text. We expect that the use of appropriate POS tags in Classifier 2 will improve accuracy over Classifier 1. For instance, among the patterns "press the link below" and "here is the website of the printing press", we are only interested in the presence of the word "press" in the former, but Classifier 1 sees both the occurrences of "press" as belonging to *ACTION*.

4.3 Classifier 3: PM + POS + Word Senses

We extend Classifier 2 by extracting the senses of words using SenseLearner [14] and taking advantage of these senses to improve classification. The goal is to reduce errors that result from ambiguity in the meaning of polysemous keywords. For instance, when "your account" appears, we are only interested in financial accounts and not in someone's account of an event. Toward this end, we performed statistical analysis on words with their POS tags and senses to train

the classifier. Then we designed this classifier to look for patterns that match selected features up to their senses whenever the classifier analyzed an email.

4.4 Classifier 4: PM + POS + Word Senses + WordNet

So far our analysis has selected a certain set of features biased to the analysis dataset. This is very similar to the way training works in machine learning based classifiers. A better way to extend the features and improve the robustness and generalization capability of our feature selection method is to find words closely associated with them so that similar patterns can be obtained. To this end, we incorporate WordNet in this classifier.

In Classifier 4, we extend the sets *PROPERTY, ACTION* and *PH-SUB* into *ext-PROPERTY, ext-ACTION* and *ext-PH-SUB* respectively by computing first the synonyms and then direct hyponyms of all synonyms of each selected feature (with its POS tag and sense), expanding the corresponding sets. Note that *PROPERTY* contains bigrams and we only extract the nouns and add their synonyms and the direct hyponyms of all the synonyms to the set. In addition, we modify the classifier as follows:

i) When we look for properties, we check to see whether the bigram that follows the word "your" includes a noun that belongs to *ext-PROPERTY*, instead of looking for the occurrence of the whole bigram in *ext-PROPERTY*.
ii) In order to detect actions, we check each sentence, that indicates the presence of a link, for the occurrence of a verb from *ext-ACTION*.
iii) When we check for *similarity*, for each noun in the email's subject, we look in the email's text for the presence of a hyponym or a synonym of the noun.

5 Analysis and Results

In this section, we present the results obtained using each of the classifiers. We also compare our classifiers to existing phishing email filters, and we present insights into the nature of our datasets.

As shown in Table 1, the results demonstrate that Classifier 4 performs the best among all the classifiers in marking both phishing and legitimate emails accurately. We used the same phishing corpus as PhishNet-NLP [3], and we tested our classifiers on 1365 phishing emails. Classifier 4 has 95.02% phishing email detection as opposed to 77.1% by the text analysis classifier of PhishNet-NLP. Of the 3000 legitimate emails tested, Classifier 4 marked 97.76% of the emails as legitimate compared to 85.1% for the text analysis classifier of PhishNet-NLP. We tested on the public Enron email database whereas the legitimate email database of PhishNet-NLP was not revealed. Furthermore, on the database of 4,000 non-phishing emails from Enron's sent mails section used only for testing, Classifier 4 obtains an accuracy of 97.58%, exhibiting potential for adaptation to a new domain. Its performance also justifies the use of semantics in classification in addition to the robustness as explained above.

Table 1: Results of using the classifiers on the test set.
P = % phishing detected on 1365 phishing emails
I = % false positives on 3000 non-phishing Enron Inbox emails
S = % false positives on 4000 non-phishing Enron Sent emails

Classifier	*P*	*I*	*S*
Classifier 1	**92.88**	**4.96**	**4.17**
Action-Detector	73.6	1.92	1.96
Nonsensical-Detector	12.84	2.87	2.21
Other	6.44	0.17	0
Classifier 2	**92.01**	**4.88**	**3.9**
Action-Detector	72.23	1.4	1.76
Nonsensical-Detector	13.34	3.31	2.14
Other	6.44	0.17	0
Classifier 3	**94.8**	**2.16**	**2.37**
Action-Detector	75.1	0.5	0.72
Nonsensical-Detector	13.3	1.49	1.65
Other	6.44	0.17	0
Classifier 4	**95.02**	**2.24**	**2.42**
Action-Detector	75.82	0.57	0.77
Nonsensical-Detector	12.74	1.5	1.65
Other	6.44	0.17	0

PILFER [16], based on machine learning, correctly marked 92% of the 860 phishing emails and 99.9% of the 6950 non-phishing emails it was tested on. Using probabilistic Latent Dirichlet Allocation, ensemble machine learning methods and cotraining, [4] claimed an F-score of 1. All these methods use features from the *entire* email, i.e., the header, the body text and the links in the email whereas our classifiers relied on the text in the body of the email only.

5.1 Performance Analysis

Dataset Characteristics After filtering the phishing emails from the analysis set using *Action-detector* and *Nonsensical-detector*, we analyzed the emails that were not detected by both of these subclassifiers. We sorted most of these emails into the following categories and took measures to correctly label them:

- **Spam Flagged:** Detected by checking if the Spam-Flag field of the email header reads YES.
- **Emails in Foreign Language:** In these emails, the text cannot be analyzed in English. So we looked for frequent occurrences of the foreign language translations of any of the words from {of, the, for, a, an} that are not present in the English vocabulary. If this check is successful, then successful detection involves finding a link.
- **Emails containing only links:** These emails do not have sufficient text for processing. We chose not to create a subclassifier that checks for validity

of links as it defeats our purpose of creating an independent NLP classifier. We marked these emails as phishing, which gave rise to some false positives.
– **Emails with no subject, no text, no link and no attachment:** Here, we checked whether both the subject line and the text were missing.

Evaluation Table 2 shows frequency of emails triggering each subclassifier of Classifier 4 on both the legitimate and phishing email test sets.

Table 2: Detailed analysis of the performance of Classifier 4 on the test sets.
P = # phishing emails detected in 1365 phishing email dataset
L = # non-phishing emails misclassified in 3000 Enron inbox email database.

Subclassifier	*P*	*L*
Action-detector	**1035**	**17**
Nonsensical-detector	**174**	**45**
Other	**88**	**5**
Spam-flagged	42	0
Foreign emails	20	1
Emails with only links	18	4
No subject and no text	8	0
Total	**1297**	**67**

6 Related Research on Phishing

Phishing has attracted significant research interest as a social engineering threat. Solutions to this problem have included angles such as: education or training, server-side and browser-side techniques, evaluation of anti-phishing tools, detection methods, and studies that focus on the reasons for the success of phishing attacks. Since [3, 4] represent the best unsupervised and supervised techniques, we refer to them for related work on these approaches for phishing email detection. Some existing works making use of these approaches include [17, 18, 16, 19, 20, 21, 22, 23, 24, 2]. Since discussing the vast literature on phishing is not feasible, we focus on prior research directly related to our work.

We can identify two ways of classifying phishing detection methods. The first classification considers the information used for detection. Here, there are two kinds of methods: those that rely on analyzing the content of the target web pages (targets of the links in the email) and methods that are based on the content of the emails. The second classification is based on the domain of the technique employed for detecting phishing attacks (emails and web pages). Here, there are detection methods based on: information retrieval, machine learning, and string/pattern/visual matching.

Applications of machine learning known as content-based filtering methods, on a feature set, are designed to highlight user-targeted deception in electronic communication [16, 19, 20, 21, 22, 23]. These methods, deployed in many phishing email detection research, involve training a classifier on a set of features extracted from the email structure and content within the training data. When training is completed, the resulting classifier is then applied to the email stream to filter phishing emails. The key differences in content-based filtering strategies are the number of features selected for training, and the types of these features.

Incorporating NLP and machine learning, [4] uses a 3-layered approach to phishing email detection. A topic model is built using Probabilistic Latent Semantic Analysis in the first layer, then Adaboost and Co-training are used to develop a robust classifier, which achieves an F-score of 1 on the test set, raising the possibility of overfitting the data. Machine learning phishing detection methods have to be updated regularly to adapt to new directions taken by phishers, making the maintenance process expensive. See [24] for a comparison of machine learning methods for detecting phishing. A non-machine learning based classifier is PhishCatch [2], which uses heuristics to analyze emails through simple header and link analyses, and a rudimentary text analysis that looks for the presence of some text filters. In [3], the authors create three independent classifiers: using NLP and WordNet to detect user actions upon receiving emails, building on the header analysis of [2], and a link analysis classifier that checks whether links in the email are fraudulent. The evolution of phishing emails is analyzed by [1], the authors classify phishing email messages into two categories: *flash* and *non-flash* attacks, and phishing features into *transitory* and *pervasive*.

In [25], a stateless rule-based phishing filter called Phishwish is proposed, which uses a small set of rules and does not need training. Although Phishwish obtains a high detection accuracy with low false positives, it is tested only on a small data set of 117 emails (81 phishing and 36 valid). For more details on phishing, please see the books by [26, 27] and [28]. Turner and Housley [29] present a detailed treatment of email operational details and security.

7 Conclusions

We presented a robust and effective semantic feature selection method for text data that is based on the t-test and generally applicable to text classification. This method was applied to automatic classification of phishing emails. We created four classifiers of increasing sophistication starting with simple pattern-matching classifiers and then designing more sophisticated ones by combining statistical methods with part-of-speech tagging, word sense, and the WordNet lexical database. Our classifiers perform significantly better than the best previous body text-based phishing classifier. When combined with header and link information it is comparable in performance with the best and most sophisticated machine learning methods that also use all the information in the email. This demonstrates the efficacy and robustness of our feature selection method.

Bibliography

[1] Irani, D., Webb, S., Giffin, J., Pu, C.: Evolutionary study of phishing. In: 3rd Anti-Phishing Working Group eCrime Researchers Summit. (2008)

[2] Yu, W., Nargundkar, S., Tiruthani, N.: Phishcatch - a phishing detection tool. In: 33rd IEEE Int'l Computer Software and Applications Conf. (2009) 451–456

[3] Verma, R., Shashidhar, N., Hossain, N.: Detecting phishing emails the natural language way. In: ESORICS. (2012) 824–841

[4] Ramanathan, V., Wechsler, H.: Phishgillnet - phishing detection using probabilistic latent semantic analysis. EURASIP J. Information Security **2012** (2012) 1

[5] Li, S., Xia, R., Zong, C., Huang, C.R.: A framework of feature selection methods for text categorization. In: ACL/AFNLP. (2009) 692–700

[6] Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: ICML. (1997) 412–420

[7] Nigam, K., McCallum, A., Thrun, S., Mitchell, T.M.: Text classification from labeled and unlabeled documents using em. Machine Learning **39**(2/3) (2000) 103–134

[8] Forman, G.: An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research **3** (2003) 1289–1305

[9] Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. **34**(1) (2002) 1–47

[10] Herzberg, A.: Combining authentication, reputation and classification to make phishing unprofitable. In: Proceedings of The IFIP 24th International Information Security Conference, IFIP SEC 2009, Springer (May 2009) 13–24

[11] Porter, M.: An algorithm for suffix stripping. Program **14**(3) (1980) 130–137

[12] Fellbaum, C., ed.: WordNet An Electronic Lexical Database. MIT Press (1998)

[13] Richens, T.: Anomalies in the wordnet verb hierarchy. In: COLING. (2008) 729–736

[14] Mihalcea, R., Csomai, A.: Senselearner: Word sense disambiguation for all words in unrestricted text. In: ACL. (2005)

[15] Nazario, J.: The online phishing corpus. `http://monkey.org/~jose/wiki/doku.php` (2004)

[16] Fette, I., Sadeh, N., Tomasic, A.: Learning to detect phishing emails. In: Proc. 16th int'l conf. on World Wide Web, ACM (2007) 649–656

[17] Ludl, C., McAllister, S., Kirda, E., Kruegel, C.: On the effectiveness of techniques to detect phishing sites. In: DIMVA. (2007) 20–39

[18] Sheng, S., Wardman, B., Warner, G., Cranor, L., Hong, J., Zhang, C.: An empirical analysis of phishing blacklists. In: Proc. 6th Conf. on Email and Anti-Spam. (2009)

[19] Chandrasekaran, M., Narayanan, K., Upadhyaya, S.: Phishing email detection based on structural properties. In: NYS CyberSecurity Conf. (2006)
[20] Bergholz, A., Chang, J., Paaß, G., Reichartz, F., Strobel, S.: Improved phishing detection using model-based features. In: Proc. Conf. on Email and Anti-Spam (CEAS). (2008)
[21] Basnet, R., Mukkamala, S., Sung, A.: Detection of phishing attacks: A machine learning approach. Soft Computing Applications in Industry (2008) 373–383
[22] Bergholz, A., Beer, J.D., Glahn, S., Moens, M.F., Paaß, G., Strobel, S.: New filtering approaches for phishing email. Journal of Computer Security **18**(1) (2010) 7–35
[23] Gansterer, W.N., Pölz, D.: E-mail classification for phishing defense. In: ECIR. (2009) 449–460
[24] Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S.: A comparison of machine learning techniques for phishing detection. In: Proc. anti-phishing working group's 2nd annual eCrime researchers summit, ACM (2007) 60–69
[25] Cook, D.L., Gurbani, V.K., Daniluk, M.: Phishwish: a simple and stateless phishing filter. Security and Communication Networks **2**(1) (2009) 29–43
[26] Jakobsson, M., Myers, S.: Phishing and countermeasures: understanding the increasing problem of electronic identity theft. Wiley-Interscience (2006)
[27] James, L.: Phishing exposed. Syngress Publishing (2005)
[28] Ollmann, G.: The phishing guide. Next Generation Security Software Ltd. (2004)
[29] Turner, S., Housley, R.: Implementing Email and Security Tokens: Current Standards, Tools, and Practices. John Wiley & Sons Inc (2008)