Phishing Detection Using Traffic Behavior, Spectral Clustering, and Random Forests

Dave DeBarr, Venkatesh Ramanathan, Harry Wechsler
Computer Science Department
George Mason University
Fairfax, Virginia, 22030, United States
{ddebarr, vramanat, wechsler}@gmu.edu

Abstract—**Phishing is an attempt to steal a user's identity. This is typically accomplished by sending an email message to a user, with a link directing the user to a web site used to collect personal information. Phishing detection systems typically rely on content filtering techniques, such as Latent Dirichlet Allocation (LDA), to identify phishing messages. In the case of spear phishing, however, this may be ineffective because messages from a trusted source may contain little content. In order to handle such emerging spear phishing behavior, we propose as a first step the use of Spectral Clustering to analyze messages based on traffic behavior. In particular, Spectral Clustering analyzes the links between URL substrings for web sites found in the message contents. Cluster membership is then used to construct a Random Forest classifier for phishing. Data from the Phishing Email Corpus and the Spam Assassin Email Corpus are used to evaluate this approach. Performance evaluation metrics include the Area Under the receiver operating characteristic Curve (AUC), as well as accuracy, precision, recall, and the (harmonic mean) F measure. Performance of the integrated Spectral Clustering and Random Forest approach is found to provide significant improvements in all the metrics listed, compared to a content filtering technique such as LDA coupled with text message deletion done randomly or in an adaptive fashion using adversarial learning. The Spectral Clustering approach is robust against the absence of content. In particular, we show that Spectral Clustering yields (99.8%, 97.8%) for (AUC, F measure) compared to LDA that yields (94.6%, 89.4%) and (79.6%, 57.9%) when the content of the messages is reduced to 10% of their original size using random and adversarial deletion, respectively. The difference is most striking at low False Positive (FP) rates.**

Keywords—Phishing; Spear Phishing; Spectral Clustering; Link Analysis; Latent Dirichlet Allocation

I. INTRODUCTION

Phishing is a form of identity theft. A typical phishing attempt consists of a phisher sending an email to a user. The email appears to come from a legitimate service, such as a financial institution, a social networking web site, or an electronic message service provider. The email contains a link to a web site that mimics the web site of the legitimate service provider. In fact, the graphics and layout of the web site may be identical to the web site of the legitimate service provider. The only difference may be the use of an intermediate link for grabbing form related data, such as account login information. This information is then passed to the legitimate service provider's web site making it difficult for the user to know their account information has just been compromised. The phisher exploits the victim's compromised financial accounts, or the victim's social contacts. The victim's social contacts potentially become the next target of the phisher.

With access to the victim's account, and a list of the victim's social contacts, the phisher can now engage in spear phishing. Rather than sending phishing messages as unsolicited messages to randomly selected addresses harvested from Internet sources, the phisher can now direct messages to specific individuals. In this form of spear phishing, brief messages are sent to known contacts along with a link to the phishing web site. Since the message will come from a known contact, individuals may be much more likely to provide their account information. This can be especially damaging if the initial victim is an administration account for a service provider. Because most phishing detection systems rely on content filtering techniques, such as Latent Dirichlet Allocation (LDA), many phishing detection systems will fail to identify the message as a phishing attempt.

In order to deliver their messages and collect the victim's information, phishers often use the same infrastructure repeatedly. Figure 1 illustrates this behavior. A message is sent from the phisher to a mail server, in order to relay the message to the victim. The user then visits the phisher's web server by clicking a link in the message, prompting the user for their account information. The web server then stores the account information for later retrieval by the phisher.

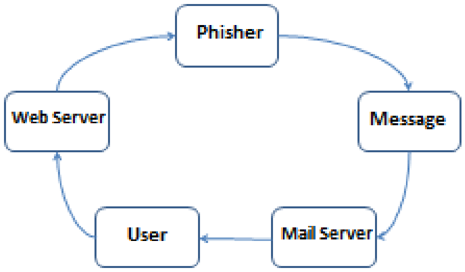


Figure 1. Phishing Infrastructure

By analyzing connections between the web server URLs, provided by the email message, one will be better able to detect spear phishing messages containing very little content. The web server connections can be used to compute a Laplacian matrix, a matrix representation of the graph consisting of messages and the URLs connecting them. Spectral clustering uses the spectrum (eigenvalues) of the Laplacian matrix to group messages together. Cluster membership, along with the spectral representation of the messages, can then be used to build a classification model for detecting phishing messages.

The remainder of this paper is organized as follows. Section II provides background information related to content filtering, focusing on earlier work related to the use of features based on Latent Dirichlet Allocation (LDA) for phishing detection. Section III presents background related to link analysis and Spectral Clustering. Section IV details our hybrid detection approach using the Random Forest classification algorithm and Spectral Clustering. Section V presents the proposed representation using traffic behavior. Section VI describes the data sets collected and experiments used to evaluate the proposed approach. Section VII shows the evaluation results; section VIII discusses the results; and section IX provides conclusions.

II. Background

Content filtering approaches to phishing detection rely on text contents, where a filter either disallows delivery of messages recognized as phishing messages or delivers suspect messages to a special folder for careful review by the intended recipient. [1] provides a comparison of machine learning techniques for phishing detection using Term Frequency – Inverse Document Frequency (TF-IDF) representation. Training and test features were a function of the product of how often a term occurs within a document (TF) and the inverse of the proportion of documents containing the term (IDF). This is known as a bag-of-words approach, because relationships between words are not directly considered in this representation. A total of 2,889 emails were used for evaluation, with 40.5% of the data set labeled as phishing messages. The phishing messages were composed of 1,171 out of 1,423 messages from the PhishingCorpus [9]. The non-phishing messages came from the researchers' personal mail boxes. Comparing neural network, logistic regression, Bayesian Additive Regression Trees (BART), Classification And Regression Tree (CART), Random Forest, and Support Vector Machine (SVM) models, the results showed that the Random Forest outperformed the other methods when misclassification errors had equal costs.

In [13], Latent Dirichlet Allocation (LDA) was explored as an improved representation for phishing message detection. LDA estimates the probability of each topic for a message based on the contents of the message, where the topic probabilities sum to one. Given Dirichlet parameters **α** and **β**, the joint probability of a topic mixture **θ**, and a set of *N* topics *z* and words *w*, is given by [2]:

$$p\left(\boldsymbol{\theta},\mathbf{z},\mathbf{w}\mid\boldsymbol{\alpha},\boldsymbol{\beta}\right)=p\left(\boldsymbol{\theta}\mid\boldsymbol{\alpha}\right)\prod\nolimits_{i=1}^{N}p\left(z_i\mid\boldsymbol{\theta}\right)p\left(w_i\mid z_i,\boldsymbol{\beta}\right)\quad(1)$$

The Topic Modeling Toolbox [15] was used to discover the distribution of terms for a topic and the distribution of topics for a document, using a collapsed Gibbs sampling approach. LDA topic distributions for phishing and non-phishing topics were used as features for classification. As shown later in this paper, performance drops when one reduces the message content, as might be the case in spear phishing, e.g., the entire message may be simply reduced to the words "Check this out" with a URL link to the phishing site.

III. Link Analysis and Spectral Clustering

A graph is a simple data structure composed of objects (aka nodes or vertices) and links (aka connections or edges). For example, a set of messages can be viewed as a graph. Each message is a node in the graph, while having similar URLs in the message body can be viewed as a link between messages. The graph formed by these connections can be compactly represented by an *n* x *n* Laplacian matrix. There are essentially 4 steps in spectral clustering [11]:

1. Compute the Laplacian matrix ***L*** to represent the set of messages
2. Derive the spectral decomposition of the Laplacian matrix: eigenvalues ***Λ*** and eigenvectors ***Q***
3. Form the new spectral representation ***S*** from the eigenvectors of ***Q***
4. Cluster the rows of the matrix ***S***

A normalized Laplacian matrix ***L*** is simply the affinity matrix ***A*** normalized by the degree matrix ***D***:

$$\mathbf{L}=\mathbf{D}^{-1/2}*\mathbf{A}*\mathbf{D}^{-1/2}\quad(2)$$

where

$$\mathbf{A}\left[i,j\right]=\begin{cases}\exp\left(-\dfrac{\text{distance}\left(i,j\right)^2}{2\sigma^2}\right), & i\neq j\\ 0, & i=j\end{cases}$$

$$\mathbf{D}\left[i,j\right]=\begin{cases}\sum\nolimits_{j=1}^{n}\mathbf{A}\left[i,j\right], & i=j\\ 0, & i\neq j\end{cases}$$

The distance() measure is described further in section V. The σ, kernel width, parameter is chosen as the expected distance between a pair of messages.

The spectral decomposition of the Laplacian matrix ***L*** provides the spectrum (eigenvalues) ***Λ*** and characteristic directions (eigenvectors) ***Q*** of the graph [6]. An *n* x *n* Laplacian matrix can be factored as follows:

$$\mathbf{L}=\mathbf{Q}*\mathbf{\Lambda}*\mathbf{Q}^{-1}\quad(3)$$

where

$$\mathbf{\Lambda}\left[i,j\right]:=\begin{cases}\text{solution of det}\left(\mathbf{L}-\mathbf{\Lambda}\left[i,j\right]*\mathbf{I}\right)=0, & i=j\\ 0, & i\neq j\end{cases}$$

$$\mathbf{Q}[i,] := \text{solution of} \left(\mathbf{L} - \mathbf{\Lambda}[i,j] * \mathbf{I}\right) * \mathbf{Q}[i,] = 0$$

The new representation matrix ***S*** is then formed as the eigenvectors associated with the *k* largest eigenvalues; e.g. all eigenvectors associated with eigenvalues greater than zero. The rows are then normalized so they have length equal to one:

$$\mathbf{S}[i,j] = \frac{\mathbf{Q}[i,j]}{\sqrt{\sum_{j=1}^{k} \mathbf{Q}[i,j]^2}} \tag{4}$$

A clustering algorithm such as Partitioning Around Medoids (PAM) [8] can then be used to identify *m* groups of similar observations. The number of groups *m* can be chosen as the value of *m* that results in the largest average "silhouette" value for an observation, where the silhouette *u* for observation *i* is defined as:

$$u_i = \frac{b(i) - a(i)}{\max\left(a(i), b(i)\right)} \tag{5}$$

where *a*(*i*) is the average distance between observation *i* and the other members of the cluster to which it is assigned, and *b*(*i*) is the average distance between observation *i* and the other (unassigned, neighbor) cluster with the smallest average distance to observation *i*. The silhouette value incorporates both the compactness *a(i)* of the assigned cluster and a measure of separation *b(i)* for the nearest cluster in assessing fit for an observation. Silhouette values range between -1 and 1, where 1 indicates an observation has been assigned to an appropriate cluster and -1 indicates a poor fit.

A classification model can then be constructed using the following features for each observation:

- cluster: the identity of the assigned cluster
- silhouette: the silhouette value for the observation
- neighbor: the identity of the nearest neighboring cluster
- spectral features: the spectral representation for the observation

In [16], spectral clustering was used to cluster spamming servers based on HoneyPot data [12]. The HoneyPot project routinely exposes email addresses on the Internet to identify the email address harvesters (observed via HTTP Get request) and the spamming email servers (observed by collecting message traffic). Spectral clustering was then used on this bipartite (harvesters, email servers) graph, to identify interesting groups of servers. For example, the authors discovered that spammers involved in phishing often use different infrastructure from spammers who are not involved in phishing, where phishing was determined by heuristic keyword classification of the associated email content (e.g. occurrence of the keyword "paypal" in the message was used to classify the message as phishing). Our work also uses spectral clustering but it differs in that the goal is email classification (discriminating phishing and non-phishing messages) rather than server classification, and web server links are analyzed instead of email address links between harvester servers and email servers.

IV. Phishing Detection Using Random Forests

The Random Forest algorithm [4] is an implementation of bootstrap aggregation (bagging) where each tree in an ensemble of decision trees is constructed from a bootstrap sample of messages from the training set. Each bootstrap sample of messages is obtained by repeated random sampling with replacement until the size of the bootstrap sample matches the size of the original training set. This helps to reduce the variance of the classifier (reducing the classifier's ability to overfit the training data) [3]. When constructing each decision tree, only a randomly selected subset of features is considered for constructing each decision node. Of the *k* randomly selected features to consider for constructing each decision node, the yes/no condition that best reduces the Gini impurity measure *g* of the data is selected for the next node in the tree:

$$g = 1 - P\left(Phish\right)^2 - P(NotPhish)^2 \tag{6}$$

The Gini impurity measure is largest when the classifier is most uncertain about whether a message is phish. Random Forest classifiers are used to construct models for both the LDA features and the Spectral Clustering features.

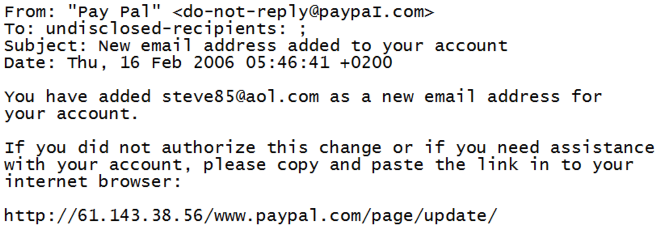
To classify new messages, each tree in the Random Forest classification model casts its vote for a class label: phishing or not phishing. The proportion of votes for the phishing class is the probability that a randomly selected tree would classify the message as a phishing message. This is interpreted as the probability of a message being a phishing message.

V. Traffic Behavior

Phishers often use web server infrastructure to capture account information from victims. The web server Uniform Resource Locator (URL) links can be found in the body of the email messages. These links are often populated by configurable software used by spammers to automate their operations. For example, the web server link is selected from a pool of available web servers/pages. The web server selection simply populates a phishing link macro for a template.

Figure 2 shows a simple example of an email message from the PhishingCorpus [10]. The web server link is found in the message body:

http://61.143.38.56/www.paypal.com/page/update/



```
From: "Pay Pal" <do-not-reply@paypaI.com>
To: undisclosed-recipients: ;
Subject: New email address added to your account
Date: Thu, 16 Feb 2006 05:46:41 +0200

You have added steve85@aol.com as a new email address for
your account.

If you did not authorize this change or if you need assistance
with your account, please copy and paste the link in to your
internet browser:

http://61.143.38.56/www.paypal.com/page/update/
```

Figure 2. Phishing Email Example

Phishers often use similar behavior to drive traffic to their sites. In many instances, phishers use similar links for many phishing messages; e.g. re-using the same web server (registered domain), or the same path generated by a commonly used phishing kit (as phishing kits may simply contain zip archives allowing for quick installation of content for a phishing web site). The web server links in a message can be broken down into "ngrams", substrings of varying lengths. For example, possible substrings of "interest" for the message in Figure 2 include "paypal" and "update". In order to discover which substrings should be used for measuring distance between messages, all substrings are derived for a "labeled" training corpus. We use all substrings where the Gini impurity measure of the phishing and non-phishing classes for the substring is less than ¼. The Jaccard distance between messages *i* and *j* is then computed as:

$$\text{distance}(i,j) = 1 - \frac{\left|substrings(i) \cap substrings(j)\right|}{\left|substrings(i) \cup substrings(j)\right|} \quad (7)$$

This is simply the complement of the proportion of URL substrings found in both messages.

VI. Experimental Design

The Phishing Corpus [10] and the 2003 Spam Assassin Archive [14] are used for our experiments. The Phishing Corpus contains 4559 phishing messages, while the 2003 Spam Assassin archive contains 3900 "easy ham" messages, and 250 "hard ham" messages. These "ham" messages are messages that are not spam nor phishing messages. Our experiments focus on using messages containing URLs and little text content.

While some of the messages in the PhishingCorpus have relatively little content, such as the message shown in Figure 2, our experiments involve simulation of spear phishing with reduced content messages by removing text from the message. Tokens are randomly selected for removal. Somewhat surprisingly, the average number of tokens per message, as delimited by spaces and punctuation, is 517 tokens for phishing messages from the Phishing Corpus and 354 tokens for "ham" messages from the Spam Assassin archive. This is probably because phishers observed in the Phishing Corpus felt the need to put in extra content to make their message appear legitimate. In a spear phishing scenario, where a "friend" (or trusted contact) has sent a link simply saying "Check this out", this will not be the case. In order to simulate reduced content messages, we report results with 5%, 10%, 30%, 50%, and 100% of the original content for the LDA approach. Furthermore, we use two different approaches to selecting content for removal. In the first approach, we use random selection of tokens for removal (RND). For the second approach, we use adversarial selection of tokens for removal (ADV). For adversarial selection, the probability of removing a token is proportional to the probability of a classifier identifying a message as a phishing message given the presence of a token within a set of "test" messages. This probability can be evaluated by an adversary by sending test messages and checking results, either via a web "beacon" (such as a web-based image) in the message for private email systems or via "test" accounts for public email systems (such as Gmail). The Spectral Clustering approach only relies on the URL links within the body of the message. URL links are not removed for either the LDA or Spectral Clustering approaches.

We provide both LDA and Spectral Clustering results for both the original message sets, and the modified message sets where a randomly selected subset of tokens is removed. A randomly selected subset of 500 messages containing URLs is selected from the combined corpus of 8709 messages, then *k*-fold ($k = 10$) cross validation is used to estimate the performance of both the LDA and the Spectral Clustering approaches.

For the LDA approach, the topic and term distributions discovered from the training corpus are applied to the test corpus. The number of topics in the training corpus is chosen as the number that minimizes the perplexity [2]. Lower perplexity scores mean a higher predicted likelihood for terms found in the corpus. A Random Forest is constructed for each training set, then evaluated on each test fold. For the Spectral Clustering approach, both the training and test sets are combined to derive the spectral representation. The spectral representation, cluster membership, and silhouette values for each training set are used to build a Random Forest classifier, which is then evaluated on each test fold. For both the LDA approach and the Spectral Clustering approach, the Random Forests had 500 trees and the number of features considered for each split node was the ceiling of the square root of the number of features used for training.

Email classification for the Spectral Clustering approach uses transduction, rather than induction, for classification. Transduction focuses on reasoning from observed training examples to classify specific (observed) test examples, rather than generalizing to unobserved test examples (induction) [5]. This is a form of Semi-Supervised Learning (SSL), as both the labeled training examples and the unlabeled test examples are analyzed together (via spectral decomposition) for classification. Batch processing of incoming messages can be performed frequently, including both messages observed earlier and newly received messages. For example, messages from the last hour can be clustered every 10 minutes to classify newly received messages. Message classification is performed as follows.

The first step in processing messages for the LDA approach is to extract tokens. Tokens are extracted by converting the text of the message subject and body to lower case; partitioning the text into tokens by breaking on spaces, tabs, and punctuation ('.', '?', '!'); and removing common stop words and tokens that occur in less than 1% of the messages. The Topic Modeling Toolkit (TMT) was then used to model the probability of each topic in the training messages, and the topic probabilities for each message were then derived for the test set.

The first step in processing messages for the Spectral Clustering approach is to extract URLs. The set of all possible substrings of length 1 through 10 is then considered for the URLs found in a message. Substrings which do not occur within at least 1% of all messages are discarded, as are substrings where the Gini impurity measure of the phishing and non-phishing message classes in the training set is greater than or equal to ¼.

VII. Performance Evaluation

Performance is evaluated based on 5 classification metrics for each class:

1. Area Under the Receiver Operating Characteristic (ROC) Curve (AUC): the probability that a randomly selected message from the phishing class will be viewed as more likely to be a phishing message than a randomly selected member of the non-phishing class
2. Accuracy: the probability that the predicted class for a randomly selected message is the actual class for that message
3. Precision: the probability that a predicted phishing message is actually a phishing message
4. Recall: the probability that an actual phishing message is predicted to be a phishing message
5. F Measure: the harmonic mean of Precision and Recall

Table 1 contains the results of our experiments.

Method	AUC	Accuracy	Precision	Recall	F Measure
SC	99.8%	97.6%	97.2%	98.6%	97.8%
LDA: 100%	99.4%	96.6%	97.1%	96.8%	96.9%
LDA: RND 50%	99.1%	95.8%	96.4%	96.0%	96.2%
LDA: RND 30%	98.8%	94.0%	94.9%	94.2%	94.6%
LDA: RND 10%	94.6%	88.2%	88.9%	89.9%	89.4%
LDA: RND 5%	90.0%	81.2%	82.1%	84.5%	83.3%
LDA: ADV 10%	79.6%	63.4%	79.7%	45.5%	57.9%
LDA: ADV 5%	71.6%	60.6%	73.5%	45.1%	55.9%

Table 1. Results for Experiments

As seen in Table 1, the Latent Dirichlet Allocation (LDA) approach is found to be just as good as the Spectral Clustering (SC) approach when using 100% of the content, but SC does significantly better than LDA when the content of the message is reduced. Performance drops most significantly when content is removed proportionally to how well the content identifies a message as a phishing message.

Figure 2 shows the Receiver Operating Characteristic (ROC) curves for the experiments. The ROC curve [10] is formed by plotting the False Positive (FP) rate on the horizontal axis and the True Positive (TP) rate on the vertical axis as the classification threshold for identifying phish is lowered from 1 to 0. The classification threshold is applied to the probability that a message is a phishing message: if the probability is above the classification threshold the message is treated as a phishing message, otherwise the message is treated as a non-phishing message. The FP rate is the proportion of not phishing messages that are predicted to be phishing messages. The TP rate is just another name for recall: the proportion of phishing messages that are predicted to be phishing messages.

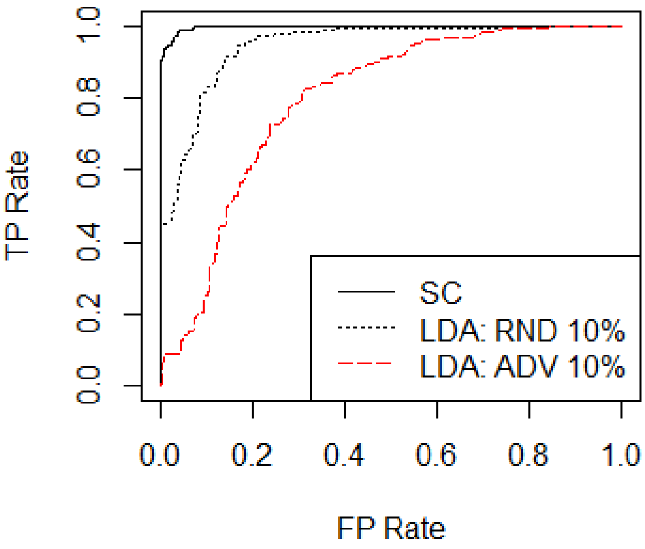


Figure 2. Receiver Operating Characteristic Curves

A perfect classifier would have a TP rate of 100%, with an FP rate of 0%. As shown in Figure 2, the Spectral Clustering (SC) approach (solid black line) appears to have the best performance. The true positive rate for a zero false positive rate drops below 50% when randomly selecting tokens for removal, and it drops below 10% when selecting tokens for removal proportional to their probability of identifying a message as a phishing message.

The optimal number of topics for the LDA approach was 45, as determined by average perplexity. An example of the top 3 terms for one of the LDA topics was “access”, “limited”, and “paypal” respectively. This makes sense as a topic that might often be found in phishing messages.

The optimal number of features for the Spectral Representation was 38, and the optimal number of clusters was 30. Examples of ngrams used for Spectral Clustering include “pay”, “ebay”, and “secur”. These also make sense as substrings that might be found in phishing URLs that want to appear legitimate.

The most important features for the Spectral Clustering model, as measured by mean Gini decrease, are shown in Figure 3. Mean Gini decrease measures the average Gini decrease per tree for each feature. Assigned cluster membership and the silhouette value (a measure of how well an observation fits with the assigned cluster) are the two most important features used by the Random Forest for classification. The V# features are simply normalized eigenvector coordinates from the spectral representation, while “neighbor” is the identity of the next closest cluster (used for computing the silhouette value). It’s interesting to note that the second eigenvector coordinate was considered to be much more important the first eigenvector coordinate, indicating that while the first coordinate captures more variance in link structure it has less utility for classification.

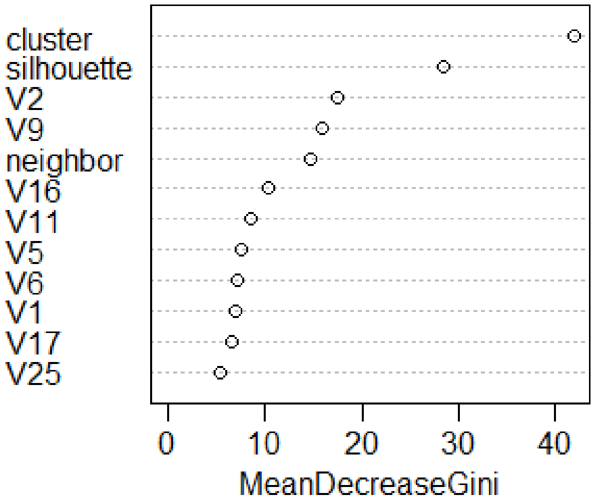


Figure 3. Variable Importance for Spectral Clustering

VIII. Discussion

Multiple extensions are possible for this work. The measurement of similarity can be enhanced to include mail server address information found in the "Received" headers. This approach can detect "colluding providers" who act as phishing message carriers, though this would need to be done with care when using different message classes from different corpora (to avoid having message headers make separating the classes a trivial exercise). It may also be useful to determine whether messages are being sent person-to-person, or whether an organization or email list is involved.

The Spectral Clustering approach can also be extended to support data streaming, where incoming messages are classified as they arrive. The challenge of spectral decomposition for large Laplacian matrices can also be addressed.

A similar Spectral Clustering approach can be evaluated for web site (rather than email) classification too, as this might be useful for a browser. The web sites may contain limited content with hyperlinks to legitimate web sites.

IX. Conclusions

Phishing is an attempt to steal a user's identity. This is typically accomplished by sending an email message to a user, with a link directing the user to a web site used to collect account information. Phishing detection systems typically rely on content filtering techniques, such as Latent Dirichlet Allocation (LDA), to identify phishing messages. In the case of spear phishing, however, this may be ineffective; because messages from a trusted source may contain little content. In order to identify this type of spear phishing behavior, we propose the use of Spectral Clustering of messages based on analysis of links between the URLs for web sites, found in the message contents. Cluster membership is then used to construct a classifier for phishing. Data from the Phishing Email Corpus and the Spam Assassin Email Corpus were used to evaluate this approach. Performance evaluation metrics included the area under the Receiver Operating Characteristic (ROC) curve, as well as accuracy, precision, recall, and the F measure. When 100% of the message content was present the Spectral Clustering approach was just as good as the LDA approach; but for experiments where the content was randomly reduced, the Spectral Clustering approach was significantly better. For the type of spear phishing studied, performance of the proposed Spectral Clustering approach is found to provide significant improvements in all metrics evaluated, including 9.4% and 68.9% improvements in the F measure when the content of the messages is reduced to 10% of their original size using random and adversarial deletion, respectively. The lift in performance is significantly larger when adversarial phishers choose which terms to remove based on whether a term is likely to increase the probability of a message being classified as spam.

References

[1] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A Comparison of Machine Learning Techniques for Phishing Detection," Proceedings of the Anti-Phishing Working Group's 2nd Annual eCrime Researchers Summit, ACM, New York, 2007, pp. 60-69.

[2] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent Dirichlet Allocation", Journal of Machine Learning Research, vol. 3, Jan 2003, pp. 993-1022.

[3] L. Breiman, "Bagging Predictors", Machine Learning, vol. 24, Aug 1996, pp. 123-140.

[4] L. Breiman, "Random Forests", Machine Learning, vol. 45, Oct 2001, pp. 5-32.

[5] O. Chapelle, B. Scholkopf, A.Zien, and V. Vapnik, "A Discussion of Semi-Supervised Learning and Transduction", Semi-Supervised Learning, MIT Press, 2006, pp. 457-462.

[6] F.R.K. Chung, Spectral Graph Theory, American Mathematical Society, Providence, Rhode Island, 1987.

[7] T. Fawcett, "An Introduction to ROC Analysis", Pattern Recognition Letters, vol. 27, Jun 2006, pp. 861-874.

[8] L. Kaufman and P.J. Rousseeuw, "Partitioning Around Medoids", Finding Groups in Data: an Introduction to Cluster Analysis, Wiley, 2005, pp. 68-125.

[9] J. Nazario, Phishing Corpus Mailbox 2, http://monkey.org/~jose /phishing/phishing2.mbox, last accessed 2013-01-03.

[10] J. Nazario, Phishing Corpus, http://monkey.org/~jose/wiki/doku.php, last accessed 2013-01-03.

[11] A.Y. Ng, M.I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm", Advances in Neural Information Processing Systems, MIT Press, 2001, pp. 849-856.

[12] Project Honey Pot, http://projecthoneypot.org, 2010.

[13] V. Ramanthan and H. Wechsler, "Phishing Detection and Impersonated Entity Discovery Using Conditional Random Field and Latent Dirichlet Allocation", Computers & Security, http://dx.doi.org/10.1016 /j.cose.2012.12.002, last accessed 2013-01-03.

[14] SpamAssassin Email Corpus, http://spamassassin.apache.org /publiccorpus/, last accessed 2013-01-03.

[15] Topic Modeling Toolbox, http://nlp.stanford.edu /software/tmt/.reference, last accessed 2013-01-03.

[16] K.S. Xu, M. Kliger, and A.O. Hero, "Identifying Spammers by their Resource Usage Patterns", Proceedings of the 7th Annual Conference on Collaboration, Electronic messaging, Anti-abuse and Spam (CEAS), Redmond, WA, 2010.