

A Learning Approach to Spam Detection based on Social Networks

Ho-Yu Lam, Dit-Yan Yeung
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{ivanlam,dyyeung}@cse.ust.hk

ABSTRACT

The massive increase of spam is posing a very serious threat to email which has become an important means of communication. Not only does it annoy users, but it also consumes much of the bandwidth of the Internet. Most spam filters in existence are based on the content of email one way or the other. While these anti-spam tools have proven very useful, they do not prevent the bandwidth from being wasted and spammers are learning to bypass them via clever manipulation of the spam content. A very different approach to spam detection is based on the behavior of email senders. In this paper, we propose a learning approach to spam sender detection based on features extracted from social networks constructed from email exchange logs. Legitimacy scores are assigned to senders based on their likelihood of being a legitimate sender. Moreover, we also explore various spam filtering and resisting possibilities.

1. INTRODUCTION

Unsolicited commercial email (UCE), a.k.a. spam, is not a new problem causing complaints from many Internet users. Spamming, i.e., the act of sending UCE, involves the sending of nearly identical emails to thousands or even millions of recipients without the recipients' prior consent or even violates recipients' explicit refusal [9, 30, 34]. Unsolicited bulk email (UBE) is another category of emails that can be considered spam. As suggested in recent reports by Spamhaus [4] and Symantec [31], spam is increasingly being used to distribute virus, spyware, links to phishing web sites, etc. The problem of spam is not only an annoyance, but is also becoming a security threat.

There is an increasing trend for both UCE and UBE. For instance, Symantec has detected a 44% increase in phishing attempts from the first half of 2005 to the second half. Statistics from the Distributed Checksum Clearinghouse (DCC) project [24] shows that 54% of the email messages checked by the DCC network in 2005 are likely to be from bulk email. Also, statistics from MX Logic [22] shows that on average 80.78% of the email messages delivered to their clients during the week of March 24–30, 2007 are considered spam, with peaks at more than 90%.

Various legal means of anti-spam attempts have been discussed in [16, 23]. Legislations specifically targeted at email spam as well as unwanted messages in general have been introduced in some countries, such as the United States of America. Before targeted legislations are introduced, some existing laws are sought for fighting spam. Possible approaches are based on laws and statutes that combat fraud, antiracketeering, trespassing and antiharassment. These

approaches are considered ineffective as they require considerable costs and efforts for the prosecutor to prove the relevance between the spam messages and the law. Another challenging problem to the legal approach is the limited jurisdiction of the law concerned. Also, many legislators are forced to leave loopholes in the legislations to avoid infringing the freedom of speech [16]. These often allow spammers to slip through and the restriction merely becomes a burden to legitimate senders.

To prevent users from being overwhelmed by spam, many Internet service providers (ISP) and organizations deploy spam filters at the email server level. The family of Naive Bayes (NB) classifiers [13, 25] is probably one of the most commonly implemented, which is also embedded in many popular email clients. They extract keywords and other indicators from email messages and determine whether the messages are spam using some statistical or heuristic scheme. However, spam senders (spammers) nowadays are using increasingly sophisticated techniques to trick content-based filters by clever manipulation of the spam content [14]. For example, random character strings are inserted to counter frequency analysis. Also, words with scrambled character order can render vocabulary-based detection techniques ineffective, yet humans can still understand the scrambled words. As a consequence, content-based filters are becoming less effective and hence other approaches are being explored to complement them.

One popular approach is based on blacklists and whitelists. A blacklist is a list of senders whose emails are blocked from getting through to the recipients. A whitelist is just the exact opposite. While a blacklist specifies who is to be kept out allowing all others to pass, a whitelist only allows those who are already on the list to get through. Since spammers almost always spoof the "From:" field of spam messages, blacklists usually keep IP addresses rather than email addresses. For incoming messages from senders not on the lists, content-based filters may be applied so that the two approaches can complement each other.

In this paper, we propose a machine learning approach to spam detection that may be regarded as partially automating the construction and maintenance of blacklists and whitelists. Specifically, the proposed framework extracts seven features from email social networks for each sender. Based on these features, a supervised learning model is used to learn the behaviors of spammers and legitimate senders given a set of training data. The model is then used to assign a legitimacy score to each sender. The score assignment works by analyzing batches of logs and thus is an off-line process. Scores are made ready in a database where online mitigation methods can query for the score of a particular sender.

The rest of this paper is organized as follows. Section 2 reviews related work on sender-based spam detection methods and social networks. Section 3 describes the spam detection problem

addressed by this paper. Section 4 details the proposed sender scoring scheme and section 5 explores possible deployment schemes for spam mitigation. Section 6 presents experimental results and finally section 7 concludes this paper.

2. RELATED WORK

A sender-based spam detection method tries to determine whether a sender is a spammer or a legitimate sender. Our proposed scheme belongs to this category.

One popular approach, and possibly the earliest one employed, is based on blacklists and whitelists. While blacklists and whitelists are effective for filtering spam without affecting legitimate email, the main problem of this approach is the effort required for constructing the lists and keeping them up to date. Some automatic whitelisting methods can be applied based on heuristics, such as whitelisting addresses that a user replies to. SpamAssassin introduced auto-whitelisting [29] since 2001. The algorithm is part of the SpamAssassin rule-based anti-spam system, in which many score generating rules are combined to give a spam score to an email. The auto-whitelisting algorithm takes into account the average score of the emails corresponding to the sender and pushes the score for a new email to be the new average score. As a result, a sender that mostly sends legitimate emails will be ensured to get a low score. This effectively whitelists the sender. The reverse is also true for a persistent spammer whom will be auto-blacklisted.

Another popular approach is the sender reputation system. Golbeck and Hendler [11] presented a reputation network scheme based on user feedback in a social network. Users assign their reputation ratings to other users on a social networking platform. The users of the reputation network are connected to each person that they have rated. A recursive algorithm was proposed to infer the reputation scores for email senders. A user, namely the source, can request for the reputation of another user, namely the sink. If the source has rated the sink then the reputation is simply the rating. Otherwise, the source requests all of his/her neighbors to recursively request for a reputation of the sink. At each recursion, the local ratings of each user in the path of the query are taken into account.

Chirita et al. [6] proposed a global reputation scheme called Mail-Rank. Email communication data are collected to construct a global email network consisting of email users. A user U_1 sending an email to another user U_2 is considered a trust vote from U_1 to U_2 . A power iteration algorithm can then be used to calculate the scores for all email addresses in the email network. A set of trusted users is predetermined by ways of email address books and autowhitelists from the users. These trusted users bootstrap the reputation system as high reputation users.

Taylor [32] discussed the domain reputation system deployed in Google’s Gmail system [1]. The reputation system maintains the reputation for each domain that sends email to Gmail. The reputations are calculated based on previous results from statistical filters and user feedback. If the reputation of a domain is good, the domain will be whitelisted and the reverse will be blacklisted. The emails from senders in neither lists are further processed with statistical anti-spam filters for making the final decision. Email classification results are logged as auto spam or auto nonspam events. Users can send feedback to the system by clicking on a button in the webmail interface for reporting misclassification. These events are also logged and used during the next update of reputations.

Taylor also discussed the problem of spoofed source addresses which can affect sender-based detection systems. The Sender Policy Framework (SPF) [35] and Domain-based Email Authentication (DomainKeys) mechanisms are used to authenticate whether an email is really sent from the domain that it claims to be from.

Besides reputation systems, heuristics-based approaches have also been explored. Harris proposed a heuristic method called Greylisting [15] to avoid receiving spam at the recipient’s Mail Transfer Agent (MTA). When a recipient MTA that uses Greylisting receives a delivery attempt, the MTA will respond with an SMTP temporary error message. As required by the SMTP RFC [17], upon receiving an SMTP temporary error, the sending host has to store the message and retry at a later time. The recipient MTA will record the identity of the recent attempts of delivery so that the next attempt will be accepted. Legitimate senders that conform to the standard will have their message delivered, as they will retry according to the RFC. Whereas spammers, who concern more about coding simplicity and speed of the spamming engine, ignore any error message and move on to the next recipient in the list instead of retrying. Thus, spam can be avoided.

Structural features in email social networks may also be exploited for sender-based spam detection. Gomes et al. [12] presented a graph-theoretic analysis of email traffic and proposed several features that can serve to distinguish between spam and legitimate email. Although they did not present any spam detection study in their paper, the features proposed can be used for spam detection. In particular, we use the features called Communication Reciprocity (CR) and Clustering Coefficient (CC) in this paper.

Boykin and Roychowdhury [3] proposed an automated anti-spam tool that exploits the properties of social networks to construct blacklists and whitelists. Based on some heuristics on the structural properties of a social network, the nodes in the network are clustered to form spam and non-spam components automatically.

3. PROBLEM STATEMENT

Given a set of n email accounts $A = \{a_1, a_2, \dots, a_n\}$, a sender set $S \subseteq A$ is defined as the set of email accounts that have sent at least one email and a receiver set $R \subseteq A$ is the set of email accounts that received at least one email. Within the set of senders, t of them are initially labeled as follows:

$$y_i = \begin{cases} 1 & \text{if } a_i \text{ is a legitimate sender,} \\ -1 & \text{if } a_i \text{ is a spammer,} \end{cases}$$

for $a_i \in S$ and $t < n$. We call this set of t labeled sender the training set $a_i \in S_t \subset S$. Although the training set may contain all the senders $S_t \subseteq S$, such a scenario will not be of interest to us as all senders are already labeled.

Logs of events in email transactions $L = \{l_i\}$ between accounts are available as a tuple of attributes:

$$(a_i, a_{j_1}, a_{j_2}, \dots, a_{j_t}, x_1, x_2, \dots, x_m)$$

where $a_i \in S$ and $\{a_{j_i} \in R\}$ are the sender and the corresponding set of receiver accounts, respectively, and x_1 through x_m are other attributes that the log may have, such as time of transaction, message size, event type, sender’s host IP, authentication status, etc. In particular, the possible event types can be {accepted, delayed, rejected sender address, rejected recipient address, unexpected connection termination, other errors}. The goal is to assign the remaining accounts $\{a_{k+1}, \dots, a_n\}$ with a score y_i in $[-1, 1]$, where the sign of the score classifies a sender as either a spammer when negative or a legitimate sender otherwise. Moreover, the magnitude of y_i reflects the confidence of the classification. The score can also be interpreted as the extent of legitimacy of the sender.

In this paper, we limit our focus on the two categories: account that spams (spammer) and account that does not (legit./non-spammer).

4. PROPOSED DETECTION SCHEME

Figure 1 is an overview of the proposed solution for detecting spam senders. Email social networks are first constructed from email transaction logs. A social network can be represented by a directed graph where senders are represented as nodes and email transactions are represented as edges. After the feature extraction and preprocessing stages, a machine learning method, such as *k-Nearest Neighbor* (k-NN) classifier, can be used for the classification task. Some postprocessing on the classifier output may yield results that are more versatile. The remainder of this section details the steps involved.

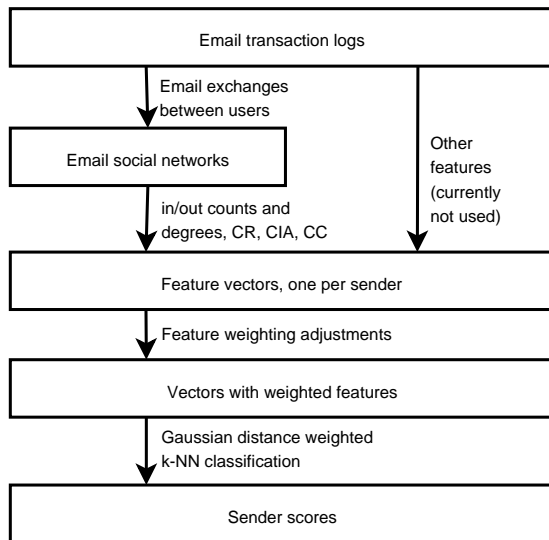


Figure 1: System flow chart

4.1 Social network from logs

Almost all popular SMTP MTA implementations keep the logs of email transactions. The logs record both normal and error SMTP transactions. In addition to the time and date of a transaction, the log also records the IP address of the SMTP client, envelope originator and recipient addresses, message-id, authentication status, etc. Email social networks can be constructed off-line by parsing the email transaction logs.

This paper focuses on events that correspond to successful email deliveries only. The potential contributions of other events, such as attempted delivery to non-existent addresses, are considered as one of the future extensions.

Information about email exchanges between users needs to be extracted from the logs. Let:

- $EmailCount(a_i, a_j)$ = number of emails sent from a_i to a_j , where $a_i \in S$, $a_j \in R$, and $S \cup R = A$
- $C = \{-1, 1\}$ as the set of class labels where -1 and 1 represent spammer and non-spammer, respectively.

It is expected that in a real social network that has interactive participants, some senders are also receivers, i.e., $S \cap R \neq \emptyset$.

An email social network is represented by a directed graph $G = (A, E)$. Each unique email user appeared in the logs is represented by exactly one node $a_i \in A$ in the graph. The sender and recipient relationship is represented by a directed edge $e_{ij} \in E$ which originates from sender a_i and terminates at recipient a_j . The edge weight $[a_i, a_j]$ may be used to store the number of emails sent from a_i to a_j , i.e., $[a_i, a_j] = EmailCount(a_i, a_j)$. The subset of senders $S_l \subset S$ is labeled where each sender has a class label.

4.2 Features from email social networks

Most email users communicate and interact within social groups. They communicate with people who have some kinds of mutual ties among them, such as friends, colleagues, common interests, etc. Their email transaction patterns thus naturally form social networks [3, 18]. On the contrary, spammers generally compose their spam recipient lists by various methods [16, 19, 28, 33] and harvest from a wide variety of sources such as websites, newsgroups, forums, public directories, etc., with robots or even spywares. According to a research study by the Federal Trade Commission (FTC) [10], more than 85% of email addresses posted to web pages and newsgroups received spam. Virtually everything presented with the '@' sign and looks like an email address are harvested. As a result, their recipients are unlikely to be socially related among themselves. This paper proposes to capture social network related features that reflect the aforementioned observation for spam detection.

4.2.1 In-count and out-count

The sum of edge weights of all out-edges represents the total number of emails sent and the sum of edge weights of all in-edges represents the total number of emails received. By definition, a non-spoofed spammer account sends emails in bulk. Such an account sends many more emails than an average sender would. Also, a normal sender sends more emails than a spoofed spammer account as a spammer switches to a new spoofed account frequently. Again, with the same argument as discussed in the previous section, we can expect a spammer account to receive much fewer emails than a normal user.

4.2.2 In-degree and out-degree

In-degree and out-degree in social networks represent the numbers of email accounts that a node receives email from and sends email to, respectively. A non-spoofed spammer account is expected to have a larger list of recipients than that of a normal sender. After all, spammers exercise mass emailing. A spoofing spammer, however, may use each spoofed originator account to send only a couple of spam before generating another spoofed account. As a result, such an account is expected to have a lower out-degree than a normal user would have.

In-degree is related to the response rate and interactivity of an email account. One would expect a human user to be engaged in more bi-directional and interactive communications and thus a higher in-degree. As for spam, the content usually directs the reader to visit web pages rather than an email reply. Also, since spam, by definition, is unsolicited email, we can expect fewer email accounts would respond to a spammer. As for spoofed non-existent spammer accounts, there can be no response at all.

4.2.3 Communication Reciprocity

The Communication Reciprocity (CR) of a node is defined as

$$CR(a_i) = \frac{|OS(a_i) \cap IS(a_i)|}{|OS(a_i)|}, \quad (1)$$

where $OS(a_i)$ is the set of accounts $\{a \in V\}$ that received at least one message from a_i and $IS(a_i)$ is the set of accounts $\{a \in V\}$ that sent at least one message to a_i .

This feature measures the percentage of interactive neighbors that a node a_i has. It aims at capturing the social behavior of human users that tend to provide responses to other users within their social groups. On the contrary, a spam source has a strong structural imbalance between the set of senders and receivers. This follows from the observation that spammers send emails to many

receivers while few of their recipients reply and few other senders send emails to spammers.

4.2.4 Communication Interaction Average

The Communication Interaction Average (CIA) of a node is defined as:

$$CIA(a_i) = \frac{\sum_{j:\exists e_{ij}} w(e_{ji})/w(e_{ij})}{\text{Out-degree of } a_i}. \quad (2)$$

This feature measures the level of interaction between a sender with each of the corresponding recipients. It is the average ratio of *receive count/send count* among the recipients of a sender.

As part of a social behavior of human users, they do not only send emails once or twice but exchange emails with other social neighbors many times. Social groups once formed tend to stay for a considerable time and witness interactive communications. Therefore, accounts that engage in solicited communications should give a higher CIA than those that do not. For instance, spam has a very low response rate. Many of the spam are simply ignored or discarded by recipients. Even if the recipient is interested in the subject described in a spam, the usual action is to follow a hyper-link in the spam instead of replying to the email. In the case of spoofed originator addresses, the difference is even more obvious. The CIA of spammers should be near zero.

4.2.5 Clustering Coefficient

The Clustering Coefficient (CC) of a node measures the friends-of-friends relationship between email accounts. This relationship exists in human user accounts because a social group is established out of some common bonding that holds the group members together. For instance, personal friends of a particular person A are likely to know each other as well, perhaps through group activities. Therefore, the friends, in addition to communicating with A , would also communicate with each other.

As for spammers, since they harvest email addresses from the public domain, such as web pages, and merge them with addresses from many other sources, the resulting recipients are unlikely to share common interests and communicate with each other. In other words, those neighbor accounts of spammers are unlikely to exhibit the friends-of-friends relationships.

Given z neighbors, it is easy to observe that the maximum number of connections among the z neighbors is $[z \times (z - 1)]/2$. CC measures the existence of such connections among neighbors of a node. It is defined as

$$CC(a_i) = \frac{n_{a_i}}{[z_{a_i}(z_{a_i} - 1)]/2}, \quad (3)$$

where z_{a_i} is the number of neighbors of a_i and n_{a_i} is the number of such connections that actually exist in the social network of a_i .

4.3 Preprocessing

For the purpose of spam detection, we propose to estimate the likelihood that an email originator is a spammer by using machine learning techniques. The goal is to assign a score to each sender based on the likelihood of that sender being a spammer. This section details the formulations of the machine learning approach given social network features extracted in previous steps. The discussion below focuses on a supervised learning approach with only two classes of senders: (1) spammers and (2) legitimate senders.

4.3.1 Sender Feature Vectors

Since a spammer that we can detect must be one of the senders in the email social network, extracting features for email accounts that sent at least one email is enough. As will be shown in section 6,

using only any one of the seven features presented in the previous section is not sufficient in detecting spammers accurately. Instead, the values of the seven features \hat{f}_l ($l = 1, \dots, 7$) are used to form a feature (row) vector $\hat{\mathbf{x}}_i = (\hat{f}_1, \dots, \hat{f}_l, \dots, \hat{f}_7)$ for each of the accounts $a_i \in S$.

Since the seven features are of different units and magnitudes, each of the feature values is normalized, the attributes \tilde{f}_l of the normalized feature vector $\tilde{\mathbf{x}}_i$ is given by:

$$\tilde{f}_l = \frac{\hat{f}_l - MEAN(\hat{f}_l)}{VAR(\hat{f}_l)}, \quad \forall l, \quad (4)$$

where $MEAN(\hat{f}_l)$ and $VAR(\hat{f}_l)$ are the mean and variance, respectively, of \hat{f}_l over all $a_i \in S$.

4.3.2 Weighted features

Among the seven features, some may be more useful than others in distinguishing spammers from legitimate senders. While the in/out-degrees and in/out-counts can be useful as auxiliary features, using these features alone can focus the learning process. After all, some legitimate users may look like spammers if one looks at these four features only. For example, there exist low traffic legitimate users who have small values for all four features, similar to seemingly low traffic spammers that employ originator spoofing. On the other hand, the three other features capture the main structural differences between spammers and legitimate senders. To prevent the influence of important features from being masked out, normalized feature values are further weighted according to their relative importance to give the weighted feature vector \mathbf{x}_i :

$$\mathbf{x}_i = \tilde{\mathbf{x}}_i \mathbf{w}, \quad (5)$$

where $\mathbf{w} = (w_1, \dots, w_l, \dots, w_7)^T$ is the weight (column) vector with higher attribute values for relatively more important attributes.

Although the description above assumes seven features, it is possible to include more (or fewer) features, $l = 1, 2, \dots, L$. A possible future extension to incorporate additional features. As a summary, each sender now has a feature vector with normalized and weighted feature values.

4.4 Machine Learning to Assign Spam Score

Each sender in an email social network is now represented by a feature vector of normalized and weighted feature values. It is assumed that similar senders have similar feature values. Given a set of labeled senders S_l , the goal is to label the rest. This section describes the formulation to use a k -Nearest Neighbor (k -NN) classifier to assign spam scores to unlabeled senders.

k -NN is an instance based supervised learning algorithm. The algorithm assumes all inputs to be points in the n -dimensional Euclidean space. To classify a new instance a_q , the Euclidean distances between the instance and other training points are calculated. One may assign a class by the majority vote among the k -nearest labeled neighbors. Let y_j be the classification of training instance a_j , where $j = 1, \dots, k$ being the k nearest neighbors of a_q . Then the k -NN classification is given by:

$$v_{kNN} = \arg \max_{c_i \in C} \sum_{j=1}^k \delta(c_i, y_j), \quad (6)$$

where $\delta(a, b) = 1$ if $a = b$, or zero otherwise. One may also weight the classifications of the training points with the similarities, such that the more similar a training point is, the more weight its classification has on a_q .

The k -NN can be a simple yet effective method. Nonetheless, there are some drawbacks. By using the Euclidean space, k -NN

assumes that each feature are of equal importance. This makes the algorithm vulnerable to irrelevant or noisy features. This can be compensated by weighting feature values according to their relative importance, as suggested in section 4.3.2. Feature weighting can be viewed as scaling the axes in the Euclidean space according to the relative importance of features.

Being a lazy learning algorithm, k -NN can be quite inefficient at classification time. To classify a new instance, k -NN has to compute the distances between the new instance and all the training instances, in order to discover the k nearest neighbors. Special data structures, such as those in [2, 5], can be used for storing and indexing training examples such that the search for nearest neighbors can be more efficiently done in the expense of some computations in the training phase and storage overhead. Details of these technique are out of the scope of this paper.

Our goal is to assign a legitimacy score to each of the senders reflecting its likelihood of being a legitimate sender. The higher the score, the more likely that the sender is legitimate. This paper proposes to use the similarity weighted k -NN method as will be discussed in the subsections that follow.

4.4.1 Gaussian Similarity

To use a distance based method like k -NN, we first define the similarity measure of neighbors. The idea is to give higher weights to labeled data points that are closer to the data point to be labeled. We assume the feature vectors to be in an Euclidean space \mathbb{R}^L , where L is the number of features included in the feature vectors. Let the Euclidean distance between two feature vectors \mathbf{x}_i and \mathbf{x}_j be $d(\mathbf{x}_i, \mathbf{x}_j)$. The Gaussian similarity is given by:

$$w_{ij} = e^{-\frac{d^2(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}}, \quad (7)$$

where σ is a parameter controlling the decay factor of the Gaussian similarity function. This similarity measure decays exponentially as the Euclidean distance increases. In other words, the close the two vectors are in the Euclidean space, the larger the Gaussian similarity.

4.4.2 Similarity Weighted Mean k -NN Scores

The score for an unlabeled feature vector is inferred from the similarity weighted mean of labels of its k nearest neighbors. Given a vector \mathbf{x}_i , we first calculate the Gaussian similarity w_{ij} for $a_j \in S_l$, i.e., the similarity between \mathbf{x}_i and each of the feature vectors that corresponds to a labeled sender. Recall that the larger the Gaussian similarity, the closer the labeled vector is from \mathbf{x}_i . Thus, the score of \mathbf{x}_i in the k -NN definition is simply the similarity weighted mean of the labels with the k largest w_{ij} . Denote the set of k \mathbf{x}_j that have the k largest Gaussian similarities as κ . The score \tilde{y}_i of \mathbf{x}_j is given by:

$$\tilde{y}_i = \frac{\sum_{j:\mathbf{x}_j \in \kappa} w_{ij} y_j}{k}. \quad (8)$$

For the case that some of the vectors are of the same similarity resulting in more than k largest neighbors, we randomly choose enough vectors to break the tie. As defined earlier, the labels are -1 for spammers and 1 for legitimate users. The sign of the resulting score can be treated as a classification of spammer or legitimate sender and the magnitude reflects the confidence level. For instance, one may set a threshold for the score below which a sender is considered a spammer. Three potential ways of using this score to counter spam are discussed in section 5.

4.4.3 Score scaling

The Gaussian similarity function, which is used to weight the scores in the previous step, is unlikely to give a weight close to unity unless the feature vectors are so tightly clustered such that most of the similarities are near one. Recall that the possible values of labels are bounded by $[-1, 1]$, weighting the labels with such a similarity function is likely to give scores that are very small and thus are not effectively utilizing the whole range of possible scores (i.e., $[-1, 1]$). To avoid the distribution of scores being too clustered around zero, we may scale the scores such that the maximum magnitude is one. Denote the set of vectors \mathbf{x}_j corresponding to unlabeled senders be λ . After all unlabeled senders are scored, the scaled score \hat{y}_i of each \mathbf{x}_i is given by:

$$\hat{y}_i = \frac{\tilde{y}_i}{\max_{\mathbf{x}_j \in \lambda} \{|\tilde{y}_j|\}}. \quad (9)$$

4.5 Section Summary

Given the feature vectors of labeled senders, this section shows how k -NN can be used to assign legitimacy scores to unlabeled senders. It is assumed that senders that share similar feature values, thus close in proximity in the Euclidean space, belong to the same class. Based on this, the score is assigned by the similarity weighted mean of k -NN's labels. The sign of the score may be used to classify a sender and the magnitude reflects the confidence. Alternatively, the higher the score, the more likely that a sender is a legitimate sender. The score can therefore be treated as a score of legitimacy.

5. POTENTIAL MITIGATION SCHEMES

A detection scheme needs a mitigation scheme to react to spam. There are more than one way to use the legitimacy scores provided by the social network based detection scheme to mitigate spam. One of the more straightforward ways is to apply a threshold to the score below which all email from the sender will be filtered. While this approach is simple, we observe that it is unlikely that the social network based detection alone is accurate enough for the purpose. Also, existing content-based schemes and rule-based schemes are still performing reasonably well. We prefer to use the social network based detection scheme to complement rather than replace content-based filtering schemes.

Different ways of combining filters have been explored in the literatures. Segal et al. [26] proposed to form a pipeline of anti-spam filter components. An email passes through each component in the pipeline one by one. Each component assigns a score to the email. An email can be directly classified by an intermediate classifier and skip all subsequent components if the classifier determined the classification of the email with high confidence. Lynam and Cormack [21] explored different ways of combining anti-spam filters. Specifically, the voting of binary classifications from spam classifiers, the log-odds averaging of spam scores, the use of Support Vector Machine on spam scores from different spam filters and the use of logistic regression to find the weights for computing the weighted average of spam scores from multiple filters.

Since the main focus of this paper is the formulation of the detection scheme as described in the previous sections, we intend to discuss only simplified views of three potential approaches in which the legitimacy sender scores may be used to complement existing score generating filters. In depth study on the benefits and effectiveness on advanced filter ensemble schemes are reserved for future work.

5.1 Parallel single thresholding approach

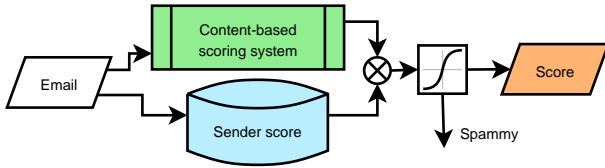


Figure 2: A parallel single thresholding system

Many of the content-based spam detection schemes are able to generate a spam score, and so does the proposed social network based scheme. A natural way to combine the two is to run the two schemes in parallel so that each of them generates a score. The two scores are combined to give a decision. Figure 2 shows such a setup.

An email is fed to both schemes. The content-based analyzer will analyze the content of the email and assign a score s_c to the email. The higher the score s_c is, the more confident that the analyzer thinks the email is spam. The proposed social network based scheme will identify the originator of the email concerned and query the score database for the sender's score. Since the score from the database is a legitimacy score, we may switch it to a spam score by a simple negation, i.e., $s_s = (-1)y_i$. This spam score can then be combined with other content-based and rule-based filters with, for example, a weighted sum, to generate a single spam score. Emails with a score higher than a certain threshold can be considered as spam.

5.2 Serial multiple thresholding approach

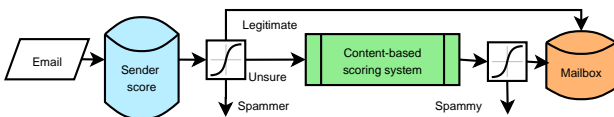


Figure 3: A serial multiple thresholding system

To cope with the advanced techniques of spamming, content-based filters are getting more and more sophisticated. The sophistication also translates to heavier load on the spam filtering module. On the contrary, the spam sender score is first determined offline. Only a lightweight query to the score database is needed during the online process. One may consider taking a serial approach by filtering spam with the lightweight sender score approach first.

Figure 3 shows a serial multiple thresholding system. During the spam filtering process, the legitimacy score for the email sender will first be fetched from the database. Two thresholds $T_l > T_s$ on this score will be defined. Emails from senders with legitimate score above T_l will be accepted directly to the inbox, skipping the content-based filter. Senders with a score lower than T_s will be considered spammers. Their emails can be rejected at this stage or flagged as spam directly. Email from senders with a score in between the two thresholds, i.e., $T_s \leq f_i \leq T_l$, will be passed to the content-based analyzer that will make the final decision. Spammy emails (i.e., emails with high spam scores) can be filtered or flagged accordingly.

This approach has several advantages. The sender based filtering scheme acts like an automatic whitelist and blacklist approach. As a result, the load on the content-based filter will be lowered. Additional resource intensive analysis on the email, such as Optical Character Recognition (OCR) on images, may now be enabled to improve the accuracy. Also, notice that some of the legitimate senders are allowed to skip the subsequent filters, an adminis-

trator may use a more aggressive threshold for those filters while maintaining the same false overall rejection rate.

5.3 Serial throttling and thresholding approach

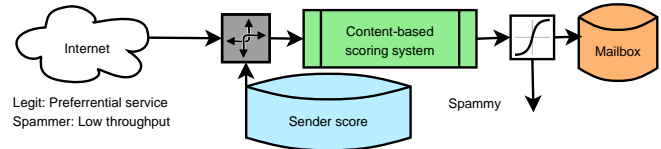


Figure 4: A serial resistance and throttling system

A variant of the serial approach is to throttle rather than to filter senders with the sender score. Figure 4 shows a conceptual diagram of such an approach. It is observed that spammers generally depend on a very high email delivery throughput to generate a revenue [7, 19]. Li et al. [20] proposed to slow down the transmission rate of a suspected spam at the TCP level. Although they did not propose a way to identify suspicious senders, the authors showed that when enough recipients are using TCP damping against emails with a high spam likelihood, it is possible to lower the delivery throughput of spammers considerably. This may be used as a deterrent that drives spammer away from the server to avoid high delays.

Also, since emails are slowed down during the delivery but not entirely dropped, false rejection of legitimate email is much less expensive. An average user may not care about some minutes of delays in delivery. However, for spammers that require high delivery throughput, a slowdown in delivery hurts their profitability.

One potential problem is that the scheme requires online determination of the spam likelihood of an email. Content-based analysis may fall short in this aspect since limited information about the email is revealed during the earlier stage of the email delivery process. By the time the content of an email is being received and analyzed, it may already be near the end of the delivery. It may be too late for TCP damping to slow down the spammer enough to make a difference.

Given that our proposed scheme gives a sender score and an online query of the score is lightweight, one can afford to implement TCP damping with the legitimacy score. Senders with a high legitimacy score would be offered normal or preferential service while others are slowed down. One of the way is to use an exponential decay function on the legitimate score to determine the extent of the damping. The delay imposed by the server grows exponentially with the decrease of the legitimate score.

6. EXPERIMENTS

The proposed method is tested with legitimate senders extracted from the Enron dataset prepared by Shetty and Adibi [27] and simulated spammers. The Enron email dataset has been released to the public by the Federal Energy Regulatory Commission. The CALO project [8] extracted and prepared a dataset for research use. Shetty and Adibi [27] further processed the dataset by removing duplicated, corrupted and system generated emails, and analyzed them. The dataset is composed of emails in the mailboxes of 150 users in the reasonably large organization. The social network characteristics are well preserved. In this paper, email exchange information is extracted from email headers to simulate email exchange logs for our evaluation purposes. If an email is addressed to more than one recipient, one transaction will be generated for each recipient.

x	$P[\text{out-degree} = x]$
1	0.664
2	0.171
3	0.070
4	0.040
5	0.024
6	0.014
7	0.010
8	0.007

Table 1: Out-degree probability distribution of simulated spam accounts

6.1 Extracting legitimate Enron senders

The enron dataset contains both spam and nonspam, thus both legitimate senders and spammers. To get legitimate senders from the Enron dataset, we first extract the email transactions within the Enron email domain (i.e., email transactions with sender and recipient addresses with @enron.com). These senders are, for the moment, labeled as legitimate senders. Social networks of these senders are constructed. Raw emails are also fed to SpamAssassin to generate scores. By examining the social networks, scores and email subjects manually, bogus senders that have forged to be from the Enron domain are identified.

Notice that the dataset only contains emails from mailboxes of 150 Enron users while there are many other Enron users exchange emails with these 150 Enron users. The transactions not involving any of the 150 Enron users are not visible. As a result, the social networks constructed will represent the full social networks between the 150 Enron users but only a partial view of the social networks of users other than the 150 users. In reality, a single organization or an ISP will have a similar situation, where only a partial view of the social networks is available for senders outside of the email domains that they have control of. By including not only the 150 Enron users but also other Enron users, we are not assuming that we have a global view of the social networks of every email user.

6.2 Simulated spam senders

Transactions from spammers are simulated by generating spam accounts that inject spam emails to the dataset of legitimate senders obtained from the previous section. The out-degree of each spam account is set according to the out-degree distribution provided in [12]. The out-degree distribution in [12] is measured from real email traffic of a university in Brazil. It is assumed that each spam account will send only one spam to each randomly chosen recipient. Table 1 summarizes the actual probabilities of spammer out-degree used. Also, since recipients usually do not reply to the spam accounts, the probability of a reply from a recipient is fixed to be 0.05.

5000 spam accounts are generated to inject spam traffic into the Enron dataset obtained as described in the last section. An email social network is generated from the resulting dataset and is used for feature extraction. Enron accounts that have out-degrees > 0 are used as legitimate sender accounts. There are 9150 senders, with 4150 of them being legitimate senders.

Unless otherwise specified, all of the following experiments are based on this dataset and are repeated 100 times to report the averages as shown. 120 senders from each class ($\approx 2.5\%$) are randomly selected to be labeled as the training set. The error bars show the standard deviations.

6.3 Number of nearest neighbors

To use k -NN, one needs to determine the number of neighbors (k) to be used in the score calculation. The value of k affects not only the accuracy of the scheme, but also the running time requirement. The larger the k , the longer it takes for the similarity calculations in the scoring phase. In this paper, we determine the value of k empirically through a series of experiments with varying k . The relative weights for the features are fixed to be 1 for in/out-degrees and in/out-counts and 5 for CR, CIA and CC. The ROC curves for $k = \{1, 3, 5, 7, 9, 11\}$ are shown in figure 5. We choose $k = 3$ in all the subsequent experiments.

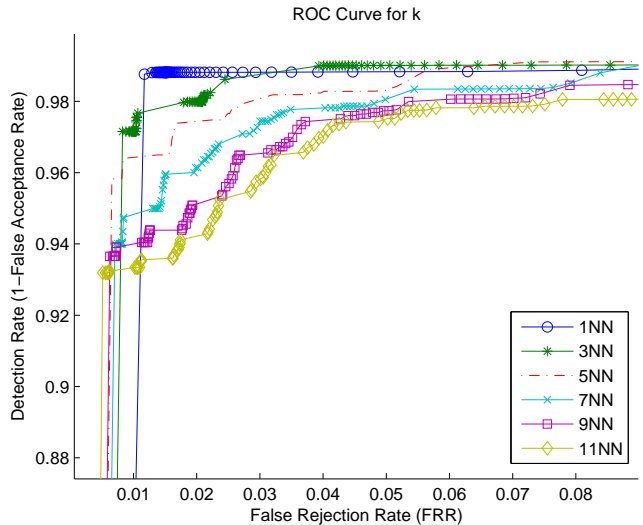


Figure 5: ROC curves: Varying k in k -NN

6.4 Feature weightings

Among the seven features, some are more important than others. The feature should be weighted accordingly to yield better performance. Extensive experiments have been done to investigate different feature weightings. This section presents some of the more representative results.

Figure 6 shows the ROC curves of six different weights of CC. As we increase the weight of CC, the classification performance, in general, increases. This shows the relative importance of this feature over others. The increase in performance becomes insignificant for weights larger than 15. Table 2 shows the area above the ROC curve while varying the weights of CIA. Considering the associated standard deviations of the results, we observe that the improvement in performance is inconclusive. To proceed on experiments on CR, we choose weighting of CC to be 15, the smallest among 15, 20 and 25 to avoid over-emphasis; and that for CIA to be 10. Figure 7 shows the results of varying CR. The trend of degrading performance is observed as the weight of CR increases. Thus, a weight of 1 is chosen for CR.

Through similar experiments on the four remaining features: in/out-counts and in/out-degrees, an increase in their weights hurts performance in general. Another set of feature selection experiments reveals that using only the three features CR, CIA and CC yields poor accuracies. Therefore, we choose the weights of the seven features to be 1 for the in/out-counts, in/out-degrees and CR, 10 for CIA, and 15 for CC.

Figure 8 shows the results for different numbers of training data

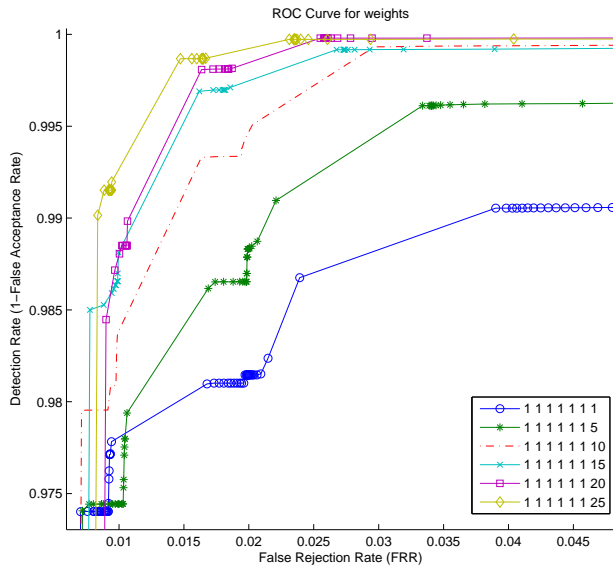


Figure 6: ROC curves: Varying Clustering Coefficients (CC)

No.	Weights	Area (%)	Std	Var
1	01 01 01 01 01 10 15	0.39184	0.15909	0.00025
2	01 01 01 01 01 15 25	0.41627	0.14438	0.00021
3	01 01 01 01 01 25 15	0.42567	0.17520	0.00031
4	01 01 01 01 01 01 15	0.43668	0.21220	0.00045
5	01 01 01 01 01 01 25	0.43872	0.18122	0.00033
6	01 01 01 01 01 05 25	0.44140	0.18848	0.00036
7	01 01 01 01 01 10 25	0.44265	0.15273	0.00023
8	01 01 01 01 01 05 15	0.44560	0.21346	0.00046
9	01 01 01 01 01 15 15	0.45097	0.24533	0.00060
10	01 01 01 01 01 25 25	0.45976	0.25064	0.00063

Table 2: Varying the feature weights of CIA: The mean area above ROC curve, standard deviation and variance. The seven values in the first column represent the weights for in-count, out-count, in-degree, out-degree, communication reciprocity, communication interaction average, and clustering coefficient.

given. As observed from the figure, the accuracy increases noticeably as we increase the number of labeled senders for each class from 10 (0.2%) to 30 (0.6%) and 90 (1.8%). The detection rate attained 99% with only 0.5% of false positives as 3% of the senders are labeled.

7. DISCUSSIONS AND CONCLUSION

The proposed scheme facilitated assignment of a legitimacy score to each sender given a small portion of labeled senders. No content of emails is needed. As shown in the previous section, encouraging results are obtained from a stand-alone setting with only 3% of the senders labeled for the training phase.

The results may seem sound especially when it is yet to be combined to existing content-based schemes. However, we have to be cautious about the actual performance of the proposed scheme considering that we are scoring senders instead of individual emails.

One of the concerns is that the experimental settings are using originator email addresses as senders. Although in reality spammers do spoof and change their originator email addresses fre-

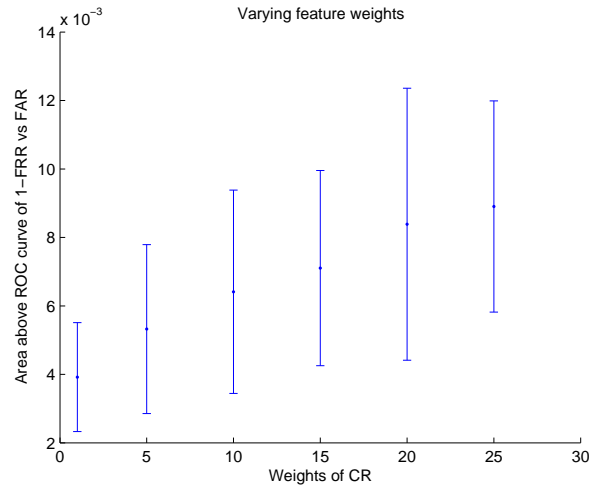


Figure 7: Area above ROC: Varying Communication Reciprocity (CR)

quently and this is reflected in our dataset, there are spammers that deliberately spoof specific addresses that may be commonly seen and likely to be whitelisted. For instance, originator addresses of email notifications from popular websites such as ebay and amazon. As discussed by Taylor in [32], for a sender-based approach, one may use the domain part of the originator email address together with the IP address of the sending host to identify a sender. This combination facilitates, to some extent, sender authentication. The recipient can authenticate the sender with Sender Policy Framework (SPF) and DomainKey.

Another concern is the relative number of emails associated with a spammer address compared to that with a legitimate sender address. As we assume that a spammer changes its sender address frequently, the relative number of emails should be small compared to an active legitimate sender. Should a false positive occurs on an active legitimate sender, many emails will be affected. As with other anti-spam schemes, one would like to push the false positive rate to near zero. To achieve this, one may need to consider various ensemble approaches of existing schemes with the proposed scheme, such as those discussed in section 5.

In this paper, we have explored the new direction of taking a learning approach on structural features extracted from the email social networks. Currently, only seven features are considered. One possible extension is to explore additional features, such as those that capture the anomaly in changes of sender behaviors over time.

8. REFERENCES

- [1] Gmail. Retrieved: Jan. 2006 <http://gmail.google.com/>.
- [2] A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 97–104, New York, NY, USA, 2006. ACM Press.
- [3] P. O. Boykin and V. P. Roychowdhury. Leveraging social networks to fight spam. *Computer*, 38:61–68, April 2005.
- [4] E. Burns. The deadly duo: Spam and viruses, Jun. 2006. Retrieved: Jul. 2006 <http://www.clickz.com/stats/sectors/email/print.php/3614491>.
- [5] Y. S. Chen, Y. P. Hung, T. F. Yen, and C. S. Fuh. Fast and versatile algorithm for nearest neighbor search based on a lower bound tree. *Pattern Recogn.*, 40(2):360–375, 2007.

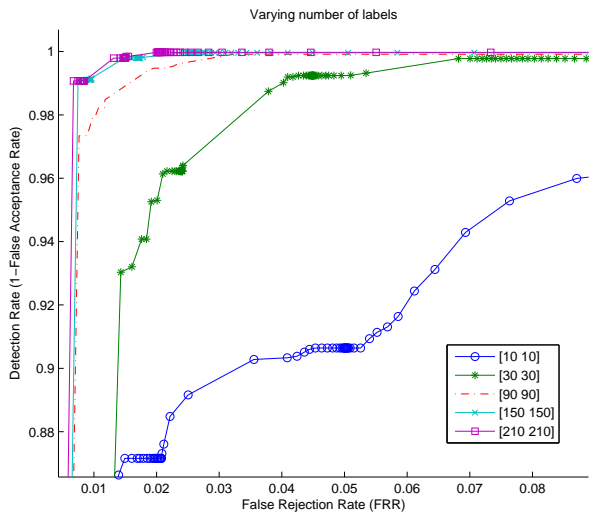


Figure 8: ROC curves: Different numbers of training data. The numbers in the legend are the numbers of labeled legitimate and spam senders given, respectively.

- [6] P. A. Chirita, J. Diederich, and W. Nejdl. Mailrank: using ranking for spam detection. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 373–380, New York, NY, USA, 2005. ACM Press.
- [7] S. Cobb. The economics of spam, Feb. 2003. Retrieved: Jun. 2006 http://www.spamhelp.org/articles/economics_of_spam.pdf.
- [8] W. Cohen. Enron email dataset, Mar. 2004. Retrieved: Jan. 2006 <http://www.cs.cmu.edu/~enron/>.
- [9] Federal Trade Commission. Unsolicited commercial e-mail. *Prepared Statement to the subcommittee on Telecommunications, Trade and Consumer Protection of the committee on Commerce*, Nov. 1999.
- [10] Federal Trade Commission. Email address harvesting: How spammers reap what you sow, Nov. 2002. Retrieved: Mar. 2006 <http://www.ftc.gov/bcp/online/pubs/alerts/spamalrt.pdf>.
- [11] J. Golbeck and J. Hendler. Reputation network analysis for email filtering. In *First Conference on Email and Anti-Spam*, Jul. 2004. <http://www.ceas.cc/papers-2004/177.pdf>.
- [12] L. H. Gomes, R. B. Almeida, L. M. A. Bettencourt, V. Almeida, and J. M. Almeida. Comparative graph theoretical characterization of networks of spam and legitimate email. In *Second Conference on Email and Anti-Spam*, Jul. 2005. <http://www.ceas.cc/papers-2005/131.pdf>.
- [13] P. Graham. A plan for spam, Aug. 2002. <http://www.paulgraham.com/spam.html>.
- [14] J. Graham-Cumming. The spammers' compendium, Jun. 2006. Retrieved: Jun. 2006 <http://www.jgc.org/tsc/>.
- [15] E. Harris. The next step in the spam control war: Greylisting, Aug. 2003. Retrieved: Aug. 2006 <http://projects.puremagic.com/greylisting/whitepaper.html>.
- [16] B. Hoanca. How good are our weapons in the spam wars? *Technology and Society Magazine, IEEE*, 25(1):22–30, 2006.
- [17] J. Klensin. Simple Mail Transfer Protocol. RFC 2821 (Proposed Standard), Apr. 2001.
- [18] J. S. Kong, P. O. Boykin, B. A. Rezaei, N. Sarshar, and V. P. Roychowdhury. Scalable and reliable collaborative spam filters: Harnessing the global social email networks. In *Second Conference on Email and Anti-Spam*, Jul. 2005. <http://www.ceas.cc/papers-2005/143.pdf>.
- [19] J. Leyden. The economics of spam, Nov. 2003. Retrieved: Jun. 2006 http://www.theregister.co.uk/2003/11/18/the_economics_of_spam/.
- [20] K. Li, C. Pu, and M. Ahamad. Resisting spam delivery by TCP damping. In *First Conference on Email and Anti-Spam CEAS 2004*, Jul. 2004. <http://www.ceas.cc/papers-2004/191.pdf>.
- [21] T. R. Lynam, G. V. Cormack, and D. R. Cheriton. On-line spam filter fusion. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 123–130, New York, NY, USA, 2006. ACM Press.
- [22] MX Logic. Overall email threat traffic trends, Mar. 2007. Retrieved: Mar. 2007 http://www.mxlogic.com/threat_center/.
- [23] S. L. Pfleeger and G. Bloom. Canning spam: Proposed solutions to unwanted email. *Security & Privacy Magazine, IEEE*, 3(2):40–47, Mar.–Apr. 2005.
- [24] Rhyolite Software. Spam ratio at most DCC servers, Jan. 2006. Retrieved: Jan. 2006 <http://www.rhyolite.com/anti-spam/dcc/graphs/big.cgi?BIG=all-spam-ratio>.
- [25] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk E-mail. In *AAAI-98 Workshop on Learning for Text Categorization*, Madison, Wisconsin, 1998.
- [26] R. Segal, J. Crawford, J. Kephart, and B. Leiba. Spanguru: An enterprise anti-spam filtering system. In *First Conference on Email and Anti-Spam CEAS 2004*, Jul. 2004. <http://www.ceas.cc/papers-2004/126.pdf>.
- [27] J. Shetty and J. Adibi. The enron email dataset database schema and brief statistical report, Mar. 2004. Retrieved: Jan. 2006 http://www.isi.edu/adibi/Enron/Enron_Dataset_Report.pdf.
- [28] Sophos Plc. Glossary of spam terms, Jul. 2006. Retrieved: Jul. 2006 <http://www.sophos.com/security/spam-glossary.html>.
- [29] Spamassassin. AutoWhitelist - Spamassassin Wiki, 2007. Retrieved: Jun. 2007 <http://wiki.apache.org/spamassassin/AutoWhitelist/>.
- [30] Spamhaus. The definition of spam, Jul. 2006. Retrieved: Jul. 2006 <http://www.spamhaus.org/definition.html>.
- [31] Symantec. Symantec internet security threat report, Mar. 2006. Retrieved: Jul. 2006 <http://www.symantec.com/enterprise/threatreport/index.jsp>.
- [32] B. Taylor. Sender reputation in a large webmail service. In *Third Conference on Email and Anti-Spam CEAS 2006*, Jul. 2006. <http://www.ceas.cc/2006/19.pdf>.
- [33] Wikipedia. Directory harvest attack, Jul. 2006. Retrieved: Jul. 2006 http://en.wikipedia.org/wiki/Directory_Harvest_Attack/.
- [34] Wikipedia. Spam (electronic), Jul. 2006. Retrieved: Jul. 2006 <http://en.wikipedia.org/wiki/Spamming/>.
- [35] M. Wong and W. Schlitt. Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1. RFC 4408 (Experimental), Apr. 2006.