ACCURATE KEYWORD BASED SPAMS FILTERING IN SOCIAL NETWORKS.

Durga Prasad JKS,
Co-Author – Dharani J
M.tech, Department of information technology,
SRM University, Chennai.

Abstract—The numerous growth of unsolicited emails are increasing day by day and it can be identified and with development of various spam filter techniques.

in that spam filter techniques Bayesian spam filter technique plays major role in internet in order to filter the unsolicited emails effectively and accurately to keyword based spam filter and continuously evolve to tackle new spam by learning keywords in spam emails.

but the major drawback of Bayesian spam filters are easily poisoned by clever spammers with different keywords and add many innocuous words in the emails as well store in anti-spam filters as a ham keywords that means (i.e., bad w) and also Bayesian spam filters are need sufficient amount of time to adapt to an new spam based on user training or feedback. moreover few spam filters that are using currently in social networks are exploiting and not properly detecting the spam messages, images and videos so on.

In order to develop an accurate user-friendly Bayesian Spam filter .we propose and enhancing and apply an Re-tokenization (Reverse Tokenization) concept in an existing SOcial network Aided Personalized and Effective spam filter(SOAP) in this paper. In SOAP, each node connects to its social friends; that is, nodes form a distributed overlay by directly using social network links as overlay links. Each node uses SOAP to collect information and check spam autonomously in a distributed manner. interests to detect spam adaptively and automatically. In each node, SOAP integrates four components into the basic **Bayesian filter: social closeness-based spam filtering, social interest-based spam filtering, adaptive trust management, and friend notification. We have evaluated the performance of SOAP using simulation based on trace data from Facebook. We also have implemented a SOAP prototype for real-world experiments and we enhance these project with the help reverse tokenization concept to improve the performance accurately stressed words detection (e.g: K!iLL, L*ottery)by using Bayesian spam filters in terms of accuracy, attack-resilience, and efficiency to avoid the spam poisoning attack on social networks.**

Index Terms: Spam Filtering, Bayesian Spam Filtering, Social Networks, Reverse Tokenization.

1. INTRODUCTION:

In Today's Modern Life Electronic Communication have revolutionized business communications. Although huge increases in use has taken some organization by surprise it is a common place now for people at work to use email as easily as using Telephone and internet access is often part of the setup of a workstation. In that Internet emails and Social Networks is a most popular medium of communication has become an almost indispensable tool for business, education, social and personal lives. However Spam emails are becoming a severe problem in email systems as well as in social networks (such as Face Book, Twitter and Wall Post etc.). Currently 120 billion spam emails are sent per day, with projected cost of $338 billion by 2013. Spam emails interfere with both email service providers and end users. A fundamental way to prevent spam email is to use Anti-spam filter Techniques engines. The Anti-Spam Engines can be deployed at strategic places based on the requirements of customer in Client Level (as POP/SMTP/IMAP Proxies and Plugins) and Server Level (in Network Boundary Gateway, Mail Routers and Message Store) of a website. By using of Anti-Spam Engines also some spam filters are suffered from spam mails by improper architecture of filters by programmer and also improper maintaining of services by third party spam service providers. Some of attack-resilient and personalized features are important to achieve high accuracy. A more accurate filter generates less false positives and false negatives. False positives are legitimate emails that are mistakenly regarded as spam emails. False negatives are spam emails that are not detected. There are two primary types of spam filter attacks: poison attacks and impersonation attacks. So, they introduced SOcial network Aided Personalized and effective spam filter (SOAP) for spam detection to meet the requirements in an social network into the email network.

SOAP in each node leverages social networks to combine four components into the basic Bayesian filter.

1. Social Closeness-based spam filtering.

2. Social Closeness-based spam filtering.

3. Adaptive trust management and

4. Friends Notifications

By using above SOAP components we can filter the spam mails and messages on relationship between each users and profile marinating in social networks But SOAP also facing keyword based poisoning attack in emails and social networks. In order to overcome this attack we are going to enhance the SOAP Architecture using reverse tokenization concept in order to resolve problems of accurate keyword poisoning attack in an SOAP architecture.

2. RELATED WORK:

2.1.Fighting Spam:

The Anti-Spam engines can be deployed at strategic places based on the requirements or needs of the customers. Following are the layers where it can be applied:

SERVER LAYER:

1. Network Boundary/Gateways:

These are the server that are listed in the DNS MX records for the recipient's/sender's domain these are most preferred places to install the Anti-spam/Antivirus filters.

2. Mail Routers:

They can be any of internal mail routers/relays. However it is not a common practice to keep the ant spam filters as part of the internal mail routers.

3. Message Store:

The Last/First Server which holds the mail boxes for recipients/sender's of an organization. These are usually kept inside the private network.

CLIENT LAYER:

POP/IMAP/SMTP Proxies:

These act as transparent/opaque proxies for user sending and receiving mails. These filter the mail and optionally tag the email or store it in a different location.

Plugins to MUAs:

These are the plugins to message sending/receiving clients and filter the mails. (Before sending-for SMTP and after receiving for POP/IMAP)

The vast quantity of spam emails are distributed among in a networks and many spam filtering approaches are came to avoid spams emails in a network. Likewise approaches are mainly classified into two classes, They are: Content Based and Identity Based Approaches.

2.2 Content-Based Approaches:

The basic approach of content-based spam filtering is the static keyword list, which however makes it easy for a spammer to evade filtering by tweaking the message. The second category of content-based approaches includes
Machine learning-based approaches such as Bayesian filters, decision trees, Support Vector Machines, Bayes Classifiers and combinations of these techniques. In this approach, a learning algorithm is used to find the characteristics of the spam and of legitimate emails. Then, future messages can be automatically categorized as highly likely to be spam, highly likely to be legitimate emails, or somewhere in between.
The third category of content-based approaches is collaborative spam filtering. Once a spam email is detected by one User, other users in the community can avoid the spam later on by querying others to see if their received emails are spam or not. Spam Net uses a central server to connect all participants of the collaborative spam filter. Spam Watch is a distributed spam filter based on the Tapestry Distributed Hash Table system. Kong et al proposed a distributed spam filter to increase the scalability of centralized collaborative
spam filters.

2.3 Identity-Based Approaches:

The simplest identity-based spam filtering approaches are blacklist and whitelist , which check the email senders for spam detection. Whitelists and blacklists both maintain a list of addresses of people whose emails should not and should be blocked by the spam filter, respectively. One server side solution records the number and frequency of the same email sent to multiple destinations from specific IP
addresses. If the number and frequency exceed thresholds, the node with the specific IP address is blocked. Boykin et al. constructed a graph in which vertices represent email addresses and direct edges represent email
interactions. Emails are identified as spam, valid, or unknown based on the local clustering coefficient of the graph subcomponent. This is based on the rationale that the social communication network of a normal node has a higher clustering coefficient than that of a spam node is a whitelist spam filtering system based on social links. It is based on the assumption that all friends and FoF are trustable. Hameed proposed LENS, which extends the FoF network by adding trusted users from outside of the FoF networks to mitigate spam beyond social circles. Only emails to a recipient that have been vouched by the trusted nodes can be sent into
the network. DeBarr et al. evaluated the use of social network analysis measures to improve the performance of a content filtering model. They tried to detect spam by measuring
the degree centrality of message relay agents and the average path length between senders and receivers. They claimed that the messages from a promiscuous mail relay or messages with unusual path lengths that deviate from the
average are more likely to be spam. Lam et al.proposed a learning approach for spam sender detection based on user interaction features (e.g., in degree/out degree and interaction
frequency) extracted from social networks constructed from email exchange logs. Legitimacy scores are assigned to senders
based on their likelihood of being a legitimate sender.Tran et al. implemented an email client called Social Email, which provides social context to messages using a social network's underlying social graph. This not only gives each email recipient control over who can message him/her, but also provides the recipient with an understanding of where the message socially originated from. However, if a spammer compromises a legitimate user's computer, the spammer can easily attack the user's friends in the social network, which is
characterized by high clustering and short paths. Also, such social interaction-based methods are not sufficiently effective in dealing with legitimate emails from sendersoutside of the social network of the receiver. Golbeck et al proposed an email scoring mechanism based on an email network augmented with reputation ratings. An email is
considered spam if the reputation score of the email sender is very low. Different from these social network based methods, SOAP focuses on personal interests in conjunction with
social relationship closeness for spam detection. There are other approaches not belonging to the above two classes. Due to space limit, we do not present the detailsof these methods.

3.SOAP:SOcial Network Based Bayesian Spam Filter:

3.1 Overview of SOAP :

By using SOAP (Social Aided network Personalized) for spam detection to meet spam detection on social networks we require four components to filter an spam emails using Bayesian filter. Such as Social Closeness Based Spam Filtering, Social Interest based spam filtering, adaptive trust management and friend notification. By using above four components we can easily identify and filter the spam mails by using one by one of SOAP components. By using SOAP based Bayesian spam filters are easily poisoned by clever spammers who avoid spam keywords and add many innocuous words in their emails as well in social networks and it can be avoided using Reverse Tokenization concepts. By using Tokenization Concept in SOAP Architecture can be easily overcome from poisoning attack. It will be very efficient and accurate spam filtering after enhancement of SOAP Architecture using Reverse Tokenization.

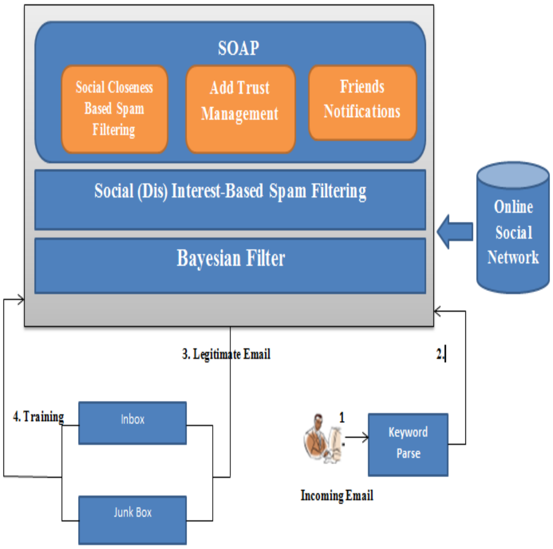

Fig 1.The Architecture of SOAP

3.2 Overview of the Basic Bayesian Spam Filter:

A Bayesian filter has a list of keywords along with their probabilities to identify an email as a spam email or a legiti-mate email. The list is built by training the filter. During training, a user is given a pool of emails, and s/he will manually indicate whether each email is spam or not. We use $P(S)$ and $P(L)$ to denote the probability that an email is a spam email and a legitimate email, respectively. The filter parses each email for spam keywords. It calculates the prob-abilities that a word w appears in a spam email and in a legitimate email, denoted by $P(w|S)$ and $P(w|L)$ respectively. After training, the calculated probabilities are used to com-pute the probability that an email with a particular set of keywords in it belongs to either category. The probability that an email including a word w is spam is:

$$P(S|w) = \frac{P(S,w)}{P(w)} = \frac{P(w|S)P(S)}{P(w)} \tag{1}$$

$$= \frac{P(w|S)P(S)}{P(w|S)P(S) + P(w|L)P(L)}. \tag{2}$$

Then, the probability that an email including a set of keywords W is spam is:

$$P(S|W) = \frac{P(S,W)}{P(W)} = \frac{\prod_i P(w_i|S)P(S)}{\prod_i P(w_i|S)P(S) + \prod_i P(w_i|L)P(L)}. \tag{3}$$

The Bayesian filter sets a threshold, denoted by T. If an email's parsed keywords are W and $P(S|W) \geq T$, then it is spam. Otherwise, it is considered as legitimate.

3.3 Social Closeness-Based Spam Filtering:

When a person receives an email from another socially close person, the email has a low probability of being spam unless the email sender's machine is under an impersonation attack. Thus, the social closeness between individuals can be utilized to improve the accuracy of spam detection. Note that, in a social network, people treat others differently based on their social closeness. People impose different levels of interest, trust, or tolerance to the emails from others with different social closeness. People with close social relationship are willing to receive emails from each other. On the other hand, receivers may have less interest in or tolerance for emails containing spam keywords from senders that are socially far away. We regard spam as emails that receivers are not interested in. Therefore, we need to differentiate emails from persons with different social closeness. SOAP loosely checks emails between individuals with high closeness and strictly checks emails between individuals with low closeness.

In this section, we propose an algorithm that is used in spam detection to numerically measure social closeness be-tween two persons to determine node closeness values. We use $c(u\,v)$ to denote the weight of a relationship between node $u$and v. Below, we introduce how to calculate the closeness of adjacent nodes and non-adjacent nodes in a social network.

3.3.1 Node Closeness

In a social network, more relationships between two adjacent nodes make them closer. Thus,

$$C(u,v) = \sum_{i=1}^{n} c_i(u,v), \tag{4}$$

where$n$ is the number of relationships between u and v, $c_i(u\,v)$ is the relationship weight of the i^{th} relationship.

For example, if node $A$is D's father and $E$is D's best friend, $A$is unlikely to send spam to E. The closeness transitivity should capture three properties in order to correctly reflect the social relationship.

Property 3.1

Closeness propagation property. The closeness between node u and node k_i exponentially decreases as their distance in-creases. As shown in Fig. 2, it can be illustrated by $C(u\,k_i) = C(u\,k_1).\varepsilon^{i-1}$, where $\varepsilon<1$ is not necessarily a constant.

Thus, the more hops that exist between node$u$ and node k_i, the less closeness between them. The closeness value is de-creased to an extremely small value when the distance ex-ceeds 3 hops. This relationship has been confirmed by other studies.

Property 3.2

Weakest link property. The weakest link in a social path (not necessarily a disjoint path) is the direct link between adjacent nodes that has the minimum closeness, denoted by $\min_{1\le i\le n} C(k_i k_i+1)$. The closeness between two non-adjacent nodes$u$ and v is upper bounded by the closeness of the weakest link between u and v. That is, for a social network path from node u to node v with n nodes in between, $C(u\,v)<\min_{1\le i\le n} C(k_i k_i+1)$, where node k_i is in the path be-tween$u$ and v.

Suppose the link between adjacent nodes k_i and k_i+1 in the path from u to v that has the smallest closeness value $C(k_i k_i+1)$. Then, $C(u\,v)<C(u\,k_i)<C(k_i k_i+1)$. That is, $C(u\,v)<\min_{1\le i\le n} C(k_i k_i+1)$..

Property 3.3

Closeness accumulation property. The more social paths that exist between node u and node v, the higher closeness they have. Specifically, if node u and node v have p social paths between them, their closeness through p paths denoted by $C(u\,v\,p)$ is

$$C(u,v,p) = \sum_{j-1}^{p} C_j(u,v). \tag{5}$$

We then design a closeness calculation formula that can meet the above three properties:

$$C(u,k_{i+1}.p) = \sum_{j-1}^{p} (C_j(u,k_i)\cdot(C_j(k_i.k_{i+1})/\varphi)^i), \tag{6}$$

whereφ is a scale parameter to control the closeness scale rate in each hop in closeness propagation, and

$$\varphi > \max_{<x<i} (C(k_{x-1},k_x) \cup C(u.k_1)). \tag{7}$$

Equ. (7) indicates that φ is larger than any closeness value between two adjacent nodes in the path from u to v, which ensures that $C_j(k_i\,k_i+1)/\varphi<1$ in Equ. (6). Therefore, for each social path$j$from node u, the social closeness value $C(u\,k_i+1)$ decreases exponentially based on i, which meets Property 3.1.

For each social path j, we have:

$$C(u,k_{i+1}) = C(u.k_i) \cdot (C(k_i,k_{i+1})/\varphi)^i. \qquad (8)$$

Since $C(u\,v) = C(u\,k_{n-1}).(C(k_{n-1}k_n)/\varphi)^{n-1}$, we canrecursively get:

$$C(u,v) = C(u,k_{x+1}) \cdot \left(\prod_{x+1 \le i \le n-1} (C(k_i,k_{i+1})/\varphi)^i \right)$$

$$< C(u\,k_{x+1}). \qquad (9)$$

Suppose

$$C(k_x,k_{x-1}) = \min_{1 \le i \le n} C(k_i.k_{i+1}). \qquad (10)$$

From Equ. (6), we can get:

$$C(u,k_{x+1}) = C(u,k_x) \cdot (C(k_x,k_{x-1})/\varphi)^x$$

$$< C(u\,k_x).(C(k_x\,k_{x+1})/\varphi) \qquad (11)$$

$$= (C(u.k_r)/\varphi) \cdot C(k_x,k_{x+1}).$$

3.2.1 Distributed Closeness Calculation Algorithm

In a social network, each person has a friend list. Based on the social relationship of his/her friends, the closeness values with adjacent friends can be calculated. Most current social networks have a central server to store all individuals' information in the social network. However, such a centralized method may generate a single point of failure, and hence is not scalable. We propose a distributed algorithm as an alternative for the closeness calculation. In the algorithm, a source node sends a query message with a specified TTL along the FoF links. Upon receiving the message, an intermediate node decreases the TTL by 1, inserts its closeness values with its neighbour's into the message, then forwards it to all its neighbour's. The process continues until the TTL becomes 0. Then, the destination nodes directly sends the message back to the source node. Subse-quently, the source node retrieves all closeness values of the nodes in the path to the destination; it can then calculate its closeness with each of node using Equ. (6).

Thus, each node should collect the closeness information of the nodes within a certain distance from itself. Hence, we can set $\mathrm{TTL}=3$ for two reasons: (1) Sending the message along more hops produces high overhead, and (2) Property 3.1 indicates that closeness decreases exponentially. The close-ness value is decreased to an extremely small value when the distance exceeds 3 hops. Therefore, the source has very low closeness to the nodes far away from itself, and the emails from these nodes should be strictly checked. Algorithm 1 shows the pseudo code of this distributed closeness calculation algorithm. The number of message transmission hops in information collection from one node is $O(n^3)$, where n denotes the number of neighbour's of a node.

Algorithm 1 Distributed closeness calculation algorithm.

```
Send a query message with TTL
if Receive a response from destinations then
    Calculate its closeness with each node using Equ. (6)
end if

if Receive a query initiated by node $i$ then
    Insert its closeness with node $i$ to the message
    $\mathrm{TTL}=\mathrm{TTL}-1$
    if TTL¿0 then
        Forward the message to its neighbours
    else
        Send the message to node $i$
    end if
end if
```

3.2.2 Integration with Bayesian Filter

In the Bayesian filter, each of an email's keywords is weighted to show the probability that an email containing the keyword is spam. checked. The keywords tuning function is:

$$P(S|w) := \begin{cases} P(S|w)e^{-\varphi_f\cdot(C(u,v)-\varphi_l)}, & if\ C(u,v) \geq \varphi_l, \\ P(S|w)\xi(\xi \geq 1), & if\ C(u\ v)<\varphi_l. \end{cases} \quad (13)$$

where $P(S|w)$ is the weight of a keyword, φ_f is a scale parameter to adjust the decreasing rate of $P(S|w)$, and φ_l is a location parameter to determine the origin for exponential decreasing. If $C(u\ v)=\varphi_l$, then the weight is not changed. If $C(u\ v)\geq\varphi_l$, then $P(S|w)$ is decreased by a factor $e^{\varphi_f\cdot(C(u,v)-\varphi_l)}$. If $C(u\ v)<\varphi_l$, then $P(S|w)$ is increased by a factor $\xi(\xi \geq 1)$. ξ can be adjusted by users with different accuracy requirements. Higher ξ requires the email to have a higher probability to be regarded as spam. ξ normally is set to be 1 in order to reduce false positives.

3.3 Social Interest-Based Spam Filtering

The social interest-based spam filtering component aims to make SOAP personalized in order to increase the spam detection accuracy. It is actually a content-based spam detection method. By matching the keywords in an email with the email receiver's social interests and disinterests, SOAP increases and decreases the probability of these keywords to be spam, respectively.

3.4 Node (Dis)Interest Inference

SOAP relies on a rule-based inference system to infer each user's (dis)interests. The inference system has three components: profiles, inference rules, and inference engines. The profile component is a database containing all useful facts parsed from the user' profile in the social network including interests, occupations, and affiliations. The inference rules component contains all the rules that are used for the inference of (dis)interests. Such rules can be rational reasoning based on non-monotonic logic or common sense such as "Most of the birds can fly". The inference engine component determines the applicability of the rules in the context of the current profile, and selects the most appropriate rules for the inference. Fig. 7 shows an example of the rule-based inference method. All the facts are built into a fact database. Numerous rules are made for the inference engine based on the non-monotonic logic.

3.4.1 Integration with Bayesian Filter

SOAP is a personalized spam filter since it considers individual (dis)interests in spam detection.

For a spam keyword within the email receiver's interests, its weight is tuned by:

$$P(S|w_{interest}) := P(S|w_{interest}) \cdot e^{-\rho_I}. \qquad (14)$$

where $w_{interest}$ is the spam keyword in interests and ρ_I is a scale parameter. As ρ_I increases, $P(w_{interest})$ decreases. There-fore, the probability that the email is considered to be spam decreases. As a result, emails within a receiver's interests usually will not be regarded as spam. Therefore, SOAP can reduce false positives in traditional spam filters that lack the personalized feature.

If a spam keyword matches the disinterests of the email receiver, the weight of the keyword is adjusted by

$$P(S|w_{disinterest}) := P(S|w_{disinterest}) \cdot e^{\rho_D}, \qquad (15)$$

Where, $w_{disinterest}$ is the spam keyword in the email receiver's disinterests.

3.5 Adaptive Trust Management

If an impersonation attack occurs, means that if an unknown user, uses others identity to send spam, in this scenario, SOAP reduces the weight of the result if it is from the friend list of the email receiver. And also the closeness is reduced and if the closeness goes below the threshold value, then the email sender will be added in blacklist.

3.6 Friends Notification Scheme:

Due to high clustering in social networks, once a spammer compromises a node in the social network in an impersonation attacks the spammer quickly send spam to close friends of the compromised node in social networks. In addition, as user j and the reporters are socially close, user j contacts the reporters offline (e.g., through telephone) to inform them to check if their computers are compromised. The user j report to the administrator about the accuse. If a node receives an email from node j in its blacklist, rather than adjusting the weights of keywords in the email based on closeness, the node increases the weights of these keywords:

$$P(S|w) = P(S|w)\xi(\xi \geq 1). \qquad (20)$$

Algorithm 2: The process of the spam detection in SOAP.

```
for each incoming email $e$ do
    K = parsed keywords in the email
    Retrieve the weights of these keywords in Bayesian filter
    //closeness-based weight adjustment when the email sender is not in the blacklist of the email receiver
    if sender is not in the blacklist of receiver then
        //calculate the social closeness between sender and receiver and update the weight of the keywords
        Calculate C(e.sender,e.receiver) based on Equ. (6)
        Adjust every keyword's weight based on Equ. (13)
    else
        //weight adjustment when the sender is in the blacklist of the email receiver
        Increase every keyword's weight based on Equ. (20)
    end if
    //execution of interest-based weight adjustment
    for each keyword $k$ in K do
        if $k$ matches the interests of e.receiver then
            Decrease $k$'s weight according to Equ. (14)
        end if
        if $k$ matches the disinterests of e.receiver then
            Increase $k$'s weight according to Equ. (15)
        end if
        Calculate the weight of $e$ according to Equ. (3)
        Email classification
        //update trust value of the email receiver to the email sender based on the classification result
        Update trust value on e.sender according to Equ. (19).
        if trust value of e.receiver on e.sender is less than $T_t$ then
            //send a notification message to inform friends with-in 2 hops ine.receiver'ssocial network
            Send friend notification message with $TTL = 2$
        end if
    end for
end for
```

CONCLUSION:

By using SOAP(Social Aided Network Personalized) for spam detection on social networks we requires four components to filter an spam emails using Bayesian Filter. Such as Social Closeness Based Spam Filtering, Social interest based spam filtering, adaptive trust management and friend notification components. By using SOAP based Bayesian Spam Filters are easily poisoned by clever spammers in order to overcome these we propose an concept Reverse Tokenization Concept by enhancement of SOAP Architecture using Reverse Tokenization.

REFERENCES:

[1].Reference Book "The Spam Evolution" author by ASSEM JHAKAR

[2].Reference Book "SPAM ASSASSIN" author by ALAN SCBWARTZ

[3].Reference Book "EMAIL HACKING" author by ANKIT FAIDA

[4].Reference Paper "Personal Email Network:An Effective Anti-Spam Tool".

[5].Spam Filtering Algorithm(Bayesian Algorithm)

[6]."Zombie," http://en.wikipedia.org/wiki/Zombie_computer, 2014.

[7].C. Binzel and D. Fehr, "How Social Distance Affects Trust, and Cooperation: Experimental Evidence from a Slum," Proc. Economic Research Forum (ERF), pp. 99–106, 2009.

[8].M. Uemura and T. Tabata, "Design, and Evaluation of a Bayesian-Filter-Based Image Spam Filtering Method," Proc. Int'l Conf. Information Security and Assurance (ISA), pp. 46–51, 2008.

[9].M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian Approach to Filtering Junk E-Mail," Proc. AAAI-98 Workshop Learning for Text Categorization, pp. 55–62, 1998.

[10].X. Carreras, L. Marquez, and J. Salgado, "Boosting Trees for Anti-Spam Email Filtering," Proc. Recent Advances in Natural Language Processing (RANLP), pp. 58–64, 2001.

[11].P. Haider, U. Brefeld, and T. Scheffer, "Supervised Clustering of Streaming Data for Email Batch Detection," Proc. Int'l Conf. Machine learning (ICML), pp. 345–352, 2007.

[12].J.A.K. Suykens and J. Vandewalle, "Least Squares Support Vector Machine Classifiers," Neural Processing Letters, vol. 9, no. 3,

[13]..R. Brachman and H. Levesque, Knowledge Representation and Reason-ing. Morgan Kaufmann, 2004.

[14]."Increasing and Decreasing Functions," http://www.mathsisfun. com/sets/functions-increasing.html.

[15].O. Boykin and V. Roychowdhury, "Personal Email Networks: An Effective Anti-Spam Tool," Arxiv Archive, 2004.

[16].H. Yu, M. Kaminsky, P.B. Gibbons, and A. Flaxman, "SybilGuard: Defending against Sybil Attacks via Social Networks," Proc. ACM Special Interest Group on Data Communication (SIGCOMM),

[17]. Spam Assassin, http://spamassassin.apache.org/, accessed in Mar. 2014.