Int. J. Communications, Network and System Sciences, **2013, 6, 88-99**
http://dx.doi.org/10.4236/ijcns.2013.62011 Published Online February 2013 (http://www.scirp.org/journal/ijcns)

Efficient Spam Filtering System Based on Smart Cooperative Subjective and Objective Methods*

Samir A. Elsagheer Mohamed[1,2]
[1]College of Computer, Qassim University, Qassim, KSA
[2]Electrical Engineering Department, Faculty of Engineering, Aswan University, Aswan, Egypt
Email: samhmd@qu.edu.sa, samirahmed@yahoo.com

Received September 17, 2012; revised January 16, 2013; accepted January 25, 2013

ABSTRACT

Most of the spam filtering techniques are based on objective methods such as the content filtering and DNS/reverse DNS checks. Recently, some cooperative subjective spam filtering techniques are proposed. Objective methods suffer from the false positive and false negative classification. Objective methods based on the content filtering are time consuming and resource demanding. They are inaccurate and require continuous update to cope with newly invented spammer's tricks. On the other side, the existing subjective proposals have some drawbacks like the attacks from malicious users that make them unreliable and the privacy. In this paper, we propose an efficient spam filtering system that is based on a smart cooperative subjective technique for content filtering in addition to the fastest and the most reliable non-content-based objective methods. The system combines several applications. The first is a web-based system that we have developed based on the proposed technique. A server application having extra features suitable for the enterprises and closed work groups is a second part of the system. Another part is a set of standard web services that allow any existing email server or email client to interact with the system. It allows the email servers to query the system for email filtering. They can also allow the users via the mail user agents to participate in the subjective spam filtering problem.

Keywords: Anti-Spam System; Objective Spam Filtering; Cooperative Subjective Spam Filtering; Web Application Web Services

1. Introduction

With the rapid growth of the Internet, email has become one of the most common media for us to exchange information. More and more people depend on it to communicate. Spam, unsolicited, undesired, bulk email (or junk email), has been a significant security issue for computer users and a massive waste of time, disk spaces, and network bandwidths. Spam has been recognized as problem since 1975 [1]. Spam is usually characterized as unsolicited commercial or bulk email, is sent in large numbers and repeatedly to individuals. According to the statistics from ITU (International Telecommunication Union), 70% to 80% of the present emails in Internet are spam, which has become a worldly problem to the information infrastructure [2,3].

Spam is a multifaceted phenomenon and therefore very complicated to address. This phenomenon is probably one of the biggest challenges the Internet will have to face in the immediate future [4]. A spammer sends a large number of messages to many different recipients who have not requested the content. (Interestingly most spammers do not care whether a particular addressee receives the message; they merely seek to get a sufficient percent of their postings delivered to some of the addressees.) Spam can conform to Internet technical standards and can contain no technical differences from legitimate—desired—messages.

Moreover, the association of spammers with hackers and virus writers poses a very real threat to the Internet security and availability. About 8 years ago, spam was sent by spammer's own e-mail servers. Approximately 45% - 60% of spam is now sent from compromised systems distributed over the Internet [5-8]. Spam relaying increases the distribution base and at the same time eludes and overwhelms spam detection systems [9].

To fight the spam, there exist many objective measures that can effectively limit the spam email impact on the end users (subjects). Traditional anti-spam techniques include the Bayesian-based filters [10-13], rule-based Scoring Systems [14-16], DNS MX Record Lookup and Reverse

*This work is supported by a grant from King Abdulaziz City for Science and Technology (KACST) via the National Science and Technology Plan, Al-Imam Muhammad bin Saud Islamic University, KSA. Project number: 08-INF438-8.

Lookup Systems [17], DNS Realtime Blackhole List (DNSRBLs) or IP Blacklists [18]. However, these objective measures fail in many situations and they can have a level of accuracy. They suffer from the false positive and the true negative problems. Where a non-spam email can be classified as spam email or the filter can classify a spam email as legitimated email (non-spam). The spam problem is very complex one. An ideal anti-spam solution cannot be found as the spammers constantly try to invent new tricks and mechanisms to bypass the objective anti-spam measures so that their spam emails be classified as legitimated emails.

Emails are sent to human (subjects). Thus, only humans can accurately identify spam emails based on the content. As a result, subjective methods are better than the objective methods for content-based filtering. If we can have a group of qualified users (subjects) to monitor and read all the incoming emails, sure this will be the ideal solution for spam filtering. However, for many reasons this solution cannot be applicable because of the privacy, the scalability, the huge volume of emails that is sent by the spammers, etc.

As we highlighted, objective measures alone cannot fight the spam problem. In addition, no pure subjective measures can be implemented effectively to work in real time. Subjective measures can check only the message content (the email body), but in no means they can check other spamming metrics like a forged sender email. Objective measures can accurately and rapidly eliminate in real time up to 70% of spam emails based on known spamming techniques. The only time-consuming and inaccurate objective measures are those filtering the spam based on the body contents.

In this Paper, we propose a new technique for the spam filtering problem. It consists in combining both subjective and objective methods. This can effectively be accurate and reliable spam filtering solution. If subjective measure is devised in such a way to maintain the privacy of the users, it can cope with almost all the newly devised spammer's tricks that can pass through the objective measures. In addition, they can compensate (correct) the false-positive objective evaluations.

In addition, we have developed a complete system that is based on the proposed technique. The system consists of several software components. The first is a web-based email client application that allows users to access their email accounts from a central location. It allows them to benefit from combining both the traditional objective spam filtering and the smart subjective spam filtering. Users are the main players in the subjective spam filtering, thus through this application, they can do that in a smart way. Another application is also presented that can be used for enterprises and companies to allow them to add all their users to the systems and make them benefit from the offered solutions as well as participate in the subjective spam filtering problem. A web service is developed that allows any email server to request from the system the subjective evaluation of the newly arrived emails. Another developed web services allow the users of any email user agent to participate in the subjective spam filtering. All these open source software are presented in the paper.

The rest of this paper is organized as follows: In Section 2, the related works and research efforts are given. The description of the proposed technique is provided in Section 3. The developed Email Client Application that is using the proposed technique is described in Section 4. In addition, the cooperative subjective application is described in Section 5. In Section 6, we provided a set of developed web services that allows any other email application to interact with the developed system. Finally, the conclusions and the future works are given in Section 7.

2. Related Works

Various techniques [3,19-21] on automatically detecting or filtering spam emails have been proposed by building comprehensive databases for blocking emails whose addresses have been reported as black-lists or whose message bodies contain specific words or phrases defined as threatening terms, etc. The most famous approach based on the statistical filtering is the Bayesian filter [11-13], which is based on the Bayes' theorem. This theorem states that the probability that an email is spam, given that it has certain words in it, is equal to the probability of finding those certain words in spam email, times the probability that any email is spam, divided by the probability of finding those words in any email. The drawback of the statistical filtering techniques is the processing time: the time required to process an email and to end up if it is a spam or not. Another problem with this kind of filtering is that it cannot fight against the new tricks of the spammers, like changing vocabulary, introducing the most recognizable terms or adding a relatively high number of random words, miss spell words, adding numbers and symbols in the middle of the word or the phrase, etc.

Another new direction for the spam filtering problem which becomes now more popular is the Collaborative filtering [11,22] approach, which is a generic approach used to describe any type of filtering in which a network of intelligence is used to identify spam. A collaborative intelligence network can take many forms, such as a collection of lexical data in which characteristics of spam are described. In most cases, collaborative networks take advantage of the misfortune of others receiving spam to build better intelligence for future filtering efforts. This idea is implemented in Vipul's Razor [23], which is a distributed, collaborative, spam detection and filtering network. Through user contribution, Razor establishes a distributed and constantly updating catalogue of spam in propagation that is consulted by email clients to filter out known spam. Detection is done with statistical and randomized signatures that efficiently spot mutating spam content. The drawback of collaborative filtering is the participating community itself [11,22]. In large communities, there is either a high maintenance loop or, if automated, a high risk of false positives. Larger networks with high maintenance loops generally experience latency in updating their databases, in the same way that blacklisting has a propagation delay. Automated networks run the risk of false information being propagated or possibly injected by a malicious party. Smaller networks are generally more accurate and more real-time but lack the ability to cover a wide pool of fresh inbound spam.

The existing anti-spam techniques are passive, that is to wait for the spam email and to do all the best to identify the spam email and move it to the Spam folder of the user or simply drop it. However, in [24], a completely different approach is proposed for the spam problem. An active technique that attack the spammers' resources. By grapping all the links from an incoming spam email and building an application that periodically download the contents from these links, the spammers resources (their servers and network bandwidth) will be wasted answering the request from this application. Once this application is widely deployed and used by many users, it will act as a massive Distributed Denial of Service (DDoS) attack on the spammers' resources. For more details see [24].

Another direction to fight the spam problem is the "machine learning" techniques. Machine learning is the ability of a machine to improve its performance based on previous results. The key feature of machine learning is that based on existing dataset training has to take place in order to learn to distinguish between spam and non spam emails. Feature extraction is a major and critical part of machine learning. It does not work on the raw content of the email, but extracting the features then feed them to the tool that can tell wither this email is spam or not based on the experience it has during the training or the learning phase. Moreover, the dataset has to be divided into three parts, one of the training, another for testing and the third for the cross validation. Once operational, it can give false positive or false negative. These false predictions can be recycled to dataset to improve the accuracy. Of course, a retraining algorithm and update criteria have to be done regularly to deal with the new tricks of the spammers.

Some of the existing machine learning based techniques are: rule learning, decision trees [25], support vector machines [26,27] or combinations of different learners [14,28]. The basic and common concept of these approaches is that using a classifier to filter out spam and the classifier is learned from training data rather than constructed by hand.

In [29], ant colony optimization (ACO) algorithm is proposed to detect spam in host level. In this work, the host link structure is first constructed by aggregating hyperlinks over web pages. Training and validation is required by ACO to distinguish between good (non spam) and bad (spam) web pages. The authors in [10] propose a social network Aided Personalized (SOAP) as a spam filter. SOAP does not focus on parsing key-words (e.g, Bayesian filter) or building blacklists, it exploits the social relationship among email correspondents to detect the spam adaptively and automatically. From the machine learning viewpoint, spam filtering based on the textual content of e-mail can be viewed as a special case of text categorization, with the categories being spam or non-spam [14,15,30].

Spam can also be sent embedded in an image [31,32] or in a PDF documents (*i.e.* the text of the message is converted to an image and sent by email or MMS to users). This kind of spam is very hard to detect as the spam detecting software has to decode the image and convert it back to text using Optical Character Recognitions (OCR) to identify if the content is a real spam or just a normal image. Only the human can read the text on the image. This trick of the spammers is one of the hardest to solve. Many research efforts concentrate to image spam detection as the ones given in [33,34]. In [33], a framework for filtering image spams by using the Fourier-Mellin invariant features is presented. Fourier-Mellin features are robust for most kinds of image spam variations. A one-class classifier, the support vector data description (SVDD), is exploited to model the boundary of image spam class in the feature space without using information of legitimate emails. However, these approaches have many limitations that limit their applicability in real time. They require too much computational power to process all the pixels in the image, usual many iterations have to executed with complex algorithms. Spammers use several techniques to make the text recognition almost impossible. They are language dependant techniques. There is no generic OCR for all the languages. This make these approaches ineffective for realistic email applications to run in real time due to the huge volume of emails coming to the server and the excessive processing complexity for each image embedded in an email.

3. The Proposed Spam Filtering Technique

As stated in the Introduction, it is known that subjective measures are better than the objective measures for spam filtering. To fight the spam, there exist many objective measures that can effectively limit the spam email impact on the end users (subjects). However, these objective measures fail in many situations and they can have a level of accuracy. They suffer from the false positive and the true negative problems. Where a non-spam email can be classified as spam email or the filter can classify a spam email as legitimated email (non-spam). The spam problem is very complex one. An ideal anti-spam solution cannot be found as the spammers constantly try to invent new tricks and mechanisms to bypass the objective anti-spam measures so that their spam emails be classified as legitimated emails.

On the other hand, if we can have a group of qualified users (subjects) to monitor and read all the incoming emails, sure they will be the ideal solution for spam filtering. However, for many reasons this solution cannot be applicable because of the privacy, the scalability, the huge volume of emails that is sent by the spammers, etc.

As we highlighted, objective measures alone cannot fight the spam problem. In addition, no pure subjective measures can be implemented effectively to work in real time. Subjective measures can check only the message content (the email body), but in no means they can check other spamming metrics like a forged sender email. Objective measures can accurately and rapidly eliminate in real time up to 70% of spam emails based on known spamming techniques.

Combining both subjective and objective methods can effectively provide better result. If subjective measure is devised in such a way to maintain the privacy of the users, it can cope with almost all the newly devised spammer's tricks that can pass through the objective measures. In addition, they can compensate (correct) the false-positive objective evaluations.

It is known that the spammers send the same email to thousands of users. In many situations, spammers send the same email every day or week. This led us to think in a way to devise a smart subjective measure so that the spam email can be viewed only by the first few email users. If, for example, four users agree that the email is spam, the system will automatically moves thousands of same-email spam to the users' spam folders. That means that only the first few people will view the spam email and all the rest of the users who received the same email will not have to view it.

However, things are not easy as that. Spammers use special bulk mailing systems and mail merge so that the same email message will have different attributes. For example, they may change the Subject of the email per user. Some spammers include HTML image or Hyperlink links to track the users. Thus, for objective measures they are completely different emails and may seems to be legitimated because they are targeted to specific user.

There is a great challenge to eliminate the customizations the spammers inject into a spam email and to correlate them objectively. When the subjects (users) view the same-email spam copies they will see them as identically. But how to identify and remove the customizations attributes objectively to correlate this same-spam email is really very difficult.

We have devised an approach that is very effective in this matter. To protect the privacy of the user (the email receiver), and to try to correlate all the emails sent from the same spammer to thousands of users, we proceed as follows. Simply remove everything from the email message except the Sender Email address and the filtered message body. In the filtered message body, we keep everything (text) except the hyperlinks (for the images and the links). For these links, we keep only the hostname of the URL. The rest of the URL is removed. Traditionally, spammers tracks the users using the GET parameters which are included in the URL after the "?" (See Examples 4-6 in **Table 1**). However, modern URL rerouting techniques put the parameters in the folder parts of the URL. Example of such techniques are given bellow (Examples 1-3 in **Table 1**). The parameters in these URLs contains information about the target user (usually each email address has a database record in the spammers' servers to know if he/she is active or not, the links he/she is interested in, the location of the user, etc.) In addition, it contains the spam campaign identifier hosted by the spamming infrastructure.

Therefore, in order to be able to correlate the same spam message, we have to remove all the link customization from the body text. This can be done by removing the tracking information in the link. Thus, we keep only the host part of the URL and remove the rest of the URL.

All the other email messages fields can be modified per user by the spammer mailing system. Example of these fields are the Received field (from which mail server at which time), the Receive Date, the Subject, Message-ID, Content Length, To Field, etc.

The proposed technique for spam filtering is as follows:

- An application has to be developed that allows any user to access all the email accounts possessed by the user. This application must have a central database to store the user's information and the email message information.
- Apply the user defined rules and filters to the newly arrived emails. This can include the use of the white, gray, and black lists. It can also combine several rules like "if the subject contains these words and the recipient fields contain some email addresses, and then move the email to this specific folder". The rule or filter action can be to leave the new email in the Inbox, move it to another folder, or simply to delete it.

Table 1. Examples of real tracking URL.

Example #	Link (URL)
1	http://ninjaasteroid.com/52646271726908237236 3.UEOVHRA.YTH4D92/197417/141514/3024-006-2-5/b01491b120f69f9534c6ba8ce7106851/euopfedb.5IS6GKTK
2	http://fandragontastic.com/12081088282704739366.ABKPNZE.CVNOL3L/591323/140557/3472-006-2-5/3b6e615291cb5839bb2928ac038fcbae/luy56abk.D639K5PO
3	http://click.countrybaby.net/PmgEoCSeXsETlZMEJXUwDzRdWKfrnQEWRijeuRyCDmHRBmlVlrqmdWGihGzV?&n=1585615822&h=a9757f2e2a4885267e6c90e108572ebbf32d26ef
4	http://engine.gtsmobidistributed.com/www/delivery/ck.php?oaparams=2__bannerid=38413__zoneid=1633__cb=d25aa30b4c__oadest=http%3A%2F%2Fwww.mporn.com%2F%3Futm_source%3Dgtsredirects%26utm_medium%3Dcpm%26utm_campaign%3Dmobile_redirects
5	http://click.vegasvapor.net/waqCcEoxTjuEeSbiNLIzRGhJlmKEGRzglvpDRYsnRJCPaEcNMdWGEEWGR?&n=1585581499&h=7f5415cf181e3cf08cbcac6837aa1dc534916258
6	http://www.x3track.com/click.track?CID=209129&AFID=21845&ADID=751949&SID=

- On the arrival of any email message, first the traditional spam filtering techniques are applied to filter these emails. No content-based objective filtering is required because they are time consuming and inaccurate. We are analyzing the spam emails since three years. About 70% of the spam emails can be accurately identified using these traditional spam filtering techniques.
- When the user open the Inbox folder of any email account and open any received email that is not filtered using the previous steps, the user can judge the email content whether it is spam or not. In the case when the user judge that the email content is spam, he/she simply clicks on a button labeled “This is a spam”. The application takes this into account and creates (if not already voted by another user) a record for this email. If several users agree that this email content is a spam, then all the other similar emails will be considered as spam. Matching (the meaning of similar) is based on the neutralized body text explained before.
- In this case only the first few users are forced to see the spam emails in their Inbox. For the rest of the users the email is placed in the Spam folder. Moreover, it is known that the spammers periodically send the same spam email (every day, week or month). Thus, the users will not be annoyed by these spam emails that cannot be cached by the objective spam filtering methods.
- On the other hand, if the user opens the Spam folder and finds that any email is not a spam, he/she clicks on a button “This is not a spam”. Then in this case, the application decreases the “Spam weight”. If the “Spam weight” is bellow some value, any similar email is considered as legitimate and placed in the Inbox. If the “Spam weight” is greater than certain value, all the similar incoming emails are considered as spam. If the “Spam weight” is between these thresholds, the email is placed temporally in the “Gray folder”. Periodically, the contents of “Gray folders” have to be reevaluated by the system to decide if the emails are spam or not.

To avoid the attacks to the system by the malicious users, a confidence factor has to be associated to each user. If the user makes correct voting (as the average votes of the email), the confidence factor is increased, otherwise it is decreased. If it is bellow some threshold, then the user’s voting will not be considered.

4. Email Client Application

In this Section, we present an Email Client Application (ECA) that we developed that can in addition to the functions offered by the classical email systems offer the proposed anti-spam solutions. We started from the fact that Internet users already have their email addresses (email accounts) and they are using them from long time ago. Any solution forcing the users to have different email address might fail as people do not prefer to change theirs as they are known by their correspondent users. Another fact is that many Internet users have more than one active email accounts (one for the work, one private, and one for web form registration or signing up, etc.). Thus, it is highly preferable for such users to have all their email accounts be read from a single web-based applications so that they can reach all these email accounts and manage them from a centralized location from anywhere in the world at any time without having to install special software.

We have successfully developed a web-based application that combines both the traditional objective measures and a clever subjective measure. Email Client Application (ECA) that we have developed implements the methodology provided in Section 3 and hence can be used by any user to access his email account (s) from a single location from the web browser. It in addition integrates the anti-spam features for all the users’ email boxes provided that these accounts have Mail Access

Protocol (IMAP) or Post Office Protocol (POP3) access. It includes both the objective (traditional) spam filters as well as a clever subjective spam filtering methodology. This developed Application included the following features.

- Open source software. All the source codes of the application are provided. It can be obtained directly from the system website from the following link (http://aspam.asites.org/packages.zip). This allows any developer to add any feature to the existing one and use it or send us the patch so that we include it in the system.
- To pace the cloud computing era, we developed the system as web application that can be accessed from the Internet using any web browser. It can be accessed from anywhere in the world, at anytime, from virtually any device (Desktop computer, laptop, Smartphone, etc.).
- The main power of the system consists in the use of the traditional objective spam filtering combined with subjective approach. In such a way, the system will be become more and more reliable and effective as the number of users increases. Up to 70% of the spam emails can be filtered using the traditional spam filtering techniques. However, the rest of the cases, especially the content have to be done based on the end-user contributions. Once the end-user opens a new email he can click on a button to say if it is spam or non spam as well as the spam category of the message. This information will help the system to automatically move any quasi-similar email to the user spam folder for the rest of the users.
- The system is developed using the latest MS web development technology such as MS. Net framework v4.0, ASP.NET, VB.NET, etc.
- The developed application can be hosted on any web hosting environment having Internet Information Services (IIS) web server and MS SQL server 2008. Any company or a group of users can obtain the application with its source code, host it on their servers if they prefer to work locally (intranet of a company or an enterprise). We provide more sophisticated solutions (the server edition of the system), where companies/enterprises/groups can add their users to our system and manage their users with all the included security measures. For more details about that see the Email Server Application developed in this project Section 5. We have also developed several web services that can allow all the private installations of our system to interact with the global centralized one. For more details about that see the Web Services developed in this solution in Section 6.
- Any user can easily create an account with a minimal sign-up form on the system. The user can then setup the account by providing his/her existing mailbox account. The user can setup unlimited number of accounts he/she owns. The system does not provide any new email account for the user, but the user will provide the access information for his/her existing email accounts. For example, if the user has an email on yahoo, another on Gmail, another on Gmx, another on his work email server, etc.
- When viewing the list of emails in any folder in any email account, the user has two modes or options: the Lightweight mode and the full processing mode. In lightweight mode, no spam analysis takes place; the headers only are fetched form the remote email servers. No data is stored on the system database. This mode may be fast in the beginning because no email body is downloaded nor spam processing. However, the user can have all the other classical features like moving the emails to another folder or deleting the emails. In addition, in this mode, the user can classify any email he/she opens as spam, gray or non-spam. His rating and spam-classification on a message will be taken into account. In the Full processing mode, the unread emails in the Inboxes are thoroughly examined using the objective and the subjective spam filtering techniques. Email contents (except the attachments) are fetched from the server only once, after being processed, they are stored on the system internal database. Viewing the emails using this mode may by slow at the beginning depending upon the number of non-fetched emails. After fetching the emails, reviewing the email folders will be very fast.
- When the user open any email in any account, he/she can classify it as Spam, Gray, or legitimated one. In the case when classified as spam, he/she can have the ability to categorize it. In the case when it is classified as a spam, lengthy processing operations take place and the email moves to the Spam folder.
- The system supports the Mail User Agent (MUA) functionality, where users could compose there emails and sent it via the Mail Delivery Agent (MDA) via the SMTP protocol. To avoid classifying the emails sent from the global system as spam by the recipient emailing systems, the system will not send directly from the system SMTP server, but using the SMTP authenticated email account of the user (set up in the email account setup process).
- The system supports Mail Retrieval Agent (MRA) functionality, where users could retrieve their emails from the email server(s). MRA supported are both Internet Mail Access Protocol (IMAP) and Post Office Protocol (POP3). In the case when IMAP protocol is used, the user can access all the email folders in that account. He/she can access all the old emails, move the emails between accounts, create/delete

email folders, etc. The system supports the access to the built-in folders such as Inbox, Sent emails, Trash, Drafts, Spam. In addition, any user-defined folder is also supported (given that IMAP protocol is used). However, POP3 email accounts can access the Inbox folder only; no folder management can be done. Depending on the user email configuration, the user can access the old email or not.

- The system provides support of all the security and the encryption standards, including the support for Secure Socket Layer (SSL) protocol, Transport Layer Security (TLS), S/Mime, APOP, NTLM/GSSAPI, and FIPS etc. When setting up an email account, using the port number, the system will automatically indentify the security and encryption standard that will be used.
- The system supports all the email servers providing IMAP and POP3 access, where any user who has a mailbox on any server supporting IMAP or POP3, could read that account from the system via IMAP or POP3. In addition, the user could receive all the emails on the system and could also send the email from the system. Examples of such email servers are: MS Exchange, Lotus Notes, GroupWise, IIS SMTP/POP3, IMail, MailEnable, AxiGen, SmarterMail, SurgeMail, MDaemon, Kerio, CommuniGate, hMailServer, Exim, Postfix, Sendmail, Qmail, Courier, Dovecot, Cyrus, Zimbra, Gmail, Hotmail, Yahoo... and any POP3/SMTP/IMAP compliant servers.
- A wide range of filters and actions is available. Each filter can have several rules: Sender email, the recipient (To and CC), the Subject. For each of these fields, the user can select: Start with, contains, end with, or exactly equal. If all the rules apply to any newly arrived email address, there are three actions: Leave it in its folder, delete it or move it to another folder in the same account or to a different email account for the user.
- Any user account has three built-in lists: Black list, Gray list and White list (per user lists). The user can create any other lists (per user lists). There is a whole list management configuration page on the system. User can create, rename and delete any of the user-defined lists. However, he/she cannot modify any of the built-in lists. Each list has an associated folder. On the creation of the list, the user must select the associated folder for it (the user has to select the Email account first, and then he/she has to select the folder from all the available folders under this email account). The user then can have full control over the content of each list. The entries of the list can be a complete email address, a domain address, or any part of the email. On the arrival of any new email, when the system process it, if the sender email address matches any of entry of the list, then that email will be moved directly to the associated folder in the associated email account for that user.
- On the arrival of new email messages, the order of processing will be as follows:
  a) First if the sender email/sender domain/or part of the sender email exists in any of the user lists, then the email message will be moved to the associated folder in the associated email account of the user.
  b) Otherwise, the filter rules are checked. If all the rules are satisfied for a specific filter, then the associated filter action will be executed. Actions may be to move the email to a specific folder in a specific email account or to delete the email completely, to move it to the gray folder or the spam folder of the user.
  c) Otherwise, the email message is checked if it has been reported by any user before as being spam or not. If so, check the *spamwieght* of the previously reported email, if it is greater than the *Spam-Threshold* then move it to the Spam Folder of the user, otherwise move it to the Gray Folder of the user.
  d) Otherwise, apply the objective spam filtering rules (valid DNS, rDNS, DKIM, etc.). If the email pass all the tests, then the email is left in its Inbox folder in that email account, if on the other side one of the tests fails, then the email is moved to the Spam Folder of the user.
  e) Regularly, emails that have been moved to the gray folder automatically are reevaluated, if the *Spamweight* is more than the threshold, then move it to the Spam folder, otherwise move it to the inbox of the user. This is to avoid stalling an email long time in the gray folder and it could be important for the user and not important to the first few users who reported it as spam. If the user find that it is spam, he can report that and the *Spamweight* will increase. This can improve the accuracy and eliminate the need of other users to view a spam email.
- Security is of great concerns. Thus the developed application is secured and proof-tested against the known security attacks such as, SQL injection attack, cross sight scripting, etc.

5. Cooperative Anti-Spam Email Server Application

The second part of the developed system is Cooperative Anti-Spam Email Server Application (CASESA). It is complementary to the Email Client Application (ECA), refer to Section 3. The server can automatically classify any email for any email server that integrates our web services for any incoming email. This is based on the subjective evaluations of the emails carried out by the users of the developed ECA, or by using any email client application with the help of the Web Services that have been developed (See Section 6).

Using the innovative technology named as the web service (part of the cloud computing), we have developed a web service that allows any existing or new email server or email client application to ask for classifying an incoming email message. In addition, for companies or user groups and lists, the system administrator can simply create accounts for the users in the system in one action. After that, all the users can benefit from the services and features offered on the system. In addition, they can participate in the subjective spam evaluation process to enrich the system and to improve the accuracy of the subjective rating.

As said before, it is known that all the existing objective anti-spam filters are not accurate and thus give false positives and false negative classification. The most accurate one is the subjective method. By allowing the user to give his feedback about the received email and by building a huge server application hosting all the accounts of the users, then once a new email is received by any user and opened, if the user subjectively classify it as a spam by clicking on the spam button, then the client application will send this email message to server saying that "this is a Spam".

The server then take this into account and any new message will be checked against the existing database of the subjectively-classified email messages. If it is found, then the server will place the email in the Spam folder of the user. Usually, the spammers send the same email to many users (bulk) but changing only the destination. Thus, in this scenario, only the first user will open the spam email and any other user in the system will not be annoyed by that email with high confidence as the classification is done by human. Our developed system included the following features.

- Opened source software, where all the source codes are available on the project website.
- The developed application can be hosted on any web hosting environment having Internet Information Services (IIS) web server and MS SQL server 2008. Any company or a group of users can obtain the application with its source code, host it on their servers if they prefer to work locally (intranet of a company or an enterprise).
- The system is developed using the latest MS web development technology such as MS. Net framework v4.0, ASP.NET, VB.NET, etc. In addition, we have used the standard Web services to communicate with the system as well be explained in Section 5.
- The system is intended mainly for servicing the email servers to help them automatically obtain objective and subjective email filtering for the users having email accounts on these servers and/or allowing a single person to create and manage all the existing users on these servers. For this reason, a single user must be nominated by the enterprise, the company, the campus, or any establishment to manage the users (adding them to the system, validating them, disabling any one of these users, etc.). We refer to this user as the ServerUser. ServerUser must contact the system Superadmins to create a server manager account for them, upon approval, an account will be created for the ServerUser and an email will be sent to his account to activate his account. We have chosen this methodology to avoid the Spamming attack to the system and to allow the service for the serious people only.
- Once the ServerUser account is created and activated, he/she can log into the system and create client access accounts on the system for all the users on the email serves in that enterprise, the company, the campus, or any establishment. The simplest way for doing that is by composing all the user information in a single Excel file and importing them from the system. We have prepared a template excel file that can be used for that purpose. One very important thing to be noted is that, the ServerUser must enter the Email Server access information (the SMTP host name, the SMTP port number, the email access server type, the email access host name, and the email access port number). This information plus the specific user information imported from the Excel file will be used for creating email client account for every user on the system.
- Upon the creation of user accounts of the users in previous step by the ServerUser, anyone of these users can log into the system (The Email Client Application described in details in Section 3).
- All the features provided in the Email Client Application are integrated on the developed Cooperative Anti-Spam Email Server Application. There is no need to repeat them here, for more details, please see Section 3.
- The system developers can make use of the developed Web Services (cf. Section 6) allow these users to remotely provide their subjective voting on the emails they receive. In addition, the system developers can include these web services to allow their email servers to automatically classify any incoming email for these users as spam or non-spam.
- Each user is assigned a confidence factor to avoid injecting bogus data by malicious users. If he/she reported a spam email with other users, this factor will be increased, otherwise, it is decreased. If the confidence factor of the user was less than a threshold,

then this user is placed on the black list and all his/her future judgments will be ignored.

- The system periodically calculates and updates the Confidence Factor for all the Email Users on the system. As this step is a lengthy process and usually the usual user can vote on around 10 emails daily, there is no need to shorten the period. We have chosen that this process can be done once per day (every 24 hours). In addition, for the optimization reasons, we have chosen to make the processes be executed by the database server as a stored procedure and called by the web server once per day. Again for optimization reasons, only the users who voted on some emails during the current period are processed (their votes are stored on a table during that period), then this table is flushed. The calculation of the confidence factor of the user is done in the database table as a Computed Column based the # of Correct Votes and the # of Wrong Votes of the user using the following formula. The constant 0.000000001 is to avoid the divide by zero problem and to avoid using an additional test condition. If the user provides less than 30% correct votes, then his confidence factor is zero (as if he/she is panned or black listed). Thus all his/her next votes will not be considered. There is no need to manually remove the user from the black list. The user can help him/her self by improving his subjective evaluation. As the script is executed daily, once he/she gives better votes (more than the 30% threshold), his/her opinion on filtering email messages subjectively will be considered weighed to the value of the confidence factor.

```
(case when [NoCorrectVotes]/
(([NoCorrectVotes]+[NoWrongVotes])+
(0.000000001))<(0.3) AND
([NoCorrectVotes]+[NoWrongVotes])>(100)
then (0) else [NoCorrectVotes]/(([NoCorrectVotes]+
[NoWrongVotes])+(0.000000001)) end)
```

- Once an email arrives on any of the Inbox of any of the Email users, if the user opens it and classify it as Spam, in this case, it will be placed on the Spam folder of that user. However, we cannot be sure that this email is Spam (we cannot base our judgment on a single user). As a result, when any user opens his Inbox folder in any email account, the new emails will be processed as follows. If the email is previously rated by any user as Spam, we check the SpamWeight. If it is >4, then the email is moved to the Spam folder, else it is moved to the Gray folder. If not previously rated by any user, the email remains in its folder (after executing the user defined filtering rules). The SpamWeight for each message is calculated based on the Confidence Factor of the voting user. If the user rate the email as Spam, then the SpamWeight is incremented by the Confidence Factor of the voting user, else, the SpamWeight is decreased by the Confidence Factor of the voting user. One problem is that once the user opens the any of the Email folders, the email will be classified based on the previously mentioned rule and it will remain in that state even if its SpamWeight is updated with time by other users. It could remain in the Inbox even if it is Spam and it could remain in the Gray or the Spam folder even if it is legitimate email. To solve this problem, we have designed a stored procedure that executes every 10 minutes on the database server that periodically updates the states of the emails of the users based on the overall user voting's. To avoid updating the status of the emails that have been moved by the user manually, we apply this procedure to those emails that are not moved manually by the user using the isMovedByUser flag.

6. AntiSpam Web Services

We have developed a complete solution consisting in two main components: the first one is the complete Email Client Application (refer to Section 3 for more details) and the second one is the Cooperative Anti-spam Server Application (refer to Section 5 for more details). We know, however, that there are many people who are not willing to migrate to our Email Client Application, even if it has many features. For example, Gmail, yahoo, live, hotmail, enterprise email servers like Exchange Servers, Exim, etc email users are billions and cannot migrate to ECA. Even if the system support the access to these email servers email accounts and their email boxes can be accessed from that system, the end users and even the owners of these email solutions may not prefer to migrate. From another point of view, the subjective anti-spam filtering accuracy depends on the number of the users voting on the viewed email. Even if we have used the confidence factor to eliminate or minimize the unreliable and the malicious users, the more the email voters, the more accurate the results will be.

Because of all these reasons, we have developed some standard web services that can be used by almost any email access client software and almost any email server software. To illustrate the idea, let's assume that a company would like to make use of the spam filtering solutions provided by the developed solution which is installed on the (http://www.aspam.org). In this case, a developer must add a simple code to the email server such that on the arrival of any email, a request to our system will be sent to know if that email is previously received by any users and classified as spam or not. The response from our system to the company email server would contain much other useful information.

The company email server then based on the received response can decide to place the email in the Inbox of the user or to move it to the Spam folder. This action is completely transparent to the end user. Taking this into account, almost all the emails servers can integrate this functionality to their servers to provide the subjective and objective email filtering solution for their end users. This is exactly the first web service that we have developed. It is named as ***ClassifyNeutralEmail***. We know that the privacy is of great importance for the users. Additionally, not all the email information is required by the system to be able to match the email in the database of the system. After several studies, we concluded that only the email sender and the message body are needed (as described in Section 5). Other information such as the recipients (To and CC fields), the subject, the receive time, the sent time, MessageID, etc. can be variable for the same spam email from user to user. In addition, the spammers can change these fields per users. Thus to increase the accuracy and to maintain the privacy of the users, no need to send these values and no data will be stored on the system on the recipient of the request from the servers. Regarding to the Body contents, it has to be neutralized (as described in Section 3).

On the other hand, what about the existing email clients applications such as Ms outlook, Thunderbird, Web-based email access clients (e.g. Gmail, yahoo, Outlook Web Access, etc), and the Smartphone email access programs (Android, Iphones, IPad, Nokia, Windows Mobile, etc)? Billions of users are using these email access programs and they liked it and cannot change to another solution easily. We can hit two birds by a single stone using the following procedure. We have developed several standard web services that can be directly integrated with almost any existing email access clients (Mail User Agents) to allow the billions of these end users to participate into the subjective email filtering problem.

The developers can simply add two simple buttons on this Mail User Agents (MUA) if they do not exist before. The first one will be shown when the user opens and email found in the Inbox labeled “This is a Spam message”. When the user open an email and decide that this is a spam and he/she clicks on that button, a call to ***reportSpamEmail*** web service will be executed. The message will be executed on the remote server (Our system). An intelligent search in the database will be performed to find if the message is reported previously by any user. If it is found, then update the message record by incrementing the number *ReportedAsSpam* counter and increase the *SpamWeight* by the Confidence Factor of that user. If the email is not found in the database, then a record for it is created and the voting counters for it are initialized.

Similarly, when the user is viewing the contents of his/her Spam folder on the MUA, he/she can decide that an email is not a spam. Thus, a developer can update this MUA software by adding a button labeled “This is not a spam” as an option. When the user clicks on that button a call to the web service named ***reportNonSpamEmail*** will be executed. The message will be executed on the remote server (Our system). An intelligent search in the database will be performed to find if the message is reported previously by any user. If it is found, then update the message record by decrementing the number *ReportedAsSpam* counter and decreasing the *SpamWeight* by the Confidence Factor of that user. If the email is not found in the database, then a record for it is created and the voting counters for it are initialized.

For security reasons, all the implemented web services cannot be used anonymously. Reporting a spam message or reporting non-spam email message must be done using a previously registered user. This is to avoid the attacks from the spammers or the malicious users. In addition, this is a must for calculating and updating the confidence factor of the reporting user. It may be thought that forcing all the users to create an account on the system can be difficult task, but using the Cooperative Anti-spam Server Application (refer to Section 5, Page 17 for more details), all the users accounts can be created in a single operation, by importing their details from a single excel file.

For the performance and availability issues, and to be able to service millions of users and thousands of email servers and email access clients, the system must be installed on a server farm.

Another factor is the intelligent technique in the search in the database. The database tables could contain millions of records. Thus, for every incoming email a search must be done on the entire database. Even using the Indexing technique cannot help because that can make inserting and updating very slow. We are using a method which consists in using Hashing technique combining both the Sender + the body. We search based on the hash code (MD5). The hash code can be indexed and as it is constant size, it cannot make performance issue on the Insert and the Update. We know that several different emails can have the same hash code, thus, we iteratively try to scan all the messages having the same hash code to find the exact match.

7. Using the Aspam Web Services

We briefly explain here how to integrate these web services within the email applications: Mail User Agent (MUA) and Mail Servers. Details on how to integrate the web services in any application is outside the scope of this paper. It is extremely simple for a professional developer. There exist many resources on the Internet showing how to use them. These web services are provided with the source code of the developed applications. Several sample applications are developed and can be obtained from the source code of the whole system.

These web services are hosted in the web site of the project under the following link:
http://www.asites.org/aspamServices.svc. If you navigate to that link, you will simple see a very simple how-to use these services. The source code of the web services are also provided and can be used in a stand-alone installation within the whole system. More details and description of the web services and the sample application code description are provided in an extended technical report [35].

8. Conclusions and Future Directions

In this paper, a new technique for spam filtering that combines both the traditional spam filtering methods and a smart cooperative subjective spam filtering method is presented. A developed email client application (ECA) that implements the proposed technique is described. This application provides the traditional email services in addition to the spam filtering features. It makes use of the user voting on the incoming emails to classify the rest of the similar emails for the other users automatically. The second developed application that is described in the paper is the Cooperative Anti-Spam Email Server Application. It complements the ECA and provides several features for the enterprises and the email service providers. Finally, a set of standard web services that we have developed are presented. One of them allows any existing email server to request for filtering any arrived email as spam or not based on the subjective spam filtering technique provided here. The other web services allow the users of the existing email client applications to participate on the subjective spam filtering activity on the centralized system. The system is validated and tested. The accuracy of the system depends on the number of users using it. As more users use the system, its accuracy increases

REFERENCES

[1] W.-F. Hsiao and T.-M. Chang, "An Incremental Cluster-Based Approach to SPAM Filtering," *Expert Systems with Applications Journal*, Vol. 34, No. 3, 1975, pp. 1-28.

[2] N. Leavitt, "Vendors Fight Spam's Sudden Rise," *IEEE Computer Magazine*, Vol. 40, No. 3, 2007, pp. 16-19.

[3] L. M. Spracklin and L. V. Saxton, "Filtering Spam Using Kolmogorov Complexity Estimates," *The Proceedings of the* 21*st International Conference on Advanced Information Networking and Applications Workshops*, Niagara Falls, 21-23 May 2007, Vol. 1, pp. 321-328.

[4] Y.-M. Wang and M. Ma, "Strider Search Ranger: Towards an Autonomic Anti-Spam Search Engine," *Fourth International Conference on Autonomic Computing*, Jacksonville, 11-15 June 2007, pp. 32-32. doi:10.1109/ICAC.2007.38

[5] M. N. Marsono, M. Watheq El-Kharashi and F. Gebali, "Binary LNS-Based Naive Bayes Inference Engine for Spam Control: Noise Analysis and FPGA Implementation," *IET Computers & Digital Techniques*, Vol. 2, No. 1, 2008, pp. 56-62.

[6] K. Saraubon and B. Limthanmaphon, "Fast Effective Botnet Spam Detection," *Fourth International Conference on Computer Sciences and Convergence Information Technology*, Seoul, 24-26 November 2009, pp. 1066-1070. doi:10.1109/ICCIT.2009.128

[7] Z. H. Duan, P. Chen, F. Sanchez, Y. F. Dong, M. Stephenson and J. Barker, "Detecting Spam Zombies by Monitoring Outgoing Messages," *IEEE INFOCOM* 2009, 19-25 April 2009, pp. 1764-1772.

[8] Z. H. Duan, P. Chen, F. Sanchez, Y. F. Dong, M. Stephenson and J. M. Barker, "Detecting Spam Zombies by Monitoring Outgoing Messages," *IEEE Transactions on Dependable and Secure Computing*, Vol. 9, No. 2, 2012, pp. 198-210. doi:10.1109/TDSC.2011.49

[9] E. Levy, "The Making of a Spam Zombie Army: Dissecting the Sobig Worms," *IEEE Security & Privacy Magazine*, Vol. 1, No. 4, 2003, pp. 58-59.

[10] Z. Li and H. Y. Shen, "SOAP: A Social Network Aided Personalized and Effective Spam Filter to Clean Your E-Mail Box," *Proceedings of IEEE Infocom*, Shanghai, 10-15 April 2011, pp. 1835-1843.

[11] J. A. Zdziarski, "Ending Spam—Bayesian Content Filtering and the Art of Statistical Language Classification," 5th Edition, No Starch Press, San Francisco, 2005.

[12] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. D. Spyropoulos and P. Stamatopoulos, "Learning to Filter Spam E-Mail: A Comparison of a Naive Bayesian and a Memory-Based Approach," *Proceedings of the Workshop*: *Machine Learning and Textual Information Access*, 2000, pp. 1-13.

[13] M. Sahami, S. Dumais, D. Heckerman and E. Horvitz, "A Bayesian Approach to Filtering Junk Email," *Learning for Text Categorization—Papers from the AAAI Workshop*, 1998, pp. 55-62.

[14] D. Karthika Renuka, T. Hamsapriya, M. Raja Chakkaravarthi and P. Lakshmi Surya, "Spam Classification Based on Supervised Learning Using Machine Learning Techniques," *International Conference on Process Automation, Control and Computing*, Coimbatore, 20-22 July 2011, pp. 1-7.

[15] H. Drucker, D. Wu and V. N. Vapnik, "Support Vector Machines for Spam Categorization," *IEEE Transactions on Neural Networks*, Vol. 10, No. 5, 1999, pp. 1048-1054. doi:10.1109/72.788645

[16] C.-Y. Tseng and M.-S. Chen, "Incremental SVM Model for Spam Detection on Dynamic Email Social Networks," *International Conference on Computational Science and*

Engineering, Vancouver, 29-31 August 2009, pp. 128-135.

[17] S. Suwa, N. Yamai, K. Okayama and M. Nakamura, "DNS Resource Record Analysis of URLs in E-Mail Messages for Improving Spam Filtering," *IEEE/IPSJ* 11*th International Symposium on Applications and the Internet* (*SAINT*), Munich, 18-21 July 2011, pp. 439-444.

[18] A. Khanal, B. S. Motlagh and T. Kocak, "Improving the Efficiency of Spam Filtering through Cache Architecture," 15*th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, Istanbul, 24-26 October 2007, pp. 303-309. doi:10.1109/MASCOTS.2007.27

[19] T. R. Surmacz, "Reliability of E-Mail Delivery in the Era of Spam," 2*nd International Conference on Dependability of Computer Systems*, Szklarska, 14-16 June 2007, pp. 198-204.

[20] B. Hoanca, "How Good Are Our Weapons in the Spam Wars?" *IEEE Technology and Society Magazine*, Vol. 25, No. 1, 2006, pp. 22-30.

[21] E. Rabinovitch, "Readers' Comments on SPAM," *IEEE Communications Magazine*, Vol. 40, No. 11, 2002, pp. 20-24.

[22] J. Yan and P. L. Cho, "Enhancing Collaborative Spam Detection with Bloom Filters," *Proceedings of the* 22*nd Annual Computer Security Applications Conference*, Miami Beach, December 2006, pp. 414-428.

[23] "Razor: A Distributed, Collaborative, Spam Detection and Filtering Network," http://razor.sourceforge.net

[24] S. A. Elsagheer Mohamed, "A Solution for Fighting Spammer's Resources and Minimizing the Impact of Spam," *International Journal of Communications*, *Network and System Sciences*, Vol. 5, No. 7, 2012, pp. 416-422. doi:10.4236/ijcns.2012.57051

[25] X. Carreras and L. Márquez, "Boosting Trees for Antispam Email Filtering," *Proceedings of* 4*th International Conference on Recent Advances in Natural Language Processing*, 2001, pp. 58-64.

[26] M. R. Islam, W. L. Zhou and M. U. Choudhury, "Dynamic Feature Selection for Spam Filtering Using Support Vector Machine," 6*th IEEE/ACIS International Conference on Computer and Information Science*, Melbourne, 11-13 July 2007, pp. 757-762. doi:10.1109/ICIS.2007.92

[27] W. W. Cohen, "Learning Rules That Classify E-Mail," *Proceedings of AAAI Spring Symposium on Machine Learning in Information Access*, Stanford, 25-27 March 1996, pp. 18-25.

[28] X.-L. Pang, Y.-Q. Feng and W. Jiang, "The Compensation Strategy of Unseen Feature Words in Naïve Bayes Text Classification," *Journal of Harbin Institute of Technology*, 2007.

[29] A. Taweesiriwate, B. Manaskasemsak and A. Rungsawang, "Web Spam Detection Using Link-Based Ant Colony Optimization," *IEEE* 26*th International Conference on Advanced Information Networking and Applications* (*AINA*), Fukuoka-shi, 26-29 March 2012, pp. 868-873.

[30] S. Mohamed, W. Ata and N. Darwish, "A New Technique for Automatic Text Categorization for Arabic Documents," *Proceedings of* 5*th International Conference on Internet and Information Technology in Modern Organizations*, Cario, December 2005.

[31] H. Q. Zuo, X. Li, O. Wu, W. M. Hu and G. Luo, "Image Spam Filtering Using Fourier-Mellin Invariant Features," *IEEE International Conference on Acoustics*, *Speech and Signal Processing*, Taipei, 19-24 April 2009, pp. 849-852.

[32] J.-H. Hsia and M.-S. Chen, "Language-Model-Based Detection Cascade for Efficient Classification of Image-Based Spam E-Mail," *IEEE International Conference on Multimedia and Expo*, New York, 28 June-3 July 2009, pp. 1182-1185.

[33] B. Biggio, G. Fumera, I. Pillai and F. Roli, "Image Spam Filtering Using Visual Information," 14*th International Conference on Image Analysis and Processing*, Modena, 10-14 September 2007, pp. 105-110.

[34] H. B. Aradhye, G. K. Myers and J. A. Herson, "Image Analysis for Categorization of Image-based Spam E-Mail," *Proceedings of the* 8*th International Conference on Document Analysis and Recognition*, Korea, 29 August-1 September 2005, pp. 914-918.

[35] S. A. E. Mohamed and M. M. Tezeghdanti, "Spam Detection and Categorization for Electronic Arabic Messages: Project Final Technical Report," Technical Report, 2012. http://aspam.asites.org/0Files/TechnicalReport-FinalV3.pdf