DESIGNING SPAM MAIL FILTERING USING DATA MINING BY ANALYZING USER AND EMAIL BEHAVIOR

By

Abdullah Ibn Nurul Islam
Student ID: 074605

Supervised by

Professor Dr. Md. Abdul Mottalib
Department of Computer Science and Engineering
Islamic University of Technology (IUT)

A thesis submitted to the department of Computer Science and Engineering (CSE)
in partial fulfillment of the requirements for the award of the degree of
Masters of Science in Computer Science and Engineering

Department of Computer Science and Engineering
Islamic University of Technology (IUT)
Gazipur, Bangladesh

24 January, 2013

Declaration of Candidate

It is hereby declared that this thesis/project report or any part of it has not been submitted elsewhere for the award of any Degree or Diploma.

_____

Prof. Dr. Md. Abdul Mottalib
Head
Department of CSE
Islamic University of Technology (IUT)
Board Bazar, Gazipur-1704, Bangladesh.
Date: 24 January 2013

_____

Abdullah Ibn Nurul Islam
Student No: 074605
Academic Year: 2011-2012
Date: 24 January 2013

Dedicated To

All of my beloved teachers of IUT and members of my family

Table of Contents

List of Tables

List of Figures

Abbreviations

SPAM	Specifically Persecuted Advertising Mail
Ham	Not Spam/Legitimate
SMTP	Simple Mail Transfer Protocol
RFC	Request for Comment
EMT	Email Mining Toolkit
MIME	Multipurpose Internet Mail Extension
DOS Attack	Denial of Service Attack
MAN method	Proposed method
OCR	Optical Character Recognition
IP	Internet Protocol
MTA	Message Transfer Agents
TCP	Transmission Control Protocol
SPF	Sender Policy Framework
PF	Post Filtering

Acknowledgements

First and foremost, I must sense grateful to and wish to acknowledge my insightful indebtedness to my supervisor Professor Dr. M. A. Mottalib, Head of the Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT). His unfathomable knowledge in the field of spam filtering influenced me to carry out this thesis up to this point. Their endless endurance, scholarly guidance, continual encouragement, constant and lively supervision, constructive criticism, priceless suggestion, reading many mediocre drafts and correcting them at all state of affair have made it possible to come up to this phase. Without his inspiring enthusiasm and encouragement, this work could not be completed.

I also wish to take opportunity to articulate my sincerest gratitude and heartiest thanks to Mr. Tareque Mahmud Chowdhury, Assistant professor of Islamic University of Technology (IUT) for his all out mutual aids and providing important pertinent documents.

I thank all the staffs, graduate students and faculty members of Department of Computer Science & Engineering of IUT for their supports and encouragements. I wish to express my gratitude to IUT for providing an excellent environment for research work.

Last, but no means least, I thank Allah for the talents and abilities I was given that made it possible to undertake this research

Abstract

Electronic Mail is the “killer network application”. It is ubiquitous and pervasive. In a relatively short timeframe, the Internet has become irrevocably and deeply entrenched in our modern society primarily due to the power of its communication substrate linking people and organizations around the globe. Much work on email technology has focused on making email easy to use, permitting a wide variety of information and information types to be conveniently, reliably, sent throughout the Internet. However, the analysis of the vast storehouse of email content accumulated or produced by individual users has received relatively little attention other than for specific tasks such as spam and virus filtering. Users in the email continuously receive spam and they get into trouble wasting their time and also harmful emails can cause harm to the computers.

This thesis presents an implemented framework for data mining behavior models from email data. The EMT is a data mining tool kit designed to analyze email corpora, including the entire set of email sent and received by an individual user, revealing much information about individual users as well as the behavior of groups of users in an organization. A number of machine learning and anomaly detection algorithms are embedded in the system to model the user’s email behavior in order to classify email for a variety of tasks. There are different methods for detection of spam through email. The main goal is to develop a method that outperforms the existing methods in terms of detection of spam, ham and wrongly classified spam, i.e. need is to improve the accuracy of the proposed method compared to the other existing methods. The other goal is to implement the proposed algorithm for reducing the time. So, to recapitulate, this thesis also deals the accuracy and process timing based on prioritization of detecting email messages.

The proposed method uses prioritization of process criterion which is unavailable in the earlier existing methods. It also uses the post-filtering concept which contributes for the enhancement of accuracy of the proposed method. Thus the proposed method, which we name as MAN is responsible for spam detection and outperforms

the existing methods. This method also provides user convenient spam detection process. So, by using the concepts of post-filtering, process prioritization and different criterion in order to detect spam, the optimum accuracy for detecting spam will be possible.

Chapter 1: Introduction

1.1 Background

Electronic mail is one of the most popular forms of communications today. The surprisingly fast acceptance of this communication medium is best exemplified by the sheer number of current users, estimated to be as close to three quarters of a billion individuals, and growing [1]. This form of communication has the simple advantage of being almost instantaneous, intuitive to use, and costing virtually nothing per message. The current email system is based on the SMTP protocol RFC 821 and 822 developed in 1982 and extended in RFC 2821 in 2001[2]. This system defines a common standard to unite the different messaging protocols in existence prior to 1982. It allowed users the ability to exchange messages with one another using a system based on the SMTP protocol and email addresses. These protocols allowed messages to pass from one user to another, making it practical and easy for different users to communicate independent of the service-provider or the client application. In 1982, Denning [3] wrote about the problem of working with email, asking" Who will save the receivers from drowning in the rising tide of information so generated?"

Emails for the most part are held in data files or folders with no structured relationship (at files), making anything more than a keyword search very slow. Users may choose to move messages into time-ordered sub-folders of related messages. Studies have shown that typical users quickly generate anywhere from tens to hundreds of folders in a relatively short amount of time. Finding a particular past message across these sub-folders can easily turn into a daunting task. Not only is the email the subject of search, but also the folder in which it might have been placed. Within these at file folders, attachments are encoded in MIME format making analysis of anything other than simple filename close to impossible. Recent tools have been released which allow indexing and searching local data including emails and parts of attachments. Above and beyond simply sending messages, studies have shown that many users have quickly adopted email to a variety of tasks including task delegation, document archiving, personal contact list, and reminder and scheduling [4]. For example, typical users will use their INBOX or main message area, as an

active “to-do list”, leaving current messages on the top of the list. Even for well-organized users who always maintain past messages in appropriate sub-folders, there remains the possibility of down-time, and hence, over a relatively short period of time, bursts of email can quickly accumulate making organization of these new messages a slow and difficult task. In addition to these organization issues, the Achilles heel of the current email system is its relative ease of abuse. The protocols were based on the assumption that email users would not abuse the privilege of sending messages to each other. The misuse and abuse of the email system has taken on many forms over the years. Typical misuse includes forged emails, unwanted emails (spam), fraudulent schemes, and identity theft and fraud through “Phishing” emails. Abuse includes virus and worm attachments, and email DOS attacks. The common denominator among all these categories is they exploit the email system’s lack of controls and authentication of sender and recipient (an inherit problem in a decentralized system). Email is not permission based, and one can simply send a message without prior approval. Users should not be expected to pay a repair bill for simply opening an email which seemed to have originated from a friend’s email address, spoofed by an abuser.

Thus, detecting spam is one of the most important criterion. In this thesis paper, our effort is to detect the spam in email. There are some existing spam filtering methods like Bayesian spam filtering, Improved Bayesian spam filtering, Naïve Bayesian spam filtering, Meta spam filtering. We compare the existing methods with the proposed method and find out the accuracy and false positive, i.e. wrongly detected spam of the proposed method with the above mentioned existing methods. It is observed that using the post filtering method of user customization increases the accuracy of the proposed method. The method of process prioritization is also used in order to detect the accuracy of the proposed method. If a process is able to detect spam more frequently than the other process than the prioritization of the process is automatically updated and thus the accuracy also increases significantly with the update of the process prioritization. To recapitulate, it can be said that the proposed method of spam filtering is the best and overwhelms the other existing methods in terms of accuracy and false positive detection.

1.2 Problem statement

Now as the number of email users is increasing day by day, so is the number of spam in the inbox. There are different methods for detection of spam. The most well known of these techniques are Bayesian, Improved Bayesian, Naïve Bayesian , Meta spam filtering, Greylist method for detection of spam. The advantage of Bayesian spam filtering is that it can be trained on per-user basis and it can perform particularly well in avoiding false positives, where legitimate email is incorrectly classified as spam. The main disadvantage is that spammer tactics include insertion of random innocuous words that are not normally associated with spam. For the improved Bayesian method, the main advantage is that the risk of loss factor is reduced. The disadvantage for the method is that calculating the weighting factor is time consuming and costly. The advantage of Naïve Bayesian spam filtering is that it only requires a small amount of training data to estimate the parameters [5]. The disadvantage of the method is that the dependence of the class conditional independence among these cannot be modeled. The advantage of Meta spam filtering is that TCP/IP blocking is used to find malicious email address, while the disadvantage of the method is that definitions of spam should be agreed on before testing. In the Greylist method, the advantage of the method is that it requires no additional configuration from user end while the disadvantage of the method is that it delays much of the mail from non-white listed mail servers. The main problems with all these existing algorithms is that none of the existing methods have accuracy of greater than 98% and all of these have higher time complexity which can be reduced with feasible implementation of a proposed method. So, our goal is to propose a method with higher accuracy and also provide a users' (receivers) customization in proposed model. So, we propose an efficient proposed method named as MAN which will drive away the disadvantages of the existing method. The MAN method has greater accuracy in order to detect email spam.

All of these have the advantages and disadvantages. Our main goal in this thesis is to develop an efficient method of spam detection named as MAN spam detection technique.

1.3 Significance of study

This thesis paper first detects the characteristics of different messages in order to find out the spam. There are different criteria for the detection of harmful messages such as presence of images, presence of hyperlinks, number of words and characters in the subject line, receiver actions towards the emails, number of emails send at a time, the set of distinct word frequently, requesting secrete information, number of unique sender addresses. If the message seems to be a harmful one then that message is detected as spam and different methods are applied to find out which method is the best one in order to detect spam.

1.4 Scope of study

This thesis mainly focuses on the spam, ham (those that are not spam) and false positive (wrongly detected spam). For this purpose, we have compared Bayesian, Improved Bayesian, Naïve Bayesian, Meta spam filtering, Greylist and our proposed MAN spam detection method. The accuracy of these methods is compared. The prioritization of the receiver in the proposed method has immense contribution to decrease the wrongly detected spam or ham. The MAN spam detection method out performs the other methods in terms of the criterion mentioned earlier.

1.5 Objectives:

We have set forth the followings as the research objectives:

a. To study different methods of spam filtering
b. To analyze the behavior of spammer (sender)
c. To analyze the behavior of emails
d. To analyze the behavior of user (receiver) towards the spam's
e. To propose a spam detector on the basis of analysis
f. To implement the proposed method in a real life mail server.

1.6 Thesis outline

In this thesis paper, the next chapter deals with the existing filtering methods, in chapter-3 we will discuss the proposed MAN method for spam detection, in chapter-4 we will implement the proposed method and in chapter-5 will produce a graphical

representation of the performance analysis of the proposed MAN spam detection method compared with Bayesian and Naïve Bayesian method. Last, but not the least, in chapter 6 we will summarize the report and will indicate the future work for the existing proposed method.

Chapter 2: Literature Review

In this chapter, we will discuss about the various email classifications of the existing methodologies. The main methodologies used for spam filtering are Bayesian spam filtering, improved Bayesian filtering, A Naive Bayes classifier, Meta spam filtering, and Greylist. We will discuss about these methodologies in the next section.

2.1 Bayesian spam filtering

It is known as statistical spam filtering method. It makes use of a naive Bayes classifier to identify spam e-mail. Bayesian classifiers work by correlating the use of tokens (typically words, or sometimes other things), with spam and non-spam e-mails and then using Bayesian inference to calculate a probability that an email is or is not spam. Bayesian spam filtering is a very powerful technique for dealing with spam, that can tailor itself to the email needs of individual users, and gives low false positive spam detection rates that are generally acceptable to users.

The first known mail-filtering program to use a Bayes classifier was Jason Rennie's ifile program, released in 1996. The program was used to sort mail into folders. The first scholarly publication on Bayesian spam filtering was by Sahami et al. in 1998[7]. That work was soon thereafter deployed in commercial spam filters. However, in 2002, Paul Graham was able to greatly improve the false positive rate, so that it could be used on its own as a single spam filter.

2.1.1 Process

Particular words have particular probabilities of occurring in spam email and in legitimate email. For instance, most email users will frequently encounter the word "Viagra" in spam email, but will seldom see it in other email. The filter doesn't know these probabilities in advance, and must first be trained so it can build them up. To train the filter, the user must manually indicate whether a new email is spam or not. For all words in each training email, the filter will adjust the probabilities that each word will appear in spam or legitimate email in its database. For instance, Bayesian spam filters will typically have learned a very high spam probability for the words

"Viagra" and "refinance", but a very low spam probability for words seen only in legitimate email, such as the names of friends and family members.

After training, the word probabilities (also known as likelihood functions) are used to compute the probability that an email with a particular set of words in it belongs to either category. Each word in the email contributes to the email's spam probability, or only the most interesting words. This contribution is called the posterior probability and is computed using Bayes' theorem. Then, the email's spam probability is computed over all words in the email, and if the total exceeds a certain threshold (say 95%), the filter will mark the email as a spam.

As in any other spam filtering technique, email marked as spam can then be automatically moved to a "Junk" email folder, or even deleted outright. Some software implements quarantine mechanisms that define a time frame during which the user is allowed to review the software's decision.

The initial training can usually be refined when wrong judgments from the software are identified (false positives or false negatives). That allows the software to dynamically adapt to the ever evolving nature of spam.

Some spam filters combine the results of both Bayesian spam filtering and other heuristics (pre-defined rules about the contents, looking at the message's envelope, etc.), resulting in even higher filtering accuracy, sometimes at the cost of addictiveness.

2.1.2 Mathematical foundation

Bayesian email filters take advantage of Bayes' theorem. Bayes' theorem is used several times in the context of spam:

- A first time, to compute the probability that the message is spam, knowing that a given word appears in this message;
- A second time, to compute the probability that the message is spam, taking into consideration all of its words (or a relevant subset of them);
- Sometimes a third time, to deal with rare words.

2.1.3 Probability computation

Let's suppose the suspected message contains the word "replica". Most people who are used to receiving e-mail know that this message is likely to be spam, more precisely a proposal to sell counterfeit copies of well-known brands of watches. The spam detection software, however, does not "know" such facts, all it can do is compute probabilities.

The formula used by the software to determine that is derived from Bayes' theorem

$$Pr(S|W) = \frac{Pr(W|S).Pr(S)}{Pr(W|S).Pr(S)+Pr(W|H).Pr(H)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

where:

- *Pr(S|W)* is the probability that a message is a spam, knowing that the word "replica" is in it;
- *Pr(S)* is the overall probability that any given message is spam;
- *Pr(S|W)* is the probability that the word "replica" appears in spam messages;
- *Pr(H)* is the overall probability that any given message is not spam (is "ham");
- *Pr(W|H)* is the probability that the word "replica" appears in ham messages

Recent statistics [8] show that the current probability of any message being spam is 80%, at the very least: $Pr(S) = 0.8$; $Pr(H) = 0.2$

However, most bayesian spam detection software makes the assumption that there is no *a priori* reason for any incoming message to be spam rather than ham, and considers both cases to have equal probabilities of 50%: $Pr(S) = 0.5$; $Pr(H) = 0.5$

The filters that use this hypothesis are said to be "not biased", meaning that they have no prejudice regarding the incoming email. This assumption permits simplifying the general formula to:

$$Pr(S|W) = \frac{Pr(W|S)}{Pr(S|W)+Pr(W|H)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

This quantity is called "spamicity" (or "spaminess") of the word "replica", and can be computed. The number *Pr(W|S)* used in this formula is approximated to the frequency of messages containing "replica" in the messages identified as spam during

the learning phase. Similarly, *Pr(W|H)* is approximated to the frequency of messages containing "replica" in the messages identified as ham during the learning phase. For these approximations to make sense, the set of learned messages needs to be big and representative enough. It is also advisable that the learned set of messages conforms to the 50% hypothesis about repartition between spam and ham, i.e. that the datasets of spam and ham are of same size [8].

Of course, determining whether a message is spam or ham based only on the presence of the word "replica" is error-prone, which is why bayesian spam software tries to consider several words and combine their spamicities to determine a message's overall probability of being spam.

2.1.4 Combining individual probabilities

The bayesian spam filtering software makes the "naïve" assumption that the words present in the message are independent events. That is wrong in natural languages like English, where the probability of finding an adjective, for example, is affected by the probability of having a noun. With that assumption, one can derive another formula from Bayes' theorem:

$$P = \frac{P_1 P_2 \ldots P_n}{P_1 P_2 \ldots P_n + (1-P_1)(1-P_2) \ldots\ldots (1-P_n)} \quad \ldots \quad (3)$$

where:

- P is the probability that the suspect message is spam;
- P_1 is the probability $P(S|W_1)$ that it is a spam knowing it contains a first word (for example "replica");
- P_2 is the probability $P(S|W_2)$ that it is a spam knowing it contains a second word (for example "watches");
- Etc. ...
- P_N is the probability $P(S|W_N)$ that it is a spam knowing it contains an Nth word (for example "home").

Such assumptions make the spam filtering software a naive Bayes classifier. The result p is usually compared to a given threshold to decide whether the message is

spam or not. If p is lower than the threshold, the message is considered as likely ham, otherwise it is considered as likely spam.

2.1.5 Other expressions for combining individual probabilities

Usually p is not directly computed using the above formula due to floating-point underflow. Instead, p can be computed in the log domain by rewriting the original equation as follows:

$$\frac{1}{P} - 1 = \frac{(1-P_1)(1-P_2)\dots(1-P_n)}{P_1 P_2 \dots P_n} \quad \dots (4)$$

Taking logs on both sides:

$$\ln\left(\frac{1}{P} - 1\right) = \sum_{i=1}^{N} [\ln(1 - P_i) - \ln P_i] \quad \dots (5)$$

Let $\eta = \sum_{i=1}^{N} [\ln(1 - P_i) - \ln P_i]$

Therefore, $\frac{1}{P} - 1 = e^{\eta}$ ... (6)

Hence the alternate formula for computing the combined probability:

$$P = \frac{1}{1+e^{\eta}} \quad \dots (7)$$

2.1.6 Dealing with rare words

In the case a word has never been met during the learning phase, both the numerator and the denominator are equal to zero, both in the general formula and in the spamicity formula. The software can decide to discard such words for which there is no information available.

More generally, the words that were encountered only a few times during the learning phase cause a problem, because it would be an error to trust blindly the information they provide. A simple solution is to simply avoid taking such unreliable words into account as well.

Applying again Bayes' theorem, and assuming the classification between spam and ham of the emails containing a given word ("replica") is a random variable with beta distribution, some programs decide to use a corrected probability:

.... (8)

where:

- is the corrected probability for the message to be spam, knowing that it contains a given word ;
- *S* is the *strength* we give to background information about incoming spam ;
- *Pr(S)* is the probability of any incoming message to be spam ;
- is the number of occurrences of this word during the learning phase ;
- *Pr*(*S*|*W*) is the spamicity of this word.

This corrected probability is used instead of the spamicity in the combining formula.

Pr(S) can again be taken equal to 0.5, to avoid being too suspicious about incoming email. 3 is a good value for *s*, meaning that the learned corpus must contain more than 3 messages with that word to put more confidence in the spamicity value than in the default value. This formula can be extended to the case where *n* is equal to zero (and where the spamicity is not defined), and evaluates in this case to *Pr(S)*.

2.1.7 Other heuristics

"Neutral" words like "the", "a", "some", or "is" (in English), or their equivalents in other languages, can be ignored. More generally, some bayesian filtering filters simply ignore all the words which have a spamicity next to 0.5, as they bring little to a good decision. The words taken into consideration are those whose spamicity is next to 0.0 (distinctive signs of legitimate messages), or next to 1.0 (distinctive signs of spam). A method can be for example to keep only those ten words, in the examined message, which have the greatest absolute value |0.5 – *pI*|.

Some software products take into account the fact that a given word appears several times in the examined message, others don't.

Some software products use *patterns* (sequences of words) instead of isolated natural languages words [9]. For example, with a "context window" of four words, they compute the spamicity of "Viagra is good for", instead of computing the

spamicities of "Viagra", "is", "good", and "for". This method gives more sensitivity to context and eliminates the Bayesian noise better, at the expense of a bigger database.

2.1.8 Mixed methods

There are other ways of combining individual probabilities for different words than using the "naive" approach. These methods differ from it on the assumptions they make on the statistical properties of the input data. These different hypotheses result in radically different formulas for combining the individual probabilities.

For example, assuming the individual probabilities follow a chi-squared distribution with $2N$ degrees of freedom, one could use the formula:

$$P = C^{-1}(-2ln(P_1P_2 \dots P_N), 2N) \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (9)$$

where C^{-1} is the inverse of the chi-squared function.

Individual probabilities can be combined with the techniques of the Markovian discrimination too.

2.1.9 Advantages of the existing Bayesian method

One of the main advantages of Bayesian spam filtering is that it can be trained on a per-user basis.

The spam that a user receives is often related to the online user's activities. For example, a user may have been subscribed to an online newsletter that the user considers to be spam. This online newsletter is likely to contain words that are common to all newsletters, such as the name of the newsletter and its originating email address. A Bayesian spam filter will eventually assign a higher probability based on the user's specific patterns.

The legitimate e-mails a user receives will tend to be different. For example, in a corporate environment, the company name and the names of clients or customers will be mentioned often. The filter will assign a lower spam probability to emails containing those names.

The word probabilities are unique to each user and can evolve over time with corrective training whenever the filter incorrectly classifies an email. As a result, Bayesian spam filtering accuracy after training is often superior to pre-defined rules.

It can perform particularly well in avoiding false positives, where legitimate email is incorrectly classified as spam. For example, if the email contains the word "Nigeria", which is frequently used in Advance fee fraud spam, a pre-defined rules filter might reject it outright. A Bayesian filter would mark the word "Nigeria" as a probable spam word, but would take into account other important words that usually indicate legitimate e-mail. For example, the name of a spouse may strongly indicate the e-mail is not spam, which could overcome the use of the word "Nigeria."

2.1.10 Limitations of the existing Bayesian method

Depending on the implementation, Bayesian spam filtering may be susceptible to Bayesian poisoning, a technique used by spammers in an attempt to degrade the effectiveness of spam filters that rely on Bayesian filtering. A spammer practicing Bayesian poisoning will send out emails with large amounts of legitimate text (gathered from legitimate news or literary sources). Spammer tactics include insertion of random innocuous words that are not normally associated with spam, thereby decreasing the email's spam score, making it more likely to slip past a Bayesian spam filter. However with (for example) Paul Graham's scheme only the most significant probabilities are used, so that padding the text out with non-spam-related words does not affect the detection probability significantly.

Words that normally appear in large quantities in spam may also be transformed by spammers. For example, « Viagra » would be replaced with « Viaagra » or « Viagra » in the spam message. The recipient of the message can still read the changed words, but each of these words is met more rarely by the bayesian filter, which hinders its learning process. As a general rule, this spamming technique does not work very well, because the derived words end up recognized by the filter just like the normal ones.

Another technique used to try to defeat Bayesian spam filters is to replace text with pictures, either directly included or linked. The whole text of the message, or some

part of it, is replaced with a picture where the same text is "drawn". The spam filter is usually unable to analyze this picture, which would contain the sensitive words like "Viagra". However, since many mail clients disable the display of linked pictures for security reasons, the spammer sending links to distant pictures might reach fewer targets. Also, a picture's size in bytes is bigger than the equivalent text's size, so the spammer needs more bandwidth to send messages directly including pictures. Some filters are more inclined to decide that a message is spam if it has mostly graphical contents. Finally, a probably more efficient solution has been proposed by Google and is used by its Gmail email system, performing an OCR to every mid to large size image, analyzing the text inside [10].

2.1.11 Applications of Bayesian filtering

While Bayesian filtering is used widely to identify spam email, the technique can classify (or "cluster") almost any sort of data. It has uses in science, medicine, and engineering. One example is a general purpose classification program called AutoClass which was originally used to classify stars according to spectral characteristics that were otherwise too subtle to notice. There is recent speculation that even the brain uses Bayesian methods to classify sensory stimuli and decide on behavioral responses [11].

2.2 Improved Bayesian filtering

Final decision is made based on the weighted score of the attributes of both attitude analysis phase and relevancy analysis phase. The attitude analysis holds 0.5 weight age for both e-mail id and subject trusted. Similarly, relevancy analysis phase holds 0.5 weight age for relevant content. If the weighted value is greater than 0.5 then the email is moved to Inbox and the pre-processed root words which are not already exist are added to positive dictionary. If the weighted value is less than 0.5 then the email is moved to spam and the pre-processed root words which are not already exist are added to negative dictionary. If the weighted value is equal to 0.5 then the e-mail is hold. The number of normal e-mail that are classified as spam and the reverse will be significantly trim down since there are a two levels of validating a e-mail in

the system. Also user can classify spam and ham e-mail according to his personal interest on a particular e-mail rather than going for a generalized spam filter.

Assumed Ham classified as *C0*, Spam classified as *C1*, decision-making text messages as legitimate risk conditions,

$R(HAM|D)=P(C_1|D)$,
$R(SPAM|D)=1-P(C_1|D)$

After calculating a probability the e-mail is spam, one need to compare with the critical value to determine whether it is a spam. Suppose *D* is spam e-mail the probability of $P(C1|D)$, the probability of the normal messages $P(C0|D)=1-P(C1|D)$. Threshold in two forms: [12]

a. Set the critical probability *t*, if $P(C1|D) > t$, then that e-mail is spam;
b. Set the critical ratio *k*, if the $(P(C1|D) / P(C0|D)) > k$, then that e-mail is spam

It is easy to get the relationship between *t* and *k* is:

$$t= \frac{k}{1+k}$$

$$k= \frac{t}{1-t}$$

Therefore the text D decision-making of risk as spam R(SPAM | D)=k(1- (P(C1 | D))

2.2.1 Advantage of improved Bayesian filtering method

Using improved Bayesian spam filtering method, the risk of loss factor of *k*, i.e. weight factor of the ham emails recognized wrongly as spam are reduced.

2.3 Naïve Bayesian spam filtering method

The well known Naïve Bayesian classifier is a method based on bayes theorem, with strong naïve independence assumptions. “Independent feature model” is known as to be a more descriptive term. In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of the other features, a naive

Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple.

Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian methods.

In spite of their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. In 2004, analysis of the Bayesian classification problem has shown that there are some theoretical reasons for the apparently unreasonable efficacy of naive Bayes classifiers[13] Still, a comprehensive comparison with other classification methods in 2006 showed that Bayes classification is outperformed by more current approaches, such as boosted trees or random forests [14].

An advantage of the naive Bayes classifier is that it only requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

2.3.1 The Naive Bayes probabilistic model

Abstractly, the probability model for a classifier is a conditional model $p(C|F_1, \dots, F_n)$ over a dependent class variable C with a small number of outcomes or *classes*, conditional on several feature variables F_1 through F_n . The problem is that if the number of features n is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable.

Using Bayes' theorem, we write

$$p(C|F_1, \dots, F_n) = \frac{p(C)\, p(F_1,\dots,F_n|C)}{p(F_1,\dots,F_n)} \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (10)$$

In plain English the above equation can be written as

$$posterior = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (11)$$

In practice we are only interested in the numerator of that fraction, since the denominator does not depend on C and the values of the features F_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model $p(C, F_i, \ldots, F_n)$ which can be rewritten as follows, using the chain rule for repeated applications of the definition of conditional probability:

$$\begin{aligned}
&p(C, F_1, \ldots, F_n) \\
&\alpha\, p(C)\, p(F_1, \ldots, F_n|C) \\
&\alpha\, p(C)\, p(F_1/C)\, p(F_2, \ldots, F_n/C,F_1) \\
&\alpha\, p(C)\, p(F_1/C)\, p(F_2/C, F_1)\, p(F_3, \ldots, F_n/C, F_1, F_2) \\
&\alpha\, p(C)\, p(F_1/C)\, p(F_2/C, F_1)\, p(F_3/C, F_1, F_2)\, p(F_4, \ldots, F_n/C, F_1, F_2, F_3) \\
&\alpha\, p(C)\, p(F_1/C)\, p(F_2/C, F_1)\, p(F_3/C, F_1, F_2) \ldots\ldots p(F_n/C, F_1, F_2, F_3, \ldots, F_{n-1}) \ldots. \quad (12)
\end{aligned}$$

Now the "naive" conditional independence assumptions come into play: assume that each feature F_i is conditionally independent of every other feature F_j for $j \neq i$. This means that $p(F_i/C, F_j) = p(F_i/C)$ for $i \neq j$, and so the joint model can be expressed as

$$p(C, F_1, \ldots, F_n)\ \alpha\ p(C)\ p(F_1|C)\ p(F_2|C)\ p(F_3|C) \ldots \alpha\ p(C) \prod_{i=1}^{n} p(F_i|C) \quad \ldots. \quad (13)$$

This means that under the above independence assumptions, the conditional distribution over the class variable C can be expressed like this:

$$p(C|F_1, \ldots, F_n) = \frac{1}{Z} p(C) \prod_{i=1}^{n} p(F_i|C) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (14)$$

Where, Z (the evidence) is a scaling factor dependent only on $F_1, \ldots, F_n$, , i.e., a constant if the values of the feature variables are known.

Models of this form are much more manageable, since they factor into a so-called *class prior* $p(C)$ and independent probability distributions $p(F_i/C)$. If there are k classes and if a model for each $p(F_i/C = c)$ can be expressed in terms of r parameters, then the corresponding naive Bayes model has $(k - 1) + n\ r\ k$ parameters. In practice,

often k = 2 (binary classification) and r = 1 (Bernoulli variables as features) are common, and so the total number of parameters of the naive Bayes model is $2n$+1, where n is the number of binary features used for classification and prediction.

2.3.2 Estimating the parameters

All model parameters (*i.e.*, class priors and feature probability distributions) can be approximated with relative frequencies from the training set. These are maximum likelihood estimates of the probabilities. A class' prior may be calculated by assuming equally probable classes, or by calculating an estimate for the class probability from the training set (i.e., (prior for a given class) = (number of samples in the class)/(total number of samples)). To estimate the parameters for a feature's distribution, one must assume a distribution or generate nonparametric models for the features from the training set [15]. If one is dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution.

For example, suppose the training data contain a continuous attribute, x. We first segment the data by the class, and then compute the mean and variance of x in each class. Let μ_c be the mean of the values in x associated with class c, and let be the variance of the values in x associated with class c. Then, the probability of some value given a class, $P(x=v|c)$, can be computed by plugging v into the equation for a Normal distribution parameterized by μ_c and . That is,

.... (15)

Another common technique for handling continuous values is to use binning to discrete the values. In general, the distribution method is a better choice if there is a small amount of training data, or if the precise distribution of the data is known. The discrete method tends to do better if there is a large amount of training data because it will learn to fit the distribution of the data. Since naive Bayes is typically used when a large amount of data is available (as more computationally expensive models can generally achieve better accuracy), the discrete method is generally preferred over the distribution method.

2.3.3 Sample correction

If given class and feature values never occur together in the training set then the frequency-based probability estimate will be zero. This is problematic since it will wipe out all information in the other probabilities when they are multiplied. It is therefore often desirable to incorporate a small-sample correction in all probability estimates such that no probability is ever set to be exactly zero [16].

2.3.4 Constructing a classifier from the probability model

The discussion so far has derived the independent feature model, that is, the naive Bayes probability model. The naive Bayes classifier combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the *maximum a posteriori* or *MAP* decision rule. The corresponding classifier is the function *classify* defined as follows:

$$\text{classify}\,(f_1, \ldots, f_n) = \underset{c}{\operatorname{argmax}}\, p(C = c) \prod_{i=1}^{n} p(F_i = f_i \,|\, C = c) \quad \ldots\ldots\ldots\ldots\ldots\ldots (16)$$

2.3.5 Discussion

Despite the fact that the far-reaching independence assumptions are often inaccurate, the naive Bayes classifier has several properties that make it surprisingly useful in practice. In particular, the decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution. This helps alleviate problems stemming from the curse of dimensionality, such as the need for data sets that scale exponentially with the number of features. While naive Bayes often fails to produce a good estimate for the correct class probabilities, this may not be a requirement for many applications. For example, the naive Bayes classifier will make the correct MAP decision rule classification so long as the correct class is more probable than any other class. This is true regardless of whether the probability estimate is slightly, or even grossly inaccurate. In this manner, the overall classifier can be robust enough to ignore serious deficiencies in its underlying naive probability model. Other reasons for the observed success of the naive Bayes classifier are discussed in the literature cited below. The naïve Bayesian classifier is used for various purposes such as gender

classification, and document classification. As the main goal of our thesis is dealt with spam detection, we will deal with the document classification.

The description of the document classification is given below:

2.3.6 Document Classification

Here is a worked example of naive Bayesian classification to the document classification problem. Consider the problem of classifying documents by their content, for example into spam and non-spam e-mails. Imagine that documents are drawn from a number of classes of documents which can be modeled as sets of words where the (independent) probability that the *i*-th word of a given document occurs in a document from class *C* can be written as $p(w_i|C)$.

(For this treatment, we simplify things further by assuming that words are randomly distributed in the document - that is, words are not dependent on the length of the document, position within the document with relation to other words, or other document-context.)

Then the probability that a given document *D* contains all of the words w_i, given a class *C*, is

$$p(D|C) = \prod_i P(w_k|C) \quad \text{.... ....} \quad (17)$$

The question that we desire to answer is: "what is the probability that a given document *D* belongs to a given class *C* ?" In other words, what is $p(C|D)$

Now by definition

$$p(D|C) = \frac{p(D \cap C)}{p(C)} \quad \text{.... ....} \quad (18)$$

And $$p(C|D) = \frac{p(D \cap C)}{p(D)} \quad \text{.... ....} \quad (19)$$

Bayes' theorem manipulates these into a statement of probability in terms of likelihood.

$$p(C|D) = \frac{p(C)}{p(D)} p(D|C) \quad \text{.... ....} \quad (20)$$

Assume for the moment that there are only two mutually exclusive classes, S and $\neg S$ (e.g. spam and not spam), such that every element (email) is in either one or the other;

$$p(D|S) = \prod_i P(w_i|S) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (21)$$

And $$p(D|\neg S) = \prod_i P(w_i|\neg S) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (22)$$

Using the Bayesian result above, we can write:

$$p(S|D) = \frac{p(S)}{p(D)} \prod_i P(w_i|S) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (23)$$

$$p(\neg S|D) = \frac{p(\neg S)}{p(D)} \prod_i P(w_i|\neg S) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (24)$$

Dividing one by the other gives:

$$\frac{p(S|D)}{p(\neg S|D)} = \frac{p(S) \prod_i P(w_i|S)}{p(\neg S) \prod_i P(w_i|\neg S)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (25)$$

Which can be re-factored as:

$$\frac{p(S|D)}{p(\neg S|D)} = \frac{p(S)}{p(\neg S)} \prod_i \frac{P(w_i|S)}{P(w_i|\neg S)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (26)$$

Thus, the probability ratio $p(S|D)$ / $p(\neg S|D)$ can be expressed in terms of a series of likelihood ratios. The actual probability $p(S|D)$ can be easily computed from log $(p(S|D)/p(\neg S|D))$ based on the observation that $p(S|D) + p(\neg S|D) = 1$.

Taking the logarithm of all these ratios, we have:

$$ln \frac{p(S|D)}{p(\neg S|D)} = ln \frac{p(S)}{p(\neg S)} \sum_i ln \frac{P(w_i|S)}{P(w_i|\neg S)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (27)$$

(This technique of "log-likelihood ratios" is a common technique in statistics. In the case of two mutually exclusive alternatives (such as this example), the conversion of a log-likelihood ratio to a probability takes the form of a sigmoid curve: see log it for details.)

Finally, the document can be classified as follows. It is spam if $p(S|D) > p(\neg S|D)$ (that is, $ln \frac{P(S|D)}{P(\neg S|D)} > 0$), otherwise it is not spam.

2.4 Meta spam filtering technique

Given the significance of the spam blight and the competitive nature of the spam-blocking vendor landscape, most organizations are diligently evaluating suppliers, and in many cases bringing in products for hands-on testing. In addition, many trade publications are doing on-site bake-offs to determine the effectiveness of various solutions, including on-premises software, appliances, and managed services. In some cases, the testing methodology is flawed, and the results do not represent the actual effectiveness of the product or service. The root cause of the invalid testing is that testers typically take a corpus of mail and forward it to the spam-blocking service or product. In such cases, because of the message forwarding, the vendor is unable to perform a series of sender IP validation tests, nor is it able to glean intelligence from the SMTP setup. In some cases, these real-time tests can contribute up to 20% of spam being blocked.

Furthermore, the header information that is forwarded along with the message is subject to spammer manipulation and is therefore not necessarily a productive interrogation target. For example, spammers now routinely add legitimate IP sending addresses to header information in hopes of the spam being allowed to pass through the blocking service without scrutiny. In fact, spammers now take great pains to hide the originating IP address in the header information, which affects not only white list performance, but also blacklist, traffic shaping, and reputation filter effectiveness. Therefore, customers need to understand spammer header tricks as well as the value of real-time spam evaluation services, and change testing methodologies accordingly to get a more accurate picture of the effectiveness of spam-blocking products.

Here, we describe various real-time blocking techniques from a sampling of vendors and conclude with best practices for accurate testing methodologies.

2.4.1 CONTEXT

It is not news to any organization using e-mail that spam threatens the effectiveness of e-mail systems. Left unattended, spam clogs inboxes, compromises user efficiency, and overwhelms system components such as message stores and MTAs. Furthermore, spam is a conduit of all types of salacious content and fraudulent come-

ons that seek to cajole users into disclosing confidential information such as credit card and Social Security numbers, bank account information, and passwords (known as phishing). Approximately 70% of most organizations' inbound SMTP traffic is spam. Therefore, it is mandatory that organizations aggressively deploy top-tier spam-blocking solutions to mitigate the risks and problems concerned and associated with spam.

Most of the spam blocking services is used in multiple strategies and techniques. These are not invoked in a mail forwarding situation.

IP-blocking reputation lists: Some spam-blocking companies filter more than 100 million messages a day, and from this large volume they are able to glean intelligence about the sending patterns of a particular IP address. If they find a high correlation between a particular IP address and an unusual volume of mail or certain types of mail coming from the same address, they will refuse connections for at least a period of time — from that address. Some companies issue a 550 SMTP error message (access denied) to the sender. In this scenario, some vendors have a technician examine the mail flow and determine whether the messages are spam and then act accordingly [17].

2.4.2 META Practice

Companies also do in-depth log analysis to determine the validity of sending IP addresses. In addition, vendors have automated the reputation process, and in cases of, for example, real-time mail flood attacks, the system can shut down connections, though only after human oversight. With these IP-based reputation filter approaches, vendors estimate that they stop between 5% and 8% of all spam flowing through its network. Although there is a common belief that IP addresses are spoof able, SMTP connections require a confirmation packet from the recipient MTA sent to the sending IP address.

Although not technically impossible, in practicality, it is extremely difficult to spoof an IP address.

2.4.3 TCP/IP blocking

Another blocking method that would not be applicable in a mail-forwarding testing scenario is blocking messages at the TCP/IP level. Vendors have determined that spammers often display certain behavior during the SMTP conversation string - when the recipient and sender MTA first establish a connection. Vendors will not disclose the specific behavior for fear of tipping off spammers, but when this common behavior is identified, vendors issue a 550 SMTP error message (access denied).

One vendor is blocking between 25 million and 30 million messages a day based on this method

This per-message blocking service is also invoked using e-mail authentication standards such as Sender Policy Framework (SPF) and directory fail attempts. [18]

2.4.4 Traffic shaping

Some vendors use a third spam-blocking method called traffic shaping or IP throttling, which would not be invoked in a mail-forwarding testing situation. In this case - called Greylisting - vendors again correlate message flow and type with a particular IP address. But instead of dropping the connection, vendors slow down delivery rates - issuing an SMTP 451 error message (connection temporary unavailable). SMTP relays of legitimate sending organizations will retry later to get the message through, but a spammer - which is typically paid on volume of messages sent - will quickly lose patience and move on to another recipient MTA[19]. This method will also effectively stop dictionary harvest attacks, where the spammer attempts to collect legitimate mail addresses by bombarding the recipient MTA with a large volume of mail addressed to common names.

2.4.5 Header walk

Other real-time techniques are emerging that help determine the validity of messages. “Header walk” services allow the interrogation of header hop data (the routing path the message took to get to the destination), enabling the discovery of the sending source IP address. After determination of the source IP address, it will

perform a real-time lookup to see whether that IP address is registered to that domain. [20]

2.4.6 Sender query

Vendors use services that validate whether the sending address is legitimate by, for example, sending a message back to the sender, enabling the company to ascertain that the sending address is legitimate by scanning for delivery error codes.

2.4.7 Header/conversation data comparison

Companies are also investigating a service that would allow the comparison of SMTP conversation data (e.g., sending domains) with header information to see whether they match. Currently, spammers will often spoof header information, and evidence of that spoofing will be revealed by comparing the SMTP conversation data with the header data the recipient sees.

2.4.8 Sender IP identification:

Although sender IP addresses can usually be found in the receive headers in a forwarding scenario, hygiene vendors take additional actions to find the appropriate IP address. Certain MTAs do not include the original IP address, and some open-source MTAs have a tendency to botch the proper placement of the sender IP address. Therefore, in a mail-forwarding scenario, it could be impossible to find the original sender's IP address.

2.4.9 Proxy server identification

At a lower level (TCP/IP), vendors can often identify when a proxy server is being used to relay mail — an almost certain indication that a spammer is the source of the messages [21].

Again, in a mail-forwarding scenario, this intelligence is lost, creating suboptimal results for the spam-blocking system under interrogation.

2.4.10 Source origin loss

Origin or source of spam refers to the geographical location of the computer from which the spam is sent; it is not the country where the spammer resides, nor the

country that hosts the spam vertised site. Because of the international nature of spam, the spammer, the hijacked spam-sending computer, the spam vertised server, and the user target of the spam are all often located in different countries. As much as 80% of spam received by Internet users in North America and Europe can be traced to fewer than 200 spammers. Losing the source origin of the message also creates other problems; some Bayesian filters look at the receive header to find legitimate hops. In a mail-forwarding situation, the final hop is always legitimate, thereby tricking the filter into awarding positive scores that are fed into the overall statistical analysis of the message.

2.4.11 Message modification

Any modification to a message before it arrives at the filter can create artificial results - whether they are changes in the headers or content. For example, adding a forward descriptor into the subject line can obscure results. Forwarding scenarios may also result in changes to the MIME (multipurpose internet mail extension) boundaries that separate messages into logical parts such as text and attachments, leading, again, to compromised results.

2.4.12 Stale mail

Although not a real-time issue, testing results can be obscured by use of obsolete messages [22]. Most vendors retire mail-blocking rules, heuristics, and signatures regularly. If an older corpus of mail is used in a test, blocking effectiveness can be compromised if the relevant rules and other data have expired.

2.4.13 Testing Methodologies

Many evaluations focus exclusively on spam capture rates and false-positive generation. A broader testing methodology is more appropriate, where factors such as end-user satisfaction, ease of administration, and operational control are considered. This approach enables greater leeway for so-called greymail (messages the recipient might have solicited at one point, but no longer wants to receive). Likewise, it is inappropriate to compare vendors representing the three major delivery modalities (hosted, appliances, and traditional software load) with the same criteria because each delivery mechanism has a different value proposition. Finally,

the plug-and-play method of spam evaluation - where testers merely turn on the service with little or no tuning - does a disservice to the vendors. Testers need to do the appropriate tuning, including quarantine conditioning - enabling recipients to set up block/allow lists to get a more accurate picture of blocking effectiveness.

The methodologies used for the following for more accurate testing: [23]

- For hosting vendors such as MessageLabs, FrontBridge, and MX Logic, the most accurate testing scenario is to change the destination MTA message exchange record to the hosted vendor, which will filter the mail and forward it to the recipient domain.
- For on-premises traditional software load and appliance vendors such as CipherTrust, an IP load balancer should be placed in front of the hygiene vendors and a real-world mail feed should be balanced equally across all the vendors being tested. These approaches have two main virtues: use of real-time, real-world e-mail feeds, and no changes to mail headers and other data typically altered in a mail-forwarding testing scenario. Testers should ensure that a statistically relevant volume of mail is tested for legitimate results.

Furthermore, for both approaches, we make the following suggestions for improving the overall testing methodology:

- Blocking engines should be appropriately tuned before the actual tests start. In the case of hosted vendors, end users should be allowed to configure their personal blocking preferences.
- Testing should be done on real business users. They should give feedback to testers when spam gets through and when false positives are detected. These users should come from various corporate departments such as human resources, accounting, and customer support. This ensures a real-world representation of a corporate mail stream.
- Definitions of spam should be agreed on before testing e.g., all messages with salacious content, regardless of sender, should be considered spam.

2.5 Greylist

A relatively new spam-filtering technique, greylists take advantage of the fact that many spammers only attempt to send a batch of junk mail once. [24] Under the greylist system, the receiving mail server initially rejects messages from unknown users and sends a failure message to the originating server. If the mail server attempts to send the message a second time — a step most legitimate servers will take - the greylist assumes the message is not spam and lets it proceed to the recipient's inbox. At this point, the greylist filter will add the recipient's email or IP address to a list of allowed senders.

Though greylist filters require fewer system resources than some other types of spam filters, they also may delay mail delivery, which could be inconvenient when you are expecting time-sensitive messages.

2.5.1 Advantages of Greylist

The main advantage from the users' point of view is that greylisting requires no additional configuration from their end. If the server utilizing greylisting is configured appropriately, the end user will only notice a delay on the first message from a given sender, so long as the sending email server is identified as belonging to the same white listed group as earlier messages. If mail from the same sender is repeatedly greylisted it may be worth contacting the mail system administrator with detailed headers of delayed mail.

From a mail administrator's point of view the benefit is twofold. Greylisting takes minimal configuration to get up and running with occasional modifications of any local white lists. The second benefit is that rejecting email with a temporary 451 error (actual error code is implementation dependent) is very cheap in system resources. Most spam filtering tools are very intensive users of CPU and memory. By stopping spam before it hits filtering processes, far fewer system resources are used. This allows more layers of spam filtering or higher throughput since greylisting can easily be configured as a first line of defense with a heuristic filter such as Spam Assassin handling messages that goes through.

Greylisting is particularly effective in many cases at weeding out miss configured MTAs, and is gaining in popularity as a very effective anti-spam tool. It is likely that those MTAs that do not correctly handle greylisting will become less numerous as greylisting spreads.

Some greylisting packages support a SQL backend which allows for a distributed multiple-server frontend to be deployed with the same greylisting data on all frontends.

2.5.2 Limitations of Greylist

The biggest disadvantage of greylisting is that for unrecognized servers, it destroys the near-instantaneous nature of email that users have come to expect. Mail from unrecognized servers is typically delayed by about 15 minutes, and could be delayed up to a few days. A customer of a greylisting ISP cannot always rely on getting every email in a pre-determined amount of time. This disadvantage is mitigated by the fact that near instantaneous mail delivery is restored once a server has been recognized and is generally maintained automatically so long as users continue exchange messages. However, this disadvantage is especially visible when a user of greylisting mail server attempts to reset his credentials to a website that uses email confirmation of password resets. In extreme cases the delivery delay imposed by the greylist can exceed the expiry time of the password reset token delivered in email. In these cases manual intervention may be required to white list the websites mail server so the email containing the reset token can be used before it expires.

Send mail, one of (if not the most) prolific internet message transport agent has a default retry interval of 15 minutes. Generally this is the maximum amount of time an email will be delayed. Experienced system administrators for email systems should tune their mail system settings to sensible values, and the biggest delays from greylisting systems are incurred when communicating with poorly configured sending systems with retry intervals left set at several hours or more.

The original specification for email states that it is not a guaranteed delivery mechanism and not an instantaneous delivery mechanism. This means that greylisting is a perfectly legitimate process and does not break any protocols or rules. Explaining this to users that have become accustomed to immediate email delivery will probably not convince them that a mail server that uses greylisting is behaving correctly.

Modern greylisting applications (such as Post grey) automatically white list senders that prove themselves capable of recovering from temporary errors [25]. Note that this is irrespective of the reputed *spamminess* of the sender.

When a mail server is greylisted, the duration of time between the initial delay and the re-transmission is variable. Some mail servers use a default of four hours, though most will retry sooner. Most open-source MTAs have retry rules set to attempt delivery after around fifteen minutes (Sendmail default is 0, 15, ..., Exim default is 0, 15, ..., Postfix default is 0, 16.6, ..., Qmail default is 0, 6:40, 26:40, ..., Courier default is 0, 5, 10, 15, 30, 35, 40, 70, 75, 80,... Microsoft Exchange defaults to 0, 1, 2, 22, 42, 62 ..., Message Systems Momentum defaults to 0, 20, 60, 100, 180, ...). Indeed, SMTP says the retry interval should be at least 30 minutes, while the give-up time needs to be at least 4–5 days.

Greylisting delays much of the mail from non-white listed mail servers - not just spam - until typical patterns of communication are recorded by the greylisting system. For best results, white listing should be used extensively. A static list of public servers worth being white listed can be found in the greylisting.org repository, though this is significantly out-of-date.

Greylisting can be a particular nuisance with websites that require an account to be created and the email address confirmed before they can be used. If the sending MTA of the site is poorly configured, greylisting may delay the initial email containing the signup confirmation link, thus introducing a waiting period even though the actual website may have attempted to send out the email confirmation code immediately. Almost all stock-configured Sendmail MTAs (sendmail being the most widely deployed MTA on the internet) will retry after a few minutes, leading to typical delays of under 10 minutes, in most cases it is still depended on the greylisting configuration.

In the above mentioned chapter, the existing best methods of spam filtering are described. In the next chapter the proposed method will be described.

Chapter-3: Proposed Method

This chapter discusses about the proposed spam filtering method. The name of the proposed method is given as MAN method. The outline of methodology, considerable email features and possible outcomes of the proposed method are described below.

3.1 Outline of Methodology/Experimental design

Naturally two kinds of emails exist in communication: ham and spam. Behavior of senders, receivers and messages are the considering issues here. Behavior means the feature or characteristics of particular matter or topics. Considering and analyzing the behaviors of emails using data mining tools, it is easy to separate spam from ham. As spam shows many abnormal behaviors in comparison to normal ham messages, these abnormalities help us to identify spam from ham. The flowchart of the new spam filtering technique is given in figure 3.1. This figure shows that the users (senders) send emails to receivers and the email is stored in to the master email database. The master email database is regularly updated based on the open source resources [26]. Also local user defined and proposed system defined white listed and blacklisted emails, domains and IP addresses. The emails go through the pre-filtering process which checks the white listed and black listed IP, domain and email addresses. This allows the spam checking time of email to be significantly reduced as well as the overall time for the email to reach the receiver. Moreover, the pre-filtering process checks the number of recipients of the email. This is done based on how many emails are sent by the sender at a time. It also checks for the number of emails send to a specific receiver on daily basis.

If the email is from white listed email address, the email goes into the ham inbox folder directly, otherwise, it will check for black listed IP, domain or email addresses. If the email matches with the black listed addresses, then the email directly goes to spam inbox folder. If it neither matches the black listed nor the white listed addresses, it will go through spam filtering process.

In the Spam filtering process, the process checks the subject of the email, the message body of the email in order to detect the spam with prioritization based subject length, mixed capital and small letter in subject line, specific words in subject and body, number of images, web links, image criteria, etc. in email body.

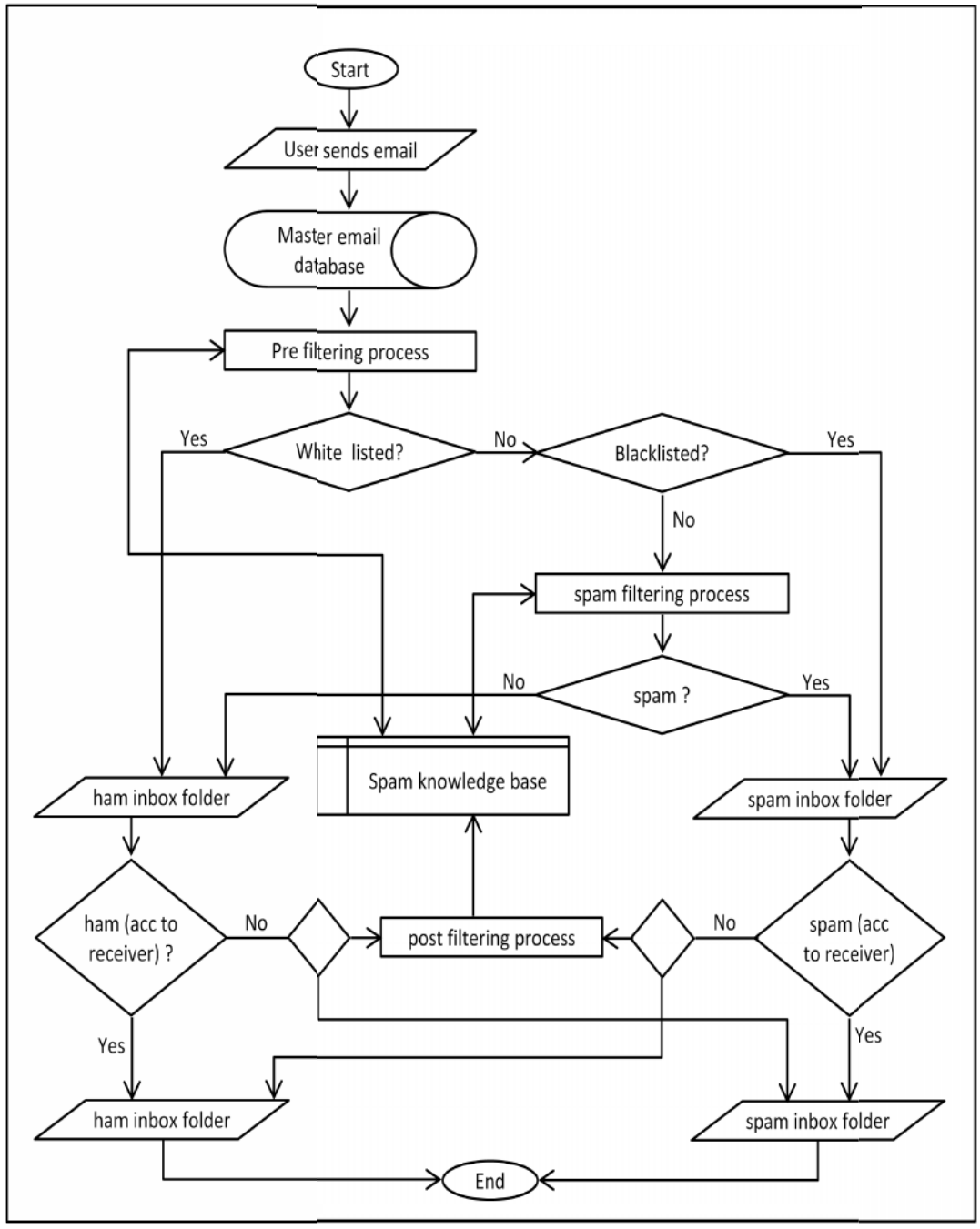

Figure 3.1: Flow chart of the proposed spam filtering method

If it is detected as spam based on these criteria, the email will go to the spam inbox and otherwise it will go to the ham inbox. The receiver will check the emails and will

detect whether it is a spam or a ham. The receiver will check the spam inbox and if the email is necessary for him, he will tag it as ham. This email address will be considered as white listed address for that specific receiver in future. This will reduce the time of spam processing. On the contrary, the ham inbox is checked and if the email is found to be useless to him, he will tag the email as spam. These email addresses will be sent as a reference to the knowledge base and will be updated as spam or ham for the future. So, from next time around, these emails will be detected as spam or ham as the receiver considers these emails to be like that.

If the percentage of a specific email address is detected as a spam by the receiver that reaches a certain significance level, the email is considered to be blacklisted by the email server in the future for all receivers. This method is known as post spam filtering method. The advantage of this post spam filtering method is that this method enhances receiver based accuracy. This accuracy will be detected compared with the other well known methods.

3.2 Proposed Spam filtering model

A pictorial representation of the proposed spam filtering method is given in figure 3.2. The figure depicts the proposed spam filtering model. This specifically demonstrate the updating method of the spam filtering process based on white listed and black listed region, analyzing pre-filtering based on sender behavior, spam filtering based on email message body, and post filtering based on receiver behavior. We use four separate spam filtering engine to connect with the central knowledge base. It will also increase the performance of the email server and will reduce the process time. The data are stored in the knowledge base that it uses support and confidence rule in order to find out the spam and ham emails. The confidence rule used as {Upper and lower case letters in email subjects, length of the subject is between 70 to 80 characters}□ {spam email} has a confidence of 0.7 which is heuristically found. This means that if these two criteria are included, there is 70% likeliness for the email to be a spam. The Apriori algorithm is used for this purpose containing two steps such as finding all frequent item sets, and then using frequent item sets to generate rule.

The post filtering method is based on the detection of spam and ham on the choice of receivers. Suppose, a single malicious email has come to 100 receivers. Among them, 60 receivers considered that email to be malicious whereas (100-60)= 40 of the receivers

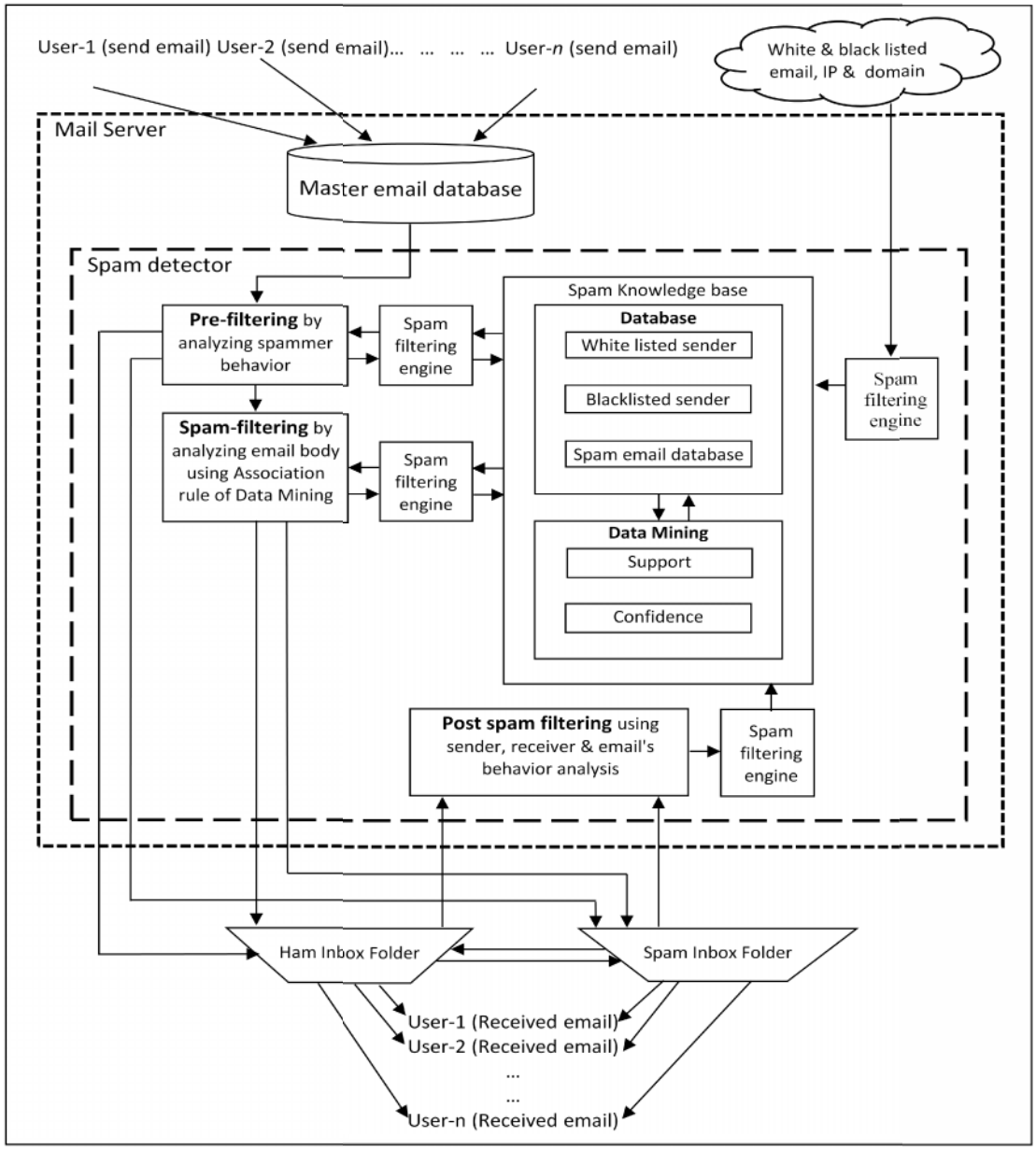

Figure 3.2 : Proposed Spam filtering model

do not take any action. In this case, the email is tagged as spam by the email server and that email will go to the receivers as spam in future by adding the email address to be black listed. Here the post filtering can be tagged as PF. The PF is detected on the ratio of the total number of users considering the emails to be spam divided by the total number of receivers receiving the same email. If PF is greater than 0.5, then

the email is considered to be malicious for the rest of the users in future. The vice versa case applies in the case of spam email where the spam is considered as ham for the receivers. So, Post Filtering, $PF = \frac{N_A(S_A|H_A)}{N}$

Where, N_A= Number of receivers taking action

N= Total number of receivers from a single sender email

S_A= Action taken to include the email in spam inbox

H_A= Action taken to include the email in ham inbox

Thus post filtering can be used in order to detect the spam and improve the accuracy for the proposed spam filtering method.

3.3 Algorithm: Process Prioritization

The algorithm of the process prioritization that is responsible for reducing the process time from others:

Set UTYPE = Process update sequence type in database
Set SYSTIME = Current System Time, UTIME = Auto update time in database
If UTYPE = Manual then
    Input sequence for each process
    Update process priority database
Else
    If SYSTIME = UTIME then
        [Load process list, current priority, total spam detection]
        PROCESS <- All process
        PSEQ <- Process sequences
        PSPAM <- No of spam detection after last sequence update by the processes
        WHILE N = 0 to *COUNT*(*PROCESS*)
            INDX = index of *MAX*(*PSPAM*)
            Set PSPAM[*INDX*] = 0 [Spam count reset]
            Set PSEQ[*N*] = *INDX*
        END WHILE
        Update priority database by the array PSEQ
    End if
End if

3.4 Algorithm: Post Filtering Method

The algorithm of the post filtering method that will improve the method is as below:

```
Procedure Spam_By_Post_Filtering (EMAIL, EMAILTYPE, SENDER, RECEIVER)
    If EMAILTYPE = HAM then
        R_ACTION = Get Receiver's Response
        If R_ACTION =1 then [1: Receiver Marked as SPAM, 0: No Action by receiver]
            Move EMAIL to SPAM inbox
            Add SENDER address to BLACK_LIST for RECEIVER
            RCOUNT = Number of receivers of the EMAIL
            MOVECOUNT = Number of receivers marked EMAIL as SPAM
            If (MOVECOUNT*100)/ RCOUNT >50 Then
                Add SENDER address to BLACK_LIST for all receivers
                under this email server
            End If
        Else
            COUNT = Count EMAIL in HAM inbox
            If COUNT=3 then
                Add SENDER address to WHITE _LIST for RECEIVER
            Else
        End if
    Else
        R_ACTION = Get Receiver's Response
        If R_ACTION =1 then [1: Receiver Marked as HAM, 0: No Action by receiver]
            Move EMAIL to HAM inbox
            Add SENDER address to WHITE_LIST for RECEIVER
            RCOUNT = Number of receivers of the EMAIL
            MOVECOUNT = Number of receivers marked EMAIL as HAM
            If (MOVECOUNT*100)/ RCOUNT >50 Then
                Add SENDER address to WHITE _LIST for all receivers
                under this email server
            End If
        Else
            COUNT = Count EMAIL in SPAM inbox
            If COUNT=3 then
                Add SENDER address to BLACK_LIST for RECEIVER
            Else
        End if
    End if
End Procedure
```

Some of the considerable email message features, descriptions and examples are given in table 3.1 in a tabular form.

Table 3.1: Email features, description and examples

Features	Description	Examples
Number of emails send at a time	It is very suspicious to send more and more email at a time.	If the number of sending email > 50 at a time, then it is obviously a spammer email account.
Number of unique sender addresses.	Many users have multiple active user accounts and they open these accounts on the same machine consecutively.	Sending messages in various addresses at a high rate from a single PC is also an indication of abnormality.
Number of words and characters in the subject line.	Spammer used more and more words in subject line and they also mixed capital and small letters in a single word. This is also notified point.	There are specific words and characters used for spam detection. If the number of characters > 70 in subject line, then it would be considered as a spam email.
Percentage of capital letters in the subject line.	Spammer used capital words in subject line and mix-ups capital and small letters in a single word. This is also notified point.	If the percentage of capital letters (except first letter of the first word) > 30 in subject line, then it would be considered as a spam email.
Presence of images	Using images in the body of email is not usual character for normal messages. Sometimes may be one or two images can be used. But more images in the email body prove its abnormality.	If the number of images > 5 then it would be considered as a spam message.

Number of hyperlinks	Hyperlinks are important characteristics of spam detection. Spam message contains really more and more hyperlinks and spam usually goes with them. It is also usual to send hyperlinks in ham but it is not usually more than 5.	If the number of hyperlink > 5 then it would be considered as a spam.
Destination of hyperlinks	Spammer uses more hyperlinks those indicate the blacklisted domain or IP addresses.	If the hyperlinked destination addresses indicate the blacklisted domain then this email will be considered as a spam.
The set of distinct word frequently	Spammer uses some certain words in their email body and subject line. So these words are the identifier of spam messages.	Words like get free, loss over weight, free training, save up to, world class, read it, protect your family, exciting career, etc.
Requesting secrete information	A special case of spamming activity is phishing, namely hunting for sensitive in-formation by imitating o□cial requests from a trusted authorities, such as banks, server administration or service providers [27]	Asking password, credit card numbers, etc.
Receiver actions towards the emails	Normally users can't receive more than 10 emails in a day from a single account. Also most of the users don't read or open some unwanted mails.	If most of the recipients of an email don't read or open the specific email from a specific sender then the mail must be a spam and sender must be a spammer.

3.5 Possible outcomes

a. At least 99% of the spam will be detected and filtered.
b. Lower complexity of behavior analysis algorithm will reduce the filtering time at least by 10%.
c. The percentage of false positive expected to reduce to almost 0%, that is, there will be almost no spam wrongly classified, which overwhelms the best existing Bayesian Spam Filter which has false positive of 1.16% [28].
d. Based on the number and percentage of hams that from a trusted region, the pre-filtering mechanism will be discarded, so that the process time will be further improved.

In the next chapter, the design and implementation details of the proposed method will be described with the aid of figures.

Chapter-4: Implementation

In this chapter, the focus will be on the performance analysis and different output that has been generated using PHP (preprocessor hypertext). Mysql has been used for the storage of the data.

4.1 Black-listed IP, domain and email addresses

The proposed method will filter the black listed email address and domain. The black listed email address and domain will be used in order to filter out the unsecured zone of the email address. The receiver himself can consider a blacklisted email address to be a white listed email address for him. Figure 4.1 is the interface that shows the blacklisted domain, IP and email addresses.

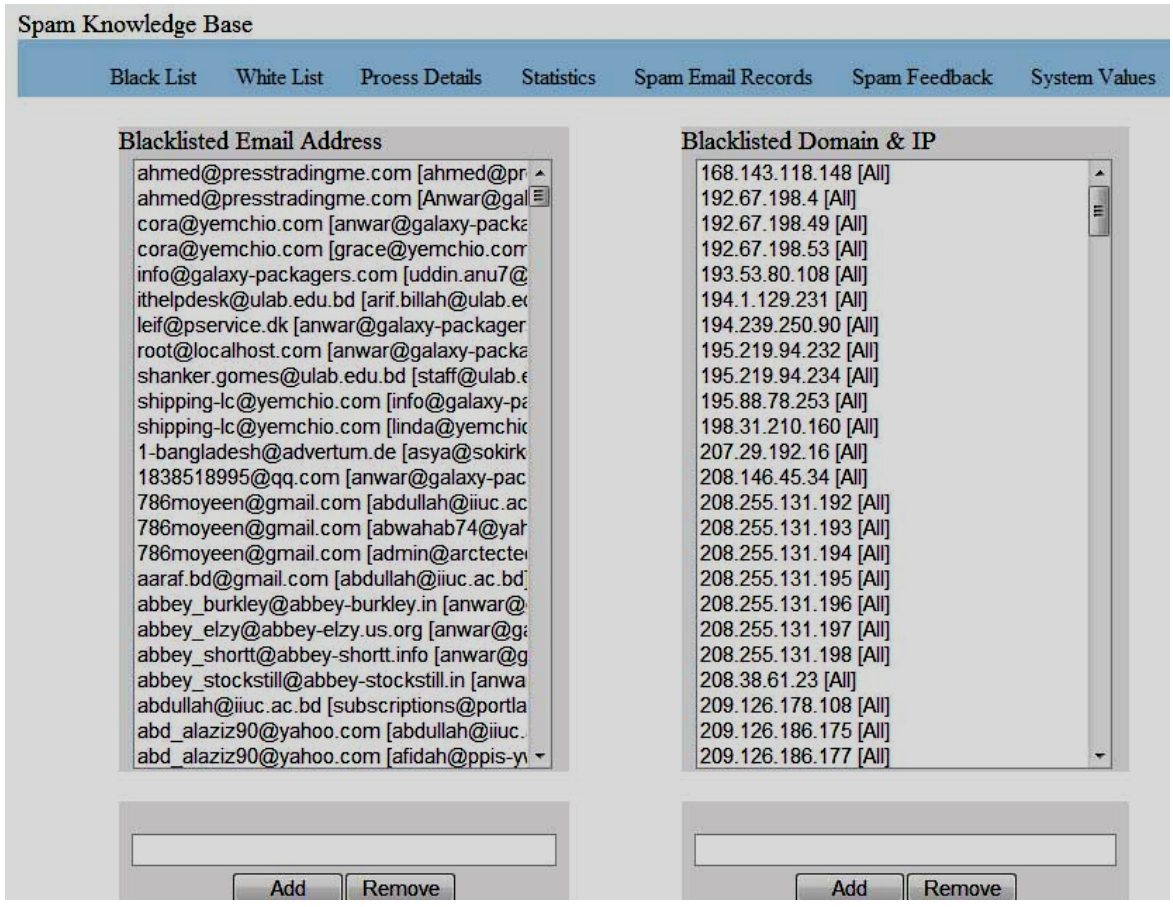

Figure 4.1: Black-listed IP, domain and email addresses

4.2 White listed IP, domain and email addresses

White listed domain, IP and email addresses are secured zone from where valid emails can come to the receiver. The white listed email addresses can also be made blacklisted by the receiver if the receiver is not intending to keep the email or the email is absolutely un-necessary for him. As user has the right to participate in the spam filtering process, this checking mechanism of the proposed method makes the proposed method much more accurate than the conventional methods.

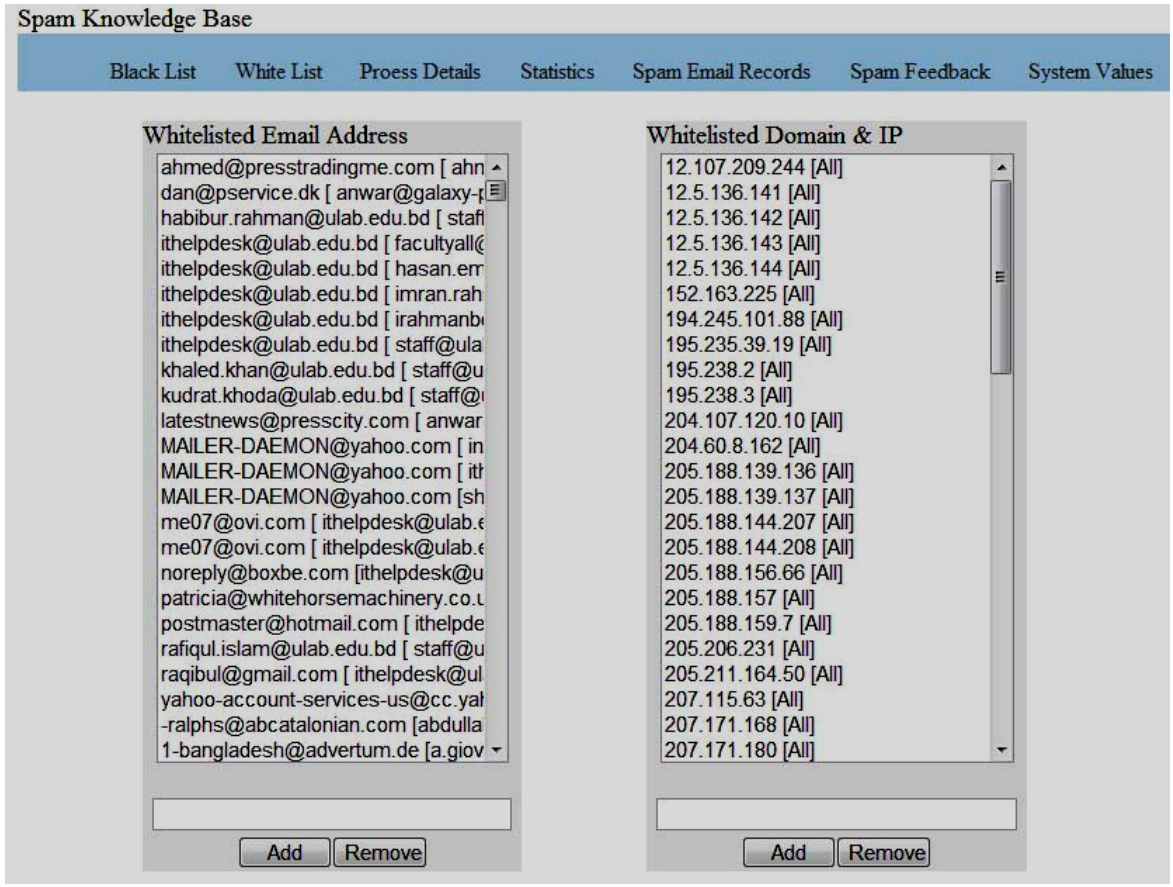

Figure 4.2: White listed IP, domain and email addresses

4.3 Email Queue

The email queue uses the FIFO approach where the emails sent by the sender are stored in the email queue and the first email goes out first from the queue. The figure 4.3 shows the interface of the email queue where email is stored. This figure shows the screen shot of the email queue. There have an option for the emails unchecked will be deleted after twelve months.

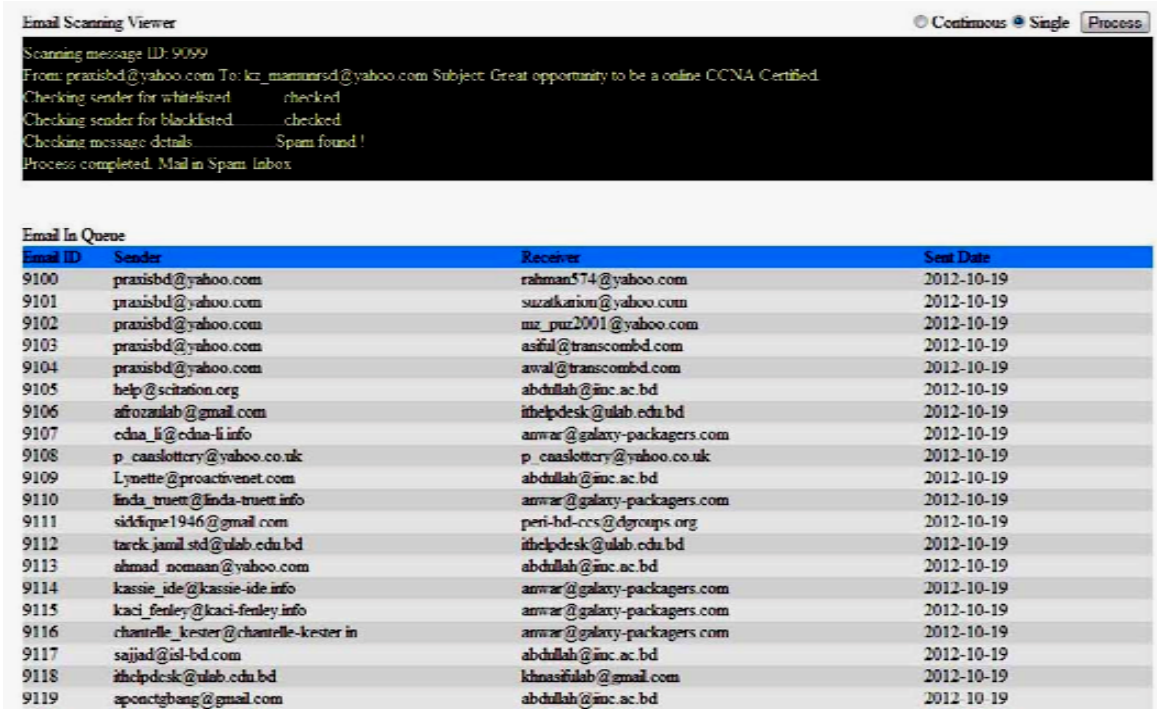

Figure 4.3: Email queue

4.4 Ham inbox

Ham inbox is the place where the receiver checks the trusted emails. The figure 4.4 indicates the ham inbox where the receiver checks for the authorized emails.

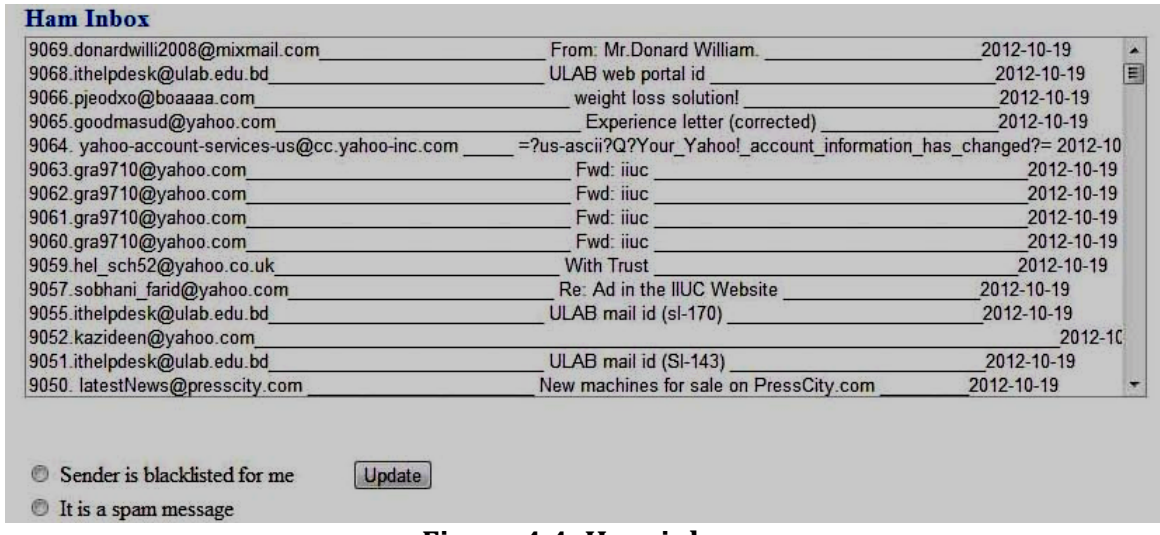

Figure 4.4: Ham inbox

In the ham inbox, as it is seen there are two options, the receiver can black list the email and send a white listed email address form ham inbox to spam inbox as it might be useless for him. So, the receiver can black list the email address.

4.5 Spam inbox

The spam inbox is the place where generally the blacklisted email address will be directed and the receiver will not read those emails considering the emails to be junk and useless emails.

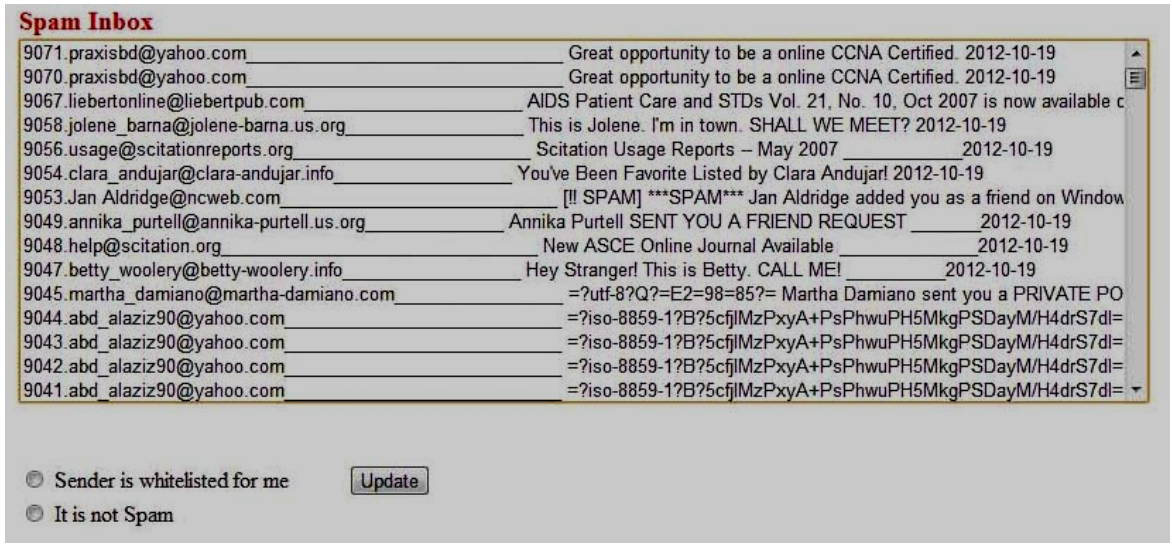

Figure 4.5: Spam inbox

Similarly, like ham inbox, in the spam inbox, the receiver can check the spam folder and tag a spam to be a ham for him based on subject of the email. This user checking mechanism in the proposed method makes the method more accurate than the existing methods.

4.6 Process prioritization and auto update

The prioritization of the process for checking the spam by default is based on the criteria of detection spam by sender behavior termed as sequence of priority-1, detection of spam by receiver behavior termed as priority-2, detection of spam considering the number of images and the criteria of images termed as priority-3, detection of spam considering the number of hyperlink and the linked addresses termed as priority-4, spam by subject, discrete words and secret information respectively termed as priority-5,6, and 7. The screenshot depicts it in figure 4.6.

The number of detected spam based on the above criteria changes the sequence of the spam detection process. The auto update feature changes the sequence or

priority of the spam detection process automatically. Thus, spam can be detected based on the criteria of the auto update policy.

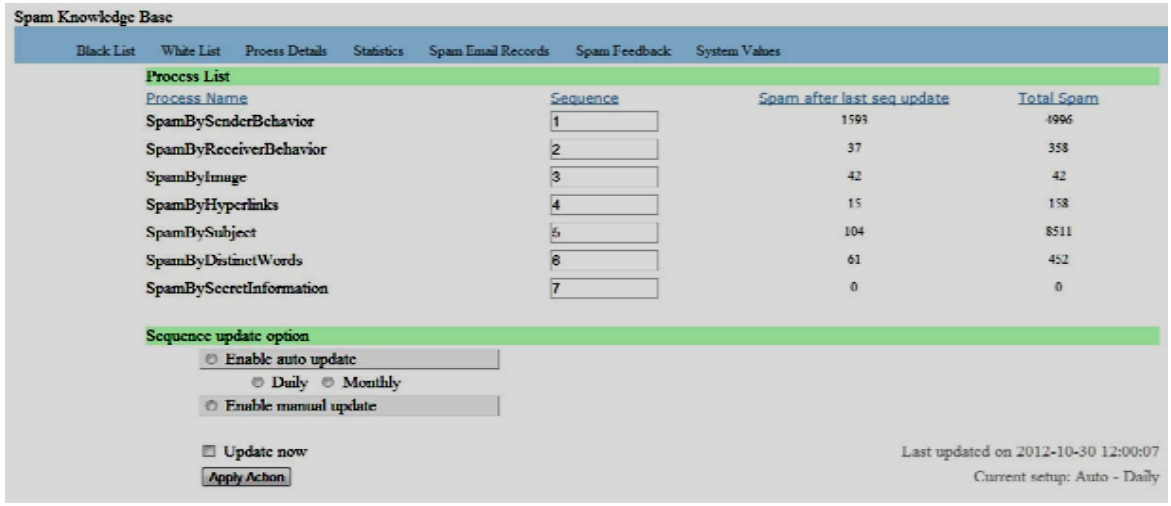

Figure 4.6: Prioritization for detecting spam

4.7 Statistics detected by the proposed method

The spam detection method is carried out based on 70,053 email messages. As the checking mechanism is done by the receiver based on black listed and white listed email addresses, the accuracy of the overall proposed method improves and overwhelms

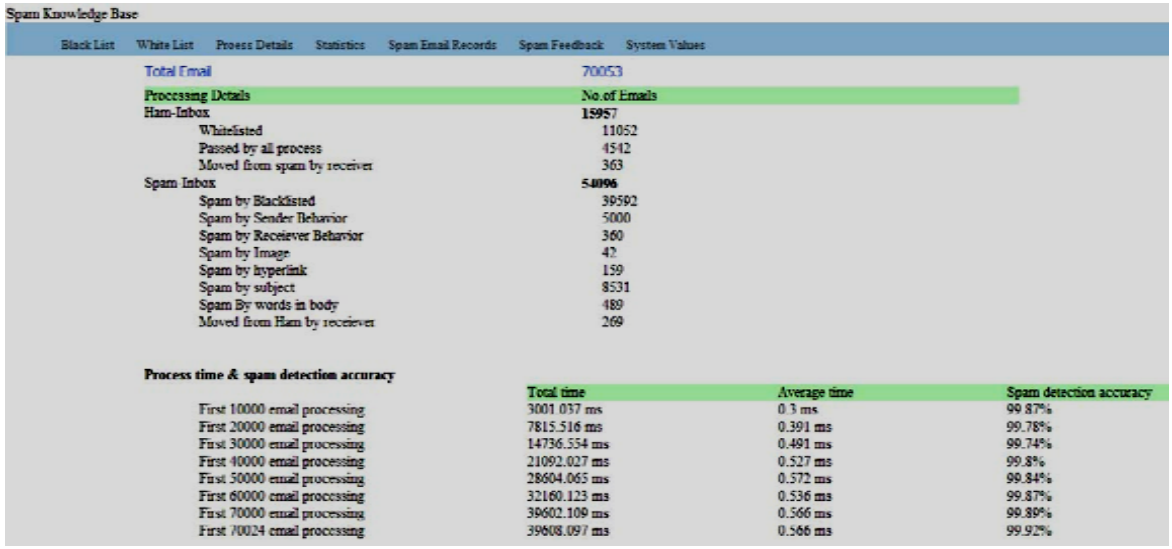

Figure 4.7: Statistics detected by the proposed method

the existing methods. It is also observed that as the number of email messages increases, the accuracy improves as well as the overall time. The reason behind this

is because the black listed and white listed email addresses are already checked and updated by the receiver and therefore, the other criterion for spam detection is not required to check. That is the reason why the overall spam detection accuracy increases and processing time of the method decreases. In the figure 4.7 the accuracy and time for different number of emails are showed:

4.8 Spam email records

It is observed from the figure 4.8 that sender, subject, receiver, spam type, etc. are used in the knowledge base. A unique message ID is used for every emails. Data mining rules are associated in to the Knowledge base.

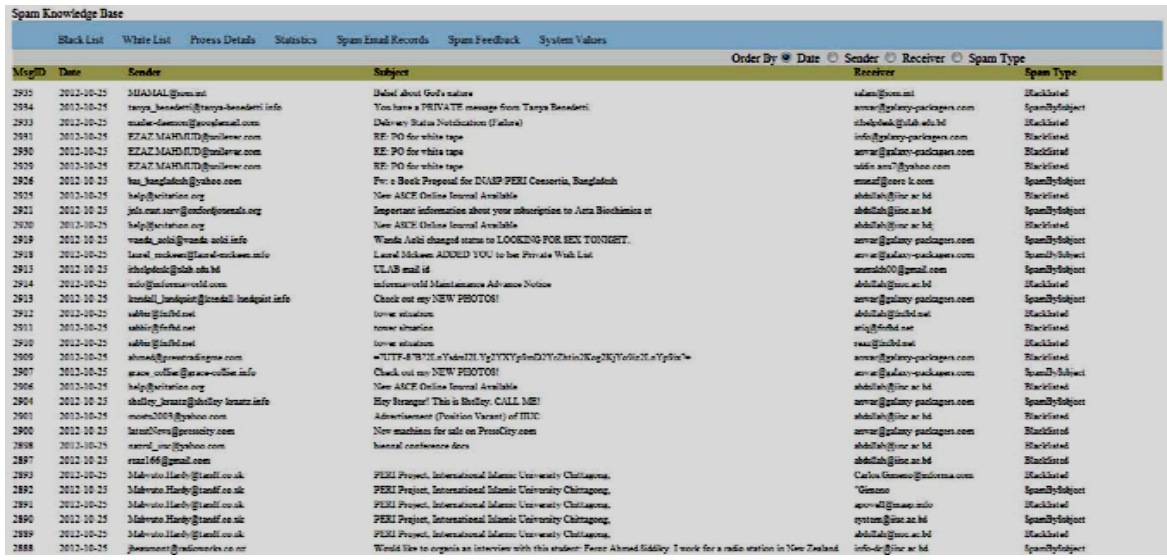

Figure 4.8: SPAM email records in knowledge base

This chapter concludes the software implementation process for the proposed MAN method. Next chapter will show the result of the proposed method and comparison among the other existing methods.

Chapter-5: Performance Analysis

In this chapter there will be shown the comparison between the proposed and existing methods. Here we will see the performance analysis among the existing and the proposed method using a large number data set. Also the comparison using the same data set among presently used well known software and the proposed method.

5.1 Heuristic detection of spam email criteria

The criteria to detect optimum number of characters in order to determine the maximum number of spam is carried out for 15,000 data sets in table 5.1

Table 5.1: Spam detection rate based on number of characters in subject.

No. of characters in subject	Spam detection (%)
10	10
20	25
30	40
40	60
50	80
60	90
70	97
80	95
90	90
100	85

It is observed that if the number of characters in the "SUBJECT" area is between 70 and 80 then the message is mostly detected as spam with maximum accuracy. This is done for 70,053 emails. The following user defined formula is used for this purpose:

Number of optimum characters =MAX ($MAXIMUM$ ($SPAM\ DETCETION(N)$));

Here $MAXIMUM$ ($SPAM\ \ DETECTION\ \ (N)$) is a subroutine call that detects the maximum number of spam and MAX indicates the maximum number of occurrence of characters in order to detect the maximum spam.

When the method is applied for several messages, it is observed that the character length of 70 to 80 is optimum for the detection of spam mostly.

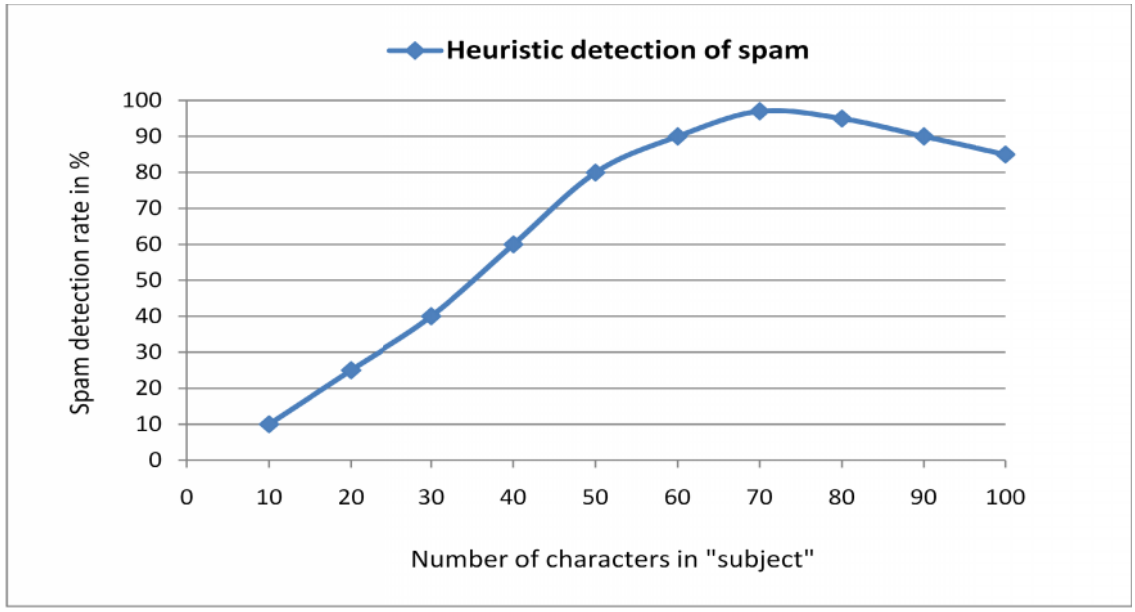

Figure 5.1: Number of optimum characters in subject to detect spam

5.2 Receivers' feedback

In the figure 5.2, receiver detects whether the email is ham or spam for his or her convenience. If the receiver thinks the email to be spam, he or she checks it as spam and if necessary checks it as ham which is added to white list and black listed email

Spam Knowledge Base

Black List | White List | Proess Details | Statistics | Spam Email Records | Spam Feedback | System Values

Process Name	Process detected as Spam but receiver marked as Ham
SpamBySenderBehavior	133
SpamByReceiverBehavior	3
SpamBySubject	115
SpamByImage	0
SpamByHyperlinks	2
SpamByDistinctWords	3
SpamBySecretInformation	0

Figure 5.2: Receivers' feedback after getting email

addresses based on user convenient. If more than 50 % of the receivers consider the email messages to be spam, then those email messages will be added as black listed email address and the vice versa is also applied. All these are done in the post filtering phase.

5.3 Optimum system values

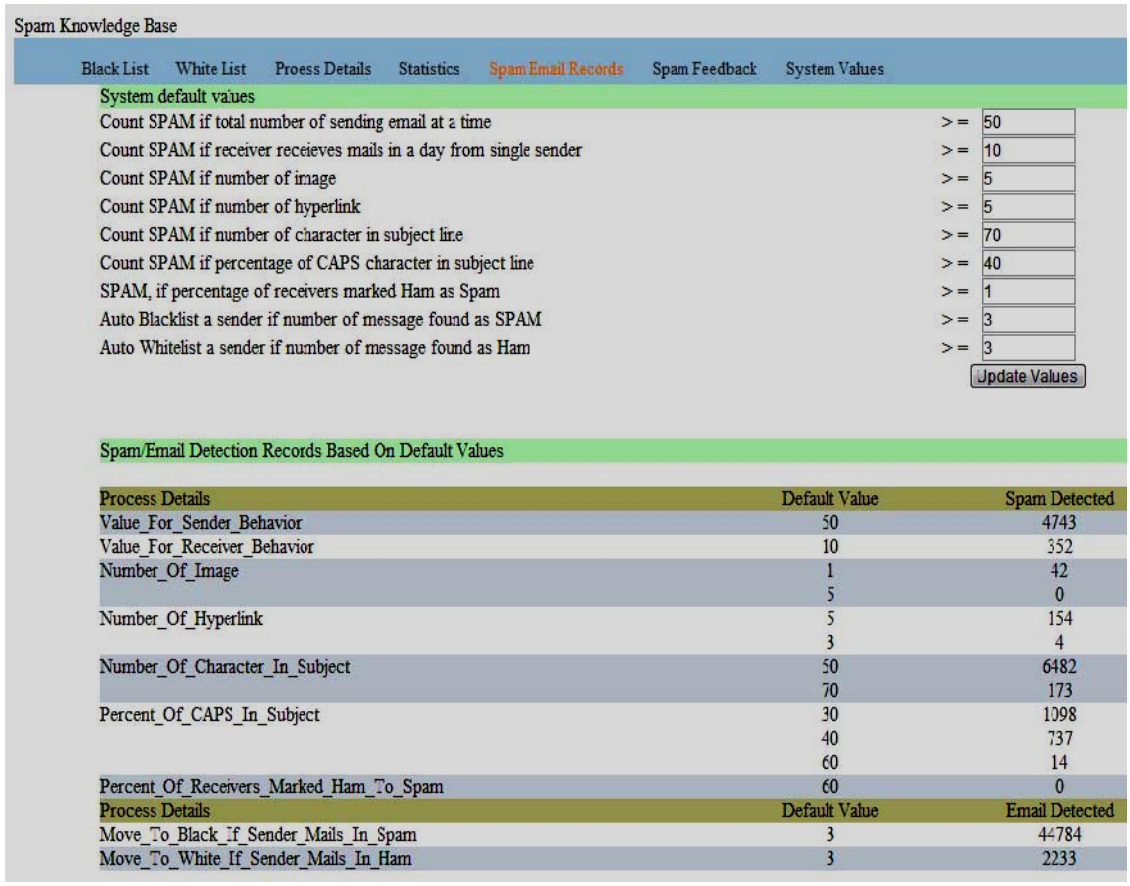

Figure 5.3: Optimum system values

The optimum system values are used in order to detect spam. Thus different criterion is used in order to detect the email messages as spam.

5.4 Accuracy for different number of emails

The accuracy of the proposed MAN method increases as the number of emails increase. The reason behind is that there is a receiver customization as well as the post filtering implementation. There is an enhancement of knowledge base (KB) which updates the black listed and white listed email addresses. The accuracy of the overall proposed MAN method overwhelms the other existing methods. If 60% of receiver says that a specific email is spam it is marked as black listed and the vice versa is true if the email is marked as white listed.

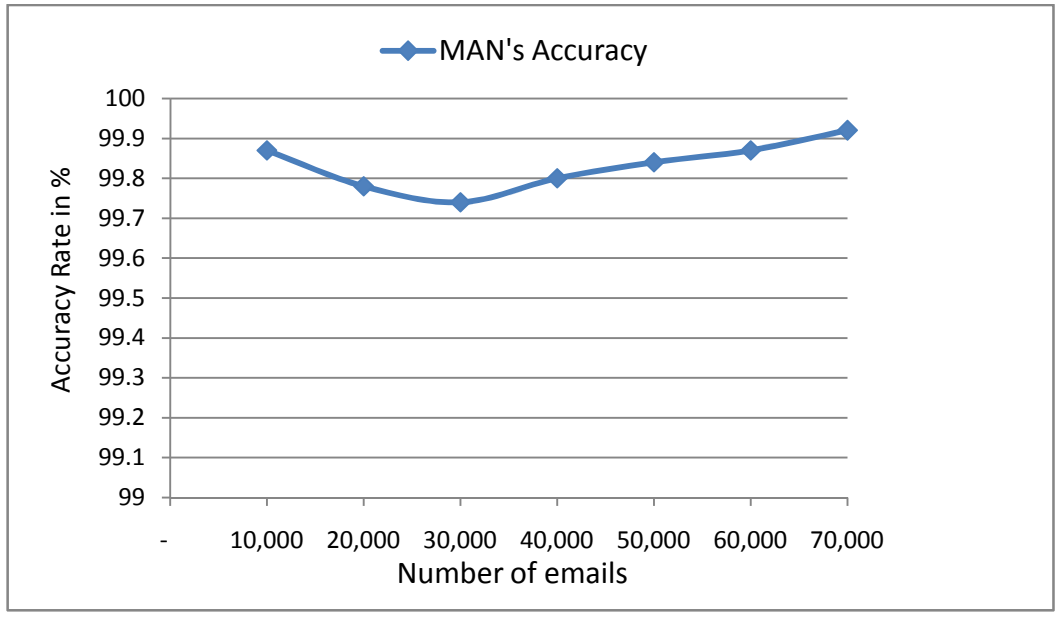

Figure 5.4: Accuracy for different number of emails using the proposed method

5.5 Post filtering method analysis

The proposed MAN method uses post filtering method to overwhelm the other existing methods and performs better in terms of accuracy and time. The time of the proposed method decreases as well compared to the other existing methods as the knowledge base is automatically updated by the user and this consideration is not being used by the earlier methods. This causes the black listed and white listed email addresses to be updated. So, as a whole the time and accuracy of our proposed MAN method is better than any other existing methods.

5.6 Comparison with the existing methods

The outcome of the proposed method is compared with the existing Bayesian and Naïve Bayesian approach and the following result was found. The accuracy is computed based on 70,053 emails.

Table 5.2: Performance analysis among the existing and proposed method

Features	Bayseian spam filter	Improved Bayseian approach	Naïve Bayseian approach	Meta spam filter	Greylist approach	Proposed method
Spam detected accuracy	98.00%	99.10%	97.30%	98.60%	96.00%	99.92%
False positive	1.16%	0.46%	1.20%	1.63%	3.50%	0.10%

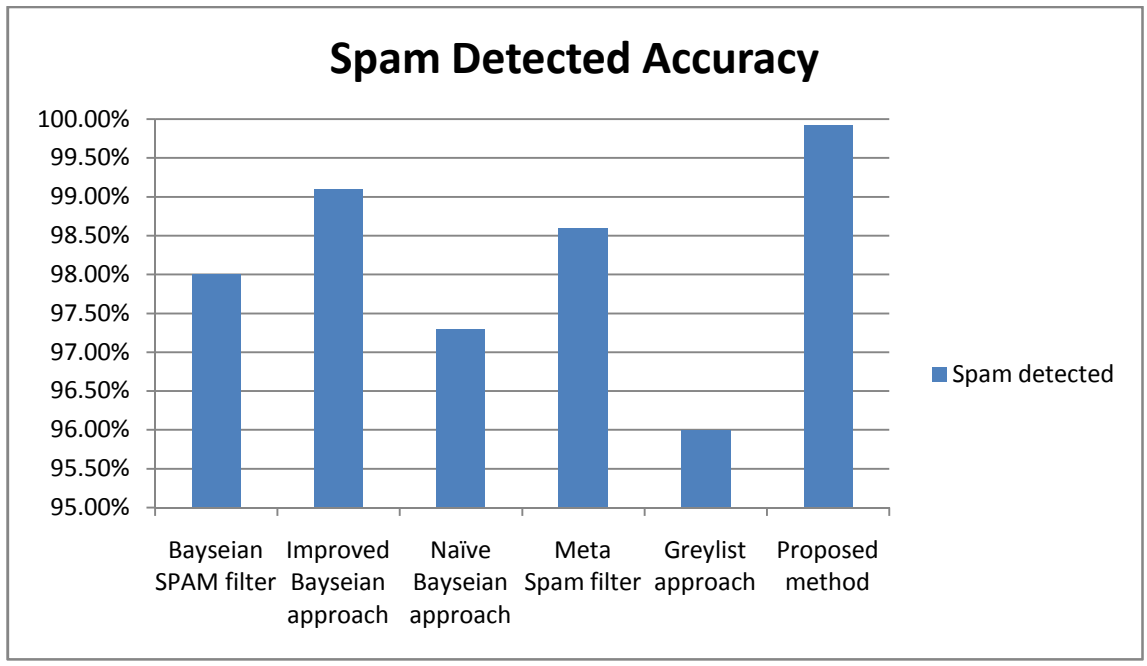

Figure 5.5: Performance analysis on the basis of spam detection

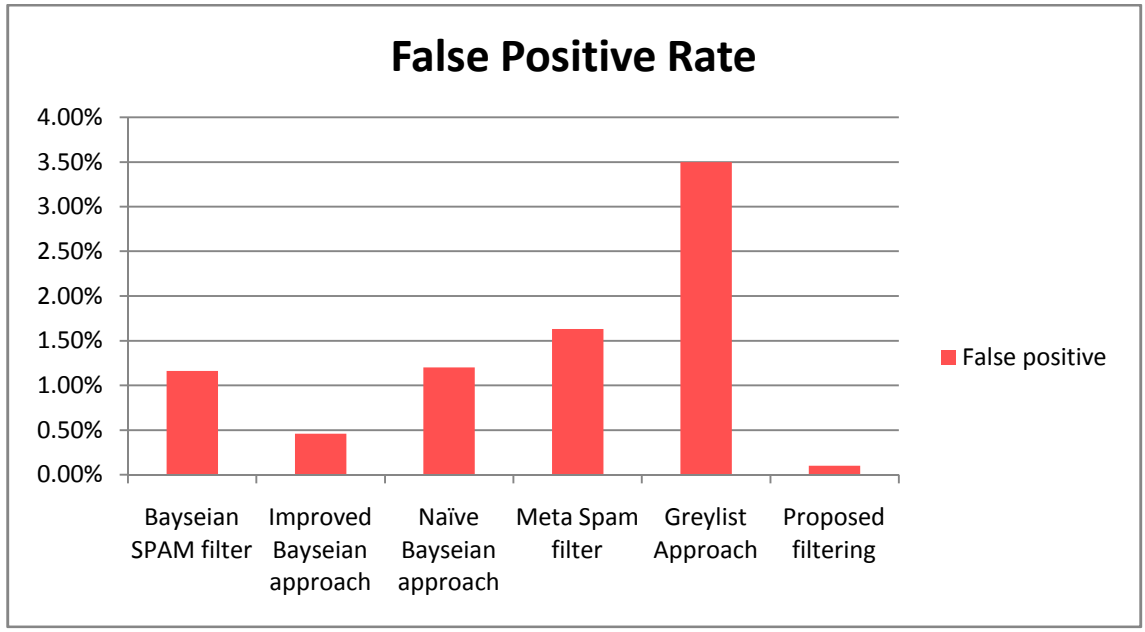

Figure 5.6: Performance analysis on the basis of false positive

5.7 Comparison with the existing software

The outcome of the proposed method is compared with the current version of Windows Live Mail 2011 (Build 15.4.3555.0308) & Gmail and following result was found. The accuracy is computed based on 8000 same data set.

Table 5.3: Performance analysis among the existing and implemented software using common data set

Features	Windows Live Mail	Gmail	Proposed method	
			Before Post Filtering	After Post Filtering
Spam detected accuracy	98.75%	99.47%	99.87%	99.92%
False positive rate	2.5%	0.26%	0.46%	0.1%

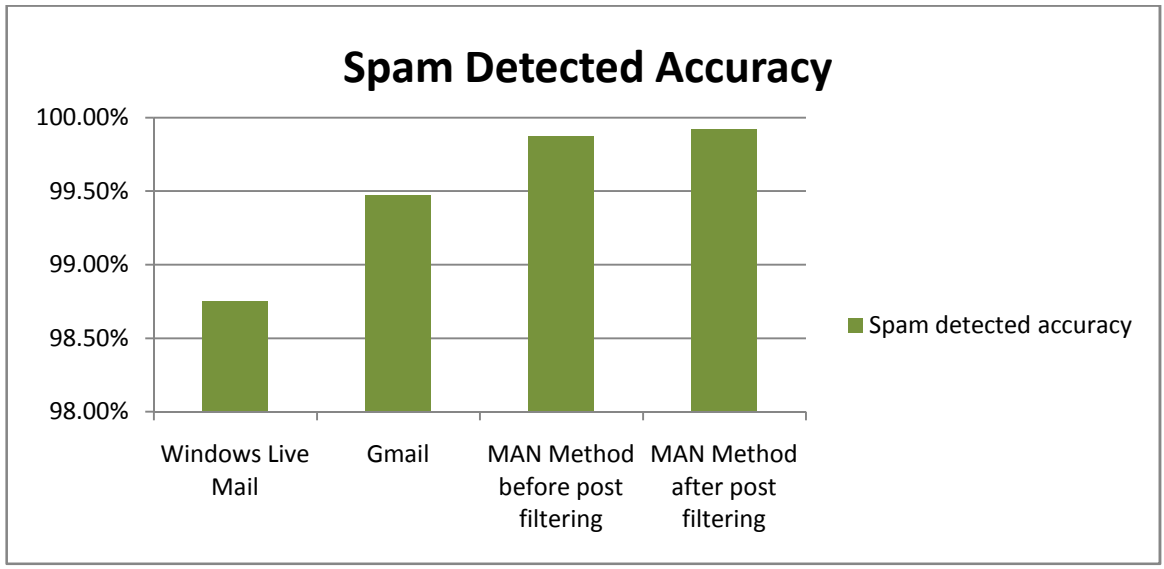

Figure 5.7: Spam detected accuracy using common data set

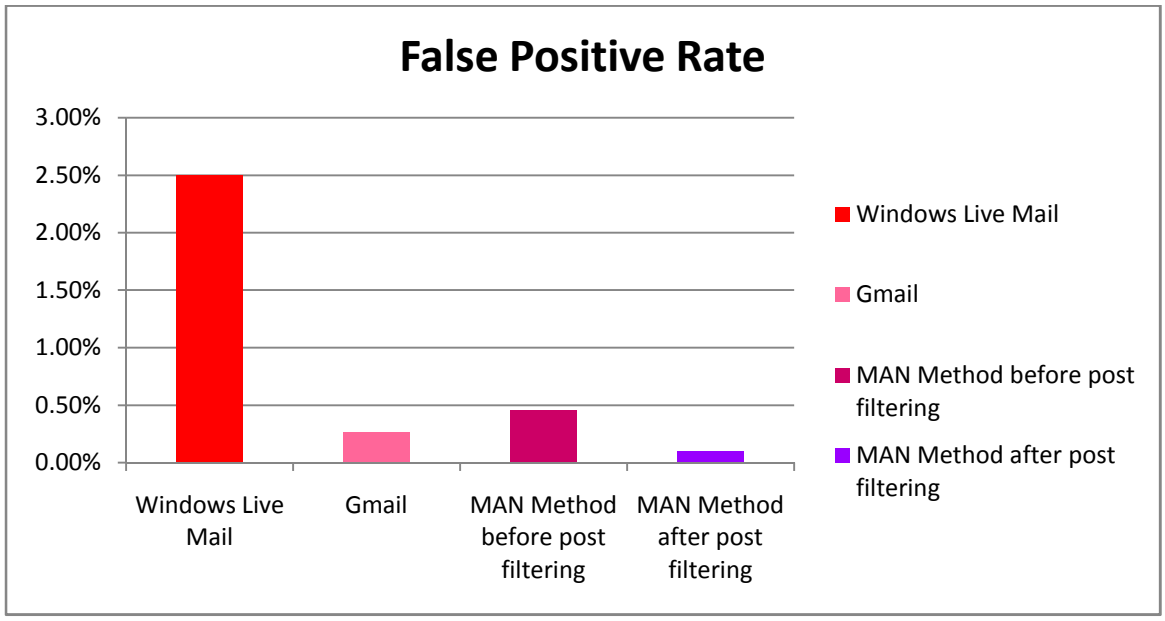

Figure 5.8: False Positive rate using common data set

5.8 Observation from the output

From all the figures above, it is seen that the proposed spam filtering method works better and overwhelms the performance of the existing method. The features and the parameters used in order to detect the performance analysis of the methods are spam detected, hams classified and the false positive, i.e wrongly detected spam by the method. It is observed that the spam detected by the proposed method is higher and performs better than the existing one. The checking mechanism of detecting the spam is carried out through 10,000 to 70,000 email messages and the proposed method was able to detect almost 99.92% of the spam. It is also observed that, the proposed method detects hams correctly, finds out the spam detected and has almost zero false positive (wrong detection of spam) which indicates the authority of the method over the other four spam detection.

So, in short, it can be said that the proposed method is able to detect spam better and able to provide user comfort.

It is observed from this chapter that the proposed method works better than the existing method in terms of detecting spam. In the next chapter, the discussion will be carried out through the further improvement of the proposed spam detection method.

Chapter-6: Conclusions

6.1 Discussion of Results

From the chapter above, it is noticed that the proposed method of spam detection overwhelms the other existing method in terms of spam detection, ham detection and false positive. Our proposed method also takes lesser time than the conventional methods of spam detection.

Email has become parts and parcel of our everyday life. Making it efficient saves significant amount of time from each of our lives. Due to it critical role in saving our time we selected the topic and came out with the idea of introducing MAN. We have successfully demonstrated the better capability of MAN in comparison to two other methods. The best anticipation and greatest satisfaction would be to put the proposed method into the real life after incorporation of the suggested improvement in the earlier paragraph. Nonetheless, we are sure that this project will be able to contribute further in the area of developing an efficient spam filter tool.

6.2 Future Works

However, there is also room for improvement on this thesis work. The concept of sender authentication with confidentiality, availability and integrity can be added to ensure the security to the receiver. Moreover, an appropriate algorithm can be used for this purpose. The knowledge base can be used to derive the age, gender, preference, area of the receiver. Based on the age, gender, preference, area of the receiver, clustering can be used in order to find out the emails that are considered to be valid to the same age group, gender, preference and area of the receiver. The spam can be used as outliers or noise. The Grid-Partitioning-Around-Medoids method can be used for this purpose, which provides less time complexity and greater accuracy. Thus, the spam will not be considered for the receiver in future. In this away, a more accurate and better spam detection method can be developed.

References

[1] *Radicati Sara, "Email Statistics Report, 2009-2013", The Radicati Group, Inc ., 2009*

[2] *Klensin J., "Technical Report RFC 2821, IETF, Simple mail transfer protocol", Network Working Group, October 2008*

[3] *Denning, P. J. "Electronic junk. Communication of the ACM", Purdue University, India, 1982, 25(3):163–165.*

[4] *Ducheneaut, N. and Bellotti, V. "E-mail as habitat: an exploration of embedded personal information management. Interactions", 2001, 8(5):30–38.*

[5] *Mackey, W. E. Diversity in the use of electronic mail: A preliminary inquiry. In ACM Transactions on Information Systems, 1988, volume 6.*

[6] *Xing Liu, Yueheng Sun, "An Adaptive Spam Filter Based on Bayesian Model and Strong Features", School of Computer Science and Technology, Tianjin University, China, IEEE, June 2012.*

[7] *Sangeetha C., Amudha P., Dr. Sivakumari S., "Feature Extraction Approach For Spam Filtering", 2012, ISSN NO: 6602 3127, IJART, Vol. 2 Issue 3, pp 89-93.*

[8] *http://eval.symantec.com/mktginfo/enterprise/other_resources/b-tate_of_spam_report_09-2009.en-us.pdf, accessed on: 23-Aug-2012.*

[9] *A. Zdziarski Jonathan, "Bayesian Noise Reduction: Contextual Symmetry Logic Utilizing Pattern Consistency Analysis", accessed on 14-Aug-2012.*

[10] *https://mail.google.com/mail/help/intl/en/fightspam/spamexplained.html, accessed on 10-Oct-2012*

[11] *Knill David C. and Pouget Alexandre, "The Bayesian brain: the role of uncertainty in neural coding and computation", Center for Visual Science and*

the Department of Brain and Cognitive Science, TRENDS in Neurosc iences Vol.27 No.12 December 2004, University of Rochester, NY 14627, USA.

[12] *Hu Yin, Zhang Chaoyang, Hubei, China, on "An improved Bayesian Algorithm for Filtering Spam E-mail", Network Center Huanggang Normal University Huangzhou, International Symposium on Intelligence Information Processing and Trusted Computing, IEEE, 2011.*

[13] *Zhang Harry, "The Optimality of Naive Bayes", Faculty of Computer Science , University of New Brunswick Fredericton, 2004*

[14] *Caruana, R.; Niculescu-Mizil, A. "An empirical comparison of supervised learning algorithms". Proceedings of the 23rd international conference on Machine learning. 2006*

[15] *Gerard Lynch, Erwan Moreau and Carl Vogel, "The Innovative Use of NLP for Building Educational Applications", Centre for Next Generation Localisation Integrated Language Technology Group School of Computer Science and Statistics, Trinity College Dublin, Ireland, June 2012, pages 257–262,*

[16] *Domingos Pedro, Pazzani Michael , "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss", Department of Information and Computer Science, University of California, Irvine, CA , 1997, Machine Learning, 29, pp 103–130*

[17] *Esquivel Holly and Akella Aditya, Tatsuya Mori, "On the Effectiveness of IP Reputation for Spam Filtering", IEEE, Year 2010.*

[18] *http://www.freebsd.org/cgi/man.cgi?query=hosts_access&sektion=5. Accessed on 10-Sep-2012*

[19] *Symantec Turn Tide Anti Spam Router, "Fighting Spam With A Multi Layered Architecture, White paper Enterprise Solution", Symantec Corporation, October 2004*

[20] *B. Templeton, "Proper Principles For Challenge/Response Anti-Spam Systems;" http://www.templetons.com/brad/spam/challengeresponse.html, accessed on 10-Aug-2012.*

[21] *Parrott Tom, "SPAM Filtering Proxy Server", Department of Electronic and Computer Engineering, University of Portsmouh, 2006.*

[22] *Ojha Gaurav, Kumar Tak Gaurav, "A Novel Approach Against E-Mail Attacks Derived From User-Awareness Based Techniques", International Journal of Information Technology Convergence and Services (IJITCS), August 2012, Vol. 2, No. 4.*

[23] *Cain Matt, "Spam Filter Testing Best Practices Content & Collaboration Strategies", 02-Jan-2005.*

[24] *M. Kucherawy, D. Crocker, Brandenburg Internet Working, Internet Engineering Task Force (IETF), ISSN: 2070-1721, June 2012*

[25] *David Schweikert, "Postgrey-Postfix Greylisting Policy Server, Clients Which Repeatedly Show To Be Able To Pass The Greylist, Are Entered In A 'Clients White list', For Which No Greylisting Is Done Anymore", March-2011*

[26] *E. -S. M. El-Alfi, "Learning Methods for Spam Filtering", International Journal of Computer Research, 2008, vol 16, No. 4.*

[27] *Drake Christine, Oliver Jonathan and Koontz Eugene. "Anatomy of a phishing email". In Proceedings of the First Conference on Email and Anti-Spam, CEAS'2004, 2004.*

[28] *Pantel Patrick and Lin. Dekang ``SpamCop-A Spam Classification & Organization Program", Proceedings of AAAI-10 Workshop on Learning for Text Categorization.*