Towards Eradication of SPAM: A Study on Intelligent Adaptive SPAM Filters

Tarek Hassan
(B.Sc. Computer Science, Egypt)

This thesis is presented for the degree of Master of Computer Science

Murdoch University, Western Australia

2006
I declare that this thesis is my own account of my research and contains as its main content work, which has not previously been submitted for a degree at any tertiary educational institution.

___________________________________________
Tarek Hassan
Abstract

As the massive increase of electronic mail (email) usage continues, SPAM (unsolicited bulk email), has continued to grow because it is a very inexpensive method of advertising. These unwanted emails can cause a serious problem by filling up the email inbox and thereby leaving no space for legitimate emails to pass through. Currently the only defense against SPAM is the use of SPAM filters. A novel SPAM filter GetEmail5 along with the design rationale, is described in this thesis. To test the efficacy of GetEmail5 SPAM filter, an experimental setup was created and a commercial bulk email program was used to send SPAM and non-SPAM emails to test the new SPAM filter.

GetEmail5’s efficiency and ability to detect SPAM was compared against two highly ranked commercial SPAM filters on different sets of emails, these included all SPAM, non-SPAM, and mixed emails, also text and HTML emails.

The results showed the superiority of GetEmail5 compared to the two commercial SPAM filters in detecting SPAM emails and reducing the user’s involvement in categorizing the incoming emails.

This thesis demonstrates the design rationale for GetEmail5 and also its greater effectiveness in comparison with the commercial SPAM filters tested.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>i</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>ii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>v</td>
</tr>
<tr>
<td>PUBLICATIONS</td>
<td>vi</td>
</tr>
<tr>
<td>CHAPTER 1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Definition of SPAM</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Sources of SPAM</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Growth of SPAM in Australia</td>
<td>5</td>
</tr>
<tr>
<td>1.5 Nuisance and costs of SPAM</td>
<td>6</td>
</tr>
<tr>
<td>1.6 Content of SPAM</td>
<td>6</td>
</tr>
<tr>
<td>1.7 Transmission of SPAM emails</td>
<td>7</td>
</tr>
<tr>
<td>1.8 Why is SPAM so common?</td>
<td>9</td>
</tr>
<tr>
<td>1.9 How to stop SPAM</td>
<td>11</td>
</tr>
<tr>
<td>1.10 Amis</td>
<td>12</td>
</tr>
<tr>
<td>1.11 Thesis Overview</td>
<td>12</td>
</tr>
<tr>
<td>CHAPTER 2. LITERATURE REVIEW</td>
<td>14</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>14</td>
</tr>
<tr>
<td>2.2 Dealing with SPAM</td>
<td>15</td>
</tr>
<tr>
<td>2.3 SPAM filters</td>
<td>18</td>
</tr>
<tr>
<td>2.3.1 Black List Filter</td>
<td>20</td>
</tr>
<tr>
<td>2.3.2 White List Filter</td>
<td>21</td>
</tr>
<tr>
<td>2.3.3 Bayesian Filtering (Content Focus)</td>
<td>21</td>
</tr>
<tr>
<td>2.3.4 Fingerprints Filter</td>
<td>22</td>
</tr>
<tr>
<td>2.3.5 Password Filter</td>
<td>23</td>
</tr>
<tr>
<td>2.3.6 Challenge/Response Filter</td>
<td>24</td>
</tr>
<tr>
<td>2.3.7 Community-base Filter</td>
<td>24</td>
</tr>
</tbody>
</table>
2.3.8 Mobile Agent  25
2.3.9 Encryption and Trust  26
2.3.10 Copyright Tokens  26
2.4 Legal Action against SPAM  30
  2.4.1 Complaining to Spammers’ ISPs  31
  2.4.2 The use of an Opt-in Mechanism  31
  2.4.3 SPAM Litigation  32

CHAPTER 3. DESIGN OF GETEMAIL5 SPAM FILTER  36

3.1 Introduction  36
3.2 History of Bayesian Probability  37
3.3 How a Bayesian Filter Works?  39
3.4 Advantages of the Bayesian Filtering  41
3.5 Design of the Proposed Filter  42
  3.5.1 Whitelist Filter  46
  3.5.2 Blacklist Filter  46
  3.5.3 Bayesian Filter Subject to HAM or SPAM  47
3.6 Implementation of the GetEmail5 SPAM Filter  48
3.7 Operation of the GetEmail5 Filter  52

CHAPTER 4. RESEARCH METHODOLOGY  56

4.1 Introduction  56
4.2 Preparing the Filter Requirements  56
  4.2.1 Creation of a new Email Account  56
  4.2.2 Obtaining a Bulk Email Sender  58
  4.2.3 Collecting SPAM Emails (Data set)  58
4.3 Sending SPAM Emails  59
4.4 Receive and Analyse the Emails  63
4.5 Evaluation  63
  4.5.1 First Experiment (Mixed Emails)  67
  4.5.2 Second Experiment (80% SPAM and 20% non-SPAM Emails)  67
CHAPTER 5. RESULTS AND DISCUSSION

5.1 Introduction
5.2 Initial Evaluation of the GetEmail5 SPAM Filter
5.3 First Experiment
5.4 Second Experiment (80% SPAM and 20% non-SPAM Emails)
5.5 Third Experiment (100% Plain Text SPAM Emails)
5.6 Fourth Experiment (100% HTML SPAM Emails)
5.7 Analysis of the five Key Performance Indicators (KPI)
  5.7.1 SPAM Detection
  5.7.2 Filtering Time
  5.7.3 False Positive
  5.7.4 False Negative
  5.7.5 User Involvement
5.8 Limitations of this Research

CHAPTER 6. CONCLUSIONS AND FURTHER WORK

6.1 Conclusions
6.2 Recommendations for Further Work

REFERENCES

APPENDIX GETEMAIL5 DOCUMENTATION
ACKNOWLEDGEMENTS

The work presented in this thesis was carried out under the supervision of Mr. Peter Cole and Associate Professor Lance C.C. Fung, to whom I extend my gratitude for their valuable and friendly guidance, encouragement and helpful suggestions throughout the progress of the research and in the preparation of this thesis.

I would like to extend my special thanks and indebtedness to Kashif Saleem for his help and support.

My gratitude and heartfelt thanks to Dr. Karin Strehlow and Anthony Horton, for their support, friendship, help and encouragement, God Bless.

Finally, my thanks and gratitude go to my wife, Eman and daughter, Amira who have been understanding, patient, supportive and making duaa(supPLICATION) for the success of my study. Thank you also to my parents, sisters and brother for their encouragement and support during my stay in Australia.


CHAPTER 1. INTRODUCTION

1.1 Introduction

Email, or electronic mail, is currently the most common, cheapest and convenient method of daily communication between individuals and organizations. It is gradually replacing the usage of traditional mail delivery services [Inovem, 2003]. Rather than having to wait for days to receive a letter, emails can be sent within seconds across the globe with the single click of a computer button. Emails have made communication very fast and easy. However, this was not always the case.

In 1971, Ray Tomlinson of ARPANET sent the world’s first email. ARPANET was created in 1969 by the United States Defense Advanced Research Project Agency (ARPA) as large wide-area network. The main task of ARPANET was to link universities and research centers together using network technologies [ARPANET, 2006]. Tomlinson sent the first text email message to himself, saying “Testing 1 2 3”, then repeated the test message by sending it to many computers. Following this he informed all ARPANET users about the existence and availability of electronic mail (email) and explained to them the instructions on how to use and address mail to another user using the custom email address format (username@domain), which is the same email address format being used today [Griffiths, 2002]. Since the early days when email was introduced to government agencies, universities, and research institutes, the use of email has increased dramatically particularly by the general public for personal and official use.

The number of the email users worldwide has increased from 505 million in 2000 to 1.2 billion in 2005 [Toffel, 2005]. The rate of increase varies between personal and
business emails. However; the rise in the use of emails has been accompanied by a serious and irritating problem, called SPAM.

1.2 Definition of SPAM

SPAM, often referred to as Unsolicited Commercial Email (UCE) or Unsolicited Bulk Commercial Email (UBCE) is the cheapest and fastest method of advertising for commercial websites [Khong, 2001]. The word “SPAM” originated during a business dinner when a group of businessmen could not have a conversation due to the presence of chorus singers advertising for a new meat product who kept singing the song (SPAM SPAM SPAM). The term SPAM became connected with computers in 1985 when a hacker harassed some employees in a large corporation by repeating SPAM SPAM SPAM on their terminals every few seconds [Falk, 2000].

The majority of SPAM emails have one or more of the following characteristics:

- They advertise goods or services of questionable quality or origin.
- They are sent untargeted and in a random way.
- They promote illegal or offensive content.
- Their purpose is to deceive people.
- They collect personal information without authorization, and sometimes they use a third party email server.
- They do not have an unsubscribe option or any combination of the above [NOIE, 2002].
Chapter 1.

Introduction

Figure 1.1: A sample of email SPAM in an individual’s Inbox

SPAM is a major problem for both email users and internet services providers. SPAM emails frequently fill up a user inbox leaving no more space for legitimate emails (An example of a user’s inbox filled with SPAM is given in Figure 1.1). SPAM is frequently credited with costing organizations substantial amounts of money as employees waste considerable time checking for and deleting SPAM from their computers.

SPAM is also responsible for the abusing bandwidth (i.e. the data rate carried by a network connection), as SPAM constantly increases network traffic, leading to slower Internet access [Speed-Guide, 1998].

These nuisance emails have a number of sources.
Chapter 1.

Introduction

1.3 Sources of SPAM

There are a number of ways spammers (people who send SPAM are commonly referred to as spammers) harvest users email addresses. Listed below are some of the more common means by which spammers can harvest email addresses and some suggested approaches to avoid it.

- **Email addresses on site**

  Spammers use special software to spider websites searching for email addresses (i.e. email addresses like “contact us”). To avoid an email being harvested, instead of using an email link, it is preferably to use forms to contact the webmaster (a person who is maintaining the website).

- **Auto reply messages**

  The use of auto-reply messages (such as “I am on a holiday, I will be back soon”) on email addresses is very attractive to spammers, as one of the spamming techniques is to send emails to random email addresses, and if spammers get any reply message from an email address, they know that the email address exists and is live. They can later use the address to send SPAM messages. It is a good idea not to use auto reply message services.

- **Common email addresses**

  There are some email addresses like “webmaster@...”, “admin@...” or “info@...” etc. Spammers send SPAM emails to those email addresses randomly, to check if these email addresses are alive. It is a good technique to use uncommon email addresses like info_company-name, admin_contact, and
so on. Also, individuals with common names sometimes get targeted (e.g. john smith@...).

- **Subscription**

  Spammers regularly sniff for new email addresses using sophisticated software. When a user subscribes a new website (e.g. newsletters, updates for certain products etc), it is often a good idea to give an email that is not the user primary email address.

- **Guest Book**

  When a user wants put a comment on or leaves a message at a website, it is not recommended to put his/her email address in a guest book, because spammers often check guest books looking for email addresses [Webnet77, 2002].

### 1.4 Growth of SPAM in Australia

According to the Australian Bureau of Statistics (ABS), 26% of all online businesses in Australia reported the use of the Internet for marketing purposes in 2002 [ABS, 2002]. One recent SPAM-monitoring agency estimated SPAM reported for as many as 10 out of every 13 messages [Postini, 2005]. A recent estimate show that SPAM messages cost businesses around $900 AUD per employee per year due to the increase of download times and Internet access [Greenway, 2004]. At the same time as the use of the internet for marketing purposes increased during this period, there was an increase of 300% in SPAM from 2001 to 2002 [Connolly, 2003] [NOIE, 2003]. Globally, the cost of SPAM is much higher.
1.5 Nuisance and costs of SPAM

Based on a European Commission study of unsolicited commercial communications and data protection, SPAM is estimated to cost Internet users approximately $16 AUD billion a year worldwide [NACPEC, 2005]. To illustrate the nuisance of SPAM, the following example estimated that on average, each employee spends around 49 minutes a day dealing with their emails, and 38% of that time (18.6 minutes) is spent checking and deleting SPAM. If a company has 1000 employees and each employee earns around $35 per hour, the approximate total cost will be 310 employee hours each day or 78,740 employee hours each year. This equates to an annual productivity loss of $2,755,900 due to SPAM [ICMI, 2004]. Many online companies such as AOL, Yahoo and AT&T are struggling to deal with billions of SPAM emails each week, because spammers have become more sophisticated in their methods [Didsbury, 2003]. In addition to the impact of SPAM on corporations, SPAM disturbs private internet users by bombarding Internet Service Providers (ISPs) with junk emails. As an example, approximately 2.4 billion SPAM emails were recorded every day by Hotmail/MSN, and 2.2 billion SPAM emails were recorded every day by AOL [SPAM-NEWS, 2004] [Gullible-Info, 2005]. The contents of these SPAM emails are mostly questionable.

1.6 Content of a SPAM

According to America Online (AOL), one of the leading providers of interactive services, internet technologies and e-commerce services, the majority of SPAM emails are advertising for goods or services online. The following subjects comprise the top ten most common SPAM, which is shown in the table below [Emailuniverse, 2004].
Table 1.1: List of the top ten most common SPAM [AOL, 2003]

<table>
<thead>
<tr>
<th>Number</th>
<th>SPAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Viagra online (also: xanax, valium, xenical, phentermine, soma, celebrex, valtrex, zyban, fioricet, adipex, etc.)</td>
</tr>
<tr>
<td>2.</td>
<td>Online pharmacy (also: &quot;online prescriptions&quot;; &quot;meds online&quot;)</td>
</tr>
<tr>
<td>3.</td>
<td>Get out of debt (also: &quot;special offer&quot;)</td>
</tr>
<tr>
<td>4.</td>
<td>Get bigger (also: &quot;satisfy your partner&quot;; &quot;improve your sex life&quot;)</td>
</tr>
<tr>
<td>5.</td>
<td>Online degree (also: &quot;online diploma&quot;)</td>
</tr>
<tr>
<td>6.</td>
<td>Lowest mortgage rates (also: &quot;lower your mortgage rates&quot;; &quot;refinance&quot;; &quot;refi&quot;)</td>
</tr>
<tr>
<td>7.</td>
<td>Lowest insurance rates (also: &quot;lower your insurance now&quot;)</td>
</tr>
<tr>
<td>8.</td>
<td>Work from home (also: &quot;be your own boss&quot;)</td>
</tr>
<tr>
<td>9.</td>
<td>Hot XXX action (also: &quot;teens&quot;; &quot;porn&quot;)</td>
</tr>
<tr>
<td>10.</td>
<td>As seen on Oprah.</td>
</tr>
</tbody>
</table>

The above table demonstrates that often spammers are attempting to attract customers with goods or services that they would normally not avail themselves of due to either embarrassment or “personal” interest, or the normal unavailability of the goods or services.

1.7 Transmission of SPAM emails

Email users are thus recipients of SPAM emails which in many instances will have no relevance to them, leading many to wonder how they managed to end up on the spammer’s mailing list.

Transmitters of SPAM emails acquire email addresses from a number of sources; including mailing lists, SPAM software capable of harvesting email addresses from the Internet, web pages and hacking into sites. Each of these will be discussed below.
• **Mailing lists**

A mailing list is used by a group of people who have common interests and conduct discussions through email messages on certain topics. When a message is sent to a mailing list, each subscriber in the list will receive a copy [Young, 2002]. Spammers join mailing lists to collect active email addresses of subscribers to the mailing list.

• **SPAM software**

Some spammers use sophisticated software to scan websites searching for email addresses. Any email addresses that can be found on a website, whether a user or a webmaster, may be located and added to spammers email lists [Comcast, 2002]. These types of software can be easily accessed on the web, where a number of websites sell software specifically designed to collect email addresses, validate those addresses, and then send bulk emails to the addresses collected. One example of such a website selling software which may be useful for legitimate commercial bulk email and also could be used by spammers is [http://www.marketing-2000.net/pm.htm](http://www.marketing-2000.net/pm.htm) [IMT, 2004].

• **Web pages**

Many websites ask for some information from their visitors, in the form of registrations forms, newsgroups and guest books etc. Some organizations sell their customer databases to other marketing companies. Often these lists have been used later for the delivery of SPAM [Junkbusters, 2005].

• **By hacking into sites.**

Some sites that provide free email addresses have been hacked into in order to get the list of email addresses [Raz, 2004]. For example, the AOL reported that
“more than 500 so-called screen names of its customers had been hacked. Those records typically contain information such as a customer's name, address and the credit card number used to open the account” [CNN, 2000].

1.8 Why is SPAM so common?

As mentioned previously, the transmission of SPAM emails is the most cost effective form of direct marketing [Leung, 2003]. SPAM can reach a huge number of Internet users with minimal internet connection cost. It is considerably less expensive than traditional mail, because there are no printing or delivery costs [Vnunet, 2004]. The cost of a newspaper advertisement, for example, can range from AUD $24 to $25,000 (full-page), while sending a catalog to 100,000 people can cost from AUD $50,000 to $150,000 depending on the size, quality of printing and type of postage of the catalog. In 1998 by comparison, an Internet connection over a 28 kbps dial-up modem could send more than 100 email messages a minute, which equals to 864,000 email messages a day, or 26 million email messages a month. If unlimited dial-up Internet connection is used, the cost may be as low as AUD $20 a month, and AUD $15 for a telephone line. A spammer can therefore send approximately 10,000 email messages for one cent. Even if the price of buying a new computer (approximately $1,000) is added, SPAM is still the most cost effective alternative to printed media [Schwartz and Garfinkel, 1998]. Today’s faster broadband Internet connections at much reduced prices have increased the capacity of spammers while reducing their cost per email. Sending SPAM is very inexpensive for the sender (spammers send millions of SPAM emails by a click in seconds). However, it can be very expensive to the recipient, because of the recipient’s time he or she spends in dealing with SPAM emails [Geller,
2003]. For example, Table 1.2 below shows an estimate cost of SPAM for both sender (spammer) and recipient based on the hourly rate of $12 for an employee.

<table>
<thead>
<tr>
<th>From</th>
<th>Cost to sender</th>
<th>Cost to recipient</th>
<th>% of cost borne by sender (Spammer)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spammer</td>
<td>$0.00001</td>
<td>$0.10</td>
<td>0.01%</td>
</tr>
</tbody>
</table>

Consequently consumers are taking action to protect themselves from SPAM. Table 1.3 illustrates the results of a survey by Fallows in 2003 investigating consumer sentiments towards SPAM (The data were collected from two sources. The first source by a telephone survey of 2,200 adults, including 1,380 Internet users, the second was a compilation of more than 4,000 first-person narratives about SPAM)

<table>
<thead>
<tr>
<th>% Email Users</th>
<th>Sentiments</th>
</tr>
</thead>
<tbody>
<tr>
<td>59%</td>
<td>SPAM is “annoying, but not a big problem”.</td>
</tr>
<tr>
<td>27%</td>
<td>SPAM is a “big problem”.</td>
</tr>
<tr>
<td>14%</td>
<td>It is no problem at all.</td>
</tr>
</tbody>
</table>

The same survey revealed some actions taken by end users to prevent SPAM (Table 1.4).

<table>
<thead>
<tr>
<th>% Email Users</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>73%</td>
<td>Avoid giving out their email addresses</td>
</tr>
<tr>
<td>69%</td>
<td>Avoid posting their email addresses on the web</td>
</tr>
<tr>
<td>62%</td>
<td>Say their employers use filters to block SPAM from their work email accounts; half of them get no SPAM at all in those accounts.</td>
</tr>
<tr>
<td>37%</td>
<td>Who have a personal email account apply their own filters to their email system; 21% of those with filters say less than one-tenth of the email they receive is SPAM.</td>
</tr>
<tr>
<td>86%</td>
<td>Report that usually they “immediately click to delete” their incoming SPAM</td>
</tr>
</tbody>
</table>
1.9 How to stop SPAM

There are different ways to stop (or at least reduce) the SPAM, like avoid posting the email address on a forum or chat rooms, use more than email address if the user wants to subscribe to online services, avoid using auto reply message (for example “out of the office” “I will be back after holiday”), do not send any confirmation when you receive a new email (some times when a user receives an email the sender asks for a confirmation that the messages has been received), and the use of SPAM filters.

SPAM filter is a program that is used to detect unsolicited email and prevent those messages from getting to a user's inbox. SPAM filter looks for certain criteria on which it based on the users configurations and needs. All spam filters available today act in one of a few techniques, or a combination of these techniques. There are many types of SPAM filters which will be discussed in details in chapter 2. This research is an attempt to design an intelligent and efficient SPAM filter.

1.10 Aims:

Clearly SPAM represents a nuisance to the community by affecting (destroying) legitimate online marketing, which leads to a big loss of the business opportunities to use such an inexpensive legitimate marketing tool for promoting their goods or services. It is very hard for the user to distinguish between an illegitimate marketing email (SPAM) and a legitimate marketing email, because both are very close in appearance to each other.

As a result the aims of this research were to:

- evaluate different methods of detecting SPAM; designing SPAM detection methods is not easy, because spammers constantly find ways around the
SPAM filter. There are some effective methods to deal with SPAM emails, some of these methods are explained in detail (see chapter 2).

- design a SPAM filter,
- test and compare the proposed SPAM filter against commercially available SPAM filters, and to
- create a more efficient SPAM filter than those currently available.

1.11 Thesis Overview

This thesis comprises six chapters.

Chapter 2 presents the literature review and examines the different methods that have been used to reduce SPAM by placing emphasis on existing SPAM filters. Some of the legal actions taken to reduce and eliminate SPAM emails are also discussed.

Chapter 3 focuses on the Bayesian method, which is the basis for the SPAM filter developed in this research. The advantages of Bayesian filtering will be outlined in this chapter. Furthermore, the different aspects of the design, mechanism and operation of a proposed SPAM filter, are illustrated.

Chapter 4 outlines the various requirements needed to make the new SPAM filter fully operational. The various methods used to test and compare the designed new SPAM filter with two other commercial filters are described.

Chapter 5 reports the results of tests performed to evaluate the effectiveness of the proposed SPAM filter in detecting SPAM emails. This chapter also presents the results and a discussion of the comparison of the new SPAM filter with two commercial SPAM filters in detecting SPAM, non-SPAM, and mixed emails. In addition, an analysis of 100% plain text and 100% HTML SPAM emails sent to the three SPAM filters is reported.

Chapter 6 includes conclusions to the research and suggestions for further work.
CHAPTER 2. LITERATURE REVIEW

2.1 Introduction

The Internet has exposed new ways of communications. Many users can now easily and quickly send an email with no charge to a friend thousands of kilometers away. This method of communication has also opened the door for bulk emailing, which can reach thousands users within seconds, with an aim of promoting certain types of goods and services [Kagstrom, 2005].

The simplest definition of SPAM is that it is any received email message that is unwelcome by the recipient. SPAM has been identified as the most widespread problem facing email users. The majority of SPAM is sent in order to achieve a profit, through the sale of goods or services. The major problem with SPAM is that it is the receiver that is paying for the SPAM in terms of spending their time to check and clean their inboxes.

As the amount of SPAM on the internet is progressively increasing, the need to eliminate SPAM is eminent. Today there are several approaches to deal with SPAM. Many nations across the world have initiated anti-SPAM legislation to deter potential spammers and punish those proven guilty. Several countries and organizations are already acting on a local and international level, implementing anti-SPAM initiatives at the technical or regulatory level [ITU, 2006].
In 2003, the Australian parliament reported that legislation on its own will not stop the spread of SPAM, while technical solution can stop some SPAM. Then the end result will be combining the technical solution with SPAM legislation together as an attempt to eliminate (as much as possible) the spreading of SPAM [APEC, 2004].

Internationally on 27 April 2005, twelve Asia-Pacific communications and Internet agencies joined the Australian Communications Authority (ACA) and the Korean Information Security Agency (KISA) in signing the Seoul-Melbourne Anti-SPAM Agreement [ACMA, 2005a]. This multiparty memorandum of understanding (MoU) on assistance in dealing with SPAM is focused on sharing knowledge, information and intelligence about known sources of SPAM, network vulnerabilities, methods of spreading SPAM, and technical, education and approach solutions to SPAM [ACMA, 2005a].

Some other notable examples of anti-SPAM legislation are described later in this chapter.

### 2.2 Dealing with SPAM

Currently, there are a number of ways to reduce the amount of SPAM. These include:

- avoid sharing email addresses,
- the use of caution when opening a new (strange) email,
- avoid unsubscribing, and
- the use of SPAM filters.
Avoid sharing email addresses

This method is most commonly used by individuals. They avoid sharing their email addresses particularly in places where it will be available online, like chat rooms. It is a good idea to create another email address (e.g. a Hotmail or Yahoo account) and use that address to subscribe or buy products online, thereby reserving the normal email address (e.g. university, work or ISP email account) for official or work related use only [Berman, 2005].

The use of caution when opening a new (strange) email

Many emails are written in HTML format. HTML stands for Hyper Text Markup Language, which is a method that describes how a collection of text and images will be displayed to the user on a web page. HTML was developed along the same lines as newspaper editing software [Marshall, 1998]. When HTML emails are opened by a recipient, spammers can get the recipient’s email address, by asking for a confirmation that the email has been received successfully [Berman, 2003]. Some spammers put a request in their emails asking for a receipt from the recipient. If the email program (Eudora-Outlook…etc) or mail server is set to automatically confirm these receipt requests, this will confirm that the email address is valid, therefore resulting in more SPAM [SpamHelp, 2006].

Table (2.1) illustrates some common titles of SPAM email messages.
Table 2.1: List of the common titles of SPAM email messages [Zedtoo, 2005].

<table>
<thead>
<tr>
<th>Number</th>
<th>Title Descriptions (SPAM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>A title containing “free pix”, “passwords”, or money-making opportunities.</td>
</tr>
<tr>
<td>2.</td>
<td>Numerous non-alphabetic characters (e.g. ***** , !!!, #### etc.), particularly at the start of the title.</td>
</tr>
<tr>
<td>3.</td>
<td>A TITLE IN CAPITALS.</td>
</tr>
<tr>
<td>4.</td>
<td>A title mentioning a filename ending in &quot;.html&quot; or &quot;.htm&quot;.</td>
</tr>
<tr>
<td>5.</td>
<td>A title containing a web site address.</td>
</tr>
<tr>
<td>6.</td>
<td>A title ending with a multi-digit number (e.g. &quot;Please help 13874&quot;).</td>
</tr>
<tr>
<td>7.</td>
<td>A title in an unexpected language (e.g. German)</td>
</tr>
<tr>
<td>8.</td>
<td>A title containing a stream of obscenities, (this either may be SPAM or worth avoiding for more obvious reasons).</td>
</tr>
<tr>
<td>9.</td>
<td>An author field consisting of a stream of random characters, (such as &quot;jsg;rhb&quot; or &quot;dkhvdjblkghsx&quot;).</td>
</tr>
<tr>
<td>10.</td>
<td>An author's field containing random numbers in an email address (e.g. &quot;<a href="mailto:ma3ry@b2ren7da4.com">ma3ry@b2ren7da4.com</a>&quot;).</td>
</tr>
<tr>
<td>11.</td>
<td>The author's name is &quot;Webmaster&quot; or when it is an invitation to a web site.</td>
</tr>
<tr>
<td>12.</td>
<td>A female author name AND a title that does not appear to be &quot;on-topic&quot;.</td>
</tr>
</tbody>
</table>

Avoid unsubscribing

One technique employed by spammers to determine whether an email addresses is active or not, is to send an email for a product/commercial to which the recipient has not subscribed [Zedtoo, 2005]. The same email contains an unsubscribe option. However, as soon as the recipient tries to unsubscribe he/she is confirming the active state of the email address. It is therefore recommended that to ignore completely the unsubscribe option.
The use of SPAM Filters

By using SPAM filters Internet users give themselves some protection from receiving SPAM [Zedtoo, 2005].

Some of the popular SPAM filters are presented below.

2.3 SPAM Filters

The dramatic increase of the SPAM in the past two years has created a real interest among researchers to investigate methods to combat SPAM. Researchers are presently working in the implementation of new filters that prevent SPAM from reaching their destination either by blocking it at the server level (i.e. organization with its own email server, for example Murdoch University) or the client level (such as a user at home) [Pelletier et al., 2004].

In January 2003 and 2004, a conference on SPAM took place at MIT in Cambridge, Massachusetts USA, and the Coalition against Unsolicited Commercial Email (CAUCE) was established [CAUCE, 2003]. While CAUCE is trying to introduce legislation that would make SPAM illegal on the international level, other research groups are trying to fight SPAM by creating SPAM filters [Pelletier et al., 2004].

A SPAM filter has a set of instructions to block emails based on the nature or the content of the email. An email filter acts like any other filter, in this case, only letting some messages through and not others depending on the filtering parameters used. As in other filters, the pore size of the filter will determine what can get through (e.g. the smaller holes the strainer has, the more pure the result of the filtering will be).
The same principle applies here. However, in this case, the filter is a set of instructions, and the pore size (strainer’s hole size) of the filter is determined by the instructions that the filter uses to decide which email to let through to the recipient.

An email filter for example, might include the following instructions:

1. If the email is from dad@home.com, save it in the family folder.
2. If the email is from newsletter@strange.com, delete it.
3. Otherwise, leave the email in the mailbox [Allman, 2003a].

The flexibility of a SPAM filter depends on the filtering software. SPAM filter analysis may be directed at the following:

*Header analysis*, here the SPAM filter will check the header of the incoming email, if the header is defined as a SPAM (for example: free gifts for you, you are the winner…etc) the SPAM filter will prevent the message from passing through to the recipient.

*Address lists analysis*, if the incoming email is from any unknown sender, the SPAM filter will check the email address of the sender against the address list that the recipient allows receiving messages from, then the message will be blocked or passed through to the user.

*Keyword lists analysis*, the SPAM filter will check the contents of the incoming message if it has any words that could be suspicious (like: Viagra, Sex…etc), then the message will be blocked [Allman, 2003b].

SPAM filters prevent SPAM from being transmitted to the recipient. The challenge is to design a filter that allows the desired email mail to pass through, while at the same
time blocking the SPAM email. Obviously, errors may occur. The filter may identify a legitimate email as a SPAM, and block it (false positive), or it may identify a SPAM email as a legitimate email, and allows it to pass through (false negative). Filters can be applied at the client level or the server level. However, dealing with SPAM at the server level is a more effective choice because it detects and stops SPAM sources quickly without delaying email delivery, but it is quite tricky. For example, in a company with 200 employees, and different departments, each department will receive email messages that could be of no use and considered to be SPAM to the other departments. Some common SPAM filters are outlined below.

2.3.1 Black List Filter

A blacklist SPAM filter operates by creating a list of common words or phrases found in the header of the email message and domain name (the main part of the address of a web site, for example murdoch.edu.au - microsoft.com etc.), which can be used to decide if an email should be prevented from passing through the SPAM filter [Turner, 2004]. However, a number of problems may occur if black list filter is the only filter used. For example if a word “result”, is blacklisted, and a user receives an email with a header (your exam “result”), and receives another email with a header (use our product for a quick “result”). Both emails will be blocked by the filter [Graham, 2002]. Spammers may try to circumvent this type of filter by either changing the contents of the email message or by using random character string for each message (e.g. Vi@gra instead of Viagra) [Moore et al., 2003]. Another significant disadvantage of blacklists is maintenance. The Internet is
unbounded and thousands of new sites are added every day and spammers continually change their identities. The problem with blacklists is the growing blacklist itself. The bigger it gets, the longer the processes required to physically check the blacklist and block a SPAM email [Sartain, 2005].

2.3.2 White List Filter

A whitelist SPAM filter is the opposite of the blacklist, and it assumes that all emails are SPAM unless they can pass through the filter. A whitelist may contain a list of email addresses that the user created to receive messages only from trusted sources. Alternatively it could be a list of domains which must be defined as legitimate before the message passes through the filter to the recipient [ExpressionEngine, 2005]. The problem with this type of filter may arise, for example, if a person wants to send an email to another person protected by this filter. The sender will have to go through the confirmation process before the message can pass through the filter. This confirmation process may cause unnecessary irritation to some users; moreover, it may block legitimate emails from new sources [Allman, 2003a].

2.3.3 Bayesian Filtering (Content Focus)

Bayesian filtering is an extension of the text classification technology. This filter is a computer program used to recognize the words in a document, and can be implemented in a SPAM filter to search the textual content of an email. Bayesian filtering method uses text categorization algorithms to determine the probability that a certain email is SPAM [Didsbury, 2003]. The algorithms are
capable of categorizing the occurrence of certain words or phrases in terms of how and where they appear in the email message, but not by their existence alone. The challenge with content filtering is that SPAM emails often contain simply image links (e.g. photographs), which download image-based content to the receiver [Androutsopoulos et al., 2000a]. Bayesian SPAM filters are capable of analysing text, but are not capable of analysing images. To carry out the analysis of images requires pattern matching techniques which is another area of research in itself. This analysis is beyond the scope of this study.

Although the Bayesian filter is quite effective, it needs to be updated regularly. The reason for this is that it divides the incoming email messages into two classes, legitimate or illegitimate. Following this, each email is split into tokens (words, html codes, etc.) so their occurrence in the body of the messages can be computed. Based on this occurrence and using a specific mathematical formula, the probability that an email is SPAM or not can be calculated [Pelletier et al., 2004].

2.3.4 Fingerprints Filter

Fingerprinting is a filtering technique that recognizes a SPAM email and assigns a distinct identifier (fingerprint) to that particular email [Didsbury, 2003]. The system then constructs a database containing all of the unique identifiers (fingerprints) and compares them with each incoming email. All matching emails are blocked by the filter. The disadvantage of this technique is that it is only effective with identifying repeated emails (i.e. after the first
one has been fingerprinted). Consequently, the system will get infected by new SPAM all the times. Another disadvantage is the speed at which the fingerprint information is obtained and distributed through the system (i.e. the amount of time it will take to update the filter about a particular email which has been identified as a SPAM plus the amount of time required to update all of the software clients). Obviously, for this filter to be effective, the amount of time to identify a SPAM email and update the SPAM database has to be short [Didsbury, 2003].

2.3.5 Password Filter

Password SPAM filters will only allow emails containing passwords to pass through the filter [Cotse, 2004]. The password may be included in the email address, the subject line, or some other parts of the email. If the password is not included, the email is simply rejected. A password filter is an effective method for blocking SPAM, but it can also block desirable emails by requiring a password on every new message from a new sender. As with the white list method, the major drawback of this filter is that it is difficult for the new users to initiate a conversation with someone whose email inbox is protected with a password, because the email will be rejected. Furthermore it is difficult to ask every new sender for a password to let his/her email pass through [Wheeler, 2003].
2.3.6 Challenge/Response Filter

Challenge/Response filters send an automated message that asks the sender to provide return confirmation of their email addresses. The aim of this is to let the system verify that the sender is an individual, not a machine generating SPAM. *This is often referred to as a "Turing Test", named after a test devised by British mathematician Alan Turing to determine if machines could think* [WhichSpamFilter, 2005].

The problem with this filter is that it may block a legitimate email. It may block many requested emails like newsletters and updates about certain products if a company is not prepared to respond manually to verification challenge/response SPAM filter. Another problem may be the nuisance factor to the new legitimate senders by requesting a return confirmation of their email addresses [Salmi, 2005].

2.3.7 Community-base Filter

Community-base filters work on the principal of "shared knowledge" of SPAM, with the filters communicating with a central server [Miti, 2003]. When a user receives a SPAM email he/she will mark it as SPAM, and so on for every user connected to the network. When the group of users (majority) decided that the message is SPAM, it will be blocked from the user's inboxes in the future [Schwartz, 2003]. This technique has several limitations:

- Before enough users decide that the message is SPAM, the other users continue to receive that SPAM email.
• Potential conflict may occur between users, as not all will agree on what is considered to be SPAM. Therefore, legitimate emails may be blocked (false positives).

• The newest SPAM emails may not be detected, given that once a new SPAM message arrives, the filter will wait until a user defines this message as a SPAM. Until this happens the system can allow too many SPAM emails to pass through [WhichSpamFilter, 2005].

2.3.8 Mobile Agent

Mobile agent is a structure of computer software and data which is able to migrate (move) from one computer to another separately and continues its execution on the destination computer to perform a number of tasks or functions. A mobile agent is designed to perform email filtering on the mail server, and identifies the SPAM emails before passing the mail to the user [Reilly, 2005]. There are some limitations with using mobile agent for security issues, because the protection of the mobile agents and the host cannot always be assured. [Li and Wang, 2002]. Also, the filtering algorithms and data list require a huge amount of code in order to be efficient. This will increase the size of the code for the agent, which will consume large amounts of bandwidth during transmission, which will lead to slow down the filtering process. In addition if enough users use such mobile agents on the server at the same time it will overload the mail server [Hoffman, 1999].
2.3.9 Encryption and Trust

Encryption and trust technique is based on a trust agreement between groups of people/companies who communicate with each other by emails [Infineon, 2005]. Under this agreement a sender cannot send an email unless the recipient gives permission to do so. Each group has a unique digital signature, created using different encryption techniques. The encrypted digital signature is very hard to fake. The digital signature is used to sign and encrypt every message that is sent out. This is to ensure that the message cannot be changed during transmission and only the intended recipient can read it. With this type of security, SPAM emails can be easily identified [Thibodeau, 2002].

The problem with this technique is that the trust agreement may not be possible to implement in a small group of users, because of the cost and time to set up the agreements. Although this method is more suitable for large groups of users, there still remain problems with the cost of implementation. Furthermore, the encryption technique is potentially too complicated for the users [Yang et al., 1997].

2.3.10 Copyright Tokens

Copyright tokens technique uses a trademark and copyright law to prevent SPAM. It uses the copyright tokens, in other words “a stamp”, as an anti-SPAM tool [O'Brien, 2002].

In the early days, this technique used some types of poem (e.g. “Haiku”, which is a type of Japanese poem, which has been devised and copyrighted by Habeas), which contained a trademark registered to the company (Habeas)
The poem is attached to the email message and serves as a form of identification that the email is not SPAM. The company has developed an agreement where individuals and ISPs (Internet Service Providers) can attach this poem to their email messages for free [habeas, 2005]. One of the limitations of this technique is that would-be spammers can attach the stamp or the token (Haiku) to the messages they send, and the receiver cannot distinguish between legitimate emails or SPAM emails. Users may also become dependent on the company and its rules which mean that the user cannot send any email unless he agreed with the company’s demands (e.g. could be paying fees for sending emails). This scenario may discourage people from using this technology. Finally, this technique may be not effective in certain countries, particularly those which do not have copyright and trademark laws, or fail to enforce them.

Table (2.2) illustrates the different methods of SPAM filters, the advantages and the disadvantages of each method.
Table 2.2: Different types of SPAM filtering methods

<table>
<thead>
<tr>
<th>Types of SPAM filters</th>
<th>Method</th>
<th>Advantages (Pros)</th>
<th>Disadvantages (Cons)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Blacklist</strong></td>
<td>Blocks mail from banned senders.</td>
<td>Blocks known SPAM messages.</td>
<td>Does not block new SPAM.</td>
</tr>
<tr>
<td><strong>Whitelist</strong></td>
<td>Allows mail only from approved senders.</td>
<td>Blocks mail from unknown senders.</td>
<td>Blocks new legitimate mail.</td>
</tr>
<tr>
<td><strong>Bayesian</strong></td>
<td>Text recognition technology.</td>
<td>• Calculate the probability of the message (SPAM or not).</td>
<td>Does not deal with HTML or image mails.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Self learning technique.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Language free.</td>
<td></td>
</tr>
<tr>
<td><strong>Fingerprints</strong></td>
<td>Assign a fingerprint for SPAM message.</td>
<td>Construct database for SPAM mails, and prevent them from passing through.</td>
<td>Only effective with identifying repeating emails (after the first one has been fingerprinted).</td>
</tr>
<tr>
<td><strong>Password</strong></td>
<td>Passwords are required to be in the email to pass through the filter.</td>
<td>Allows only the emails that have password to pass through.</td>
<td>Blocks new legitimate emails that does not have password yet.</td>
</tr>
<tr>
<td><strong>Challenge/Response</strong></td>
<td>Blocks unapproved mail until response arrives.</td>
<td>Allows only legitimate senders to pass through after their response.</td>
<td>• Blocks new legitimate mails.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Annoy legitimate senders by asking for a response with each message.</td>
</tr>
<tr>
<td><strong>Community-base</strong></td>
<td>Blocks mail based on community agreement.</td>
<td>Block a SPAM that a group decides to block.</td>
<td>Does not block a new SPAM.</td>
</tr>
<tr>
<td><strong>Mobile Agent</strong></td>
<td>Works on remote system to perform filtering on the mail server.</td>
<td>Filters the mail before passing it to the user.</td>
<td>The filtering algorithms and data list require a huge amount of code in order to be efficient, which will consume plenty bandwidth during transmission, and overload the mail server.</td>
</tr>
<tr>
<td><strong>Encryption and Trust</strong></td>
<td>Send mail with digital signature.</td>
<td>Digital signature is very hard to fake, and also used to sign and encrypt every message that is sent out.</td>
<td>The encryption technique is too complicated for the users.</td>
</tr>
<tr>
<td><strong>Copyright Tokens</strong></td>
<td>Uses the copyright tokens as an anti-SPAM tool.</td>
<td>Emails cannot be received without their own tokens.</td>
<td>Spammers can attach the same tokens to the messages they send.</td>
</tr>
</tbody>
</table>

The different types of SPAM filters described can be classified according to their mechanism as follows:
• **Blacklist mechanism**

The blacklist SPAM filtering mechanism blocks all of the incoming emails which are defined in the black list, and allows any other emails to pass through.

*Table 2.3: SPAM filter methods that use Blacklist mechanism*

<table>
<thead>
<tr>
<th>Types of SPAM filters</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blacklist</td>
<td>Block known SPAM</td>
</tr>
<tr>
<td>Fingerprints</td>
<td></td>
</tr>
<tr>
<td>Community-base</td>
<td></td>
</tr>
</tbody>
</table>

The blacklist, fingerprints, and community based SPAM filters use the blacklist mechanism. It is obvious that the same results will be obtained (block known SPAM), and the easiest method to implement with the lowest user intervention is the blacklist method compared to the other two methods which shown in table (2.3) above, which has been chosen in the design of the approached SPAM filter GetEmail5.

• **Whitelist mechanism**

The whitelist SPAM filtering mechanism allows all the incoming emails which are defined in the white list, and blocks any other emails from passing through.

*Table 2.4: SPAM filter methods that use Whitelist mechanism*

<table>
<thead>
<tr>
<th>Types of SPAM filters</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whitelist</td>
<td>Block unknown Emails</td>
</tr>
<tr>
<td>Password</td>
<td></td>
</tr>
<tr>
<td>Encryption and Trust</td>
<td></td>
</tr>
<tr>
<td>Copyright Tokens</td>
<td></td>
</tr>
</tbody>
</table>
The whitelist, password, encryption and trust, and copyright tokens use whitelist mechanism it is obvious that the same results will be obtained (block emails from unknown sender “block unknown emails”), and the easiest method to implement with the lowest user intervention is the whitelist method compared to the other three methods as shown on table (2.4), which has been chosen in the design of the approached SPAM filter GetEmail5.

- **Text Recognition mechanism**

  Text recognition mechanism is a unique mechanism used to differentiate between the SPAM and non SPAM emails by calculating the probabilities of the incoming emails. The more training the filter receives the more accurate results, and less user intervention will be required. Because of these features the Bayesian filter has been chosen to be incorporated into the design of the SPAM filter GetEmail5.

### 2.4 Legal Action against SPAM

SPAM has become a widespread problem because it is financially profitable to spammers. The huge number of Internet users around the world, the growing amount of daily email traffic, makes any technical solution particularly complex. Using SPAM filters is a widely used practice, but there is a continual effort by spammers to bypass filters. Legislation is an important tool in the anti-SPAM combat. The goals for legislation should be to identify and prosecute spammers, and support ISPs and user action against SPAM [ITU, 2004].
Some countries have laws that outlaw SPAM and allow the recipient to sue for damages. In Australia, for example, a man was convicted and sentenced two years in prison after charges of sending false investment information to discussion boards and bulk "SPAM" emails in October 2000 was proved [InternetNews, 2000].

### 2.4.1 Complaining to Spammers' ISPs

Complaints can be an effective tool against SPAM, but most spammers falsify the headers of their emails to hide their source. They will, for example, use fake email addresses with fake domains (e.g. spammer@unknown.com), making it nearly impossible to determine their origins [Mulligan, 2005]. Moreover, spammers expect to be stopped by the recipients. After receiving SPAM messages, recipients may stop SPAM by using SPAM filters or by complaining to the ISP. As a consequence spammers usually have a number of new accounts lined up to use instead of the blocked ones [Graham, 2003].

### 2.4.2 The use of an Opt-in mechanism

"Opt-in" is a term that refers to marketing emails that has been asked to be received by the individuals [Jennings, 2005]. Opt-in emails are only sent to users who particularly registered with commercial websites to receive these promotional emails (e.g. registered users with a website that sells books, music CDs etc.). User can "Opt-in" to receive notification emails when any new favorite products have been released in the market. If any user receives these kinds of emails without his/her request, then the user has the right to report the message and take legal action [Pcwebopedia, 2005].
2.4.3 SPAM Litigation

As SPAM is a worldwide problem, there is a need to join efforts from the governments and industry to stop the growth of SPAM [Jones, 2003]. Establishing a domestic policy for dealing with SPAM is therefore only going to solve the problem within Australia. Internationally co-operative procedures are required to be more effective, especially with the SPAM that comes from outside Australia. The Australian Federal Government has taken steps to fight the growth of SPAM by introducing the SPAM Bill 2003 into Parliament. The proposed method is intended to send a powerful message to spammers that spamming is not going to be tolerated in Australia [Parlinfoweb, 2003].

The bill contains several mechanisms to reduce SPAM. It has set some rules about sending commercial electronic messages. Commercial electronic messages must contain correct sender details, and must contain unsubscribe facility. Furthermore there are some rules about address-harvesting software and harvested-address lists. Address-harvesting software and harvested-address lists must not be supplied, acquired, and used.

The bill establishes a multi-level civil authorised organization. In the first instance individuals can be liable for fines up to $44,000; organizations can be fined up to $220,000. These fines can rise to $220,000 and $1.1 million respectively for repeat offenders [Parlinfoweb, 2003]. One example of this is ACMA (Australian Communications and Media Authority).

ACMA has been targeting companies that send unwanted commercial messages (SPAM). ACMA is responsible for enforcing the SPAM Act 2003
[DCITA, 2005], which outlaws the sending of unsolicited commercial electronic messages with an Australian link. Recently, ACMA has:

- “issued Global Racing Group Pty Ltd, based in Queensland, with infringement notices for penalties of $11,000 for sending unsolicited commercial SMS”.

- “fined Australian SMS Pty Ltd $2,200 for breaching the SPAM Act” [ACMA, 2005c].

- “issued infringement notices to three companies and individuals for sending unsolicited commercial email or SMS messages:
  - infringement notices for $660.00 were issued to a Queensland individual (a sole proprietor) for sending unsolicited commercial email and for use of harvested-address lists;
  - infringement notices for $660.00 were issued to a South Australian individual for sending unsolicited commercial email and for use of harvested-address lists; and
  - infringement notices for $6,600.00 were issued to caresales.com.au Ltd, a company headquartered in Victoria, for sending unsolicited commercial SMS messages” [ACMA, 2005b].

- “issued formal warnings to three companies for sending unsolicited commercial emails;

- received enforceable undertakings from two persons to cease sending unsolicited commercial emails;
• reported two live 'phishing' frauds to police at the Australian High Tech Crime Centre” [ACMA, 2005b]; and

• “executed five search warrants on premises and interviewed four other business proprietors for possible breaches of the Spam Act 2003” [ACMA, 2005b].

The National Office for the Information Economy (NOIE) has made some recommendations to reduce SPAM:

1. No commercial emails are to be sent without prior permission of the recipient unless there is an opt-in agreement.

2. All commercial emails must have full details of the sender.

3. The internet Industry Association (IIA), the Australian Information Industries Association (AIIA), and their members should require ISPs to make obtainable to clients alternatives from SPAM filtering tools at reasonable price, and assess and publicise SPAM filtering options and products.

4. The Internet industry should extend and use a list of known Spammers so that the ISPs can make better informed decisions about dealing with customers who have a record of Spamming [NOIE, 2002; NOIE, 2003].
5. NOIE should continue to monitor and publish reports on SPAM volumes and characteristics, which it will help in developing SPAM counter-measure [NOIE, 2002; NOIE, 2003].
CHAPTER 3. DESIGN OF GETEMAIL5 SPAM FILTER

3.1 Introduction

From the previous chapter it is clear that the most effective way in dealing with SPAM is by using SPAM filters [Schwartz, 2003]. Table 3.1 illustrates the comparison between different techniques of SPAM filters, and the Bayesian method was chosen as the most efficient solution to detect SPAM emails. The history, mechanism and advantages of Bayesian method will be discussed in this chapter. Moreover, the design of the proposed SPAM filter (GetEmail5) will be described in detail. In addition, the operation of the SPAM filter GetEmail5 will be illustrated.

Table 3.1: Comparison between different methods of SPAM filters

<table>
<thead>
<tr>
<th>SPAM filter Methods</th>
<th>Block known SPAM</th>
<th>Block unknown emails</th>
<th>Self Learning</th>
<th>Easy to Use</th>
<th>Low user intervention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blacklist</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Whitelist</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Bayesian</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Fingerprints</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Password</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Challenge/Response</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Community-base</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mobile Agent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encryption and Trust</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Copyright Tokens</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The expectations of designing and implementing the approached SPAM filter GetEMail5 will be based on:

- The accuracy of detecting SPAM emails: differentiate between SPAM emails (by using blacklist method), and legitimate emails (by using whitelist method)
- Less user intervention: by choosing a self learning method (Bayesian method)
- Ease of use (no complicated processes required, or too many menus to choose from just run GetEmail5 SPAM filter).

The blacklist and the whitelist mechanisms are simple and clear. The blacklist blocks the incoming emails which have been defined in the blacklist. The whitelist allows the incoming emails which were defined on the whitelist. The next section will illustrate the mechanism of the Bayesian methods.

### 3.2 History of Bayesian Probability

The word “Probability” is derived from the Latin “probare” to prove, or to test [worldwidewebfind, 2004]. “Bayesian” theory is named after Thomas Bayes (a British mathematician and Presbyterian minister, known for having formulated a special case of Bayes' theorem). However, the term "Bayesian" came into popular use in approximately 1950 [Howie, 2002]. The Bayesian theory is a statistical concept which holds the probability of a certain event occurring under a set of given conditions. The probability can be expected based on how regularly the same event has occurred under the same conditions [Bradley and Louis, 1996]. Bayes’ theorem states that new information can be used to update the conditional probability of an event [Bernardo and Smith, 2000]. While the conditional probability is defined as the
probability of some event A, assuming event B, written $P(A|B)$, and is read "the probability of A, given B".

The probability that an event $E$ occurs given the known occurrence of an event $F$ is the conditional probability of $E$ given $F$; its numerical value is $\frac{P(E \cap F)}{P(F)}$ (as long as $P(F)$ is non zero). If the conditional probability of $E$ given $F$ is the same as the ("unconditional") probability of $E$, then $E$ and $F$ are said to be independent events. That this relation between $E$ and $F$ is symmetric may be seen more readily by realizing that it is the same as saying $P(E \cap F) = P(E)P(F)$ [Joyce, 1999].

Bayes’ theorem is a relationship between conditional and minor probabilities. It can be perceived as a method of combining/merging information. It can be defined by $P(A|B)P(B) = P(A \text{ and } B) = P(B|A)P(A)$ i.e. the probability of $A$ given $B$ times the probability of $B$ is equal to the probability of both event $A$ and $B$ occurring together. This is also equal to the probability of $B$ given $A$ times the probability of $A$ symbolized by $P(A|B)$, which is the joint probability of $A$ and $B$. Dividing the left and right hand sides by $P(B)$, the following will be obtained:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

i.e. the probability of $A$ given $B$ is equal to the probability of $B$ given $A$ times the probability of $A$, divided by the probability of $B$ [Berry, 1996].

In term of SPAM the Bayesian formula can also be written as follows [Joyce, 1999].

$$P(\text{SPAM}|\text{Words}) = \frac{P(\text{Words}|\text{SPAM})P(\text{SPAM})}{P(\text{Words})}$$
The application of Bayesian theory to SPAM detection is useful because it can be used to identify certain features in the email messages, how often they occur, and the probability that each message is a SPAM. By storing this information in a database, this data can then be used to predict the probability of any incoming email message [Androutsopoulos et al., 2000].

3.3 How a Bayesian Filter works?

Before the Bayesian filter can be used to filter the emails, the user has to create two databases with individual words and tokens (for example the #, $, *... signs, IP addresses, some specific words “Viagra” and domains…etc), gathered from two representative samples of SPAM emails (SPAM word list) and legitimate emails (HAM word list), where HAM refers to legal emails.

The filter will assign a probability value for each word or token based on how often that word or token occurs in the email message. The probability value designated to each word or token is commonly known as spamicity, and ranges from 0.0 to 1.0 [Process, 2005] as shown in Figure 3.1 below.

![Figure 3.1: Spamicity value](image-url)
The spamicity could be one of the following possibilities

- Spamicity > 0.5 then the word (or the message) is most likely a SPAM.
- Spamicity < 0.5 then the word (or the message) is most likely a HAM.
- Spamicity = 0.5 then the word (or the message) is neutral, meaning that it has no effect (like “for”, “what”, “your”…etc).

For example if the word “finance” appeared 60 times in SPAM messages but only 3 times in legitimate messages. Therefore the spamicity of the word “finance” is increased and added to the SPAM word list. The opposite is true if we use as an example the word “inquiry” occurs 60 times in legitimate messages but only 3 times in SPAM messages. By calculating the spamicity value, the word “inquiry” will be added to the HAM word list.

After the SPAM and the HAM databases have been created, the probabilities of any word in the email message can be calculated and the filter will be prepared for use. This is illustrated in Figure 3.2.

![Figure 3.2: Creating a word database for the Bayesian filter [GFI-Support, 2005]](image-url)
When the new email message arrives, the filter will break down the incoming message into words, and calculate the spamicity value for each word, and then calculates the probability of the newly arrived message. If the probability is greater than 0.5 then the message is SPAM. Obviously more training the filter receives the more accurate the results will be [Process, 2005].

3.4 Advantages of the Bayesian filtering

Some of the advantages of the Bayesian filtering system include:

- The Bayesian method scans every word in the email message, and takes into consideration all the words. Based on this information the filter calculates the probability of whether the email message is SPAM or not. A message may, for example, contain suspicious words like (sex, free, cash…etc). The Bayesian filter will consider the whole message, and calculates the spamicity of that message. Depending on the spamicity value, the message will be considered as a SPAM or not.

- The Bayesian method is a self-learning technique, and it keeps building the database whilst filtering messages. Some spammers, for example, may want to trick the filter by twisting the words, (5ex instead of sex, or CaSh instead of cash) [Graham, 2004]. However, the Bayesian filter can be trained by updating the SPAM database with those twisted words to recognize these tricks, and detect the twisted words if they are going to be included in the next email message. The updating process could be done manually at this stage (the user
will enter the twisted words in the SPAM words database), or automatically (a future enhancement of GetEmail5 SPAM filter).

- The Bayesian method takes into consideration the user’s email profile (individuals or companies). If the SPAM messages have their own words and characters, the Bayesian filter can detect them. Spammers have found that it is not easy to put the recipient’s email profile into consideration [Dietz, 2003].

- The Bayesian filter is language free, and can be used in non English-speaking countries. Consequently, the filter can detect SPAM emails in other languages.

- The Bayesian filter is hard to deceive. Some spammers try to fool the filter by using words that normally designated to a valid email (e.g. important, update…etc), or they may break up the word (e.g. f-r-e-e instead of free). However, with the Bayesian filter, these techniques will only increase the possibility of the message being identified as SPAM, because it is uncommon that the legitimate users will write (free) as (f-r-e-e-) [GFI, 2005].

3.5 Design of the Proposed Filter

Currently, no perfect SPAM filter has been found [WhichSpamFilter, 2005]. However, the method proposed by the author is an attempt to find an intelligent, trainable SPAM filter that can block SPAM emails and let legitimate emails pass through using a combination of techniques including the use of Bayesian filtering [Hassan et al., 2006].
The proposed SPAM filter (GetEmail5) was coded using Java™ programming language, and was designed to be compatible with IMAP (Internet Message Access Protocol) protocols only.

IMAP is a protocol for retrieving email messages remotely from the email server. Using IMAP will allow the user not only to access his/her email but to also control messages stored on the email server without downloading the email messages. Thus, messages can be deleted, have their status changed (read, new, unread…etc referred to as a flag) and multiple mail boxes can be managed [UnitedYellowPages, 2005].

To simplify the concept of flag, a flag indicates the status of an email message. For example if an email message arrived, the flag will indicate that this message is unread (new message), and if an email message has been read, the flag will indicate that this message is checked (old message).

When a new email message arrives the IMAP protocol will set a flag for it as a new message, and when the user reads it, the protocol will set the flag to read message. This makes it easy to differentiate between new and old messages (read and unread).

By using the IMAP protocol the filter will process only the newly arrived messages and not to all messages in the inbox.

The architecture and the algorithm used in the proposed approach are illustrated in Figure 3.3. It can be seen that, GetEmail5 is composed of three filters which act in tandem. The first filter (Whitelist filter) will allow only the trusted email messages to pass through to the inbox. The second filter (Blacklist filter) will block the known SPAM messages. The third filter (Bayesian filter) will determine whether the incoming new email is going to be recognized as a legitimate or SPAM message.
The sequence of filtering methods used in GetEmail5 aims to:

- Speed up the process of receiving the legitimate emails, by using the whitelist method at the beginning. If a new message arrives the GetEmail5 will process the message against the whitelist filter, and sends the message to the inbox.

- Block the known SPAM emails, by using the blacklist method. If a new SPAM message arrives the GetEmail5 will process the message against the second process which is the blacklist filter.

- Reduce the user intervention, and builds up an auto-update technique for both SPAM, and legitimate emails, by using the Bayesian filter.
Figure 3.3: The design algorithm of GetEmail5
3.5.1 Whitelist filter

The first step in the GetEmail5 filter checks the incoming email address (or IP, domain), against a text file of the legitimate email addresses. The file (whitelist.txt) is created and stores the information (email addresses) of the email that the user wants to allow to pass through to the inbox. The filter checks the FROM field (which contains the incoming email address) against the whitelist database, and if the filter finds that the incoming email address is in the whitelist the filter will allow the message to pass through to the inbox. The filter will give a classifier [A] for the legitimate email messages. The main reason for making the whitelist filter the first filter is to make the checking process for the legitimate emails faster. Moreover, there is no waiting time until the filter finishes checking the other (non-legitimate) emails. If the incoming email is not in the whitelist then it will move to the next step which is checking the incoming email against the blacklist database.

3.5.2 Blacklist filter

The second step in the filter checks the incoming email address against a text file of the SPAM email addresses. The file (blacklist.txt) is created and used as storage of the SPAM (unwanted) email addresses that the user wants to block, and to prevent them from reaching the inbox. The filter checks the FROM field of the email against the blacklist file, if the filter finds that the incoming email address matches any one in the blacklist database, it will block the email message, and prevents the message form reaching the inbox. The filter will give a classifier [D] for the SPAM email messages. At this stage the filter checks only the email addresses against the whitelist or the blacklist databases.
3.5.3 Bayesian Filter subject to HAM or SPAM

If the filter has not recognized the incoming message as a whitelist or a blacklist, the Bayesian filter will be applied on `<SUBJECT>` field and the content `<BODY>` of the message. The filter scans through the message, and creates a probability of every word (spamicity). This spamicity value is assigned to each word, and ranges from 0.0 to 1.0. [Process, 2005]. If the spamicity value is greater than or equal to 0.5 then the message containing the word is likely to be SPAM. The filtering process checks (using Bayesian method) the incoming email message against the SPAM words text file which is called (defaultStopWords.txt). The content of this file will be similar to (Vi ag r a, ¼aŒÁžÍºé1šÔÜ­5ŠÔÜ…etc). Furthermore, the users may include their own custom SPAM phrases. The content of this text file is updated manually at this stage, but it may also be updated (in future work) form the web as an automated update. The filter will decide if the incoming email is a SPAM email. Then the filter will prompt the user (the incoming message is subject to SPAM do you want to add to blacklist “Y/N”). If the user entered Y, the message will be blocked, and the email address will be added to the blacklist file. When another email is received from the same sender, the email message will be blocked. If the user entered N, then the message will pass through to the INBOX, and the email address will be added to the whitelist file.

If the spamicity value is less than 0.5, the message containing the word is likely to be HAM. This is achieved by checking the words against a text file of HAM words (non-suspicious words) called (ham.txt), which is customized by the user. Thus, the user will be prompted with the message (the incoming message is subject to HAM do you want to add to whitelist “Y/N”). If the user entered Y, then the message will pass
through to the INBOX, and the email address will be added to the whitelist. The whitelist is a text file for the legitimate email addresses called (whitelist.txt), and it will store the incoming legitimate email addresses. If another email is received from the same email address, it will go directly to the inbox.

If the user entered N, the message will be blocked, and the email address will be added to the blacklist file, and further emails from the same source will be blocked as discussed above.

3.6 Implementation of the GetEmail5 SPAM Filter

GetEmail5 SPAM filter was written by using Java™ programming language. Java™ programming language was chosen because of the following reasons:

- Platform Independence: write a program once, and run it anywhere.
- The flexibility of the syntax makes Java™ programs easy to write and read.
- The Java Virtual Machine (JVM), which is used to run Java™ programs, is available online for free.
- Multithreaded: A program written in Java™ can perform several tasks at the same time.
- The Java Application Programming Interface (API) contains many packages for many different purposes, which made programming easy and flexible.
- Automatic garbage collection: (The term garbage collection refers to the reclamation of unused memory space). [Reilly, 2006]

“When applications create objects, the JVM allocates memory space for their
storage. When the object is no longer needed (no reference to the object exists),
the memory space can be reclaimed for later use” [Reilly, 2006].

GetEmail5 applies Bayesian method to classify emails. Bayesian classifier was
obtained from http://sourceforge.net, which is one of the world’s largest open source
software development website [SourceForge, 2003] [Lothian, 2004].

The implementation steps of GetEmail5 are described below:

Step 1: Message classification

Email messages would determine one of the four program outputs below:-

- Message [A] = the message is definitely a Whitelist Message
- Message [B] = the message is not in Whitelist but subject to HAM
- Message [C] = the message is not in Whitelist but subject to SPAM
- Message [D] = the message is definitely a Blacklist Message

Step2: Variables declaration:

- String variables: to store the names of the files
  “blacklist.txt”, “whitelist.txt”, “defaultStopWords.txt”, “ham.txt”
- Integer variables: as counters
  blackListCount, whiteListCount, stopWordListCount, hamWordListCount
- Arrays: to store the content of the files
  blackListArray, whiteListArray, stopWordListArray, hamWordListArray.
- Boolean variable: (flag) which is to identify the status of the message.
Step 3: Checking status:

- Check the number and status of the email message (flag) i.e. new or old message.
- Output the number of new email messages to the user.

Step 4: Filtering processes:

- Whitelist check:
  
  If the incoming email address is found in whitelist array
  
  Then the message is a legitimate message.
  
  display ("[A] Definitely Whitelist ").
  
  display the email address, and the content of the message.

- Blacklist check:
  
  If the incoming email address found in blacklist array.
  
  Then the message is an illegitimate message
  
  display ("[D] Definitely a Blacklist Message ").

- Bayesian check:
  
  If the incoming email address is not in the whitelist or the blacklist
  
  Then calculate the spamicity of the entire email message.
  
  If spamicity < 0.5 then message is subject to HAM
  
  display ([B] not in the whitelist but subject to HAM)
  
  display (Do you want to add to whitelist Y/N)
If (Y) then add the incoming email address to the whitelist.

If (N) then add the incoming email address to the blacklist.

if spamicity >= 0.5 the message is subject to SPAM

display ([C] not in the blacklist but subject to SPAM)

display (Do you want to add to blacklist Y/N)

If (Y) then add the incoming email address to the blacklist.

If (N) then add the incoming email address to the whitelist.
3.7 Operation of the GetEmail5 SPAM Filter

GetEmail5 is operated in command line environment. This design decision was made as the emphasis of the research was to test the efficiency and the accuracy of our method to detect SPAM emails not emphasizing on the design of the graphical user interface (GUI).

1- GetEmail5 class was created (GetEmail5.class), and the filter would be operated by typing the command shown below (Figure 3.4).

![Figure 3.4: Run the filter](image1)

2- Then the GetEmail5 filter displayed the number of messages that have been received, as shown below (Figure 3.5).

![Figure 3.5: Number of messages](image2)
3- The filter then checks the incoming messages against the whitelist. If the filter found the incoming email address in the whitelist, the filter will produce the message showing the category of the message [A] (whitelist), the sender’s email address, the subject, and the content of the message (Figure 3.6).

![Figure 3.6: A whitelist email message](image)

4- If the incoming email address was not in the whitelist, then the filter will check the incoming message against the blacklist. If the incoming email address was in the blacklist, the filter will prompt ([D] this is a blacklist Email). For example if a user wanted to read newly found emails on the blacklist, the filter will display the content as shown in Figure 3.7 below.

![Figure 3.7: A blacklist email message](image)
6. If the filter could not recognize the incoming email message as a whitelist message or blacklist message, then the Bayesian method was applied to the incoming message. The filter scanned through the message, and created a probability of every word.

7. If the spamicity value was greater than or equal to 0.5 then the message containing the word was likely to be SPAM. Consequently, the following prompt was sent to the user (e.g. the incoming message is subject to SPAM do you want to add to blacklist “Y/N”) (Figure 3.8). If the user replied Y, the message was blocked, and the email address was added to the blacklist array. If on the other hand the user replied N, then the message passed through to the INBOX, and the email address was added to the whitelist array.

8. If the spamicity value was less than 0.5, the message containing the word was likely to be HAM. Therefore the user received the following prompt message; the incoming message is subject to HAM do you want to add to whitelist “Y/N” (Figure 3.9).
If the user replied Y, then the message passed through to the INBOX, and the email address was added to the whitelist array. If the user replied N, the message was blocked, and the email address was added to the blacklist array.

Figure 3.9: Subject to HAM email message

C:\SpamFilter> java GetEmail5
[E] Not in White List but Subject is in HAM
You have received a message from a new user
From: Web Master <admin@aic.wa.edu.au>
Subject: SPAM Warning

To all AIC users,
Be careful, there is a new VIRUS attacking our network.
DO NOT open any strange email.

Best Regards,
Web Master  Do you want to add it to White List? [Y/N]
CHAPTER 4. METHODOLOGY

4.1 Introduction

The designed SPAM filter GetEmail5, was evaluated to illustrate its accuracy and efficiency in detecting SPAM emails. In this chapter the steps taken to prepare the conditions needed by the SPAM filter to run are examined. Following this, the processes of sending batch of email messages to the SPAM filter are described. Moreover, the result of a comparison between GetEmail5 SPAM filter and two commercial SPAM filters are presented. The comparison between filters included sending emails (SPAM and non-SPAM emails), as well as a batch of pure plain text and HTML emails to the filters.

4.2 Preparing the filter requirements

There were some essential requirements that had to be fulfilled before the GetEmail5 could be used. These included the creation of a new email account to receive batch of emails and collecting data set of SPAM emails.

4.2.1 Creation of a new email account

GetEmail5 SPAM filter was designed to work with the Internet Message Access Protocol (IMAP). IMAP is a method of accessing emails on the email server without downloading the emails to the local hard drive. The IMAP method was used because of its flexibility with using flags.
FastMail (email server) is a freeware IMAP server. FastMail was selected due to its speed as it does not allow for images to be displayed (downloading pictures take time), or advertisements. It has lean HTML pages, which also affords fast display. It also provides both external Post Office Protocol (POP3) and IMAP access which can be used with Outlook Express or Netscape Communicator [Mueller and Howard, 2004].

An IMAP email account getemail5@fastmail.fm was created to receive SPAM emails as shown in figure 4.1.

Figure 4.1: Fastmail (IMAP) email account
4.2.2 Obtaining a bulk Email Sender

To test the effectiveness of the GetEmail5 it was necessary to create and send SPAM emails to the getemail5@fastmail.fm email account. “Prospect Mailer® V7” was purchased online to achieve this purpose. The “Prospect Mailer®” software, turned the computer into a personal email server, (i.e. there is no need to have an outgoing mail server). It can accurately send out up to 50,000 emails per hour on a standard 56k modem line. In addition the software has the facility of using multiple channels, which facilitates sending out email messages at a high speed. Prospect Mailer® sends email messages using either plain text or HTML (rich text and graphics) and imports email addresses from standard text files, word documents, or database files [IMT, 2004].

4.2.3 Collecting SPAM emails (Data set)

SPAM email messages were downloaded from a SPAM archive website. “http://www.spamarchive.org/” is a community resource that provides a database of known SPAM to be used for testing, developing, and benchmarking anti-SPAM tools. The goal of this website is to provide a large repository of SPAM that can be used by researchers and tool developers [Spam-Archive, 2005].
4.3 Sending SPAM emails

SPAM email messages were then sent using “Prospect Mailer© V7”, using the following procedure:

- Firstly, the email address to which SPAM messages were to be sent were added to the program, which can be achieved either by typing the email address in manually and adding the address to the email list, or importing a file of email addresses, as shown in Figure 4.2.

![Figure 4.2: First step of sending SPAM message](image)

- Secondly, a SPAM message was typed, or copied and/or pasted from the SPAM archive website. The email messages could be plain text messages as shown on Figure 4.3 or HTML email messages as shown in Figure 4.4.
Figure 4.3: Plain text SPAM message ready to send

Dear Sir/Madam,

we have logged your IP-address on more than 40 illegal Websites.

Important: Please answer our questions!

The list of questions are attached.

Yours faithfully,

M. John Stellford

Figure 4.4: HTML SPAM message ready to send

Super CHEAAP Softwares & Shipp to All Countries
We have every POPULAR softwares u NEED!
You name it & we got it!

- Microsoft Windows XP Professional - my price: $50; normal: $299.00
- Adobe Acrobat v6.0 Professional PC - my price: $100; normal: $449.95
- you save $249.99

You name it & we got it!
• Then the “From Address” field is filled in using the sender email address, (either real or false), and the subject is added also (Figure 4.5).

![Figure 4.5: Information before sending the SPAM message](image)

• Once all the fields had been filled in, the mailing list was processed (Figure 4.6).

![Figure 4.6: Process the mailing List](image)
Finally, the email message was sent (Figure 4.7).

The above steps were repeated numerous times with different email contents (i.e. from “sender address”, Subject, and Body fields).

For the experiment the results of each attempted send was recorded (see section: 4.3) after each batch of SPAM email messages were sent using “Prospect Mailer© V7” software [IMT, 2004].

After sending all the messages to the email account (getemail5@fastmail.fm), the GetEmail5 SPAM filter was run by typing the following command (java GetEmail5). GetEmail5 then started filtering the email messages.
4.4 Receive and analyse the emails.

Table 4.1 shows the complete set of emails sent to GetEmail5 SPAM filter. GetEmail5 was operated to sift through the received email messages, and the results were recorded.

<table>
<thead>
<tr>
<th>Test Number</th>
<th>Total No of Emails</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19 (2 non-SPAM + 17 SPAM)</td>
</tr>
<tr>
<td>2</td>
<td>12 (ALL SPAM)</td>
</tr>
<tr>
<td>3</td>
<td>26 (ALL SPAM)</td>
</tr>
<tr>
<td>4</td>
<td>21 (ALL SPAM)</td>
</tr>
<tr>
<td>5</td>
<td>20 (2 non-SPAM + 18 SPAM)</td>
</tr>
<tr>
<td>6</td>
<td>29 (1 non-SPAM + 28 SPAM)</td>
</tr>
<tr>
<td>7</td>
<td>56 (2 non-SPAM + 54 SPAM)</td>
</tr>
<tr>
<td>8</td>
<td>36 (1 non-SPAM + 35 SPAM)</td>
</tr>
<tr>
<td>9</td>
<td>38 (ALL SPAM)</td>
</tr>
<tr>
<td>10</td>
<td>103 (1 non-SPAM + 102 SPAM)</td>
</tr>
<tr>
<td>11</td>
<td>65 (ALL SPAM)</td>
</tr>
<tr>
<td>12</td>
<td>107 (4 non-SPAM + 103 SPAM)</td>
</tr>
<tr>
<td>13</td>
<td>85 (ALL SPAM)</td>
</tr>
<tr>
<td>Total</td>
<td>615</td>
</tr>
</tbody>
</table>

4.5 Evaluation

The evaluation part of this research required a comparison of the GetEmail5 SPAM filter against two existing commercial SPAM filters. The commercial SPAM filters were chosen from the Top Ten Reviews website (http://spam-filter-review.toptenreviews.com). This website tests and ranks the commercial SPAM filters according to specific criteria as shown in Figure 4.8.
Effective at filtering: SPAM filtering software needed to have a good balance between being able to filter SPAM and allowing valid emails to pass the inbox. It also needed to be able to customize the sensitivity of the SPAM filtering.

Ease of Use: The filters needed to be easy to setup and the filtering rules also needed to be easy to set up.

Email Processing Steps: The SPAM filter should not require any additional steps to filter the incoming emails (i.e. extra commands are not needed).

Allow/Blocking of Email: The files needed to give the user the ability to block or allow email based on the senders email address, IP addresses, server or domain addresses, or country of origin.
• **Content Categories**: SPAM filters had to include the option to specify the blocking or non-blocking SPAM messages with content categories. For example, contents might include pornography, financial, games, gambling, services, health, insurance, adult, etc. Content categories will make it easier for the user to recognize and identify what should be allowed or blocked, especially if the user doesn’t know who the sender is.

• **Rule Creation**: The SPAM filter needed to allow the user to easily define and customize certain rules. These rules might be in combination with allow/block options.

• **Quarantine Area**: The SPAM filters should provide a quarantine area where all blocked email is stored. This allows the user the opportunity to retrieve email that may have been accidentally blocked.

Two filters were chosen for this experiment; the first SPAM filter was *EmailProtect* shown in Figure 4.9 below.
The second SPAM filter was *SpamEater Pro*© shown in Figure 4.10 below.

![SpamEater Pro SPAM Filter](image)

**Figure 4.10: SpamEater Pro© SPAM Filter**

*EmailProtect*© has been awarded the golden rank by Customer Relationship Management (CRM), which it is a pre-eminent meeting place for professionals and companies that provide CRM-related products and services [CRMToday, 2002].

*EmailProtect*© SPAM filter is widely used in association with the Eudora software environment [ContentWatch, 2005].

*SpamEater Pro*© has been awarded the silver award. This filter was chosen because the filter can be used as a stand alone filter (e.g. no other software required), which gives a different environment of filtering to evaluate [Yurasov, 2005 ], [TopShareware, 2005].
To ensure adequate comparison of the filters, all three filters were set up in the same environment. All three SPAM filters were installed and configured on the same personal computer, running under the same operating system (Windows XP with service pack 2), using the same speed Internet connection (dialup connection 56 kbit/s), and none of those SPAM filters had been run previously (i.e. all of the rules were not configured as yet).

Three different email accounts were created to perform the comparison. (getemail5@fastmail.fm) email address for GetEmail5 SPAM filter, (emailprotect@fastmail.fm) email address for EmailProtect© SPAM filter and (spameater@fastmail.fm) email address for SpamEater Pro© SPAM filter.

4.5.1 First experiment (mixed emails)

For general training a batch of 391 mixed emails (SPAM and non-SPAM) was sent to the three email accounts using Prospect Mailer©.

4.5.2 Second experiment (80% SPAM and 20% non-SPAM emails)

1000 mixed emails (SPAM and non-SPAM) were sent in batch of 100 emails at a time (80 SPAM and 20 non-SPAM). The reason behind sending a batch of 100 is that it is simpler to calculate the percentage of the SPAM emails detected and also because the 80-20 percent SPAM to HAM ratio is a common make up of normal email traffic [Foley, 2006].
4.5.3 Third experiment (100% plain text SPAM emails)
All of the emails that were used in the previous experiments were mixed between HTML and plain text. This experiment was performed to check the ability of the three SPAM filters on dealing with pure plain text SPAM messages, by sending a batch of 100 plain text SPAM email messages.

4.5.4 Fourth experiment (100% HTML SPAM emails)
This experiment was performed to check the ability of the three SPAM filters on dealing with pure HTML SPAM messages, by sending a batch of 100 HTML SPAM email messages.
CHAPTER 5. RESULTS AND DISCUSSION

5.1 Introduction

A mixture of SPAM and non-SPAM emails was sent as a first run to examine the performance of the proposed SPAM filter GetEmail5. Following this a combination of plain text and HTML emails was used to evaluate the precision of the designed SPAM filter against two commercial SPAM filters. The results of the comparison between our SPAM filter and the two commercial SPAM filters are discussed in detail in this chapter.

5.2 Initial evaluation of the GetEmail5 SPAM filter

A total of 615 emails were used to test the operation and the efficiency of GetEmail5 SPAM filter Table 5.1 shown below.

<table>
<thead>
<tr>
<th>Test Number</th>
<th>Total No of Emails</th>
<th>Subject to HAM (User)</th>
<th>Subject to SPAM (User)</th>
<th>SPAM Detected</th>
<th>White List</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19 (5 non-SPAM + 14 SPAM)</td>
<td>7</td>
<td>8</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>12 (All SPAM)</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>26 (All SPAM)</td>
<td>12</td>
<td>13</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>19 (All SPAM)</td>
<td>7</td>
<td>8</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>20 (3 non-SPAM + 17 SPAM)</td>
<td>1</td>
<td>2</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>29 (All SPAM)</td>
<td>0</td>
<td>1</td>
<td>27</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>56 (4 non-SPAM + 52 SPAM)</td>
<td>1</td>
<td>2</td>
<td>50</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>36 (3 non-SPAM + 33 SPAM)</td>
<td>0</td>
<td>1</td>
<td>33</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>38 (All SPAM)</td>
<td>0</td>
<td>1</td>
<td>37</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>103 (3 non-SPAM + 100 SPAM)</td>
<td>1</td>
<td>2</td>
<td>98</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>65 (2 non-SPAM + 63 SPAM )</td>
<td>2</td>
<td>3</td>
<td>59</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>107 (5 non-SPAM + 102 SPAM)</td>
<td>1</td>
<td>2</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>85 (All SPAM)</td>
<td>1</td>
<td>2</td>
<td>82</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>615</td>
<td>38</td>
<td>51</td>
<td>508</td>
<td>18</td>
</tr>
</tbody>
</table>
The initial results indicated that the filter was not very effective in blocking SPAM at the beginning of the testing process. The first trial comprised 19 email messages (5 non-SPAM and 14 SPAM), and it was evident that in this first trial the GetEmail5 detected only one SPAM message out of 14 SPAM messages sent. This was due to the filter not being fully trained. Table 5.1 demonstrates that as training progressed GetEmail5 results improved significantly.

*False positive* is defined as a non-SPAM “legitimate email” recognized as a SPAM. *False negative* is defined as a SPAM email recognized as non-SPAM email.

Table 5.1 also shows how many emails were noted to be checked by the user (Subject to SPAM and Subject to HAM). User intervention was required to prevent any false positive or false negative results.

![Figure 5.1: Number of SPAM detected by GetEmail5](image)

After this initial testing phase GetEmail5 was re-installed and reset to allow its evaluation against the two chosen commercial SPAM filters.
5.3 First experiment

A total of 391 emails (comprising of SPAM and non-SPAM emails) was sent simultaneously to the three email accounts. The effectiveness of three filters in detecting and stopping SPAM was varied as shown in Table 5.2 overleaf.

In the first batch of 30 emails that were sent to GetEmail5 only 5 SPAM were detected (approximate accuracy of 15%). EmailProtect\textregistered detected 11 SPAM emails (approximate accuracy of 33%). SpamEater\textregistered failed to detect any SPAM emails at all.

As more emails (SPAM and non-SPAM) were sent to all of the three SPAM filters, their accuracy increased. However, GetEmail5 showed the greatest improvement in detecting SPAM. In the last set of 84 SPAM emails, GetEmail5 for example, detected 69 SPAM out of 84 (82% accuracy) while EmailProtect\textregistered only detected 65% and SpamEater\textregistered 23% of the SPAM emails.

User involvement in GetEmail5 during this stage was higher than that of the other two SPAM filters. This was in order to reduce the number of false positives (non-SPAM email recognized as SPAM email).

Furthermore the number of false positives in GetEmail5 was the lowest compared with the other two SPAM filters as shown in Table 5.3 (Page 72).
Table 5.2: Performance of the three filters in identifying SPAM messages.

<table>
<thead>
<tr>
<th>No of Emails</th>
<th>Filtering Time (Sec.)</th>
<th>SPAM detected</th>
<th>False Positive</th>
<th>False Negative</th>
<th>User Involved</th>
<th>White List</th>
<th>Filtering Time (Sec.)</th>
<th>SPAM detected</th>
<th>False Positive</th>
<th>False Negative</th>
<th>User Involved</th>
<th>White List</th>
<th>Filtering Time (Sec.)</th>
<th>SPAM detected</th>
<th>False Positive</th>
<th>False Negative</th>
<th>User Involved</th>
<th>White List</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 All SPAM</td>
<td>40</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>25</td>
<td></td>
<td>35</td>
<td>11</td>
<td>0</td>
<td>17</td>
<td>2</td>
<td></td>
<td>75</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>70 All SPAM</td>
<td>90</td>
<td>48</td>
<td>0</td>
<td>0</td>
<td>22</td>
<td></td>
<td>97</td>
<td>28</td>
<td>0</td>
<td>41</td>
<td>1</td>
<td></td>
<td>176</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>55 All SPAM</td>
<td>75</td>
<td>52</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td></td>
<td>79</td>
<td>31</td>
<td>0</td>
<td>23</td>
<td>1</td>
<td></td>
<td>117</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>32 All non-SPAM</td>
<td>36</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>9</td>
<td></td>
<td>90</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>17</td>
<td></td>
<td>105</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>58 All MIX</td>
<td>84</td>
<td>28</td>
<td>0</td>
<td>4</td>
<td>22</td>
<td>4</td>
<td>95</td>
<td>29</td>
<td>0</td>
<td>26</td>
<td>3</td>
<td></td>
<td>164</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>62 All MIX</td>
<td>190</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>4</td>
<td>115</td>
<td>33</td>
<td>3</td>
<td>26</td>
<td>0</td>
<td></td>
<td>185</td>
<td>21</td>
<td>1</td>
<td>36</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>84 All SPAM</td>
<td>110</td>
<td>69</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td></td>
<td>133</td>
<td>55</td>
<td>0</td>
<td>29</td>
<td>0</td>
<td></td>
<td>190</td>
<td>20</td>
<td>0</td>
<td>40</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Total = 391</td>
<td>625</td>
<td>244</td>
<td>23</td>
<td>4</td>
<td>112</td>
<td>8</td>
<td>644</td>
<td>187</td>
<td>18</td>
<td>162</td>
<td>21</td>
<td>3</td>
<td>1012</td>
<td>78</td>
<td>21</td>
<td>91</td>
<td>194</td>
<td>7</td>
</tr>
</tbody>
</table>
Table 5.3: A comparison of first experiment between the three SPAM filters

<table>
<thead>
<tr>
<th>SPAM Filter</th>
<th>Filtering Time (Sec.)</th>
<th>No. of SPAM Detected</th>
<th>False Positive</th>
<th>False Negative</th>
<th>User Involve.</th>
<th>Whitelist</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetEmail5</td>
<td>625</td>
<td>244</td>
<td>23</td>
<td>4</td>
<td>112</td>
<td>8</td>
</tr>
<tr>
<td>EmailProtect®</td>
<td>644</td>
<td>187</td>
<td>18</td>
<td>162</td>
<td>21</td>
<td>3</td>
</tr>
<tr>
<td>SpamEater®</td>
<td>1012</td>
<td>78</td>
<td>21</td>
<td>91</td>
<td>194</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 5.3 above shows a comparison of the first experiment (a total of 391 mixed emails), which can be describes as follows:-

- GetEmail5 has the lowest filtering time compare to the other two filters, because there is no Graphical User Interface (GUI), as GetEmail5 operated via command prompt environment.

- The filtering design of GetEmail5 detected the largest amount of SPAM emails compared to the commercial filters. As the focus of this research was to design a trainable SPAM filter that outperformed existing methods we considered this result as very promising.

- GetEmail5 has the largest amount of false positive (legitimate emails recognized as SPAM) compared to the other two filters. This trait can be reduced by further training of the filter as demonstrated in later experiments.

- GetEmail5 has the lowest amount of false negative (SPAM email recognized as legitimate emails), because of the learning technique, which give the ability to detect unknown SPAM emails.
• Number of emails that the user has to deal with manually is quite high in GetEmail5, as noted earlier. This we note can be improved with further training of the filter.

• GetEmail5 detected the largest amount of the legitimate emails. Considered attributable to the learning techniques (by defining words, email addresses…etc) employed in our design.

Overall, in the initial evaluation, GetEmail5 proved to have superior performance in some aspects of its operation.

GetEmail5’s weak performance indicators were all affected by the amount of training that the filter had undergone. Our belief that further training would improve GetEmail5’s performance in theses areas were born out by the results of later experiments.
5.4 Second experiment (80% SPAM and 20% non-SPAM emails)

A total of 1000 emails (800 SPAM and 200 non-SPAM) were sent to the three email accounts in batches of 100 emails (80 SPAM and 20 non-SPAM). As mentioned earlier the reason of choosing (80% SPAM and 20% non-SPAM) is because this mix can be considered normal email traffic [Foley, 2006].

Table 5.4: Comparison between the three SPAM filters dealing with 1000 mixed emails.

<table>
<thead>
<tr>
<th>SPAM Filter</th>
<th>GetEmail5</th>
<th>EmailProtect®</th>
<th>SpamEater®</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. Emails</td>
<td>1000 (800 SPAM and 200 non-SPAM)</td>
<td>1000 (800 SPAM and 200 non-SPAM)</td>
<td>1000 (800 SPAM and 200 non-SPAM)</td>
</tr>
<tr>
<td>Filtering Time</td>
<td>28.45 min</td>
<td>28.10 min</td>
<td>52.20 min</td>
</tr>
<tr>
<td>Maintenance Time</td>
<td>49.47 min</td>
<td>57.19 min</td>
<td>109 min</td>
</tr>
<tr>
<td>Number of SPAM Detected</td>
<td>687</td>
<td>606</td>
<td>382</td>
</tr>
<tr>
<td>% of SPAM Detection</td>
<td>85.9 %</td>
<td>75.8 %</td>
<td>47.8 %</td>
</tr>
<tr>
<td>False Negative</td>
<td>12</td>
<td>186</td>
<td>424</td>
</tr>
<tr>
<td>False Positive</td>
<td>12</td>
<td>38</td>
<td>79</td>
</tr>
<tr>
<td>White List (non-SPAM)</td>
<td>175</td>
<td>170</td>
<td>115</td>
</tr>
<tr>
<td>Comments</td>
<td>(114 emails subject to the user)</td>
<td>(186 emails subject to the user)</td>
<td>(503 emails subject to the user)</td>
</tr>
</tbody>
</table>

Table 5.4 above illustrates the results of the second experiment between the three SPAM filters of sending a batch of 1000 mixed emails.
The results can be described as follows:-

- The filtering time that GetEmail5 took to filter 1000 emails was higher. This is because GetEmail5 scans the entire contents of the messages word by word.

- The maintenance time (user intervention) of GetEmail5 was the lowest, due to the learning techniques employed in GetEMail5, and also because the filtering and categorization processes in GetEmail5 (whether the email was SPAM or non-SPAM) occurred simultaneously. The two commercial filters on the other hand, first processed all the emails and then produced a list of emails to be checked by the user.

- GetEmail5 detected the largest amount of SPAM emails; with the highest percentage 85.9 % of SPAM detected compared to 75.8 % and 47.8 % detection for EmailProtect® and SpamEater® respectively.

- GetEmail5 also had the lowest amount of false positive, and false negative results compared to the two commercial filters.

If we compare the results of the first experiment (see table 5.3 in section 5.3) and the results of this experiment, we observe significant improvement in this indicator. This is due to the increased training effect on GetEmail5 subsequently resulting in a more accurate performance.

- GetEmail5 detected more legitimate emails compared to the other two filters also.
Overall GetEmail5 outperformed the other two SPAM filters. Figure 5.2 below shows the superior performance of GetEmail5 in detecting SPAM.

Figure 5.2: Number of SPAM detected of each filter

Figure 5.3 above shows that GetEmail5 has the lowest maintenance time compared to the commercial filters, also GetEmail5 has lowest filtering time compared to SpamEater©, but was a little higher than EmailProtect©.
Of the three filters tested, GetEmail5 had the lowest ratio of false positives and false negatives as shown in Figure 5.4 below.

![Figure 5.4: Number of the false positive and false positive of the three filters](image)

The design of GetEmail5 SPAM filter allowed the user to decide (during the filtering process) if the incoming email was SPAM or non-SPAM, and as a result, the possibility of giving false negative or false positive was less than the other two SPAM filters.
5.5 Third experiment (100% plain text SPAM emails)

One hundred plain text SPAM messages were sent to the three email accounts simultaneously, so as to check the performance and the accuracy with plain text email messages only as shown in Table 5.5 below.

Table 5.5: Comparison of the three SPAM filters dealing with 100 SPAM emails sent as plain text.

<table>
<thead>
<tr>
<th>SPAM Filter</th>
<th>GetEmail5</th>
<th>EmailProtect©</th>
<th>SpamEater©</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>No. Emails</strong></td>
<td>100 (ALL SPAM Plain TXT)</td>
<td>100 (ALL SPAM Plain TXT)</td>
<td>100 (ALL SPAM Plain TXT)</td>
</tr>
<tr>
<td><strong>Filtering Time</strong></td>
<td>6.00 min</td>
<td>2.43 min</td>
<td>4.40 min</td>
</tr>
<tr>
<td><strong>Maintenance Time</strong></td>
<td>3.23 min</td>
<td>5.40 min</td>
<td>9.01 min</td>
</tr>
<tr>
<td><strong>No. SPAM Detected</strong></td>
<td>89</td>
<td>65</td>
<td>31</td>
</tr>
<tr>
<td><strong>% of SPAM Detection</strong></td>
<td>89 %</td>
<td>65 %</td>
<td>31 %</td>
</tr>
<tr>
<td><strong>False Negative</strong></td>
<td>5</td>
<td>35</td>
<td>0</td>
</tr>
<tr>
<td><strong>False Positive</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>White List (non-SPAM)</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Comments</strong></td>
<td>6 emails subject to HAM</td>
<td>User has to deal with 35 SPAM manually</td>
<td>User has to deal with 69 SPAM manually</td>
</tr>
</tbody>
</table>


The results can be analyzed as follows:-

- Of the 100 plain text SPAM emails, GetEmail5 identified 89 compared to only 65 and 31 for EmailProtect® and SpamEater® respectively. This is due to GetEmail5 being designed mainly to deal with text messages (see Figure 5.4).

- GetEmail5 only had 5 false negative and 6 emails subject to HAM. In comparison EmailProtect® had 35 false negatives and the operator had to manually decide whether these 35 emails were SPAM or not. With SpamEater®, the user had to decide whether 69 emails were SPAM. Clearly, this affected the maintenance time the user was required to spend checking email status, therefore, maintenance time of GetEmail5 was obviously the lowest amongst the other two SPAM filters. The reason for this was that GetEmail5 categorized the messages during filtering.

- The filtering time was longest for GetEmail5 as it was designed mainly to deal with the plain text messages. The GetEmail5 SPAM filter recorded the longest filtering time as it had to read the entire incoming message word by word looking for the suspicious words, and then calculating the probability of the whole message being SPAM.

- It is also clear that, the more training all of the three filters received, the less false positive and false negative results were obtained.
GetEmail5 once again outperformed the other two filters in detecting SPAM messages as shown in Figure 5.5 above.

Also GetEmail5 had the shortest maintenance as shown in Figure 5.6 below.

This result reinforced the accuracy and the effectiveness of the design of GetEmail5 SPAM filter.
5.6 Fourth experiment (100% HTML SPAM emails)

In the final experiment, an additional 100 SPAM emails in HTML format were sent simultaneously to the three emails accounts of the SPAM filters. Although the GetEmail5 SPAM filter was designed to deal with plain text email messages it provided satisfactory results in the case of HTML email messages, and was more effective in identifying and blocking SPAM emails (Table 5.6).

Table 5.6: Comparison of the three SPAM filters dealing with 100 SPAM emails sent as HTML.

<table>
<thead>
<tr>
<th>SPAM Filter</th>
<th>GetEmail5</th>
<th>EmailProtect©</th>
<th>SpamEater©</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. Emails</td>
<td>100 (ALL SPAM HTML)</td>
<td>100 (ALL SPAM HTML)</td>
<td>100 (ALL SPAM HTML)</td>
</tr>
<tr>
<td>Filtering Time</td>
<td>2.00 min</td>
<td>2.50 min</td>
<td>5.00 min</td>
</tr>
<tr>
<td>Maintenance Time</td>
<td>7.23 min</td>
<td>4.50 min</td>
<td>7.50 min</td>
</tr>
<tr>
<td>No. SPAM Detected</td>
<td>63</td>
<td>61</td>
<td>36</td>
</tr>
<tr>
<td>% of SPAM Detection</td>
<td>63 %</td>
<td>61 %</td>
<td>36 %</td>
</tr>
<tr>
<td>False Negative</td>
<td>5</td>
<td>39</td>
<td>0</td>
</tr>
<tr>
<td>False Positive</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>White List (GOOD)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Comments</td>
<td>32 emails subject to HAM</td>
<td>The user has to deal with 39 SPAM</td>
<td>The user has to deal with 69 SPAM</td>
</tr>
</tbody>
</table>
The results of this experiment can be analyzed as follows:

- GetEmail5 identified the highest number of SPAM emails (Figure 5.5) and only recorded 5 false negatives.

- Moreover the user intervention was only required for 32 emails as opposed to 39 and 69 with EmailProtect© and SpamEater© filters respectively.

- Filtering time was the lowest, as the design of GetEmail5 scans first the FROM field (incoming email address). GetEmail5 is less able to understand the contents of HTML messages.

- The maintenance time was the longest, as GetEmail5 SPAM filter could not decide whether the incoming messages were SPAM or non-SPAM, and left the decision to the user. The user has to deal with many of the incoming emails manually until the filter built its database for these email address.

- As in the third experiment the effect of increased training was evident in all of the three SPAM filters in regard to false positive and false negative results.
The results of the experiment were unexpected as GetEmail5 was designed mainly to deal with plain text SPAM emails, and not specifically HTML messages. However, GetEmail5 outperformed the other two filters in this experiment and detected the largest amount of HTML SPAM emails as shown in Figure 5.7 above. Also it had the shortest filtering time as shown in Figure 5.8 above.
5.7 Analysis of the five Key Performance Indicators (KPI)

This research presented five important KPIs for the evaluation of SPAM filters.

5.7.1 SPAM detection

Figure 5.9: Percentage of SPAM detection of the three SPAM filters across four experiments

Figure 5.9 above illustrates a comparison of the three SPAM filters in detecting SPAM messages across the four experiments carried out in this research. The Y axis represents the percentage of the number of SPAM messages which obtained by calculating \( \frac{\text{Detected SPAM messages}}{\text{Total number of SPAM messages sent}} \times 100 \).

GetEmail5 reached the peak of SPAM detection in experiment three, where all of the SPAM messages were sent in plain text format (GetEmail5 was specifically designed for this), and its performance declined in experiment four, where all the SPAM message were sent in HTML format (GetEmail5 not designed for HTML).

It is very clear that GetEmail5 outperformed the two commercial SPAM filters in detecting SPAM messages across the four experiments.
5.7.2 Filtering time

Figure 5.10 above shows the average of the filtering of the three SPAM filters across the four experiments. The Y axis represents the average time in seconds to filter one email, which was obtained by calculating the average:

\[
\text{Average time (sec.) to filter one email} = \frac{\text{Total number of messages sent}}{\text{Number of minutes for processing a batch of messages} \times 60}
\]

It is obvious that in experiment three GetEmail5 spent the largest amount of time to filter the messages, as all of the SPAM messages that were sent in experiment three were plain text, and GetEmail5 is designed mainly to deal with plain text messages. The average filtering time declined in experiment four, because of all the messages sent were HTML format, and GetEmail5 not designed to deal with HTML contents, as this carried out offline, the user is not actually aware of this cost.
5.7.3 False positive

![Graph showing false positive trend for three SPAM filters across four experiments]

Figure 5.11: Average of false positive of the three SPAM filters across four experiments

Figure 5.11 above demonstrates the false positive (legitimate email detected as SPAM) trend for all the three SPAM filters across the four experiments.

It is obvious that all the improved with training. The more training the better results were.

The Y axis represents the average of the false positives. The average was obtained by calculating \[ \frac{\text{Number of false positives}}{\text{Total number of SPAM messages sent}}. \]

The results obtained from experiment three illustrated that all the three filters reached the “Confidence Point”, which means that after this point the user can have a high degree of confidence in using the filter without any worries of getting false positives any more.
5.7.4 False negative

Figure 5.12: Average of false negative of the three SPAM filters across four experiments

Figure 5.12 above shows the false negative (SPAM email detected as legitimate) trend for all the three SPAM filters across the four experiments. The more false negatives the user gets, the more time the user has to spend checking the email messages.

The Y axis represents the average of the false positive indicator. The average was obtained by calculating \( \frac{\text{Number of false negatives}}{\text{Total number of SPAM messages sent}} \).

GetEmail5 recorded the lowest average of false negatives. However, from the third experiment on the average of false negatives increased slightly. We expect that, this will improve with further training of the filter.
Chapter 5.  
Results and Discussion

5.7.5 User involvement

Figure 5.13: Average of user involvement of the three SPAM filters across four experiments

Figure 5.13 illustrates the average of the user involvement in dealing with the email messages that the filter could not decide whether the incoming message is SPAM or not. The average was obtained by calculating the following fraction:

\[
\left( \frac{\text{Number of messages subject to the users}}{\text{Total number of SPAM messages sent}} \right)
\]

GetEmail5 has the lowest average of user intervention compared to the commercial filters. The user is very much aware of this measure as this is an added cost that is involved in SPAM detection.

In experiment three (plain text) GetEmail5 achieved the lowest average of user involvement. However, user involvement increased in experiment four, because of the use of HTML messages.
This indicator determines the amount of time that the user has to spend to check the email messages, and also considered as a cost factor (the cost of user’s time inside an organization).

*SpamEater*© has the largest average of the user involvement compared to the GetEmail5 and *EmailProtect*©.

Overall GetEmail5 outperformed the two commercial SPAM filters in the number of SPAM detection, and lowest user involvement, but had the largest processing time in dealing with plain text messages. However, this was not a significant disadvantage, because GetEmail5 became more accurate in detecting SPAM.
5.8 Limitations of this research

The results of this research have shown that the performance of GetEmail5 SPAM filter was far more efficient in detecting SPAM email messages than the two commercial filters chosen. However, it is important to consider the limitations of the design, development and testing of the new SPAM filter. The most obvious limitation of the design used was its sole reliance on the Bayesian method, although the Bayesian method is the most powerful tool for text analysis [Graham, 2003], the proposed SPAM filter GetEmail5 was not designed to deal with HTML and image files [Hassan et al., 2006]. Nevertheless, the findings of this study have shown that our SPAM filter GetEmail5 was sufficiently robust to deal with HTML files. However, it is acknowledged that further research and development is needed for HTML emails and emails with images in their content to be more effectively dealt with.

Another limitation of this study was that the developed SPAM filter, GetEmail5, was only tested against two commercial SPAM filters. However, other common filters such as SPAM Assassin, Mail-Washer and McAfee SpamKiller [Bass, 2002], [TomDownload, 2005] for example could be included in a further study of the effectiveness of GetEmail5. However, at the time the comparison was performed, EmailProtect© and SpamEater© were the most powerful filters available on the market according to the criteria used by “Top Ten SPAM Review website” [TopTenReviews, 2005], and therefore considered sufficient for this research.
Finally, the number of SPAM emails used in the testing of the filter may be considered by some to be too small as it has been clearly demonstrated that GetEmail5 becomes more effective with further training. However, we note that other notable studies have used much smaller sampling size than that used in this research. For example O’Brien and Vogel only used 826 emails in the testing of their method for identifying SPAM [O’Brien, 2003]. Similarly, Crawford et al only used 525 emails in their study of building a system to assist users in managing emails [Crawford et al., 2001], and Gupta et al only used 500 emails in designing an anti-SPAM filter [Gupta et al., 2003]. Compared to these other studies, our use of a test set of 2206 emails in our experiments is considered to be more than sufficient to evaluate the performance of GetEmail5 against the two highly ranked commercial SPAM filters.
CHAPTER 6. CONCLUSIONS AND FURTHER WORK

6.1 Conclusions

SPAM is not an easy problem to solve. SPAM has become very popular due to a variety of reasons. Unscrupulous companies and individuals can reap high rewards from unsuspecting victims without having a high ingoing and ongoing investment cost. The most commonly used method for stopping SPAM in use today is the deployment of SPAM filters [Le Zhang, 2004]. SPAM filters use a variety of methods to detect SPAM. However, this research has shown that the Bayesian method, in combination with blacklist and whitelist filters, is most effective in dealing with SPAM. The other two commercial SPAM filters (EmailProtect© and SpamEater©) are not based on the Bayesian method; however they do incorporate black or white filters and a number of other methods.

The GetEmail5 SPAM filter demonstrated superior performance compared to the two commercial filters in both the number of SPAM detected and reduced maintenance time (no complicated maintenance was required after the filter was installed). The time required to filter plain text messages was longer than the other two filters due to the use of the Bayesian method. However, this was not a significant disadvantage because it made the GetEmail5 SPAM filter more effective and accurate in detecting SPAM messages. This research demonstrated that GetEmail5 outperformed the other two filters in terms of detecting SPAM in HTML messages also, however the
difference was not significant between GetEmail5 and EmailProtect©. But GetEmail5 significantly outperformed SpamEater© in dealing with HTML messages.

6.2 Recommendations for future work

To further improve GetEmail5’s performance across the whole range of SPAM types, image recognition could be added to GetEmail5. In addition, updating the SPAM database could also be automated (i.e. online update) to reduce maintenance time.

Currently, GetEmail5 is a stand alone SPAM filter, however, it could be incorporated into existing email packages as a plug-in in future extension to its functionality.

It is hoped that with the advance of filtering technology, such as the proposed SPAM filter GetEmail5, and maturity of anti-SPAM legislation, the problem of SPAM can truly be eradicated one day and therefore restore the efficiency of user and user confidence in the email communication technology.
REFERENCES


References


References


Appendix

GetEmail5 Documentation

Class GetEmail5

java.lang.Object
   GetEmail5

public class GetEmail5
extends java.lang.Object

This Class Filters SPAM by using Java Mail API.

Constructor Summary

GetEmail5()

Method Summary

void appendFile(java.lang.String s, java.lang.String f)
   This is the code to appends to a text file

int blackCountLines(java.lang.String f)
   This method counts the lines in BLACK LIST text files Accepts file name as a string Update the static variable blackListCount (i.e.counter) Returns number of lines as int

int hamWordCountLines(java.lang.String f)
   This method counts the lines in HAM WORD TEXT FILE text files Accepts file name as a string Update the static variable hamWordListCount (i.e.counter) Returns number of lines as int

static boolean isValidEmailAddress(java.lang.String email)

static void main(java.lang.String[] args)

java.lang.String[] readFile(java.lang.String f, int c)
This is the code to read the text file Copies it in a String array Display the array using System.println() Returns an array of String

```java
int stopWordCountLines(java.lang.String f)
```

This method counts the lines in STOP WORD TEXT FILE text files Accepts file name as a string Update the static variable stopWordListCount (i.e.counter) Returns number of lines as int

```java
int whiteCountLines(java.lang.String f)
```

This method counts the lines in White LIST text files Accepts file name as a string Update the static variable whiteListCount (i.e.counter) Returns number of lines as int

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Constructor Detail**

**GetEmail5**

```java
public GetEmail5()
```

**Method Detail**

**main**

```java
public static void main(java.lang.String[] args)

Throws:
java.lang.Exception
```

**isValidEmailAddress**

```java
public static boolean isValidEmailAddress(java.lang.String email)
```
blackCountLines

public int blackCountLines(java.lang.String f)
        throws java.io.IOException
    This method counts the lines in BLACK LIST text files Accepts file name as a
    string Update the static variable blackListCount (i.e.counter) Returns number
    of lines as int
    Parameters:
    f - name of the file
    Returns:
    number of lines as int
    Throws:
    java.io.IOException

whiteCountLines

public int whiteCountLines(java.lang.String f)
        throws java.io.IOException
    This method counts the lines in White LIST text files Accepts file name as a
    string Update the static variable whiteListCount (i.e.counter) Returns number
    of lines as int
    Parameters:
    f - name of the file
    Returns:
    number of lines as int
    Throws:
    java.io.IOException

stopWordCountLines

public int stopWordCountLines(java.lang.String f)
        throws java.io.IOException
    This method counts the lines in STOP WORD TEXT FILE text files Accepts file
    name as a string Update the static variable stopWordListCount (i.e.counter) Returns
    number of lines as int
    Parameters:
    f - name of the file
    Returns:
    number of lines as int
    Throws:
    java.io.IOException
hamWordCountLines

```java
public int hamWordCountLines(java.lang.String f)
    throws java.io.IOException

This method counts the lines in HAM WORD TEXT FILE text files. Accepts
file name as a string. Update the static variable hamWordListCount
(i.e. counter). Returns number of lines as int.

Parameters:
f - name of the file

Returns:
number of lines as int

Throws:
java.io.IOException
```

readFile

```java
public java.lang.String[] readFile(java.lang.String f, int c)
    throws java.io.IOException

This is the code to read the text file. Copies it in a String array. Display the
array using System.println(). Returns an array of String.

Parameters:
f - name of the file

Returns:
listArray as an array of String

Throws:
java.io.IOException
```

appendFile

```java
public void appendFile(java.lang.String s, java.lang.String f)
    throws java.io.IOException

This is the code to append to a text file.

Parameters:
f - name of the file
s - name of the word as a string to be appended

Returns:
void

Throws:
java.io.IOException
```