Strategies and Algorithms for the Effective Control of E-mail Spam

THESIS

Submitted in partial fulfillment of
the requirements for the degree of
DOCTOR OF PHILOSOPHY

by
K MANJUSHA

Under the Supervision of
Prof. Rahul Banerjee

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE
PILANI (RAJASTHAN) INDIA
2015

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE

PILANI (RAJASTHAN)

CERTIFICATE

This is to certify that the thesis entitled "**Strategies and Algorithms for the Effective Control of E-mail Spam"** and submitted by Ms. K MANJUSHA, ID No.2005PHXF003 for the award of Ph.D. degree of the Institute embodies the original work done by her under my supervision.

Signature of the Supervisor: ___________________________________

Name of the Supervisor : **Prof. Rahul Banerjee**

Designation : **Professor**
Department of Computer Science & Information Systems
BITS, Pilani (Rajasthan)

Date:

Acknowledgements

I would like to place on record my profound sense of appreciation for my supervisor Prof.Rahul Banerjee for his intellectual, attentive and valuable guidance. I wish to express my heartfelt thankfulness for his valuable time and patience.. Above all and the most importantly, he provided me unflinching encouragement and support in various ways.

I would like to express my deep sense of gratitude to Prof. V. S. Rao Vice-Chancellor (Acting), BITS, Pilani for providing me an opportunity to work in an area of my choice and for making available all the necessary facilities for the successful completion of this study. I also express my special thanks to Prof. B. N. Jain and Prof.L.K. Maheshwari, former Vice Chancellors, BITS, Pilani for having permitted to pursue my research work at BITS, Pilani. I express my sincere gratitude to Prof. A. K. Sarkar, Director, Pilani Campus, BITS, Pilani for his continuous encouragement throughout the course of this work.

It is my pleasant duty to thank Prof. S. K. Verma, Dean, and Prof. Hemanth Jadhav, Associate Dean, Academic Research Division for their constant encouragement and timely conveyance of all evaluation components.

With gratefulness, I acknowledge the valuable suggestions and support extended by the Doctoral Advisory Committee (DAC) members Prof. Mukesh Kumar Rohil and Dr.K.Hari Babu. I am highly indebted to Prof. Sudeept Mohan, Convenor, Departmental Research Committee (DRC), for his constant encouragement. I place on record my gratitude to Prof. J. P. Misra for his advice and support. I also thank my fellow research scholars for their encouragement. I owe a lot to all the professors and faculty of the Department of Computer Science & Information Systems for helping me in the process of selection of the topic and providing helpful suggestions.

A special thanks to my husband, Prof. N V M Rao for his unconditional and unflinching cooperation. My daughter Saisreya deserve special thanks from me as she is the one who has to endure my absence the most. I am at a loss of words to acknowledge my profound thanks to my parents Dr. K. P Rao and Smt. Saroja who have been a constant source of inspiration throughout the course of this work. I am thankful to my sister, Ms. Usha and my brothers, Rajesh and Srikanth, for cheering me up throughout.

I wholeheartedly thank the nucleus members of the Academic Research Division for their cooperation and support provided at the various stages of this work. I wish to render my thankfulness to the library staff for providing me all the necessary material for this study. I thank the supporting staff of the Department of Computer Science and Information Systems for their timely help. I also thank the reprography section of the Institute for their help.

Finally, I would like to thank everybody who was important to the successful realization of this thesis, as well as express my sincere apology that I could not mention individually every one.

K Manjusha

STRATEGIES AND ALGORITHMS FOR THE EFFECTIVE CONTROL OF E-MAIL SPAM

ABSTRACT

The thesis, presented here, is aimed at identifying different strategies and algorithms for the effective control of E-mail spam. The volume of spam has been increasing significantly over last few decades and therefore there is an urgent need to address the E-mail spam problem. As a part of this work, various impacts of spam mails on organizations and individuals were identified and duly analysed. Consequently, existing technologies, solutions and approaches were duly examined as part of a thorough literature analysis carried out in the processes. As part of this phase of work, various anti-spam techniques were identified and classified. Some of these involve list-based filters, cost-based filters, content-based filters, IP-based filters, etc. Based on this review, research gaps were identified before defining the problem statement.

E-mail spam problem has been defined, in the presented work, as a text classification problem, but in the specific context of typical Internet E-mail generation, relaying, receipt and processing. The choice was made in favour of select machine learning algorithms for spam mail classification. This work presents a set of new algorithms which duly benefit from a careful combination of relevant classification strategies.

The first algorithm presented in this thesis involves a hybrid approach of Bayesian and Neural Network (NN) for classification and Genetic Algorithm for training the NN. In this case, neural network was trained with a specific Genetic Algorithm to speed up the training process. This strategy helps since neural network training is slower, but good in terms of efficiency; whereas, genetic algorithm is good at optimization relevant to this kind of problem space. As a result, careful combination of these two mechanisms leads to both computing efficiency and less training time.

The second algorithm presented here is a combination of Support Vector Machine (SVM) and Decision Tree. SVM proved very efficient in text classification. It provides high accuracy, but is very slow in terms of training time. Decision Trees, on the other hand, classify new instances faster compared to SVMs while SVMs outperform Decision Trees in terms of accuracy. In order to exploit the advantages of both the algorithms, the strategy evolved employs a combined approach of SVM and DT to classify E-mails.

The third algorithm presented here uses an ensemble approach of Map-Reduce based SVM. The intention of this ensemble approach was to reduce training time of the filtering process as well as maintain the high degree of accuracy. SVM was employed as a classifier, as it offers high classification accuracy. This alone, however, was not a good choice since it needs more training time for high dimensional data. With the Map-Reduce process, the training time can be decreased dramatically and this way in this algorithm SVM and Map-Reduce were strategically combined for use.

The fourth and the last algorithm presented in this thesis leads to a parallel and distributed scheme based classification. K-Nearest Neighbourhood (kNN) Join algorithm was used for classification since it works well for multi-dimensional data. A parallel distributed environment was created to reduce the dimensionality of the data. This approach was devised with a distributive strategy that exploits parallel characteristics of kNN which further improved the resultant classification efficiency.

Each of these algorithms was tested as a part of experimental work on four e-mail data sets comprising of three public and one private data sets. The performances of the presented algorithms were compared with those of existing approaches. All these algorithms were evaluated in an experimental parallel and distributed computing environment with respect to optimizing the data, improving accuracy, etc. Combining different relevant techniques with due care was found to improve chosen performance measures and this was precisely what was attempted in this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

ANN	Artificial Neural Network
API	Application Programming Interface
ARPANET	Advanced Research Projects Agency Network
ASIN	Active Strength of Internal Node
BDT-MSVM	Binary Decision Tree Multi-Class Support Vector Machine
BNNC	Bayesian and Neural Network classifiers
CALO	Cognitive Assistant that Learns and Organizes
CAPTCHA	Completely Automatic Public Turing test to tell Computer and Humans Apart
CBN	Cantered Bayesian Network
CPD	Conditional probability Distribution
CPU	Central Processing Unit
DHT	Distributed Hash Table
DK	Domain Keys
DKIE	Domain Keys Identified Email
DNS	Domain Name System
DNSBL	DNS-based Black-hole List
DT	Decision Trees
ESP	Email Service Providers

HDFS	Hadoop Distributed File System
IIM	Identified Internet Mail
IP	The Internet Protocol
ISP	Internet Service Provider
JVM	Java Virtual Machine
Knn	K Nearest Neighbour
LCP	Lightweight currency protocol
LIBSVM	A Library for Support Vector Machines
MAPS	Mail Abuse Prevention System
MIME	Multi-Purpose Internet Mail Extensions
ML	Machine learning
MR	Map Reduce
MSN	Microsoft Network
MTA	Mail Transfer Agent
MUA	Mail User Agent
P2P	Peer to Peer
PBDB	Priority Based Decision Box
RBL	Real-time blacklist
SIDF	Sender ID Framework
SMO	Sequential Minimal Optimization
SMTP	Simple Mail Transfer Protocol

SCL	Spam Confidence Level
SPEWS	Spam Prevention Early Warning System
SPF	Sender Policy Framework
SVM	Support Vector Machine
TF	Term frequency
TF-IDF	Term frequency–inverse document frequency
URL	Uniform Resource Identifier
WEKA	Waikato Environment for Knowledge Analysis
XBL	Exploits Block List
XML	Extensible Mark-up Language
YDK	Yahoo Domain Key

CHAPTER 1
INTRODUCTION

1.1 Introduction

Electronic mail (E-mail) has become the lifeline of the majority of modern business as well as a common vehicle for inter-personal communication between connected people. E-mail is so popular, since it is simple, cost effective and support merely instantaneous delivery. Also, it supports users to keep a record of communication. E-mail has, over the years, revolutionized the way people communicate. According to an estimate by the Radicati Group (2012), the total number of worldwide E-mail accounts is expected to increase from 3.3 billion accounts back in 2012 to over 4.3 billion accounts by the end of 2016.

With such an increase in the use of electronic mail as a means of communication, the volume of unwanted e-mail messages (mostly spam E-mail) that received, annually, is growing significantly as shown in Table 1.1 as reported by Sara Radicati (2015). Spam problem which first surfaced in the first decade of the 21st century, remains a challenge till this day. The Symantec report (2014) places the global spam rate at 63.7 percent for the month of July 2014. Spam e-mails often arrives in the form of continuous, high-volume, fast and time-varying data streams which quietly adapt to any counter-measures. Correct classification of such data streams in a dynamic environment with frequent updates poses a major challenge.

Table 1.1: Daily E-mail Traffic Worldwide

Daily E-mail Traffic	2014	2015	2016	2017	2018
Total Worldwide E-mail Sent/Received Per day (B)	196.3	204.1	212.1	220.4	227.7
% Growth	-	4%	4%	4%	3%
Business E-mail Sent/Received Per Day (B)	108.7	116.2	123.9	132.1	139.4
% Growth	-	7%	7%	7%	6%
Consumer E-mails Sent/Received Per Day (B)	87.6	87.9	88.2	88.3	88.3
% Growth	-	0.3%	0.3%	0.1%	0.0%

The Radicati report (2012) also estimates that over the next four years, the amount of spam received is expected to be roughly 15% of E-mails received. In fact, it has been reported that MSN™ has to block about 2,400,000,000 spam messages a day. Table 1.2 provides a glimpse of select E-mail statistics that reflects legitimate as well as spam E-mail quantum as per the published Sara Radicati study (2014). Spam can cost companies dearly in terms of net loss in productivity, at times, to the tune of about USD 800 on an average, per employee, annually. Companies have to purchase additional servers and storage to cope with the pressure caused by E-mail spam on their mailing systems. It has been estimated that about USD 25.5 Billion shall be spent by companies around the world in purchasing such additional equipment. Most contemporary companies advertise their products through the Internet. This leads to many never-ending streams of spam. Therefore, finding an efficient method to prevent E-mail spam assumes significance and is to be treated as a high priority.

Table 1.2: Magnitude of E-mails Sent/Received Per User/Day

Business E-mail	**2014**	**2015**	**2016**	**2017**	**2018**
Average Number of E-mails Sent/Received Per Day	**121**	**126**	**131**	**136**	**140**
Average Number of E-mails Received Per Day	**85**	**88**	**91**	**95**	**97**
Average Number of Legitimate E-mails Per Day	75	77	79	83	83
Average Number of Spam E-mails Per Day *	10	11	12	12	14
Average Number of E-mails Sent Per Day	**36**	**38**	**40**	**41**	**43**

Here, an attempt has been made to briefly list some of the problems that the Internet users are currently facing due to E-mail spam. It also briefly mentions select existing solutions or suggested broad solution approaches along with their corresponding advantages and disadvantages. The treatment also includes document classification problem, in the context of spam E-mail filtering, with the main challenges that the problem faces.

Abuse of E-mail technology is a vehicle for spreading viruses and worms of various kinds apart from propagation of fraudulently created E-mails in the names of people or organizations who have not really sent them creates yet another dimension to the problem.

Figure 1.1 gives a brief idea of spam hosting countries and Figure 1.2 presents statistics of volume of E-mail vs. spam (McAfee Threats Report, 2012).

It is important to outline here that not every unsolicited E-mail is necessarily unwelcome. In fact, in many cases such mails may serve fairly reasonable purposes like being used for invitations for starting a new collaboration, participation in a conference or a seminar, requesting to work as a referee or a reviewer as well as authorized alerts or remainders set by oneself or one's own organization as well as some other select agencies. Similarly, there may exist situations that may involve moderated list services that may be used by a university for communicating with its students registered in select courses and programs. Even if unsolicited these kinds of use of emailing technology are perfectly valid.

Therefore, the principal problem stems from E-mails which do not fall into the categories of solicited emails and unsolicited but reasonable E-mails of the kind mentioned above. These unwarranted E-mails are what we term here as 'spam' what we seek to achieve through the proposed research is therefore a reasonable control over spam. This research focuses on research and development of trustworthy, reliable and user friendly computing technologies and strategies.

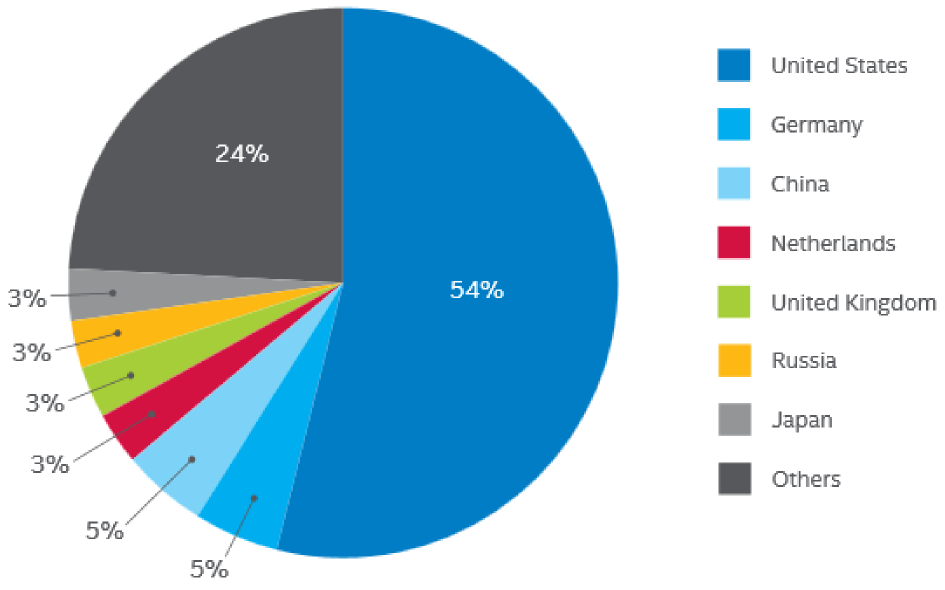


Figure 1.1: Top Countries Hosting Spam Domain

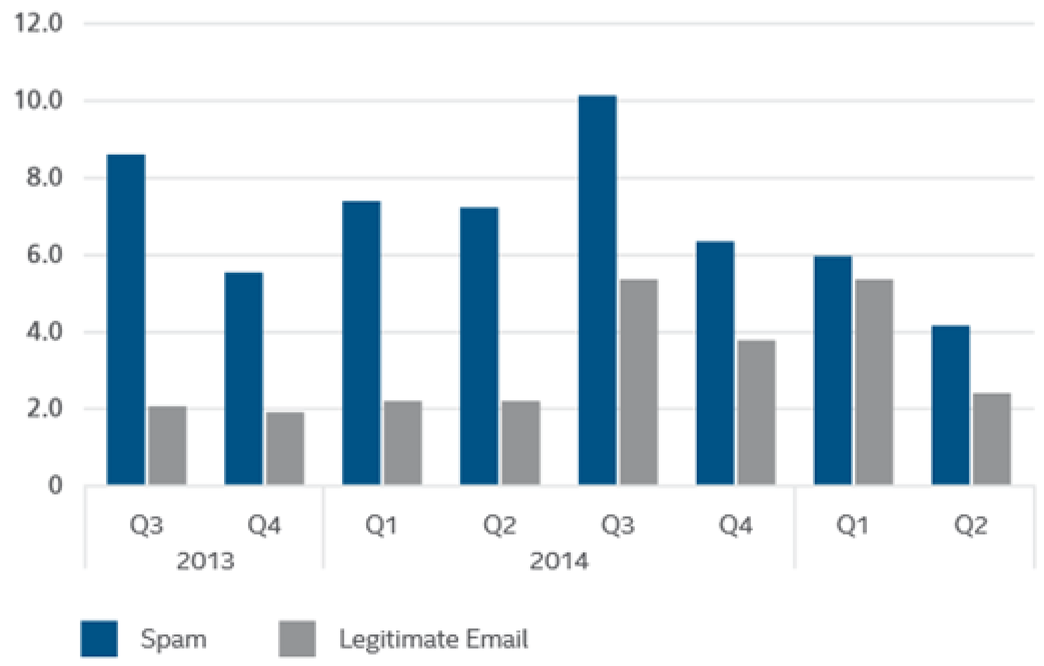


Figure 1.2: E-mail vs. Spam volume

1.2 DEFINITION OF SPAM

The ‘E-mail Spam’ doesn’t yet has a universally accepted formal definition, but it can be broadly described as unwanted E-mail message(s) or unsolicited commercial E-mail(s) (except the kinds specifically mentioned in the Section 1.1). Of late, the volume of certain types of E-mails has significantly increased, which, strictly speaking, may not be necessarily considered to be commercial in nature, but are sent in bulk without consent (expressed or implied) of recipients. Thus, they too are considered to be spam E-mails.

However, E-mails, for which one has signed up or explicitly agreed, cannot be technically classified as spam, even if not useful. It is up to the recipient to consider such mails in one way or another. This kind of interpretation may, thus, vary based on individuals’ perspectives; some might call them spam if they are of no benefit whereas, to others, it might be useful.

Table 1.3 presents some definitions of E-mail Spam by different individuals/organizations.

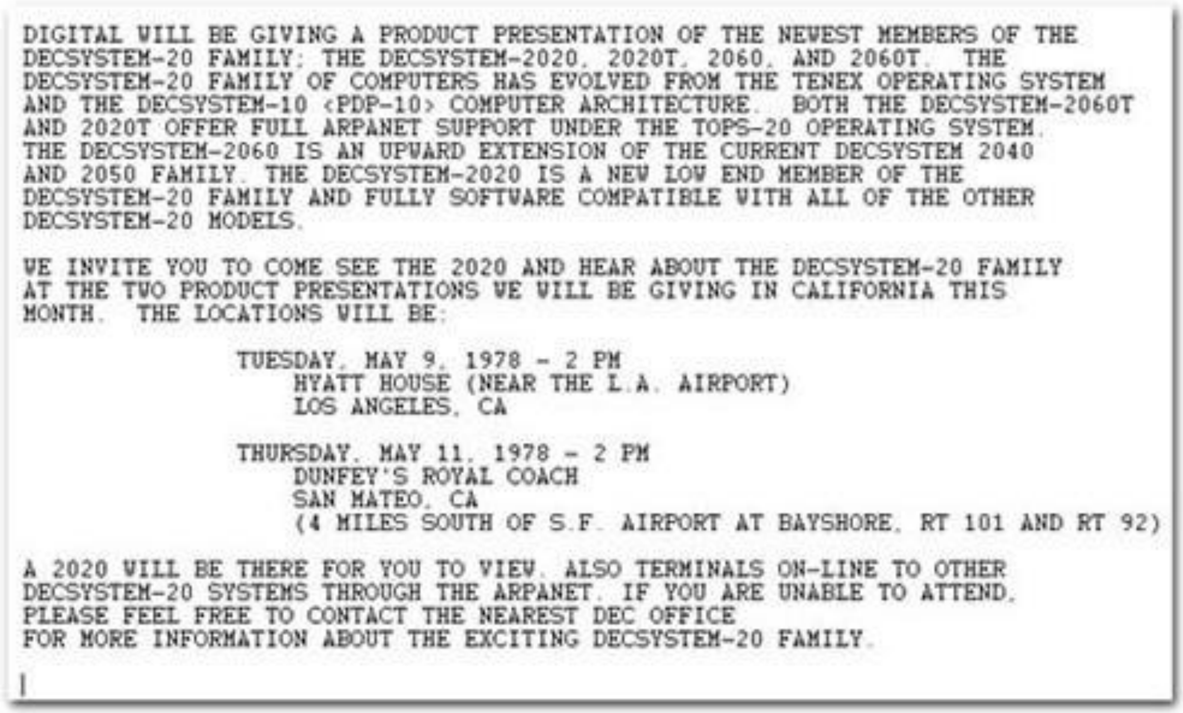
Table 1.3: Definition of E-mail Spam from Different source

Author(s)/(year)	**Definition of Spam**
Vapnik et al. (1999)	An e-mail message that is unwanted: Basically it is the electronic version of junk mail that is delivered by the postal service.
Khong(2001)	SPAM often referred to as unsolicited commercial E-mail (UCE) or unsolicited Bulk commercial E-mail (UBCE) is cheapest and fastest method of advertising for commercial websites
Oda and White (2003)	The electronic equivalent of junk e-mail which typically covers a range of unsolicited and undesired advertisements and bulk e-mail messages.
Lazzari et al. (2005)	Electronic messages posted blindly to thousands of recipients, and represent one of the most serious and urgent information overload problems.
Zhao and Zhang (2005)	Spam or junk mail, is an unauthorized intrusion into a virtual space - the E-mail box
Youn and McLeod (2007)	Spam as bulk e-mail - e-mail that was not asked for which is send to multiple recipients.
Wu and Deng (2008)	Spam e-mails, also known as ‘junk e-mails’, are unsolicited ones sent in bulk (unsolicited bulk E-mail) with hidden or forged identity of the sender, address, and header information.
Amayri and Bouguil (2009)	Spam e-mails can be recognized either by content or delivery manner and indicated that spam e-mails were recognized according to the volume of dissemination and permissible delivery.
Spamhaus (2010)	An electronic message is "spam" if A). the recipient's personal identity and context are irrelevant because the message is equally applicable to many other potential recipients; AND B). the recipient has not verifiably granted deliberate, explicit, and still-revocable permission for it to be sent; AND C). the transmission and reception of the message appears to the recipient to give a disproportionate benefit to the sender.
Text REtrieval Conference (TREC 2005)	Unsolicited, unwanted E-mail that was sent indiscriminately, directly or indirectly, by a sender having no current relationship with the recipient.
Michalson (2003)	"the mosquitoes of the Internet – numerous, annoying and often carrying objectionable content and nasty viruses"
Wikipedia.org (2015)	“Spamming is the abuse of electronic messaging systems to send unsolicited bulk messages, which are generally undesired.”

1.3. Brief History of E-mail Spam

The first ever spam message was sent by Gary Thuerk, a Digital Equipment Corporation (DEC) marketer manually. On May 1, 1978, he sent a message to around 400 people on ARPANET (2006) seeking attention about their product demonstration of the DECSYSTEM-2020, 2020T, 2060 and 2060T computers. Figure 1.3 shows the content of the first ARPANET spam. E-mail chains were introduced in 1982, and in 1989, for multi-user interactive environments (MUD's) a different kind of spam began. These spam mails were tolerable until 1993. It was in the year 1994, when spammers started using machines for spamming instead of doing it manually. Canter and Siegel, were the first major spammers in 1994. They had posted a message regarding US immigration status involving Green Card lottery and this was called Green Card Spam.

On March 31, 1993, the case of accidental use of USENet for bulk E-mail was the first scientifically documented case. Richard Depew unknowingly had posted around 200 duplicate messages to a newsgroup. On 18 January 1994, USENet based bulk mail became the first truly major spam. A lot of newsgroups had got it as religious messages which had declared that Jesus was coming soon. It had caused lots of debate and controversy. The one who sent it, Clarence Thomas had to answer a barrage of questions, but got away with a mild punishment. The other major spam was the Nigerian spam. It had started around 2000 which had annoyed the Nigerian government as they had received a lot of negative heavy publicity. Many variants of such scams continued to appear in later years as well.



```
DIGITAL WILL BE GIVING A PRODUCT PRESENTATION OF THE NEWEST MEMBERS OF THE
DECSYSTEM-20 FAMILY; THE DECSYSTEM-2020, 2020T, 2060, AND 2060T.  THE
DECSYSTEM-20 FAMILY OF COMPUTERS HAS EVOLVED FROM THE TENEX OPERATING SYSTEM
AND THE DECSYSTEM-10 <PDP-10> COMPUTER ARCHITECTURE.  BOTH THE DECSYSTEM-2060T
AND 2020T OFFER FULL ARPANET SUPPORT UNDER THE TOPS-20 OPERATING SYSTEM.
THE DECSYSTEM-2060 IS AN UPWARD EXTENSION OF THE CURRENT DECSYSTEM 2040
AND 2050 FAMILY. THE DECSYSTEM-2020 IS A NEW LOW END MEMBER OF THE
DECSYSTEM-20 FAMILY AND FULLY SOFTWARE COMPATIBLE WITH ALL OF THE OTHER
DECSYSTEM-20 MODELS.

WE INVITE YOU TO COME SEE THE 2020 AND HEAR ABOUT THE DECSYSTEM-20 FAMILY
AT THE TWO PRODUCT PRESENTATIONS WE WILL BE GIVING IN CALIFORNIA THIS
MONTH.  THE LOCATIONS WILL BE:

                TUESDAY, MAY 9, 1978 - 2 PM
                    HYATT HOUSE (NEAR THE L.A. AIRPORT)
                    LOS ANGELES, CA

                THURSDAY, MAY 11, 1978 - 2 PM
                    DUNFEY'S ROYAL COACH
                    SAN MATEO, CA
                    (4 MILES SOUTH OF S.F. AIRPORT AT BAYSHORE, RT 101 AND RT 92)

A 2020 WILL BE THERE FOR YOU TO VIEW. ALSO TERMINALS ON-LINE TO OTHER
DECSYSTEM-20 SYSTEMS THROUGH THE ARPANET. IF YOU ARE UNABLE TO ATTEND,
PLEASE FEEL FREE TO CONTACT THE NEAREST DEC OFFICE
FOR MORE INFORMATION ABOUT THE EXCITING DECSYSTEM-20 FAMILY.
```

Figure 1.3: First ARPANET Spam

1.4. Other Impacts of E-mail Spam

E-mail spam is, at times, considered as the Plague of Internet technology. Individuals, or groups of users, are easily targeted by E-mail spam. The people sending such mails have taken away assets from others. Such abuse has had instances of major economic and social consequences on both private and public systems. According to the reports of The Working Group on Internet Governance (WGIG) the spam has imposed various other costs and causes nuisance reducing the utility of the internet and obstructing internet development (Drake, William J., 2005).

Different types of viruses, worms, Trojans, Spyware, etc. were also spread through spam E-mails, typically as attachments which had an adverse impact on individual users. Because of these malicious mail data on personal personal computers can be damaged. At times, a spam mail based Denial of Service (DoS) attack uses spam E-mail messages to launch an attack on one’s E-mail account. Internet Service Providers (ISPs) and E-mail Service Providers (ESPs) have long felt the adverse impact of spam. Because of growth in spam there is, often a significant decrease in the speed access to the Internet , and because of the increase in E-mail volume E-mail servers are, often, overloaded; and ESPs ISPs has to incur extra costs for network bandwidth, data storage etc.

Spammer’s targets E-mail accounts of school going children also. The Symantec Corporation reported that eighty percent children and youth E-mail users receive spam. Spammer didn’t leave differently abled E-mail users too. Spamcon reports that there was enough evidence that spam affected inboxes of such users as well. Visually-impaired users take the help of speech-synthesis devices to read E-mail, but due to such spam mails they have to spend extra time and resources. For physical mobility-impaired users, deleting excess E-mail is a painful and time-consuming job. Hearing-impaired E-mail users adopt the E-mail-enabled text pagers for immediate remote communications. Because of such spam in their inboxes, the utility of these pagers is, often, decreased to a large extent. Rural E-mail users don’t have much choice for selection of Internet Service Providers and at an even more disadvantage since they are charged higher and for every spam mail they receive they have to pay an extra cost.

Employees in most of the organizations, have to prioritize their inboxes sorting out the important E-mails and scraping the rest, which takes substantial amounts of productive working time due to spam mails. Because of this loss of productivity, there could be a loss of millions of dollars each year per organization. Ferris Research (Michael Sampson, 2003) report states that an employee on an average spends around 115 hours each year and prioritizing their inboxes for an employee wastes $4,000 each year just for their E-mails. In addition to the cost of missed opportunities, there is a bigger issue of an organization's reputation when there is a risk of compromise in a company's E-mail system either via spyware or the ever growing threat of viruses and worms that are engineered to hijack unsuspecting workstations and use them as "zombies" for spreading additional Spam (Email sifter, 2005).

The spam has raised a lot of concerns for the society too, including ethical and select societal concerns. WGIG reported that the spam could cause serious damage to the social life of people if it has not stopped. Spam mails potentially create serious issues of dishonesty and trust in the society (Drake, William J, 2005).

In many perceived cyber threats, the spam is often the first vehicle of attack. The communication between the users alongside the creation of awareness is one possible remedy. Identity theft adds yet another angle to the whole situation. Spammers usually conceal their own identity. Not only individuals, but also organizations could get affected as due to potential identity thefts and source header manipulation, etc. they may face brand name dilution even their domain names might be added to ISP black lists. Many spammers send their messages by using someone else's account without their knowledge as using their own ISPs prohibits this act.

Spam E-mails are a significant drain on the global environment. According to McAfee study (2009), just receiving a spam message unopened has an environmental impact of 0.3gm of carbon dioxide (CO_2). Every year an average employee releases 131 kg of carbon dioxide because of these emissions released by an E-mail. Out of these releases 22 percent of total carbon dioxide was emitted because of spam E-mails.

A typical, normal organization's annual global spam footprint is equivalent to the greenhouse gases pumped out by approximately 3 million passenger cars using twelve billion liters of petrol in a year (McAfee Study, 2009) 18 billion kWh of total spam energy was drained just by viewing and deleting of spam E-mails. 33 billion kWh of spam energy was used every year, with this energy one can supply electricity for 2.4 million homes every year, and one can save 135 tWh of electricity per year by filtering spam.

McColo Inc., a US-based service provider tried to shut down all E-mail activities on November 11, 2008 which resulted a day without spam. Due to this act there was a drastic reduce in global spam volume by seventy percent and consequently a major reduction in electricity use, obviously reduction in carbon emissions and environmental benefit (DiBenedetto, Steve et. al. 2009)

1.5. Categorization of E-Mail Spam

Spam E-mails varies in terms of their types/forms. These unwanted, unsolicited, uninvited, commercial messages which could be annoying, harmless chain letters to potentially harmful mails containing Trojans, viruses, worms, spyware, etc. Many of them are advertisements or marketing spam mails just to promote their products. Spam comes in different formats, styles, only header, normal plain texts, intentionally misspelled texts, HTML format, with or without images, attachments, containing URLs and presented in Figure 1.4. Spam doesn’t have any language barrier, even though most of spam uses the English language.

1.5.1 E-mail Frauds

The messages that tempt us to pay large amount of money and hence ask about our bank account details and the scams that makes us give them our different card or bank account details are called as E-mail fraud. Nigerian scam is one of the most famous and well known money laundering scams. Here an E-mail is received in the name of government from an anonymous source with send posting himself as a government official or a relative of a recipient or a representative of any wealthy deceased. In some cases they request for money on some pretext and assure to repay the money in the coming days.

1.5.2 Chain letter spam

A Chain letter is messages that were asked to pass on to friends circle, generally contained emotionally blackmailing content with a request to forward the mail to a given number of people, at times within a stipulated time. These chain letters are, usually harmless but annoying. These are of two types: a) Hoaxes: Hoaxes are messages trying to deceive users asking to delete a file by claiming it is a virus. And b) Phishing

1.5.3 Malicious mails

The E-mails which misleadingly claim themselves to be from any prestigious organization or enterprise, manipulate the user into providing certain privileged details and then use it for robbing that target of his/her hard earned money are called as phishing scams. They often ask for the information like PIN or pass codes of credit cards or bank account details. But it is very dangerous as such mails often lead the website looks so very similar to the legitimate ones that we are not able to distinguish. The name phishing comes as a variation to the word fishing as the idea that uses mail as the bait which is used to catch a poor victim. The maliciously infected program or document attached with any E-mail which, when opened could forward it to everyone in our list of E-mail contacts. It is called E-mail virus and is dangerous once opened.

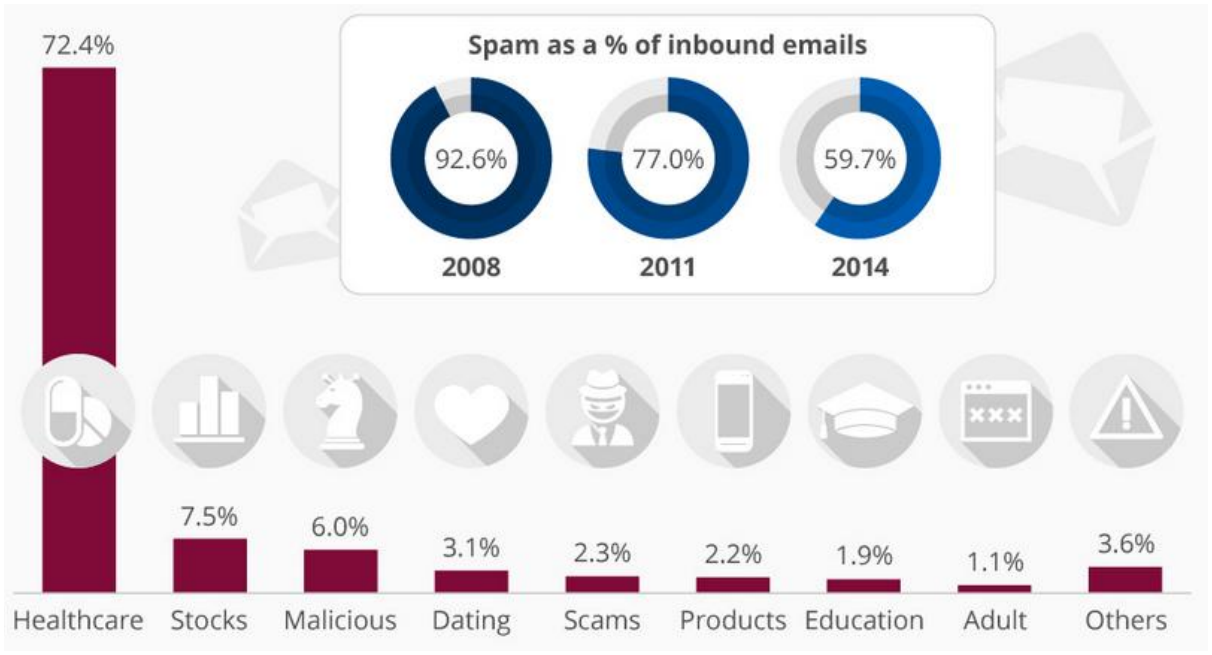


Figure 1.4.: Categories of E-mail Spam

1.5.4 Commercial Advertisements

Technically, if the unsolicited mail is used with a commercial purpose like advertising, promoting products, special offers or any financial investment related issues, they are categorized as commercial spam. Major spam volume comprises of such advertisement spam mails. In spite of the companies being large and having a high brand value, if a mail is received without permission and is meant either for any advertising purpose or insisting for a new sign up, it is considered as spam mail. Some examples include discount offers on software or hardware products, promotions of universities to join a program, stock market or lotteries, health related products etc.

1.6. Motivation behind this Work

The primary motivation of this work is to help prevent or minimize E-mail spam so as to not only reduce the inconvenience caused to individual users by such undesired communication, but also help organizations and corporations to minimize the loss of productivity caused by such spam mails.

The presented work attempts to help achieve such a reduction in spam mails by the way of devising new strategies for effective control of E-mail spam, evolving methods that shall allow these strategies to be implemented and building a proof-of-concept system for evaluation of these strategies in a controlled environment.

1.7. Scope and Objectives of the Present Research

The work focuses on the study and analysis of the contemporary anti-spam technologies as applicable to E-mails and devising new strategies that can reduce spam mails. It also involves development of efficient methods and their implementation aspects. The scope of the presented work does not include creation of a field tested, ready for use software. Analysis and solution of non-textual components of E-mails has not been included in the scope of work.

The specific objectives of the study are:

- Devising and developing new strategies for effective control of E-mail Spam;
- Evolving methods that shall allow these strategies to be implemented and building a proof of concept software system for evaluation of these strategies in a controlled environment.

1.8. Organization of the Thesis

The content of this thesis has been organized in the form of five chapters. While the current chapter (Chapter 1) presents an introduction to the area and provides an overview of the principal motivation, objective and scope of the work done, brief description about the rest of the chapters is as follows.

Chapter 2: Literature Review

This chapter presents a thorough literature survey which involves due analysis of anti-spam strategies, examining the types of solutions available so far for controlling E-mail spam as well as related algorithms. This chapter also analyses the resource requirements of existing algorithms vis-à-vis ways in which they carry out identification of the reasons. It also lists, select situations which may render the prevalent strategies and algorithms less effective or ineffective.

Chapter 3: Modeling the E-mail Spam

In this chapter, brief experimental modeling of E-mail spam has been carried out along with discussions, providing the basis involved and any alternative approach against which the chosen method stands out.

Chapter 4: Strategies and Algorithms

This chapter discusses the chosen strategies, schemes and methods for controlling spam and analysis of their effectiveness and presents corresponding algorithms along with evidence of their performance vis-à-vis known algorithmic approaches and strategies.

Chapter 5: Conclusions

This chapter presents the conclusions drawn on the results obtained from the experiments, compares them briefly with other related works in the spam-mail area and also summarizes both, the principal contributions of the work done as well as its limitations. Planned future work has also been briefly presented at the end.

CHAPTER 2
LITERATURE SURVEY AND ANALYSIS OF PROBLEM DOMAIN

2.1. Introduction

Along with the surge of spam E-mails, consequent issues and problems, *as pointed out in Sections 1.1, 1.4, 1.6 and 1.7*, have also grown manifold. Spam E-mails need to be effectively dealt with without affecting other regular services or functions of organizations. There already exist several approaches to deal with E-mail Spam. Nations have even considered resorting to varying legal measures for dealing with many categories of spam communication. In most of the organizations, anti-spam measures and policies are in place in one or other way either at the technical or regulatory level or both.

In this chapter, a detailed review of the various techniques and technologies employed to protect against the receipt of spam messages available to combat spam has been presented. Researchers have been suggesting the implementation of different types of E-mail spam filters to prevent Spam either by blocking it at the server level or the client level (Pelletier, L et. al 2004; Haskins, R and Nielsen, D., 2005).

E-mail filtering systems can be further classified based on the parts of the E-mail messages that they use for spam detection. Origin-based or Header-based or address-based filters typically use network related information for classification of spam, while content-based filters examine the actual contents of E-mail messages. Sender authentication systems use network information along with changes to the E-mail sending system in order to identify spam messages (Haskins, R and Nielsen, D., 2005).

2.2. Literature Survey and Analysis of Problem Domain

The dramatic increase of the E-mail spam over the previous years has created a real interest among researchers to investigate methods to combat it. This section will provide the review of literature and the analysis of the research problems with the different types of methods/technologies evolved for controlling spam E-mails.

2.2.1 List-Based Filters

List-based spam filters attempt to classify E-mail senders based on their IP addresses or domain names. The various types of list based/IP based filters are Blacklists, Realtime Blackhole list and Whitelists. Based on the previous history of IP addresses or based on reputation of an IP address or domain name these lists were updated.

2.2.1.1. Blacklists

Blacklists are identity based filters also known as domain name system blacklists (DNSBL). The aim is to stop E-mail spam by filtering mails from suspected mail servers or domains based on their previous spam history. These maintain a list of suspicious IP addresses and try to block them before reaching inboxes. Cook et al. (2006), Ramachandran et al. (2007) suggest that with the help of blacklists spam can be blocked at the SMTP connection phase. In order to prevent spam, ISPs and e-mail administrators can subscribe to these blacklist databases. Blacklists are centrally maintained databases with a list of IP addresses/domain names of known or suspected spammers. There are many spam blacklists that are publicly available, created and maintained by popular companies apart from some independently maintained blacklists. Some of these are the SPEWS list (SPEW, 2005), the Spamhaus (Spamhus, 2005) SBL and XBL lists, Barracuda Reputation Block List: BRBL (BRBL, 2015), Trend Micro’s RBL+ (TM, 2005), MAPS (Mail Abuse Prevention System) and the SpamCop Blocking List (SCBL) lists. A blacklist maintains a database of known open relays too.

The Spamhaus Project keeps a track of spammers. It works with ISPs and law agencies to add spammers’ IPs to the blacklist. Advantages of blacklists are that these are easy to implement, require low CPU overhead and also block the spam at the SMTP connection phase itself. Major limitation of blacklist filter is its continual maintenance. Spammers routinely change their identities along with the exponentially growing new websites. With the exponentially growing new websites, spammers change their identities, even more often these days. Updating the list is, thus, a tedious job. While comparing with other spam technologies, blacklists are not that adept at recognizing the spammer tricks. A blocked IP addresses might lead to false positives if blocked for much longer than they need to be and thus it remains a tricky matter to handle. With the increase of volume of spam, the number of DNS lookups increases. Mail servers that use more than one blacklist are particularly affected by this. Sometimes, the policies for adding/deleting were not specified clearly by the blacklist providers (SPEWS, 2005), in turn, forcing network administrators to trust the judgment of others. Sanz et al. (2008) stated that substantial DNS traffic might be generated by this approach as most of the blacklists are usually queried via the Domain Name System (DNS). Dudley et al. (2008) state that the effectiveness of blacklists depends on the people who manage the lists and on their timeliness. This simply indicates that such an approach requires timely manual intervention, something which does not always scale as well as an automated mechanism of the appropriate kind could.

2.2.1.2. Whitelists

Whitelists are the “trusted” addresses that will immediately allow mails from those addresses, classifying as legitimate all E-mail (Pfleeger, SL and Bloom, G., 2005; Orbach, Julian J. 2012). Whitelist is either a list of trusted E-mail addresses created by the user or a list of domains from legitimate sources based on the previous history. (Erickson, David, et al. 2008). Compared to blacklists, Whitelists are small in size and number of contact addreses is less in count, so they are easy to maintain. If an E-mail is whitelisted, it bypasses the spam filters, effectively reducing the load on those filters. Limitation of whitelist is that if any new contact sends a legitimate mail, it would not be passed through the filter and instead sent to the low priority inbox, for user confirmation. As white lists are already being authenticated, there is a possibility of sending spam easily, reaching inboxes without any filtration process. It’s difficult for mailer programs to learn about contacts of the individual users. *In addition, if a legitimate contact changes the originally marked E-mail identity, such a mechanism might end up blocking the mail from this user*. In a nutshell, it was not considered as a filter that is good enough to combat quite a few kinds of illegitimate attempts as well as handle select types of legitimate changes *(like the one mentioned above)*. Wieneke et. al. (2015) presented a probability based whitelist, which is the system and also a method for maintaining the whitelist. This method includes obtaining the sender message data based on an email sent; extracting recipient information from message data and updating the whitelist using the recipient information.

2.2.1.3. Greylisting

Greylisting is an identity based filtering method used to block spam at the mail-server level. Greylisting (2006) temporarily rejects an E-mail which has come from unknown sender, if the sender attempts to send the message again within a given time period, then the message will be delivered and that address will be added to the list, for passing unhindered, during future communication. The advantage of this listing is that it blocks spam messages significantly, as spammers cannot ever send the same messages again. This results in lower burden on E-mail traffic and processor load on the mail-server. No special configuration or any extra resources are required at the end users' level.

2.2.1.4. Challenge/Response Systems

Challenge/response systems are extended version of whitelists, wherein an automated message is sent to the sender's address, which were not whitelisted, as a challenge and a legitimate response is awaited. If a legitimate or valid response arrives, the test is considered to be cleared and therefore the e-mail passes the filter. The aim, here, is just to block machine-generated spam. If the sender is not a machine, the person would click onto the 'reply' link and send ID number of the response message, leading to the E-mail's passing the challenge/response mechanism (Pfleeger, SL and Bloom, G., 2005). This filter is also helpful in blocking manually sent spam, as the spammers, instead of attempting the challenge, search for other additional contact addresses to send spam.

Problem also, often, arises with the subscribed E-mails, as in some cases, subscribed newsletters or commercial E-mails do not have an option to respond manually to the involved verification process required by this filter, leading to such communications' finally ending up in getting discarded. Cook et al. (2006) observed that one of the disadvantages of challenge response filters is that if both sending and receiving mail servers implement them, and then deadlock occurs. As a result, both the servers have to wait for one-another to respond to the challenge involved. Harris (2003) proposed a solution to stop reaching the deadlock state which involved a combination of challenge response with grey-listing filter. In this case, the challenge is sent only to the suspected email server with the possible spam e-mail. Yet another problem in such case relates to their being time-consuming and in addition, at times, the process of responding to the mail or going through the verification process could be irritable. Salmi, T. (2005). Microsoft's Penny Black Project (2003) is based on the challenge response system.

Barracuda Networks (2005) observed that if both sender and receiver never correspond beforehand, the issue of deadlock arises. The recipient sends a challenge to the sender, which, if caught by sender's challenge/response system, could end up without any appropriate response from the sender to the challenging receiver. This situation can be solved if the sender adds the recipient's contact details to its whitelist immediately after sending mail.

2.2.2. Content Filters

Content based spam filters analyze the data that is present in the body or the content of the E-mail to classify the E-mail as either spam or legitimate. In this case, the header section is totally ignored, unlike in the case of whitelists/blacklists. In order to identify spam, many of these filters try to go through the text thoroughly (Garcia, F D., 2004).

2.2.2.1. Bayesian Filters

Bayesian filters are content-based filters used for binary classification. By analyzing the words of the E-mail content, Bayesian filters classify the E-mails, by calculating the probability. Both the words from spam and non-spam messages are taken into consideration for calculating probability. Bayesian filters need to be trained with both spam and legitimate E-mails for achieving good accuracy. The advantage of this technique is that it is self-adaptive with a proper training. In addition, they also achieve high accuracy with a low percentage of false positives (Graham, P., 2003). Low false positives adds the credit to Bayesian filters since the users don't want to misclassify legitimate mail as spam and are, at times, willing to accept spam in their inbox instead. Bayesian filters, however, need to be updated regularly.

Content filtering is considered to be one of the best approaches for identifying spam by many researchers. If a mail containing simply images, or image links are sent, then it's a challenge to content based filter (Androutsopoulos et. al. 2000). Bayesian SPAM filters fails here because these are not capable of analyzing images. Even if blacklists maintain a very large list of blocked IP addresses, they have low CPU overheads because the procedure involves just querying a list, whereas Bayesian filters are generally more CPU-intensive since calculating Bayesian probabilities requires significantly more processing power.

2.2.2.2. Rule-Based Filters

A heuristic filter or Rule-based filters, search the E-mail message for the presence of certain patterns that indicate spam such as HTML, specific words or phrases, special characters, etc. This filter applies a set of rules which have a value associated with it. If the total value exceeds the limit threshold set by the system administrator, it is marked as spam. Rule-based filters vary from simple filters to complex filters based on number of set of rules. Heuristics filters are usually accurate and perform relatively fast compared to other approaches. They are easy to install and configure for an individual user too. Rule-based filters were the common spam filters (Garcia, F D., 2004) before the Bayesian filter became popular.

Spammers try to get the rules from these filters and accordingly alter their messages so that they can bypass the filter and reach the targets. If the rules are secretly maintained and constantly updating the rules can reduce these spammers counter attacks. They had also proved successful in identifying unknown viruses. These filters takes many other attributes into consideration like date and time, number of attachments, MIME-types, etc., other than the content of E-mail (Vipul's Razor, 2007). Appropriate rules, effective scoring and threshold mechanism is the key success to the success of rule based filters.

2.2.3. Collaborative Filters

The collaborative filtering is, by design, a form of recommendation system. Collaborative filtering explores techniques with similar interests for group of people and makes recommendations based on this. Collaborative filtering algorithms require users' active participation and algorithms for matching similar interests. They include memory-based or user-based collaborative filters.

2.2.3.1. Distributed Checksum Clearinghouse

Distributed Checksum Clearinghouse (DCC) from Rhyolite Software (2005) is a client/server system which is basically a bulk emailer detector. A DCC-enabled E-mail server reports the received message it to designated DCC servers. These DCC servers, after receiving the mail from E-mail server(s), start counting the number of such similar E-mail messages and send the count back. If the count is high, the mechanism involved asks the server to mark the traffic as spam / bulk E-mail. DCC identifies bulk E-mail by use of a specific variant of hashing technique. The fuzzy hashing logic, involved, compares different E-mail contents and identifies similar messages. The hashes of slight variances are then collated at the Clearinghouse and counted. The E-mails with highest count are likely to be spam.

2.2.3.2. Fingerprint Analysis

Theoretically, once a spam message is received, it is possible for the recipient to create a fingerprint of this message and share it with his / her contacts. In this way, anyone in a given contact list who receives this message fingerprint before receiving original message of such type can identify and filter out such messages. Allman (2003) presented a spam filter based on such fingerprints. He propagated analyzed fingerprint lists to mail servers and any declared a message received to be spam based on matching fingerprint from the lists.

Fingerprint analysis, however, is considered to work best for blocking image spam. *(Incidentally, Image spam is out of scope of the present work, as indicated in the Scope section itself and as such, would not be examined in greater detail here.)* Interestingly, Spam Fingerprint method detects spam without looking at the E-mail content. It can also block select types of malicious software (malware), viruses, etc. by matching the similar fingerprints. Spammers apply tricks to counterattack such fingerprints with slight variations. Damiani et al. (2004) proposed a digest-based filter which is an example of exact or approximate digest. Attenberg et al. (2009) proposed a spam filter based on select hashing tricks.

2.2.3.3. Razor

Razor (2007) is content E-mail filter with a distributed hash database. The *Razor System* maintains a '*Razor Server'* on the Internet and once a hash is generated (based on the E-mail content), its checked against the available distributed database resident in these servers. As for continual update process, in this scheme, users report the hash of spam, the servers keep on updating based on these hashes. Hash values get changed even if a single character in the input varies and consequently, as expected, do not match with hash string generated by the respective users. Taking all these problems into consideration, Razor, uses *Nilsimsa (2002) Signatures*. One advantage of this algorithm is that instead of considering the entire data, it can consider only a few lines from an E-mail, and yet generate the hash for matching with entries in the distributed hash database in the given Razor Server. It also involves the creation of a statistical model with duly labelled false positive status markers for ensuring the correctness of its diagnosis.

2.2.3.4. Lexical Analysis

Lexical analysis works by examining the context of all of the words and phrases in a given E-mail message. Not every suspicious word/phrase in an E-mail means a spam. In order to identify spam, each word/phrase is given a weight, primarily based on the involved context. After a whole message is analyzed, combined weights are calculated and compared with the threshold limit. If the score exceeds the limit, it is considered as spam. Lexical analysis can also be applied to spammer counter attacks of different variations of words and phrases. Lexical analysis algorithms remain same for any language and depends on the dictionary with variation in associated rules.

2.2.4. Machine Learning Techniques

A great deal of work has gone into the use of Machine Learning techniques to fight against spam. Learning here refers to training the machine with sample E-mails, both spam and legitimate. There are two types of machine learning approaches, supervised and unsupervised. In supervised approach, classes are predefined and training the machines is carried out with these predefined sample data. In unsupervised learning, no classes are defined in advance and instead depends upon letting the machine to learn on its own.

Spam E-mail classification often involves multiple steps. First step is the corpus selection. There are many options available, which were open for public. The data will be available in both raw format and pre-processed. Next step is cleaning the data, using different pre-processing techniques, like, removing stop words, stemming etc. feature selection plays a key role in the E-mail classification and efficiency of the classifiers depends on selecting appropriate features. Feature reduction, reducing the number of features, also plays a dominant role. After finalizing the features, the next step is training the data. Final step leads to e-mail spam classification where the tested / examined data are classified as spam E-mail or legitimate E-mail.

2.2.4.1. Bayesian Classifier

The Bayesian classification (Androutsopoulos., 2000; Metsis, Vangelis, (2006); Lowd, Daniel, and Christopher Meek 2005) is based on both supervised learning as well as a statistical method. It's a probabilistic model. This Classification is named after Thomas, who proposed the Bayes Theorem. Bayesian filtering is based on the probability of how often a word/phrase that likely to occur in the future can be learnt from the previous occurrences of that word/phrase. The same technique that is also used to classify E-mail spam. Here, a probability value is assigned to each word or token; if some words occur often in spam, but not in legitimate Emails, then this incoming e-mail is considered as spam. This is done by analyzing the user's legitimate messages and spam messages. All the words, of both the spam and legitimate kinds, are analyzed to generate the probability for classification. The Bayesian method takes the whole message into account to classify an E-mail as spam E-mail or legitimate E-mail. It constantly self-adapts and is considered to be robust to noise in input data.

2.2.4.2. K-Nearest Neighbor Classifier

The *k-Nearest Neighbor (kNN)* (Cunningham, Padraig, and Sarah Jane Delany (2007); Trudgian, Dave C. (2004) classifier is considered as an instance-based classifier. There is, as such, no training phase in the kNN mechanism and instead training documents are used for comparison. When a new document needs to be classified, the 'k' number of similar documents is found. If many of such similar documents fall under a class, then the new document is assigned that class. For classifying E-mails using kNN, the training data have to be converted into document vector representation. While classifying a new E-mail, similarity between its document vector and each document vector in the training set has to be computed. Finally, the class is chosen based on the class of k nearest neighbors.

2.2.4.3. Neural Networks based Classifiers

Neural Networks (Clark, James. et. al, 2003; Stuart, Ian, et. al. 2004) based approaches are, often considered to belong to the basket of Artificial Intelligence techniques that involve computational mechanisms built around a collection of interconnected artificial neurons. The basic neural network structure has input and output layers interconnected by a hidden layer in-between. Input layer represents the source data and every input node is connected to any of the output nodes. In E-mail spam classification case, every word/phrase in an E-mail represents an input node. Consequently, the number of features in a mail represents those many numbers of neurons/nodes in the corresponding input layer. Since all the content of the E-mail is taken into consideration for prediction, Neural Networks have proved to be the fairly good to very good predictors. Accuracy of this class of the algorithm increases with the large set of features and specifically, in case of E-mail classification, the larger the content of the E-mail fed as an input, the better has been the accuracy of classification. The accuracy of this category of techniques also depends on how well the network is trained. Neural Networks can achieve a substantially high accuracy rate, provided a sufficiently large body of messages is supplied to the Neural Network during training.

2.2.4.4. Support Vector Machines (SVMs)

Support Vector Machines (SVMs) introduced by Vapnik Cortes, Corinna, and Vladimir Vapnik. (1995) and refined by Drucker et. al (1999), are based on the concept of *statistical learning theory*. SVMs map the input data into a *high dimensional feature space*. They define decision boundaries on the basis of the concept of *decision planes*. A *hyperplane* separates a set of objects having different classes. An optimal hyper plane, here, refers to the one with large margin for separating the two classes for solving the chosen optimization problem. If classes are not linearly separable, SVMs make use of the so-called *kernel functions* to separate them. They have been proved to be statistically robust for general text classification. SVMs have very good generalization performance as well.

2.2.5. Honeypot-based E-mail Address Creation Mechanism as a Tool to Attract Spam E-mails for Classification / Filtering Support

Honeypot mechanisms allow use of specifically crafted *Spam-Mail Collection Honeypots* (*SMCH)* for use as a fairly practical anti-spam technique. Honeypots, in such cases, are designed to look attractive to spammers. Their main goal is to divert the attention of spammers from real E-mail addresses, and instead draw their attention towards the fake E-mail addresses created by honeypots. Therefore, these are designed in such a way that they attract the *E-mail Harvesters* easily. These fake addresses mix up with the original spammer's list database and dilute the spammer's effective address-lists. Honeypot, in a way, may also be seen as a fingerprinting system for collecting known spam E-mail messages. *Oudot Honeypots* (2003) are machines that are created just to collect spam.

E-mail Harvesters are the tools which aim to collect targeted Users' E-mail addresses which spammers then use for actually sending the spam E-mails. An E-mail harvester may, thus, be seen as to be a mechanism for collecting E-mail addresses by using built-in or invoked *spambots* or *web-spiders*. A simple and fairly common solution used by web sites to avoid a common class of *spambots* is to avoid using @ and instead using AT or an image depicting either '@' or 'AT' in publishing the E-mail address on a public page. Project Honey Pot (2004) was an interesting experimental initiative by Unspam Technologies. All the websites participating in this project, installed their software and some honeypot addresses were consequently generated, by design, on the participating websites. Once the E-mail harvesters collected the addresses from these websites and started spamming, their IPs were tracked. Andreolini et al. (2005) observed that through honeypots one can identify many new species of emerging spam. BrightMail solution is based on a similar honeypot method which filters email addresses before adding them to the identified / designated Post Office Protocol (POP) mailbox.

2.2.6 Spam Zombie Detection Systems: Spam-Filters that Prevent Spam E-mails by Identifying Compromised Machines acting as Spam Zombies

Spam Zombies or simply zombies are the compromised machines that are used for sending spam messages. As per Hayati & Potdar (2008), spammers can send their emails by spambots or zombie machines. Lieven et al. (2007) point out that several zombie machines exploit the SMTP protocol and often use non-standard configurations. As a consequence, receiving side suffered. Spam Zombies can be, however, detected by receiving side SMTP server with due care. Duan et al. (2012) developed an effective *Spam Zombie Detection System* named *SPOT* based on a powerful statistical tool called *Sequential Probability Ratio Test*. This tool was found to be helpful for system administrators who wanted to automatically detect the compromised machines (i.e. Spam Zombies) on their networks.

2.2.7. IP-based Authentication: A Way to Control Select E-mail Spam by IP-based Authentication and Filtering

The SMTP protocol allows everyone to send an E-mail using the *Message Transfer Agent (MTA).* Many MTAs allow open E-mails relay mainly because of mirrors during configuration while the user authentication is frequently used to identify users. To relay spam, open relays are manipulated by exploiting well-known loopholes in the specification as well as select list-based defense systems that actually intend to minimize such abuse. Blacklists and Whitelists, sometimes used as associated tools, are the refused lists and allowed addresses and domains respectively. The effectiveness of blacklists and whitelists, in practice, is, however, limited by the spammers because they alter their internet domains from time to time and work around the MTA’s IP address. Also, detection of spam E-mail based on the history of the sender is not effective for the detection of spam.

2.2.8. Cryptographic Authentication

With the help of cryptographic authentication, every E-mail can be improved, in terms of its content-integrity and authenticity by use of digital signature. When any digitally signed E-mail is received by the user, the receiving MTA inquires the sending MTA for the public key to validate both E-mail and the sender.

2.2.9. Cost Based Models

There exist proposals to alter MTAs in a way such that the “physical world” method of paying for sending messages is used as a reference. Such proposals vary from using monetary based “*micropayments*” for digital stamps (Capek et al. 2003; Rivest R. and A.

Shamir, 2001) to "*virtual currencies*" used for requesting resources (Abadi et. al. 2003; Turner D. A. and D. M. Havey, 2004) and "*proofofwork*" computational puzzles (Back., 2002; Dwork et.al. 2003). The *Lightweight Currency Protocol (LCP)* (Schneier B. and N Ferguson, 2004) a virtual currency model, and Adam Back's *Hashcash* (Back 2002), a computational puzzle model, are both explored below as examples of related interest.

2.2.9.1. Lightweight Currency Protocol (LCP)

The Lightweight Currency Protocol LCP (Turner D. A. and D. M. Havey, 2004) is a simple, secure protocol based on *virtual tokens* which represent a type of money. Because of the high transaction costs of real-world currency, it doesn't go well with micro-payment-based systems. The end-users started avoiding the decision requirements for micro-payments. To solve this problem, the Lightweight Currency Protocol (LCP) (Poutanen, Tomi, 1998) was designed. They have low-transaction cost and low-risk compared to real-world currency.

The basic concept, here, is that each server adds a certain number of digitally signed tokens, for accessing the resources. These servers should have proper authentication for these tokens. These protocols are ideal for cost-based E-mail, as it is easy for E-mail service providers to issue their own tokens for delivery payments. The main goal of this approach is to avoid bulk spam, since this approach is. By design, not profitable to them and systems deploying such mechanisms have shown that this provision is not attractive to spammers. An undesirable impact of this scheme is however also visible to legitimate users as they too directly or indirectly have to share / bear the increased costs of mail servers employing such a scheme as overhead to them. One benefit of LCP is that only virtual currency comes into the picture. Thus, service providers generate currency through the sale of a service in one market, and those currencies were used for services in another market.

2.2.9.2. Adam Back's Hashcash

Hashcash proposed by Adam Back (2002) is a proof-of-work algorithm. This is a counter attack to handle select types of denial-of-service attacks as well as E-mail spam. The basic concept here is it is significantly expensive for client to compute a hashcash tag/stamp, but server side stamp validation is extremely cheap, almost negligible and therefore the involved economics would deter the spammers of certain types from sending large number of individual mails as their spamming strategy as it would incur high computation overheads to them. In this process, cycle based payment method (CPU) is used in place of real currency for resource utilization. The main use of the hashcash stamp is, that textual encoding in the header acts as a whitelist, and that E-mail is sent directly to the inbox without undergoing the other processes.

An E-mail sender, in this case, has to spend a pretty amount of time in calculating the stamp. The textual encoding of this stamp is added to the header of the E-mail before sending. Unfortunately, while such techniques do deter customized spam E-mail senders likely to create the phishing kind of frauds, it does not really prove effective when it comes to preventing a single mail being sent to a carefully crafted mailing list. Since sending a single E-mail is a time taking process, spammers attempt sending bulk spam.

2.2.9.3. Micropayments

A micropayment is an e-commerce transaction with small amounts of money. This is applicable to many things like downloading any application, file, or sending an E-mail, or for any Web-based content. In a micropayment model, the recipient once receives an E-mail from a sender it charges a small amount of money through the online banking system. If the mail received is a spam, the receiver does not return the money, else, it does. First generation micropayments weren’t successful. The limitations were, small amount transactions were impractical. Another problem was to handle transactions worldwide, as well as maintaining and keeping track of these records (Rivest R and A. Shamir, 2001). In 2010, the second generation micropayment emerged which have been fairly successfully used in certain settings, although they have not yet found favour with the largest providers of E-mail services.

2.2.9.4. E-postage Stamps: As an Example of Micropayment Scheme of Specific Kind

Electronic Postage (*E-postage)* (Abadi et. al. 2003) too involves a small amount of cash payment for each E-mail sent. Intention, here, is that only the legitimate or willing senders would want to send the E-mail to the intended receiver by bypassing the standard filters, which would otherwise delay or even block them. These are also called *certified e-mails* as E-mail to be sent gets certified once payment is done. America Online and yahoo have implemented these certified E-mails, on optional basis, on the payment of 0.25 cent per E-mail. Even advertisers can post these certified E-mails as they bypass the spam filters. E-postage is, thus, an example of micropayments, but the one with a twist of exclusion from filtering upon payment.

2.2.10. Meng Wong's Sender Policy Framework

The *Sender Policy Framework (SPF)* is an example of the use of lightweight authentication techniques. That aimed at preventing E-mail Spoofing and Spamming. (Wong, Meng, and Wayne Schlitt, 2006) This technique compare the domain of the sent message against the IP address of the connecting machine. If they get matched, the E-mail is authenticated as a legitimate mail. If they do not match, an E-mail is considered to be of spoofed kind and can hence be rejected. SPF's key goal was to minimize the spammers sending spam E-mail in the disguise of the legitimate E-mail. The SPF had proved its accuracy in validating E-mails from their domain origins. By this act, SPFs could prevent select types of Phishing (Phishing Activity Trends Report, 2004) which is one of the categories of E-mail spam. However, it was also reported that it was unable to prevent relaying of spam, in select cases, by the zombie systems that exploited the sender MTAs' SPF settings.

2.3. Microsoft's Anti-Spam Work and Strategies: As a Sample Industry Approach

In the mid of 90's Spam volume had increased significantly and had become a major issue for computer users and business organizations alike. Gradually they started adapting to anti-spam technologies for protecting their brand-value and reduce burden on their infrastructure.

Microsoft came up with different strategies to help prevent and detect these annoying mails with the help of collaborations with different companies, service providers, academia, etc. Microsoft's E-mail safety roadmap solutions are available in all Microsoft e-mail clients, servers, MSN Hotmail, Microsoft Office Outlook Microsoft Exchange Server etc. Some of the relevant work and mechanisms deployed have been briefly analyzed below.

2.3.1 Microsoft's Sender ID

Microsoft's Sender ID was based the concept of on Meng Wong's Sender Policy Framework (2006) which was discussed in one of the earlier sub-sections. It is an e-mail authentication technology protocol. The SPF implementation was embedded into Mail User Agents (MUAs). It addresses the spam problem efficiently along with the other E-mail related problems like phishing and spoofing by verifying domain addresses of sent E-mails. Here origin of the E-mail is verified by matching IP address against the domain name of the sender. Finally discards if the mail is falsely claimed. Figure 2.1 outlines the steps involved in the Sender ID verification process. Microsoft's Sender ID is now adopted by many domains globally.

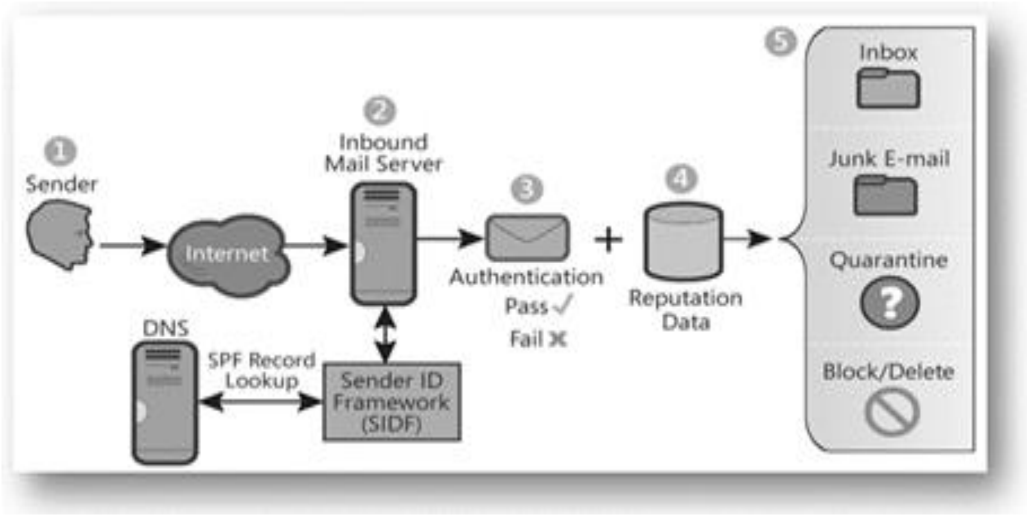


Figure 2.1 Sender ID Framework Process Steps

2.3.2 Microsoft SmartScreen

Microsoft SmartScreen is anti-spam filter based on machine-learning technology, which is patented by Microsoft Research. This intelligent spam-filtering solution has been integrated with all of the Microsoft e-mail platforms and offerings. This technology learns to detect phishing URLs embedded in e-mails. It learns from different legitimate and spam E-mails and, consequently, can distinguish between them. Learning, here, is done from the user input that is received from the volunteers of the *Feedback Loop Program*, which form a sizeable sub-set of the millions of MSN Hotmail and Windows Live Mail users.

2.3.3. Microsoft Office Outlook's Junk E-Mail Filter

The Outlook Junk E-Mail Filter uses *Microsoft SmartScreen* technology mentioned in the previous section. One can get enhanced spam protection by customizing the filter with personalized antispam rules. These rules could be a list of safe senders, safe recipients, a Block list of senders, etc. Figure 2.2 gives an overview of the referred junk E-mail filter.

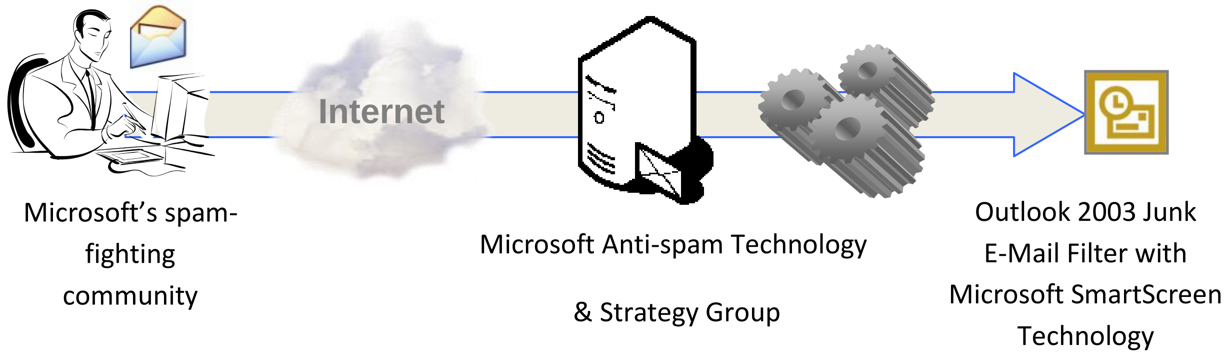

Internet
Microsoft's spam-fighting community
Microsoft Anti-spam Technology & Strategy Group
Outlook 2003 Junk E-Mail Filter with Microsoft SmartScreen Technology

Figure 2.2. Overview of the Microsoft Office Outlook Junk E-Mail Filter

Once an e-mail message is received by Outlook, it tries to examine the E-mail based on its elements, and determines if any content-scan is required. If it is from any of the safe lists, it bypasses the Junk E-Mail filter. If the address is not from any of the safe lists it directs it for scanning to the Junk E-Mail filter. Finally, an E-mail, identified as spam is sent to spam folder on the basis of Outlook Junk E-mail Options.

2.3.4. Windows Live Mail/MSN Hotmail

Windows Live Mail or MSN Hotmail also works with *Microsoft SmartScreen* Technology in attacking spam but is handled bit differently. In this case, once a mail is received from a external user, SmartScreen goes through the textual content of the e-mail and assigns a rating. The rating is given on the basis of the probability that the message is spam or not. This rating is stored as a special property called the *Spam Confidence Level (SCL)* in the E-mail. These mails are now handled on the basis of the threshold of SCL. If the SCL rating exceeds the threshold value, its discarded claiming as spam, otherwise sent to the inbox.

2.4. IBM’s Research related to Spam E-mails

IBM Research’s *SpamGuru* (Richard Segal, 2004) is an enterprise-class anti-spam filter. The strategy of SpamGuru is to attack spam, and reduce spam related problems. Different filtering technologies are involved in SpamGuru and they are intelligently combined to get a score called *Spamminess*. It is a score given to an incoming e-mail by different technologies. Basic idea, here, is to increase effectiveness of filter by using multiple algorithms.

Figure 2.3 gives a brief idea of *SpamGuru Anti-Spam Architecture*, and multiple algorithms used by it for filtering spam E-mails. SpamGuru is now a testbed for many IBM research projects like *JClassifier* (Richard Segal, 2004) which is a *Bayesian text classifier* based on Paul Graham's ((Abadi et. al. 2003) work. Another tool involved is the *Chung-Kwei’s (2004) Spam Detection Algorithm* that was based on pattern matching (developed by IBM's Bioinformatics Group).

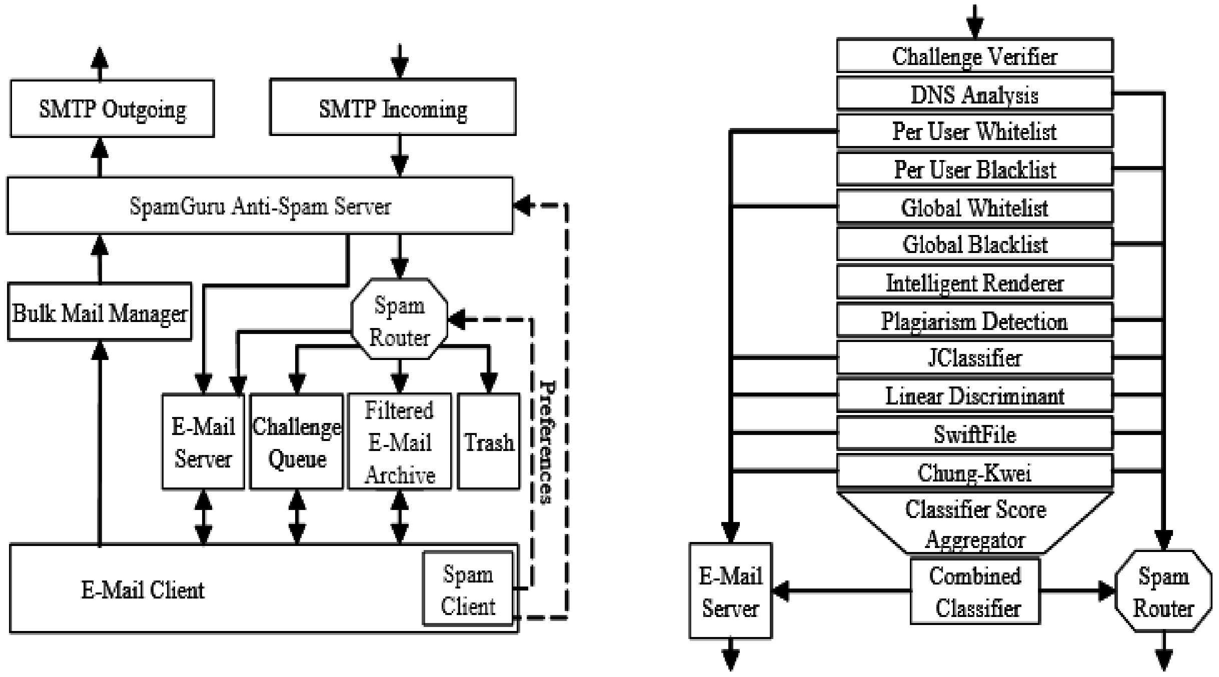


Figure 2.3 SpamGuru Anti-Spam Architecture, Server Filtering Pipeline

2.5. Research by CISCO Systems on Spam E-mail Prevention

Jim Fenton and Mike Thomas (2005) from Cisco Systems proposed *signature-based e-mail authentication system* known as *Identified Internet Mail (IIM)*. The main intention, in this case, was to identify unwanted E-mails. This system attaches significance to domain addresses of originating E-mails. It helps, preventing spammers from forging return addresses. IIM verifies the authentication of the sender. A mechanism to verify the integrity of the message using digital signatures and public key cryptography was provided by IIM and also a public key authentication mechanism was used to validate the sender's address. IIM-Signature and IIM-Verification are the two headers added to the message. IIM E-mail authentication was a great success. *IIM* was, later, merged with the *DomainKeys (DK)* scheme to form the *Domain Keys Identified Mail (DKIM)* standard.

2.6. YAHOO's Research on Spam E-mail Prevention

DomainKeys scheme proposed by Mark Delany (2007) from Yahoo! is a domain-level authentication framework. The *DomainKeys'* main objective was to verify integrity of the received message and authenticity of the sender message with public key cryptography, digital signature and DNS. Yahoo Domain Key (YDK) solution (Delany, Mark, 2006; 2007) is a cryptographic approach to validate an E-mail by applying a digital signature (Goldwasser, Shafi, 1988). Compared to some other authentication systems, it was more successful in many complex operational requirements like forwarded E-mail, distributed sending systems and authorized third-party sending.

2.7. Post Arrival Adaptive Strategy Based Approaches

It is not only the spammers' own MTAs, who send spam but also by transmitting by spam-zombies, as discussed earlier. Also, in some of the recent spam control models, the outbound spam-control is not implemented regularly for reducing spam control cost for the people. This, unfortunately, leads to situations wherein outbound spam spreads with minimal restrictions on the Internet until the processes of receiving MTAs' ingress points during both relay and receptions. Hence, controlling of spam is, in such cases, reduced to inbound E-mails and spam-control becomes the receiver's responsibility. An effective solution, here, could be that to detect spam E-mails during the transport of E-mails (by checking the sender's history and by blacklisting) with the help of relay or proxy MTAs.

Unfortunately, there have not been another way of receiving or relaying MTAs, other than to all the arriving mail for lining up by performing the spam controlling system with the equal priorities. The delaying of the transfer of E-mail, thus, depends on the queuing delay. In the traffic of E-mails that consist mostly of spam, there is a delay for the non spam E-mails too because of presence of spam in that queue. Also, many non spam E-mails are routinely lost or delayed beyond acceptable periods during this process because of heavy traffic caused by spam E-mails. During the mass-nailed worm outbreak as well as select kinds of *Denial of Service (DoS)* attacks to the MTA operations are slowed down or even disabled.

2.8. Brief History of Anti-Spam Research (non-chronological, organized based on techniques' evolution and inter-relationship)

Primitive Language Analysis (Rule Based Filtering) was one of the first solutions of the spam filtering history. In this technique, the filter simply scans the subject of the incoming e-mails and looks for the specific phrases. Although this method seems very straightforward, filtering on even a single word had a potential success rate of around 80%. Past 1994, some spam prevention tools began to emerge in response to the spammers who started to automate the process of sending spam e-mails. The very first spam prevention tools or filters used a simple approach to language analysis by simply scanning e-mails for some suspicious words or for phrases such as "click here to buy" and "free of charge".

In later years, blacklisting, white listing Orbach, Julian J, (2012); Erickson, David, (2008), and throttling methods were implemented at the Internet Service Provider (ISP) level. However, these methods suffered some maintenance problems. Furthermore, whitelisting approach was open to forgeries. Some more complex approaches were also proposed against spam problem. Many machine learning based approaches involved text classification since the task of spam E-mail classification can be efficiently handled through different algorithms which include Naïve Bayes, Support Vector Machines, Neural Networks, K-Nearest Neighbour, Rough Sets and the Artificial Immune System. Naïve Bayes Network algorithms were used frequently and they have shown a considerable success in filtering English spam e-mails Goldwasser, Shafi, (1988). Numerous variations of the Naïve Bayes classifier have been presented for spam filtering by Sahami et. al.(1998), Graham (2003), Schneider (2003), Lewis et. al. (1994), McCallum and Nigam (1998), Androutsopoulos et al (2000). Such filters are, as described earlier, self-adapting and can stop new spam once trained properly, and the false positive percentage too is low. Sculley, (2006), Cook et al. (2006) Based on the probability of key words in the training set, Bayesian filters calculate the final probability that the incoming E-mail, and decides the mail is spam or not. Several open-source anti-spam E-mail filters have been developed based on Naïve Bayesian algorithms. Pantel and Lin

(1998), developed SpamCop, T Meyer & Whateley (2004), developed SpamBayes which filter spam e-mails using Naïve Bayesian methods. Bogofilter, another open-source filter based on the same Naïve Bayesian to classify email spam was developed by Raymond (2005). Ming et al. (2007) developed a recognition filtering, where the method identifies the spam E-mail according to the behavior of sent mail. Balakumar and Vaidehi (2008) used ontology for Statistical based filtering: understanding the content of the E-mail and Bayesian approach for making the classification.

Rule-based systems spam filters, classified spam E-mail from legitimate on the basis of user-specified rules. RIPPER a rule-based spam filter developed by Cohen (1995). Ahmed, et.al. (2004) Freschi, et.al, (2006) developed spam filters on the occurrence of specific keywords both from the content and E-mail header. Zhang and Yao (2004) and Khorsi (2007) presented a Maximum Entropy technique to effect junk mail filtering. Boosting Tree came across as an algorithm that was based on the idea of combination of many weak hypotheses. Carreras and Marquez (2001) involved finding the appropriate probability distribution by using Boosting trees and compared its performance with other spam filters, such as Naïve Bayesian and Decision Trees. Jin et al. (2003), Ali & Yang (2007). He & Thiesson (2007) applied Boosting Trees for spam classification.

In yet another set of efforts, O'Brien & Vogel (2003) developed a Chi-squared filters based on chi-based authorship identification technique for the spam classification and compared the results with a Bayesian classifier. Sakkis et al. (2003) Zhang, Zhu & Yao (2004) adopted Memory-based learning filters for spam classification. Chhabra et al (2005) presented a spam classifier based on a Markov Random Field (MRF) model. This approach was an inductive learning paradigm with no parameters involved. It stored the training instances in a memory and then made predictions of new instances based on this memory data.

Yoshida et al. (2004) proposed method depended on the analysis of the document space density information to detect spam. Clustering method has also been applied for spam classification to detect email spam by clustering emails into clusters (groups). The classifier trains each group, and finally tries to detect spam by identifying its cluster Saeedian & Beigy

(2009). Clustering techniques were classified into two variations and been applied to spam classification are K-NN and density-based clustering.

Vinther (2002) describes a filter for junk mail that uses a non linear feed forward neural network trained using the back propagation algorithm. Clark et al. (2003) construct a back propagation trained artificial neural network (ANN) classifier named LINGER. In this approach feature selection was employed to select the key features, such as Information Gain (IG) and Variance (V). An alternative approach using a neural network (NN) classifier on a corpus of e-mail messages from one user was used to identify spammer tricks (Stuart, Ian et. al. 2004). Chuan et al. (2005) proposed an LVQ-based neural network approach and compared Naive Bayes to two neural network approaches and found that the neural network approaches both performed better than Naive Bayes. Goweder, et.al (2008) applied Genetic algorithms along with neural networks, in the spam classification of emails. Nosseiret. Al. (2013) presented an intelligent characters-word-Based Spam Filter Detection using Multi-Neural classifier. Shrivastava.J, Hima Bindhu. M (2014) presented, a genetic algorithm based method for spam email filtering is discussed with its advantages and disadvantages. Puniskis (2015), in his work, applied the Neural Network based approach to the classification of spam.

Decision Tree has been around as a machine learning based classification technique, where the observations are represented by the internal nodes of the tree and leaf nodes represent decisions. Sheu (2009) and Carreras & Marquez (2001), applied the Decision Tree technique to classify E-mail spam. There are several variations of Decision Tree methods were applied in spam classification. For instance, the C4.5 tree considers attribute values from the training dataset, to create a tree.

Support Vector Machine (SVM) had been yet another machine learning algorithm designed to be robust for classification especially binary classification proposed by Vapnik et. al. (1995). SVMS have been gaining popularity due to many attractive features and promising performance in select situations. Drucker et al.(1999). Cristianini & Shawe-Taylor (2000) applied SVMs to the real-world applications such as text categorization, hand-written character recognition, Spam classification, etc. Kunlun and Houkuan (2002) used SVM to classify e-mails.Woitaszek et al. (2003) have used learning approach to filter out spam mails by Support Vector Machine (SVM). Renuka et. al. (2014) presented an SVM spam E-mail classifier with a feature extraction method using Latent Semantic Indexing (LSI) to select the suitable feature space for improving the accuracy. Li, Wenjuan, and Weizhi Meng (2015) performed an empirical study with three different environments with supervised machine learning classifiers like a decision tree and SVMs. Boujnouni (2015), presented a new method to classify automatically legitimate email from spam, based on the combination of two improved versions of Support Vector Domain Description (SVDD).

In yet another direction of work, Ying et. al (2010) proposed, an ensemble approach to classify spam e-mails, based on decision tree, support vector machine and back-propagation network. Wang, et. al (2010) Spamcooling a parallel heterogeneous ensemble spam filtering system based on active learning techniques was also developed in the meanwhile. Zhen Yang (2006) An interesting approach to detect spam by Naive Bayes Ensemble Based on Decision Induction was also tried. Zi-Qiang Wang (2006) proposed an efficient spam filtering method using feature selection and Support Vector Machine technique. Chih-Ping (2008) The work focused on the effective spam filtering by using a single-class learning and ensemble approach. Amayri (2009), presented a work wherein the investigation was how different spam filters behaved with respect to different SVM kernels. Jason and Rennie (2000) had, in the mean time, demonstrated that the SVM is costly to train and requires significant time to classify. Caruana et al (2011, 2010) proposed a MapReduce based parallel SVM for large scale spam filtering and SpamCloud, a MapReduce based anti-spam architecture. Tao Yong-cai et.al (2011) proposed a MapReduce-based Bayesian anti-spam filtering mechanism. Zhao et.al. (2012) presented MRESVM, a MapReduce-based distributed SVM ensemble algorithm for scalable image annotation. Kakade (2014) surveys different spam filtering techniques, Support Vector Machine (SVM) training problems and the need to introduce MapReduce. Karol Wrótniak (2013) designed an effective spam filters using combined Näive Bayes classifiers by applying different methods of feature extraction to ensure the desirable diversity of classifier ensemble. Pătraşcu,(2015) presented a based on a cloud supported infrastructure of a service oriented architecture for spam detection and classification, which was able to scan and classify a large stream of emails.

In the literature, many studies can be found based on applying kNN Joins in the traditional method with a centralized single server. Böhm, C. and Krebs, F., (2004), Yao, Bin et.al. (2010). It is an important and frequently used operation for numerous applications including knowledge discovery, data mining, and spatial databases (Yu, Cui et.al. 2010). Gorder used a novel kNN-join algorithm (Xia, Chenyi et. al. 2004) involving a block nested loop join method that exploits sorting, join scheduling as well as distance computation filtering and reduction for reducing both I/O and CPU costs. LU, Wei (2012) investigated the performance of kNN Join using MapReduce which is a well-accepted framework for data-intensive applications over clusters of computers. Zhang et.al. (2012) proposed a novel (exact and approximate) algorithms by using MapReduce to perform efficient parallel kNN Joins on large data demonstrated using Hadoop.

2.9. Research Gaps

Following research gaps were identified, in course of literature review, related to E-mail spam and associated strategies:

- Absence of any reasonably accepted single set of cohesive strategies that could take care of all major categories of Internet based E-mail spam.
- Absence of a scalable text based mail spam classification strategy that could suitably scale by taking advantage of currently available distributed computing resources,
- Near-absence of dynamic pattern based spam filter,
- Absence of computationally efficient spam classification and handling strategies that could benefit from combination of tools and techniques available in the world of computational intelligence and concurrent programming.

These were the above-mentioned research gaps which helped in refinement of the original research objectives and consequently choose the research methodology that unfolds alongside the consequent work in the following two chapters.

CHAPTER 3
EXPERIMENTAL MODEL DESIGN

3.1. Introduction

In a broader perspective, E-mail Classification problem can be treated as a text classification problem, wherein 'spam E-mail' and 'legitimate E-mail' are two broadly different available classes into which any E-mail Text could be to be classified.

There exist multiple supervised machine-learning algorithms; those have been successfully applied to this problem like Naïve Bayesian, Support Vector Machine, Decision tree, k-NN etc. as mentioned in Chapter-2 along with their respective merits, demerits and select instances of their historical application in different Spam E-mail identification and filtering strategies in research experiments as well as in commercial solutions.

This chapter presents select strategies and algorithms proposed for the effective control of E-mail spam, ensuring the required degree of experimentally demonstrable accuracy. For each algorithm presented here, the brief design, design details as well as chosen public and /or private datasets used for the algorithm have been presented along with experimental tools/instruments details. Relevant classification algorithms have been duly described in relevant experimental setups in adequate detail.

3.2. Involved Design Approach and the Corresponding Experimental Modeling

This section and its sub-sections, together, describe strategies used in arriving the right combination of select techniques that led to the design and subsequent validation of the resultant algorithms along with the design of the respective experimental model. Relevant justification as well as performance analysis was also duly performed in each case as presented in later sections.

3.2.1. Combined Approach of Bayesian and Neural Network

The work proposed here is a hybrid approach to E-mail spam classification. Reason for hybrid approach is, both advantages and limitations in each of the base algorithm chosen and the relative benefits or strengths of the created hybrid algorithm. The Key idea behind this hybrid approach is the combination of different algorithms leads to improved results in terms of efficiency, accuracy, percentage of false positives, training time, etc. All machine learning based approaches involve supervised or unsupervised learning from / training with the data of the right kind and in appropriate quantity. Reducing the training time without compromising on the classification accuracy is, therefore, a key factor in spam classification. A reduction in one or other form of computing complexity is always a dream goal of any designer, more so when the reduction in complexity comes with ease of implementation. In this work, we have used a combination of different algorithms, centered Bayesian networks, neural networks trained with genetic algorithm.

Learning in a Centred Bayesian Network (CBN) is much simpler compared to that in the other variations of a Bayesian network. The reason behind this simplicity lies in the network structure of any CBN which is almost fixed. All the edge nodes from respective parent nodes point towards the center node. The Conditional Probability Distributions of the center node can be easily learnt based on Hebb rules. Header parts of the E-mails like Sender Address, Sender Name/ID and Carbon Copy (cc) details are trained in three different CBNs.

Subject and content are dealt separately with neural networks. These neural networks are trained with appropriate Genetic Algorithms (GA). By combining theses algorithms, the workload on the NN is reduced to a large extent since, if used alone, in case of the high dimensional data, the training time of neural networks increases.

An attempt to reduce training time is by opting genetic algorithm. GA was employed here for training the neural network. The training time of a classifier is heavily dependent on the dimensionality of the inputs and also the efficiency of the classifier and optimizing the classifier's performance to the desired level was what was primarily attempted here.

3.2.1.1 Centered Bayesian Network (CBN)

A Centered Bayesian Network (CBN) is a centered graph with a pair of (G, P), Where G = (V, E), and P is a set of Conditional Probabilistic Distribution (CPD) of the centre node of G given its parents (Shi, Zhiwei et.al. 2007). A centered graph is a directed acyclic graph, with only one center node, and there are no edges ending at non-center nodes. The centre nodes of the CBNs are called internal nodes and all other nodes are called external nodes. The following figure 3.1 presents the CBN.

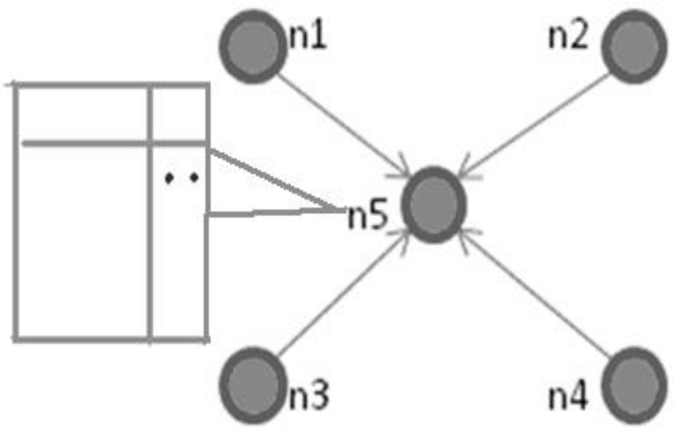


Figure 3.1. Centered Bayesian Network (CBN) Topology

3.2.1.2 The Neural Networks Element

Neural network (NN), more often referred to as an 'artificial' neural network (ANN), is a powerful computational data model that made upon parallel information processing systems. It is directed graphs whose nodes represent the simple neurons organized in different layers and are connected by links. There are a number of different types of NN, whereby the basic feed forward network configuration was employed as a classifier and discussed in this study.

The network has a series of layers with an input layer which communicates to one or more hidden layers. The hidden layers, then link to an output layer. In this feed forward network, the information moves in only one direction because the links only go forward, never backward. They go forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. Figure 3.2 shows the Neural Network structure.

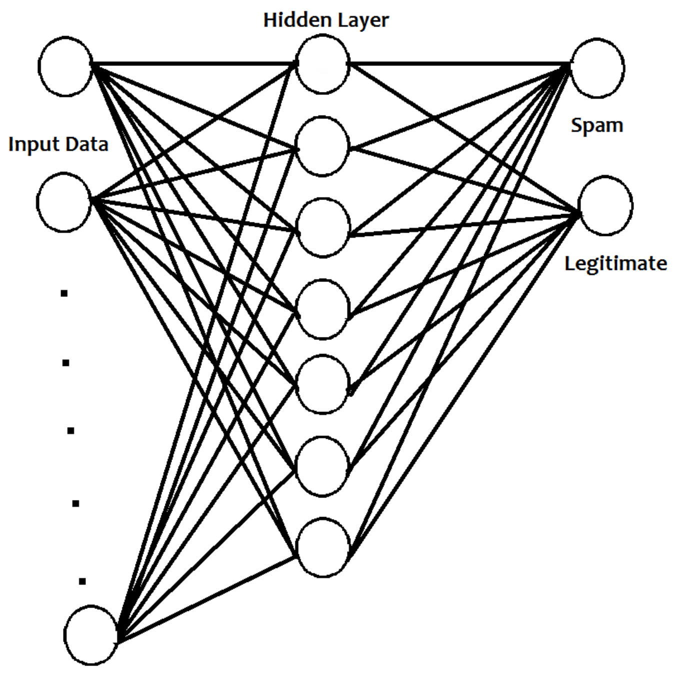


Figure 3.2. Neural Network Structure for E-mail Subject Classification

3.2.1.3 Genetic Algorithms (GAs)

Genetic algorithm (GA) is an optimization technique inspired by natural evolution. Genetic algorithms were formally introduced by (John Holland, 1975) at the University of Michigan. They have been, since, applied extensively in different types of optimization problems.

An implementation of genetic algorithms, typically, begins with a population of chromosomes, then followed by the fundamental genetic operations like fitness function, selection, recombination and the evolution scheme and finally the most appropriate offspring is selected to be passed on to succeeding generations.. GA operates on a population of chromosomes, which represents the parameters that are to be optimized. Each parameter is encoded in a binary string called gene. Thus, within a chromosome, there will genes available based on the number of parameters to be optimized. All the genes were randomly assigned "1" and "0" initially.

The parameters to be optimized are represented by a chromosome whereby each parameter is encoded in a binary string called gene. Thus, a chromosome consists of as many genes as parameters to be optimized. A population, which consists of a given number of chromosomes, is initially created by randomly assigning "1" and "0" to all genes. The fitness function evaluates the best chromosomes, based on the highest probability. GA carries out the fitness based selection process, from the randomly generated chromosome population, to produce the next generation, successor. By mating these parent chromosomes, child chromosomes are produced. These once again be passed on to the successor population. This process of evaluation and reproduction are repeated until a certain number of generations, defined fitness, a convergence criterion of the population are met. Finally, all chromosomes of the last generation have the same genes represent the optimal solution. The following Figure 3.3 explains the Genetic Algorithm operating steps.

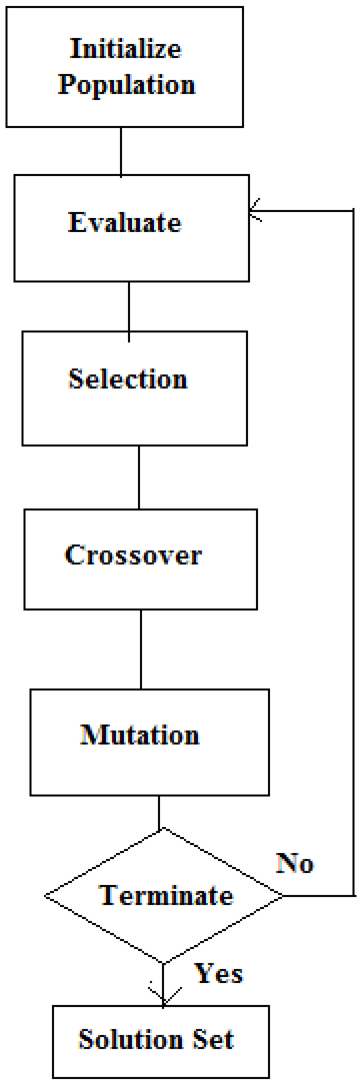


Figure 3.3. Genetic Algorithm: The Way It Works

3.2.1.4. Optimization of Neural Networks with Genetic Algorithms

The neural networks can become complex because of involved parameters, namely the number of weights and the number of biases. Number of hidden layers also plays a key role in complexity, with the dimensionality of the optimization hyperspace. while the major complexity rises, by the number of adjustable parameters. Consequently, the number of parameters can be recognized as the sum of links, hidden neurons and output neurons. Thus,by optimizing the number of input variables, the number of hidden layers and the number of neurons in the hidden layers by applying different optimizing strategies can minimize the complexity of NNs. Thereby an optimal variable subset helps in avoiding the overfitting and reduces the noisy and redundant variables.

Genetic Algorithms (GAs) are proved to be one of the successful optimization techniques. The idea of combining GA and NN came up first in the late 80s. By using a gradient based approach, the training process is slow, and even sometimes it gets hanged with local minimum. There are many optimization solutions which can replace these gradient approaches, such as Genetic Algorithms (GAs), Particle swarm optimization (PSO), Ant Colony optimization. By combining of applying genetic algorithms to neural network features, GA is used to give an optimal subset of features that would be best to train NN. With these optimal features, NN training would be faster as there is a reduction in dimensionality, and hence improves accuracy. The Neural network and genetic algorithm combination have been applied in different problems, Hancock (1991) in face recognition, James Clark (2013) e-mail classification, Mohammad et.al, (2011) for spam classification, Schiffmann (1993) for classification of Thyroid glands normality, Bishop (1993) colour prediction etc.

Two separate Neural Networks were considered, one for classifying the subject of the E-mail, and the other network for the content of an E-mail. Neural networks of subject of an E-mail contain the input nodes equal to the feature vector size of the subject input data. Same structure is maintained for the content related network. So each input unit corresponds to a single feature. In order to avoid a form of after fitting, a configurable single hidden layer is used with fewer neurons compared to the input layer. As the network is used for binary classification problem only one output layer with one neuron per class is used. The output of this neuron should be either 0 for legitimate keyword or 1 for spam keywords.

GA is most appropriate for training the neural network instead of directly acting as a classifier. In this context the chromosome is the E-mail content for the first network and the subject of an E-mail will be chromosome for the other networks. The genes of the chromosomes are replaced by keywords/features of the subject and content of the E-mail.

The fitness function which is the main point of control for a GA taken as the sum of the conditional probabilities that each word has if it were to be present in spam and the message is indeed a spam. In this specific case:

P(B|A) = (P(A|B)*P(A))/P(B)

where,

B = spam mail, A = word taken from spam mail

P (B|A) = probability that the E-mail is spam given the word is present in the E-mail

P (A|B) = probability that the word is present in spam (prior analysis)

P (A) and P(B) = prior probabilities which are also assigned by prior analysis

Here, genetic algorithm is applied just to train the neural networks. Data preprocessing is done by feature extraction process which comprises of process of word stemming and stopping to remove the unimportant words so as to reduce the overhead for the effective functioning of the GA.

The conditional probabilities of the keywords related to the content of an E-mail and subject of an E-mail to the other network were summed up separately to get the fitness. By adding the conditional probabilities of the all the words present in both the subject and content fields, the sum can be maximized. To normalize, the fitness function can be adopted by setting the number of ideal words by hand, for normalizing. Thus, the normalized value will be within a range of 0 -1.

New Fitness Value = (Old value)*(selected words / [(ideal words* Total No. of words)].

3.2.1.5. Design Model

The hybrid scheme leading to the *Bayesian and Neural Network Classifiers (BNNC) algorithm* has been presented in this section. The scheme involves what has been termed here as the *Bayesian and Neural Network* (Manjusha et.al 2010) is a hybrid network as the name itself indicates. It comprises of a collection of different networks, with corresponding algorithms and respective inputs. This network consists of three Centered Bayesian Networks (CBNs): first one for the Sender Address, second one for the Sender ID and the last one for Carbon Copy fields of the E-mail Headers. Two Neural Networks (NNs) are also embedded in BNNC along with these three CBNs. BNNC is a one way directed network with a two-layer architecture with two internal nodes. Layer 1 internal node is the internal node of the CBNs. Layer 0 internal node is the internal node of the BNNC.

Each Centered Bayesian Network (CBN) has its own Conditional Probabilistic Distributions (CPDs) for every internal node. CPDs define the updating rule for the active strength of the internal nodes. There is a priority-based decision box that provides the final output. This priority-box acts as a center node/internal node for BNNC. Every internal node as well as all the networks gives a binary output. All the networks of internal nodes, direct their output to the priority box which is in Layer 0 and it finally classifies an E-mail as spam or legitimate.

All the external nodes of the CBN represent some entity of the external world. Every external node of CBN has either a value of 1 or 0; i.e. 1 for active high and 0 for active low. These values represent the presence or absence of the entity or concept in the Input fed to the Bayesian Network against the training set. The CBN nodes can be viewed as a neuron column and the directed links can be viewed as the connections between neurons. These directed links establish causal relationships among the entities/concepts of the external world.

The Layer 1 internal nodes modify their active strengths according to the states of their parent nodes (external nodes) and their own CPDs. NNs updates the final binary output to its FB box. In the next stage, Layer 0 internal nodes modify their states based on their CPDs and the active strength of Layer 1 internal nodes. The active strength of the Layer 0 internal node which is a priority decision box takes the binary values of all the networks and finally decides the type of E-mail. BNNC or the Layer 0 internal node alone represents the kind of E-mail being received.

3.2.1.6. Experimental Tools/Instruments Data Sets Used

This section describes the environment, experimental tools, scenarios and instruments used in the study.

3.2.1.6.1. WEKA

WEKA stands for Waikato Environment for Knowledge Analysis. It is a well-known open source software suite for machine learning, data analysis and predictive modeling that supports several data mining tasks, like data pre-processing, clustering, classification, regression, visualization and feature selection etc. WEKA was developed by the University of Waikato in New Zealand. It has been implemented in JAVA and allows provides access to SQL databases by making use of Java Data Base Connectivity (JDBC). It has provision for processing the results returned by a database query.

The WEKA suite has three separate graphical interfaces the Explorer, Experimenter and the Knowledge Flow. The main user interface is the Explorer, where the user can gain quick access to all the features in WEKA and can freely analyze the data. The Experimenter provides a setup to conduct machine learning experiments where one can compare different algorithms against each other more easily. The similar functionality can be accessed from through the command line interface or the Knowledge Flow interface. The Knowledge flow interface also provides the user to utilize the same methods as in the Explorer. The whole process, right from reading the data to final results and plotting these evaluations in a graph, can be carried out with the help of this interface. A text-based terminal mode is also available, wherein one can invoke the different methods directly by making use of function calls. WEKA uses its own file format called Attribute Relationship File Format (ARFF) (Witten & Frank 2005). The ARFF format presents the data according to the requirement of the applied algorithms during the modelling stage.

3.2.1.6.2. E-mail Dataset Preparation

While the research reported here has made use of four (three public and one personal) data sets, in all, the first experiment had involved the personal data set. This dataset was collected from the scholar's personal inbox and consisted of both spam and legitimate mails. Spam mail collected was consisted, mostly, of advertisements for health products/web sites/online degrees/adult products, fast money-making schemes, chain letters etc. The legitimate e-mails included mostly personal and some individually or group-addressed official e-mails. The first step involved data cleaning, resulting in the elimination of some irrelevant data. By using a MIME parser, information from the email was generated. Parsed file, thus generated, contained one record for each email with the E-mail record's name, E-mail's content and information specific to the E-mail fields like from, subject etc. It, thereafter, was converted into the ARFF (Attribute Relation File Format) for use with WEKA.

In the pre-processing stage, first, all the irrelevant features are filtered out and then stemming is applied to stem out some of the generic terms. By using this method, all suffixes are removed. Consequently, the Stop Words removal process removes stop words which are functional words and carry no significant information usable in text classification. These words are generally pronouns, prepositions or conjunctions. Such a pre-processing lead to a reduced number of features, saves memory space and also the training time.

3.2.1.6.3. Evaluation Metrics

For evaluating the effectiveness of the proposed approach, standard information retrieval techniques were used. Precision and recall were the basic measures used in evaluating search strategies. Relevant documents retrieved by the algorithm were classified as True Positives (TP), while irrelevant documents retrieved by the algorithm were termed as False Positives (FP). Relevant documents not retrieved by an algorithm were considered to be False Negatives (FN) and irrelevant documents that also could not be retrieved by an algorithm were considered to be True Negatives (TN).

Precision, also called positive predictive value, is a measure of exactness. It is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. It is usually expressed as a percentage. Precision ***p*** is calculated by the following formula:

$$p = \frac{TP}{TP + FP}$$

Recall, also known as sensitivity, is the measure of completeness. It is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is also usually expressed as a percentage. Recall ***r*** is calculated as follows:

$$r = \frac{TP}{TP + FN}$$

The ***F-Measure*** [Lewis 94] attempts to combine Precision and Recall into a single value for comparison purposes. Van Rijsbergen proposed a balanced weighting measure between precision and recall called the ***F***-measure which is expressed as:

$$F = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

3.2.2. Spam Classification by Multiclass Support Vector Machine Classifier

The problem of spam was addressed as a simple binary classification problem where the main goal was to filter out or separate spam from legitimate. *(Effectively extending SVM for multiclass classification is still a research problem, that being actively investigated by contemporary researchers.)* Compared to many other classifiers, SVM provides better generalization capacity; but, it has major limitations like high computational complexity and training time. SVM decision function depends on the number of support vectors. Therefore, if the number of support vectors increases, it leads to a consequent rise in training and testing time. *Decision Tree* is another important classifier which can be used in classification problems in such settings. Once a decision tree has been built, classifying a test record gets extremely fast, with a worst-case time complexity of *O(w)*, where *w* is the maximum depth of the tree. When comparing both learning methodologies, Decision Trees classify new instances much faster while compared to SVMs whereas SVMs outperform Decision Trees in terms of accuracy. In order to exploit the advantages of both the algorithms, after a careful analysis and some initial experimentation, a combined approach of SVM and DT was attempted to classify E-mails. Abbreviated as *BDT-MSVM classification algorithm*, the *Binary Decision Tree for Multiclass Support Vector Machine classifier* (Manjusha 2013) represents this evolved hybrid approach, aimed at speeding up the training of SVMs over large-scale E-mail data sets. Following sub-sections provide greater details along with relevant information in this context.

3.2.2.1. Support Vector Machine (SVMs)

Support Vector Machine (Cortes, Corinna, and Vladimir Vapnik., 1995) is a supervised machine learning algorithm applied for classification and regression. SVMs are based on the principle of *Structural Risk Minimization*. They can train classifiers based on training classifiers based on polynomial functions, radial basis functions, etc., both, linearly and non-linearly. A Support Vector Machine constructs a hyper-plane in a high dimensional space, such that the margin of separation between the two classes in the data is maximized for classification. SVM linear classifier tries to maximize the margins of their decision boundaries in order to ensure that the generalization error is reduced. SVM techniques have been widely used in many real world applications such as text categorization, image classification, E-mail spam classification, etc. SVM classifiers have proven better classification accuracy in comparison to select other classification methods. This technique overcomes the curse of dimensionality. However, when the size of the training dataset becomes large, SVMs proves to be computationally intensive and lead to high algorithmic complexity.

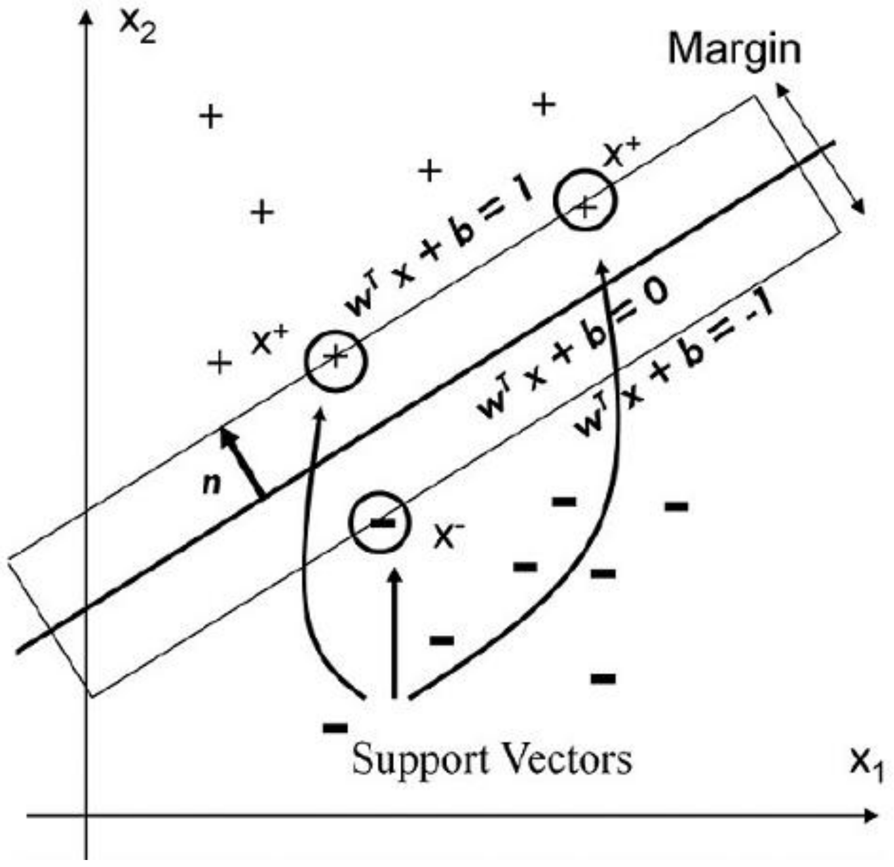


Figure 3.4 SVM Hyperplane

The SVM has been widely used to address a binary classification problem. Both linear and non-linear SVM constructs can be represented for binary SVM by a set of hyperplane in an infinite dimensional space as presented in Figure 3.4. The concept of the SVM is to maximize the geometric margin between the optimal hyper plane and the support vector and minimizing classification errors.

For an SVM binary classification, determining an optimal separating hyperplane that gives low generalization error is the primary concern. For two classes, the labels are represented by -1, +1. The ***decision function*** for linearly separable classification is represented by the relation:

$$f(\vec{x}) = \operatorname{sign}(\vec{w}^T \vec{x} + b)$$

where, w is the normal vector to the hyper plane, b is the scalar quantity which is the perpendicular distance, from hyper plane and the label of the sample is given by the sign of ***f(x)***.

The linear SVM finds the ***optimal separating margin*** by solving the following optimization task

$$\textbf{\textit{Maximum Margin (M)}} = 2\,\|w\|$$

For the binary classification, the *decision function* equation by using the ***Lagrangian Multiplier*** $\boldsymbol{\alpha_i}$ is used so that the target optimization problem could be solved with respect to the ***corresponding data point*** $\boldsymbol{x_i}$, ***Support Vector (SV)***; and, therefore, the ***linear decision function*** equation can be expressed (along with the *optimal hyper plane parameters)* as follows:

$$f(x) = sign\left(\sum_{i=1}^{p} y_i \alpha_i \langle x.x_i \rangle + b\right)$$

In the case of nonlinearly separable classes, the classes cannot be separated by a linear decision boundary. In order to adapt to the noisy data, the margin has to be prepared accordingly and thereafter the relevant kernel is to be applied to it. Here, the dot product is replaced by a nonlinear kernel method that can be computed. *The linear SVM can further be expanded to non-linear SVM by replacing* x_i *with a mapping into the feature space, in other words, the dot product (x, xi) is replaced by the function:*

$$k(x_i, x_j) = \langle \emptyset(x_i).\emptyset(x_j) \rangle$$

Then, the *new decision function equation* to classify the data can be written as:

$$f(x) = sign\left(\sum_{i=1}^{p} y_i \alpha_i k(x, x_i) + b\right)$$

The following are the most commonly used *kernel functions*:

1. *Linear :* $k(x_i, x_j) = x_i.x_j$
2. *Polynomial:* $k(x_i, x_j) = (x_i.x_j + 1)^d$
3. *Radial (RBF):* $k(x_i, x_j) = exp\left(-\|x_i - x_j\|^2 / 2\sigma^2\right)$
4. *Sigmoid:* $k(x_i, x_j) = tanh(k(x_i.x_j)| - c)$

3.2.2.2. SVM Multi-Class Approach

For multi-class problems, binary SVMs are combined in form of either 'one-against-one' or 'one-against-all' strategies. The key concept of SVM was, originally, described for a binary classification; but, many of the real world problems are not binary in nature. For multi-class problems, there exist different algorithms like *One-Against-All (OAA)* and *One-Against-One (OAO)*. If we consider a problem with *n* classes, the corresponding OAA algorithm would consist of *n* separate binary classifiers for *n-class* classification. It constructs *n* hyper planes which separates one class and other remaining *(n-1)* classes. OAO algorithm, on the other hand, evaluates all possible pairwise classifiers (Knerr et al. 1990). It constructs individual binary classifiers with *n(n−1)/2* hyper plane, which separate these binary classes. In the two cases, the final label is the one that is mainly chosen. The size of classifiers created by the OAO approach is much larger than that by the OAA approach. *Directed Acyclic Graph SVM (DAGSVM)* is an improvised version of OAO (Platt et al., 1999) which is a tree-like structure to speed up the testing phase. By this, for deciding the label of a test record, only *k−1* independent evaluations are required. Platt et. al. (1999) presented a new learning architecture which operates in a kernel-induced feature space and uses two-class maximal margin hyperplanes at each decision-node of the Decision Directed Acyclic Graph (DDAG). This algorithm is used to combine many two-class classifiers into a multiclass classifier. Chih-Wei Hsu et. al (2002) gave a decomposition implementations for two such "*all-together*" methods and then compare their performance with three methods based on binary classifications: "one-against-all", "*one-against-one*" and *directed acyclic graph SVM (DAGSVM)*. Tang et. al. (2005) proposed the multiclass SVM methods based on binary trees.

3.2.2.3. Binary Decision Tree

Decision tree classifier is one of the most important classifier used in data mining field. A technique used for constructing a decision tree is comparatively inexpensive, and once the decision tree is constructed classifying the test records are done quickly. Another major advantage for decision tree is that it is quite robust to noise and since it uses top-down approach, number of records to search decreases gradually. In the decision tree as shown in the Figure 3.5, each node, except the leaf, contains some attribute test condition. The outgoing link corresponds to the agreement and failure to the condition. The interior node can have binary split or multiple split. As we come down the level of the hierarchy the number of records satisfying the sub tree gets reduced and finally reaches a state where all the records belong to the same class. That node becomes the leaf node that specifies the class of the record. Once the complexity of the decision tree increases beyond some threshold, generalization error goes high and fails to classify previously unseen records accurately. In SVM classifier, testing time increases as the size of dataset increases (with training set proportion larger than testing set).

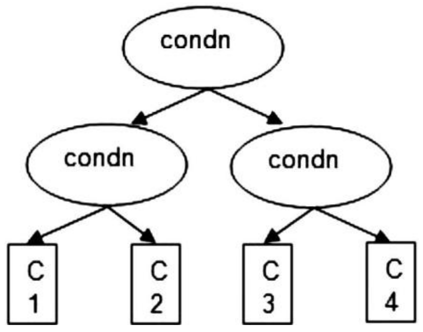


Figure 3.5. Decision Tree.

Thus, the number of support vectors is directly proportionate to training size and therefore, the increasing number of support vectors dominates over number of testing data points, thereby increasing the testing time of SVM. On the contrary, the size of the DT (total number of decision nodes and leafs) increases as we increase the size of dataset similar to increasing number of support vectors in SVM. However, the decreasing number of testing instances dominates the increasing tree size and hence the testing time of DT decreases.

3.2.2.4. Design Model

The basic Support Vector Machines was developed for binary classification and later it has been extended to multi-class problem, (*n* different classes, where *n>2*). There now exist several such proposed extended solutions for multi-class classifications. However, these solutions have many implementation issues, as they require many binary SVMs to solve such problems. There are many limitations of these approaches; for instance, at times more than one class receives the highest vote and there might be situations wherein some instances are left out with class unassigned.

This research work tries to solve this ambiguity by assigning some confidence value to instances, instead of a class voting. This confidence value is assigned by a classifier and here SVM was employed as a classifier. Many such works focused in reducing the number of support vectors, while our key focus was to approximate the decision boundary of SVM by reducing the number of test data points that aids in maximizing SVM margin by using binary decision trees. The crucial data points lying near the decision boundary were classified by using SVM and data points which are away from decision boundary were immediately classified as uni-variate nodes by Decision Tree. Here, the threshold value plays a major role in getting better classification accuracy. Finally, based on this threshold parameter, a hybrid tree is generated with both uni- and multi-variant nodes. With this approach, both, the efficient computation and high classification accuracy of SVMs can be reached.

By using an ensemble approach, a hierarchical binary tree is constructed in a way that the classification results obtained in first level will be passed on to the next level and so on in a hierarchy. *An SVM based decision tree for spam multi class classification* was employed to speedup SVMs training time. Spam dataset was classified, in this case, into three classes based on the threshold value, i.e. *spam, legitimate* and *likelihood spam*.

3.2.2.5. Experimental Tools/Instruments

The relevant experimental tools and datasets have been duly presented in the following sub-sections.

3.2.2.5.1. Enron Corpus

The *Enron Corpus* (Klimt, Bryan, and Yiming Yang., 2004) is a large database of E-mails originally collected at Enron Corporation. Once this corporation collapsed because of some scandal, for the investigation, the E-mails from 158 employees were generated and acquired by the Federal Energy Regulatory Commission and later made public. There were lots of integrity problems. Melinda Gervasio at SRI for the CALO (a *Cognitive Assistant that Learns and Organizes*) project had taken up the issue. In this process, some E-mails have been deleted, on the request of any affected employees of the company. The original dataset was cleaned up and edited. The corpus has a combination of both personal and official E-mails. The raw Enron dataset contains 619,446 messages belonging to 158 users. As the dataset is easily accessible, with original E-mails sent/received makes it unique compared to other datasets which have lot many legal and privacy issues. This makes it one of the most suitable data sets for the experimental evaluation of E-mail classification.

3.2.2.5.2. IBM® SPSS® Modeler

IBM SPSS Modeler is a data mining workbench that enables to create powerful predictive models with an intuitive graphical interface. Through the inbuilt modeling assistance, it allows to build predictive models quickly and intuitively without any programming. A wide variety of techniques can be analyzed in a single run with ease by using this type of modeling support and their results can be compared. This Modeler software is user-friendly in drawing Decision Trees. It uses C5.0 an upgraded version of C4.5 to draw a DT.

3.2.3. The Ensemble-Based SVM Spam-mail Classifier

In this part of research work, a Map-Reduce based non-linear distributed algorithm was used for solving such large scale e-mail classification problems. The presented solution approach involves splitting the large data set into pieces using Map-Reduce and subsequently applying an SVM classifier in each map phase. Support vectors received from the parent map process have been used as training data for the second SVM. Results, such obtained, demonstrate substantial improvement over comparable techniques reported in contemporary literature. The following sub-section present greater details about various involved aspects of this specific approach.

3.2.3.1. Ensemble Support Vector Machine

The basic concept of *ensemble classifiers* is an integrated approach, wherein the output of several classifiers is combined together to give the final output. This approach was applied in many research problems for the efficient classification results. Ensemble SVM is an aggregation of several SVMs, which were trained with subsamples of the training data sets. It was based on divide-and-conquer strategy. With data partitioning, total training time decreases significantly. For constructing ensemble SVM, bagging technique was adopted. In bagging technique, the partitioned data was trained independently, by bootstrap technique. While training, in ensemble SVM, *n* training datasets are needed for *n* partitions of the data. The quality of the result, in this case, typically depends on how different each training sample data is. Here, the bootstrap method helps in building the replicate training datasets, depending on the number of partitions made by using the random resampling method. Finally, each of these bootstrap built training datasets help in training certain SVM.

Once ensemble is built and the training is done, there is a need to combine all these independently trained SVMs. Aggregating several classifiers in an ensembling approach is a crucial element, so as to obtain the best predictions. In ensemble SVM, independent SVMs are combined, in order to get a final decision. There exist different aggregation techniques applied in both linear and nonlinear combinations, like majority voting, least squares estimation based weighting (LSE-based weighting), and the double layer hierarchical combining. The linear aggregation method involves combining several SVMs linearly, through majority voting or the LSE-based weighting methods. These methods often go well with bagging and the boosting, respectively. A nonlinear aggregation approach is a nonlinear combination of several SVMs, like the double-layer hierarchical combining which involves two layers, the upper and the lower layer. Here, the upper layer SVMs combine with several lower layer SVMs. A very basic aggregation method is majority voting, where the class prediction was made on the basis of majority predictions of base SVM models. The majority voting aggregation technique was employed in this work because of its simplicity. Ensembles of SVM models have been used in several fields. A neural network was used to aggregate the base classifiers in this ensemble approach for large data learning (Collobert et al., 2002). (Valentini and Dietterich, 2003) experimented with heterogenous kernel models for ensembling and concluded that for efficient results, an ensemble of homogeneous kernel functions is better.

Yu et al. came up with a cluster-based data selection and parallelization as a solution for the high computational time. (Wang et al. 2009) made an empirical analysis of different ensemble SVM classifiers with different types of boosting and bagging SVMs. (Meyer et al. used bagging methods, cascade ensemble SVMs and even combination of both these methods for large data sets. (Li et al. 2008) used weak learners for studying Adaboost SVMs. Weak learners were obtained by adapting the kernel parameters for each SVM. (Valentini, 2004) analyzed the bias-variance of the bagged SVM ensembles and found that bias-variance is consistently reduced in bagged SVMs while compared to single SVMs.

3.2.3.2. Parallel Support Vector Machine

It is well recognized that while Support Vector Machine (SVM) is extremely powerful and widely accepted classifier in the field of machine learning due to its better generalization capability and whereas for small data sets, they are robust and offer fast training and testing , they are not suitable for large scale dataset due to its high computational complexity. With the increased data size, SVM has a training complexity of $O(n^2)$, where n represents training instances. With this training, a SVM becomes very expensive. Thus, applying SVMs to large datasets, the major problems include both high memory consumption and large computation delay.

Larger datasets can be, however, effectively taclked by applying data paralellization. Consequently, in case of Parallel SVM, the training time can also be decreased. Storage problems associated with very large data volumes can be also solved better in case of parallel and distributed computing environment as compared to a centralized system, since there is enough aggregated memory available on parallel machines. The proper analysis of large scale data and obtaining effective results is still a hot topic. This process requires extending the data mining applications suitable for analyzing large scale data which itself is a challenge. Various parallel algorithms are available as of now based on different paralellization techniques for different architectural environments like OpenMP, CUDA, MapReduce, threads, etc. which yield varying degrees of performance and demonstrate different usability characteristics. For the large-scale data analysis, choosing the appropriate parallel algorithms helps in meeting the efficiency, scalability and other performance requirements. MPI model is found to be efficient in simulation based computing problems, rather than real practical problems. The MapReduce is practically well suited for the large scale data analysis among all these techniques and this is why it was chosen here.

3.2.3.3. MapReduce

MapReduce is a programming paradigm intended for many large scale computational problems. Some of the most popular implementations are Hadoop (Shvachko, Konstantin et.al. 2010), Google's Implementation and Mars. Though many of the implementations are very efficient for batch processing, they also provide the parallel processing capability which proves to be quite of use in techniques like spam E-mail filtering. At the core of each MapReduce implementation, it is divided into two phases, Mapper and Reducer in the form of map and reduce functions. From a conceptual point of view, all the data becomes available in the form of what is known as *Key (K) and Value (V) pair*s. At the finest atomic level, each map function takes *(K, V)* as input and generates an output which may be intermediate or direct input to reducer phase in some cases. The intermediate phase is called Combiner in case of Hadoop. This phase is performed for better optimization. The output *P* of the Partitioner function returns the reducer number to which this key-value pair (received from

Map function by the Partitioner function) must be sent to for further processing. Figure 3.6 depicts the MapReduce Framework.

Functional programming style with two functions to be given:

$$\boldsymbol{Map(K_1,V_1) \rightarrow List(K_2,V_2)}$$

$$\boldsymbol{Reduce(K_2,List[V_2]) \rightarrow List(v_3)}$$

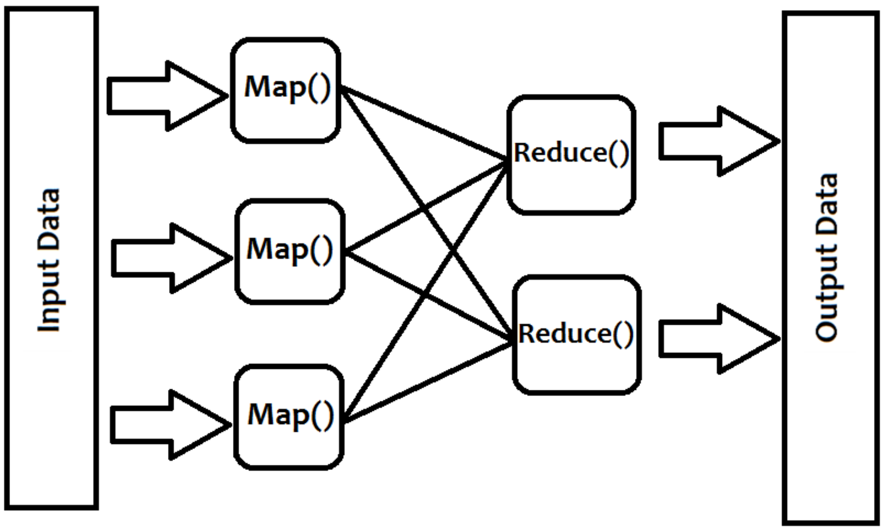


Figure 3.6. MapReduce Framework

3.2.3.4. Design Model

The Process of Spam Filtering can easily be parallelized using the MapReduce programming framework and by adapting this method SVM training performance can be improved. Apache's MapReduce Implementation 'Hadoop' is highly suitable to handle large chunks of data in parallel. A large E-mail dataset can be used in Hadoop and training process for building models becomes significantly faster as models are built in parallel over subsets of the original spam dataset. The approach followed in this work termed as '*the ES^{2}C Technique*' employed *Ensemble based SVM as the Classifier* to help separate spam mails from legitimate ones.

3.2.3.5. Experimental Tools/Instruments

The experimental tools and datasets related to this part of the work have been presented in the following sections. Two large public data sets known as the Enron Corpus and the Ling-Spam Corpus were used here.

3.2.3.5.1. Enron Corpus

The Enron Corpus presented in section 3.2.2.5.1 in detail was used here as well for the same reasons as described initially.

3.2.3.5.2. Ling-Spam Corpus

Ling-spam corpus is the combination of both spam messages which were collected from personal mailboxes and legitimate messages from the Linguistic mailing list. A total of 2,893 e-mails in the dataset, of which 2,412 legitimate e-mails and 481 spam e-mails were retrieved (Androutsopoulos, Ion, 2000). E-mails in this dataset have only the subject line and mail body text. All others were removed along with the attachments. One can easily distinguish spam and legitimate mails from the corpus as the mail distribution is totally unlike of normal user inboxes.

3.2.3.5.3. LIBSVM

LIBSVM (Chang, Chih-Chung, and Chih-Jen Lin., 2011) is a popular implementation of SVM, a machine learning classifier. LIBSVM supports various SVM formulations for classification, regression, and distribution estimation. LIBSVM provides a simple interface where users can easily link it with their own programs. A typical use of LIBSVM involves two steps: first, training a data set to obtain a model; and, second, using the model to predict information about a test data set. It was developed at the National Taiwan University and has been implemented in C++ though with a C API.

3.2.3.5.4. Apache Hadoop

Hadoop, mainly, contains two parts: Hadoop Distributed File System (HDFS) and the MapReduce. HDFS is a distributed and scalable Hadoop file system. HDFS can manage and process huge amounts of data (petabytes). A HDFS cluster consists of two different types of nodes, the NameNode and the DataNode. NameNode also known as Master Node manages the entire file system which maps the blocks to DataNodes. DataNode manages several tasks like it serves clients by accessing them the requested data. They act upon the instructions from the NameNodes like creation, deletion etc. and performing block operations. NameNode is the only master node in a Hadoop Cluster. It should be the most reliable node as if any service break in this node might cause the whole cluster unserviceable. In the simplest of senses, the idea behind use of the HDFS is to be able to deal with large datasets by dividing them into several blocks. These sets are then sorted and each sorted sets is passed to the reducer.

Hadoop’s infrastructure is designed based on the assumption that data will be distributed with hardware that is subject to failure. Therefore, each partitioned data is replicated three times. Two replicas out of the three are placed in the same rack and the other one in outside rack. While the tasks are in progress on multiple nodes concurrently, failure in any node can be automatically detected. The failed tasks are restarted on other healthy nodes. The NameNode is expected to be aware of all the blocks. It finds out block location and informs the location to the DataNode once needed. Once a node fails, automatically, a copy of the missing data is created from available replicas. The major failure will be if the failure occurs in the only NameNode, as there is no replicated data here. Hadoop allows users to run executables as mappers and reducers. This is based on fault-tolerant, share–nothing architecture where tasks are independent of each other.

The Apache Hadoop open-source software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver highavailability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

3.2.3.5.5. Experimental Setup

A test bed with homogeneous cluster consisting of 5 nodes, each was used. In this case, nodes were connected to a Gigabit Ethernet switch and ran Fedora Linux (v 12 with Hadoop (v 0.20.2). One machine was selected as the master node and the rest were used as slave nodes. The Hadoop cluster was configured to use up to 300GB of hard drive space on each slave and 1GB memory was allocated for each Hadoop daemon.

3.2.4. A Parallel and Distributed Based Spam Classification

Computing similarity measures like distance, closeness, affinity, etc. between different samples are the ones which enable knowledge discovery, pattern recognition, etc. Similarity search is generally based on the distance comparison in sample datasets. Euclidean, Manhattan and Cosine distances are common similarity metrics which are used in many machine learning algorithms. The shortest distance between any two data points shows how strong or close their relationship is. There have been many similarity search algorithms (Agrawal et al. 1995). In data mining, similarity search is, typically, a single join query rather than depending on several similarity queries (Böhm et al. 2000). Similarity join is a join of a number of similarity queries together. The most commonly used similarity join is the distance range join *R*.

The *k-nearest neighbor join (kNN-join)* is a combination of the *k* nearest neighbor (*kNN*) query and the Join operation. kNN join operation combines each point of one dataset with its K-nearest neighbors in another dataset. This is widely used in many data mining and analytic applications. In the traditional setup performing kNN joins has been studied extensively in the literature. In such a setup, a single thread was used on a centralized server (C. Bohm and F. Krebs., 2004) (Yao et.al. 2010) The k-nearest neighbor join (kNN join) has been used in several applications involving knowledge discovery, data mining etc. (Yu et.al, 2010) (Plaku et.al. 2007). Gorder kNN-join algorithm (Xia et.al. 2004) was applied to reduce both I/O and CPU costs. This used a block nested loop join approach which exploited sorting, join scheduling and distance computation. (Lu et.al. 2012) investigated the performance of kNN join using the MapReduce framework for large data applications over the clusters of computers. (Zhang et.al. 2012) proposed parallel kNN joins using Hadoop for big data, in MapReduce framework.

This section deals with Map-Reduce based kNN Join and Classification process for large scale e-mail classification problem. In kNN Join and Classification, most intensive calculations are those where distances are calculated. If we have n number of training records, then each testing record will require n distance calculations. So, if m number of testing records were considered, complexity of distance computations will be of the order O(n*m). Once these distance values are available, computational overhead for retrieval of top K training records for every test record will be very low in comparison to the computational overhead that we mentioned for distance calculations. It is obvious that distance computations between one test record and all training records is independent from the distance computations between other test records and corresponding training records. Taking this fact into account, distance computations between different testing records and training records can be executed in parallel.

3.2.4.1. kNN Classification

k-Nearest Neighbors classification algorithm is also one among the the other best classification algorithms. It takes the input value K along with a training dataset and a testing record. The algorithm proceeds as follows: Firstly, the algorithm takes the test record and uses some similarity measure to find out individual distance of every record in the training dataset from the test record. Using these distance values, the algorithm comes out with top K training records which are most closest/nearest (distance wise) to the testing record, these are also referred to as K-Nearest Neighbors(kNN) of this record. Then, using class labels, distance values or some other measures/attributes of these kNN along with a voting scheme, class label of the test record is decided. This concludes the classification process for a single test record.

3.2.4.2. kNN Join

In kNN Join [13] process, formally, for with two given datasets *R* and *S* in R^d where R^d is a *d-dimensional* dataset. Each record, say *r*, that belongs to *R* and each record, say *s*, that belong to *S* can be interpreted as a d-dimensional point. Here, the similarity distance between any two records is their Euclidean distance *d(r, s)*. Naturally, *kNN(r, S)* will return a set of *k* nearest neighbors of record *r* from *S* where ties are broken arbitrarily. Figure 3.7 shows how a kNN Join operation gives a set of kNNs.

Similarly, for each record *r* in *R* we can obtain it's *kNN in S*. This whole process concludes the complete kNN Join process. So, in brief KNN Join can be described as:

kNN Join: The kNN Join kNNJ(R, S) of R and S is: kNNJ(R, S) = {(r, kNN(r, S))| for all r belongs to R}

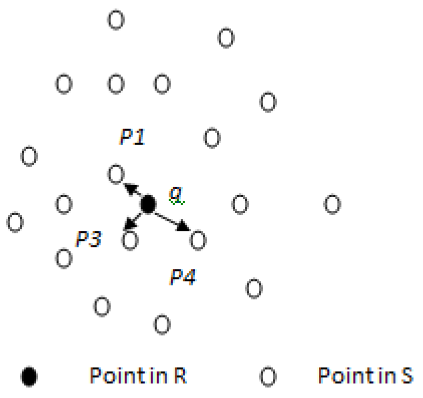


Fig 3.7. kNN Join Operation for K = 3

3.2.4.3. kNN Join and Classification on Map-Reduce

This section deals with the basic methodology employed to mould kNN Join and Classification process into Map-Reduce framework. Initially, two datasets were taken into consideration, one consisting of training records and another one consisting of testing records. For each record belonging to the testing dataset, the aim of the KNN Join and Classification process is to find its kNN in the training dataset and on the basis of these kNN assign a class label to that testing record.

3.2.4.3.1. Pre-Processing

Signature of the Pre-Processing function is *Pre-Processing (Dataset, R)*. The function treats the input dataset as R partitions and adds a partition number in the beginning of each record of the dataset indicating the partition number (virtually) of the dataset to which that record belongs to.

```
PreProcessing(Testing_Dataset, R)
{
        Integer Partition_Number
        Integer L
        Integer I,J
        L ← no. of records in the Testing_Dataset
        for(I = 1 ; I <= R ; I++ )
        {
                Partition_Number ← I
                for( J=I*(L/R) ; J <(I+1)*(L/R); J++)
                        {
   Testing_Dataset[Jth record] ← Partition_Number*Testing_Dataset[Jth record]
                (where '*' indicates concatenation)
                        }
        }
}
```

3.2.4.3.2. MAP Function

Signature of the Map function is *Map (key,value)*, i.e. the Map function receives a key-value pair. Here, the key is the record offset in the testing dataset and the corresponding value is the record string pre-appended with the partition number to which it belongs.

```
Map(record_offset, record_string)
    {
        Integer Partition_No
        String Record
    Partition_No ← Extract the partition number from the record_string
    Record ← Put the remaining record_string (without partition number)
        New_Key ← Partition_No
        New_Value ← Record
        return(New_key,New_Value)
    }
```

The output of the Map function is a new key-value pair as shown in figure 3.8, where the new key is the test record's partition number and the corresponding new value is the test record string without the partition number value, i.e. the original record string.

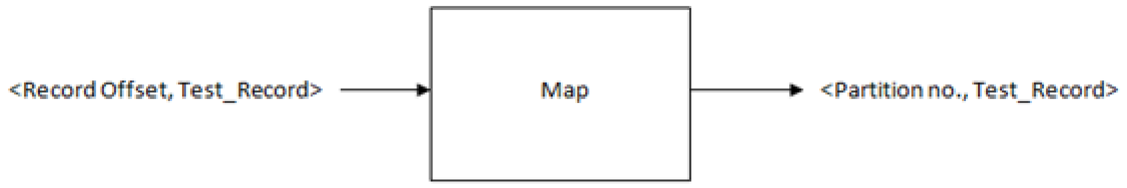


Figure 3.8. Map Function

3.2.4.3.3. Partitioner Function

Signature of the partitioner function is *Partitioner(key,value)*. The new key-value pair output from the Map function is received by the partitioner function where the key is represented by partition_number and the value is represented by record_string

```
Partitioner(partition_number,record_string)
{
    Integer P ← partition_number mod R
    return P
}
```

The output *P* of the Partitioner function returns the reducer number to which this key-value pair (received from Map function by the Partitioner function) must be sent to for further processing.

3.2.4.3.4. Reduce Function

Signature of the Reduce function is *Reduce(Key,Value)*. The reduce function receives a key-value pair where the key is the partition number to which the test e-mail record belongs to and the corresponding value is the test e-mail record string itself. The key, which indicates the partition number to which the test e-mail record should belong to, have done it's part once the record reaches to its reducer, hence is no more required and is discarded in the reduce function.

```
Reduce(partition_number, record_string)
{
        String test_email
        test_email ← record_string
```

Below, *T* represents the Training Dataset of e-mail records. *Distance_array* is an array of double values that will store the distance of each training e-mail record from the test_email. *euc_dist(a,b)* is a function that finds the Euclidean distance between the records a and b.

```
        Integer L
        Integer I
        Integer J
        L ← length of T
        For( I= 1; I < = L ; I++)
        {
Distance_array[I] ← euc_dist(test-email, Ith record of T)
        }
```

TopKRecords is an array of String values that will be containing K-nearest neighbor training e-mail records of test_email from T along with their distances from test-email.

```
For(I=1; I<= K ; I++)
        {
J<- Check Distance_Array for the Ith smallest value and  retrieve its index
TopKRecords[I] ← Jth record of T + (String)Distance_array[J]
        }
        Double MAX
```

```
          Double MIN
          Double Distance
     MAX ← Retrieve the distance value of the Kth neighbor of test_email
     MIN ← Retrieve the distance value of the 1st neighbor of test-email
     Neighbor_weight is an array of Double values that will store the weight of each
     neighbor of test_email for weighted voting scheme
     For(I=1; I<= K; I++)
     {
      Distance ← Extract distance value from TopKRecords[I]
      Neighbor_weight[I] ← (MAX – Distance)/(MAX-MIN)
     }
     String Class
     String Neighbor_class
     Double Record_weight
     Integer Count ← 1
```

Distinct_class is an array of Strings which will contain distinct class labels coming from K-nearest neighbors out of which one is to be assigned to the test_email.

Distinct_class_weight is an array of Double values which will contain the total weight of each class label coming from K-nearest neighbors on the basis of which the class label will be assigned to the test_email.

```
     For(I=1 ; I <= K ; I++)
     {
          Neighbor_class ← Retrieve the class of TopKRecords[I]

        If(Neighbor_class do not exist in Distinct_class)
               {
                    Distinct_class[Count] ← Neighbor_class
                    Distinct_class_weight ← Neighbor_weight[I]
                    Count ← Count+1
               }
          Else
               {
      J ← Retrieve index of the Neighbor_class in  Distinct_class
          Distinct_class_weight ← Neighbor_weight[I]
                    }
          }
     J ← Retrieve the index of maximum weight from the  Distinct_class_weight

          Class ← Distinct_classs[J]
          String New_key
          String New_value
          New_Key ← test-email
```

```
New_Value ← (Each record from TopKRecords in order 1 to K) + Class
        return(New_key, New_Value)
}
```

The output of the Reduce function is a new key-value pair as shown in figure 3.9. The new key is the test e-mail record itself. The new value consists of each of the K-nearest neighbors along with their individual distances from the test e-mail record followed by the class label assigned to the test e-mail record on the basis of a weighted voting scheme, which is the predicted class label for it.

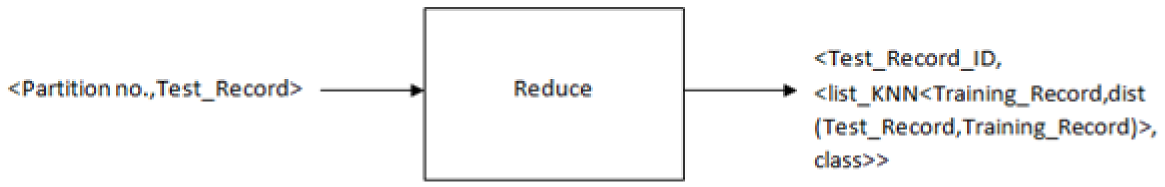

<Partition no.,Test_Record>
Reduce
<Test_Record_ID,
<list_KNN<Training_Record,dist
(Test_Record,Training_Record)>,
class>>

Figure 3.9. Reduce Function

3.2.4.4. Design Model

k-Nearest Neighbors classification algorithm is one of the most well known and commonly used classification methods. It takes the input value *k* along with a training dataset and a testing record. The algorithm proceeds as follows: Firstly, the algorithm takes the test record and uses some similarity measure to find out individual distance of every record in the training dataset from the test record. Using these distance values, the algorithm comes out with top *k* training records which are most closest/nearest (distance wise) to the testing record, these are also referred to as k-Nearest Neighbors(kNN) of this record. Then, using class labels, distance values or some other measures/attributes of these kNN along with a voting scheme, class label of the test record is decided. This concludes the classification process for a single test record.

3.2.4.5. Experimental Tools/Instruments

The experimental tools and datasets specific to this part of experimentation have been already presented in the earlier discussions. In this case, one large public data sets known as the Enron Corpus were used.

3.2.4.5.1. Enron Corpus

The Enron Corpus presented in section 3.2.2.5.1 in detail was used here as well for the same reasons as described initially.

CHAPTER 4
STRATEGIES AND ALGORITHMS

4.1 Introduction

This chapter gives a detailed description of the algorithms discussed in Chapter 3 along with the evaluation measure like Accuracy, Spam Recall and Spam Precision and their comparison with other prominent algorithms known in relevant literature.

4.2 Bayesian and Neural Network Classifier (BNNC)

Centered Bayesian networks and neural networks were combined together to classify an E-mail.

As already stated, BNNC is made up of a set of CBNs and NNs. Each of these CBNs can be used to represent and learn specific information about the E-mail. The BNNC (Manjusha et.al 2010) system architecture can be viewed in Figure 4.1. Here, separate CBNs for representing E-mail's From address, From name/ID, CC have been used. The other information comes from both the neural networks of subject and content of E-mail. The final decision is made by the decision box which is in layer 0. In the first stage the external nodes receive stimulus from the incoming E-mail. Depending on the CPDs of the Layer 1 internal nodes and the active strength of the external nodes the active strength of the Layer 1 internal nodes is decided. CPDs of the various CBNs would be different depending on the type of the data dealt with. For example, CPD of the CBN concerned with the From Address would follow an OR relationship. So if any of the external nodes of this CBN is set to active High i.e. 1 then that E-mail has to be considered as SPAM. Therefore, the active strength of the Layer 1 internal node would be set to 1. Likewise, CPDs of CBNs concerned with From Name and CC Address would follow an OR relationship for determining the state of their internal nodes. But, the CPDs for the Subject and the content of the E-mail would be much more complex. Therefore, two different Neural Networks for Body and Subject of an E-mail NN1 and NN2 were chosen and trained using Genetic Algorithm as presented earlier.

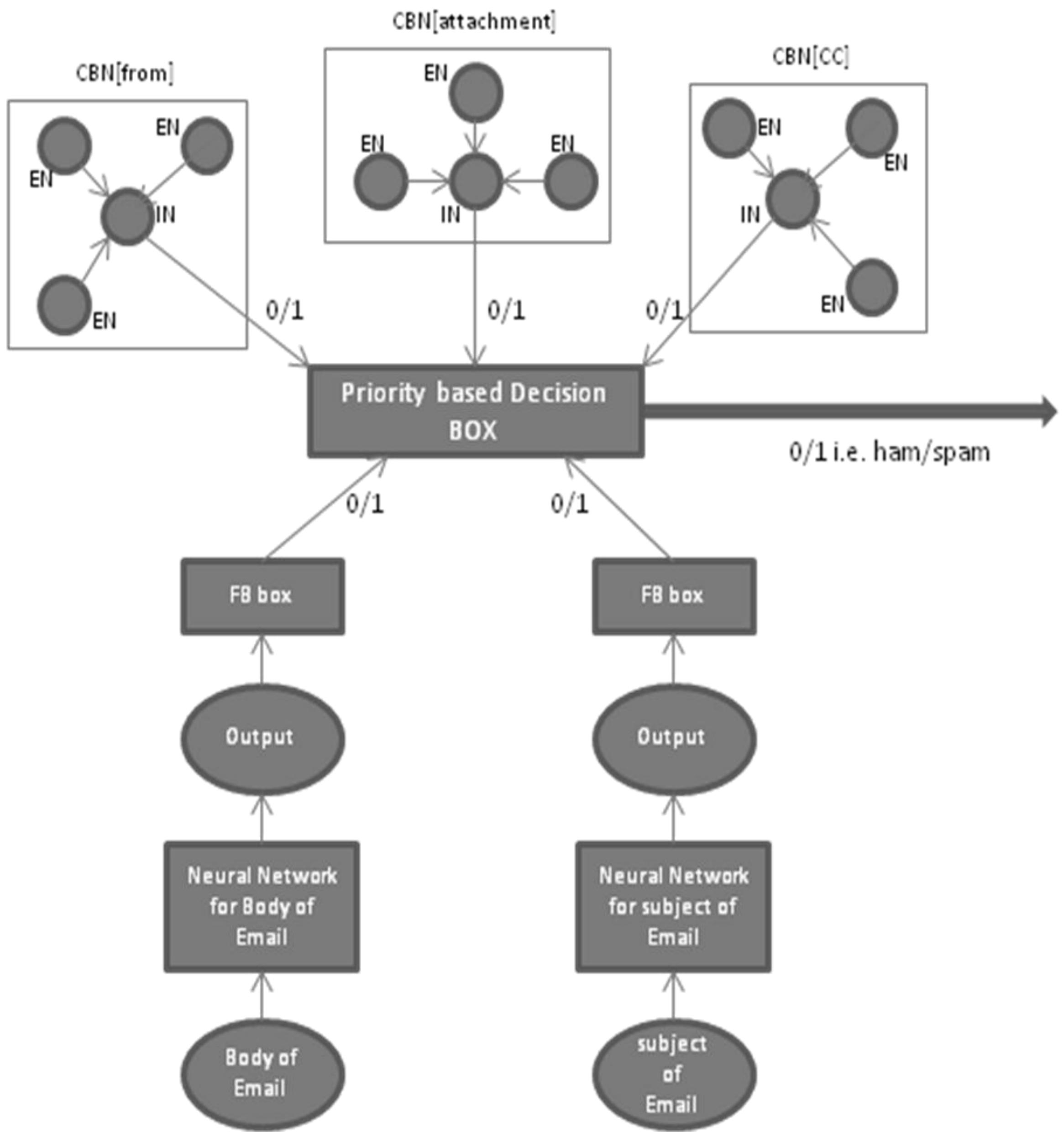


Figure 4.1. Bayesian and Neural Network classifier (BNNC)
[CBN - Centred Bayesian Network, EN - External Nodes, IN - Internal Nodes, FB - Fraction to Binary]

Body and Subject are being fed to two different Neural Networks which are trained using a Genetic algorithm. From address, CC list and file attachment information is being fed to Centered Bayesian Networks with their CPDs defined. The decision to decide which content of E-mail, subject should go to Neural Network and their respective from name, address and cc should go to which Bayesian network is based on deterministic and non-deterministic properties of this networks, the kind of information they are best able to process and the way they do learning. The output from these two Neural Networks and three Centered Bayesian Networks will finally classify the E-mail based on the decision box which is configurable by the server administrator.

4.2.1 Verification of Correctness

As mentioned in the earlier chapter, in the first of four data sets used, the corpus comprised of a collection of 2000 E-mails collected from scholar’s personal inbox is used for the purpose of this research. The model consists of separate networks for dealing with different parts of an E-mail like sender address, sender name, and CC details are dealt with separate CBNs and content, subject details are to be dealt by NNs. So the E-mail data was split into different parts and fed to their respective networks.

The classification of e-mail is mainly of three steps: data preprocessing, training the classifier and testing it. In data preprocessing WEKA (Hall, Mark., 2009) String To Word-Vector filter was used to convert the data into a new data set with word frequency to each word and finally to identify the most relevant word features within the given data by removing irrelevant or redundant features. To avoid over fitting 10-fold cross-validation method was used. By this method, all the data will be utilized for model evaluation. The data were divided into ten equal sized partitions, each of which was in turn used as an independent test set, while the other partitions were used for training the classifier. There will be a total of ten training and testing sets.

The classification error is the result of average performance over these ten test sets. As a last step testing was performed and the performance was measured by evaluating the false positive rate, false negative rate, precision and recall. For classification *weka.classifiers.meta.vote* was used for combining three CBNs (sender ID, Sender Name, CC) and two NNs (Subject, Content).

Table 4.1 Performance Measures Over 10 Stratified Fold-Cross Validation

	TP Rate	FP Rate	Precision	Recall	F-Measure
	1	1	0.99	1	0.995
	0	0	0	0	0
Weighted Avg.	0.99	0.99	0.98	0.99	0.985

These performances of BNNC were compared with a simple Bayesian classification network where the subject and the content of the E-mail were also considered along with the header data. Table 4.1 gives the performance measures over 10 stratified fold classification. It was proven better false positive rate and false negative rate. Table 4.2 is a final decision table where ASIN stands for active strength of internal node, PBDB stands for the priority based decision box.

Table 4.2. Final Decision Table

FB [Content] output	FB [Subject] Output	ASIN CBN [From Address]	ASIN CBN [CC]	ASIN CBN [From Name]	PBD output
1	X	X	X	X	1
0	1	X	X	X	1
0	0	1	X	X	1
0	0	0	1	X	1
0	0	0	0	X	X

4.3. Binary Decision Tree – Multi Class Support Vector Machine (BDT-MSVM)

After analyzing the performance of SVM and decision trees and trying to overcome the problems of both the classifiers. Figure 4.2 gives the BDT-MSVM (Manjusha et. al. 2013) architecture. In this approach both the advantages of SVM and decision tree were considered, i.e. DTs are much faster than SVMs in classifying new instances while SVM performs better then DTs in terms of classification accuracy. To include both these advantages the size of record set was reduced which will be fed to the SVM. Initially after preprocessing the total dataset is fed to the decision tree classifier to classify the records after creating the model based on the training data.

Decision Tree classifies the test record into their respective classes. But the classification confidence of several records by the decision tree is much lower than expected. Such records, if classified using some other classifier, may generate some other output. As a result, such records (for further processing) whose classification confidence is less than some threshold value were filtered. It amounted to less than 20% of the entire data set.

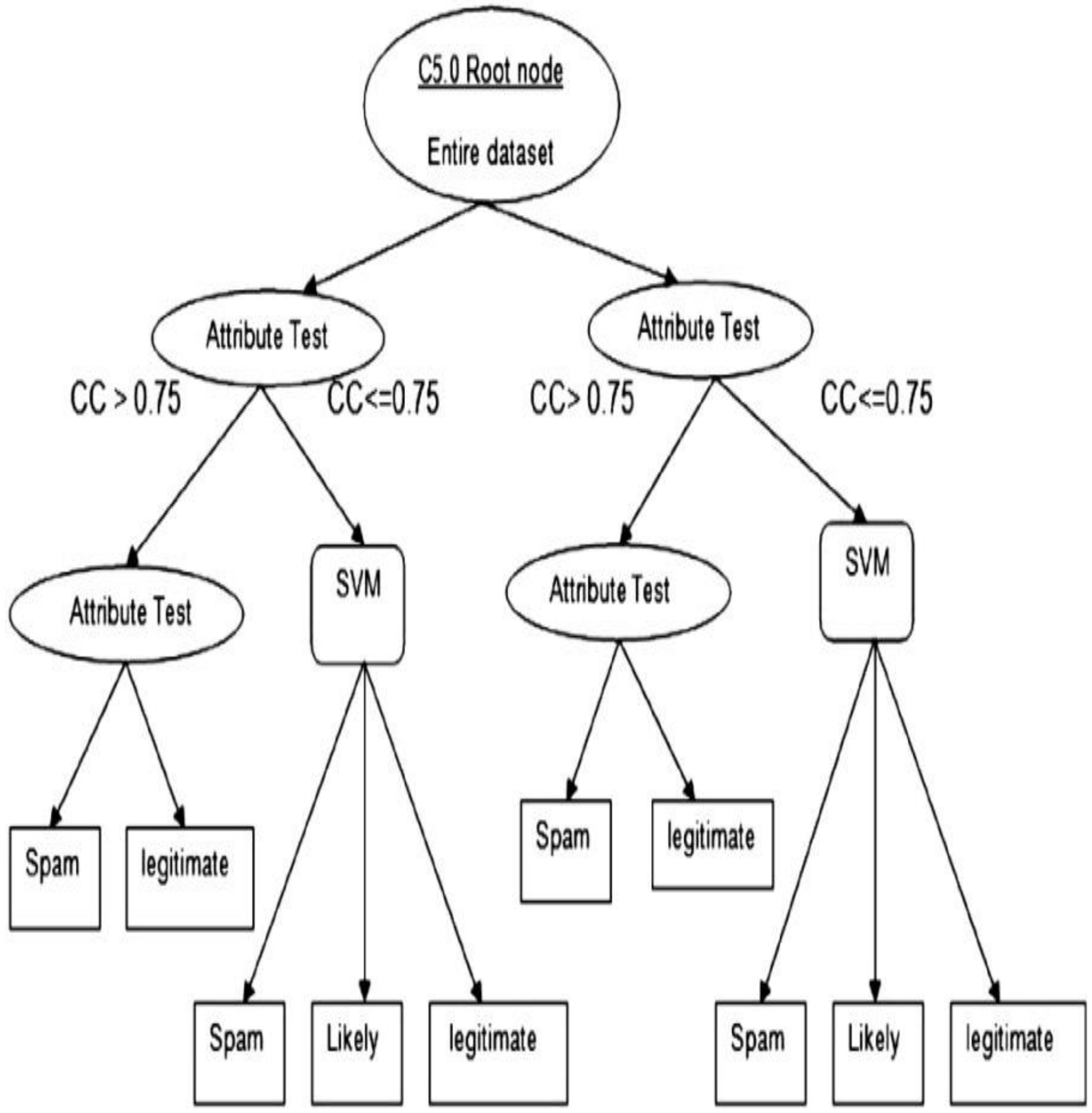


Figure 4.2 BDT-MSVM Architecture

These records are classified using SVM to classify such records precisely. This work comprises of a hybrid classifier technique employed for e-mail classification. The main focus here is to classify the e-mails efficiently into multiple classes (spam, not spam, likelihood). Basically, most spam classifier is binary classifies. The idea of probable spam class is used because after analyzing some e-mails classified by binary classifiers, it was found that many records are easily misclassified and as a result there are many false positives. Using a multiclass SVM classifier, an attempt was made to classify the records to classify spam, legitimate and likelihood class in order to reduce False Positives. This helps to point out the records which are classified as likelihood to analyze further for better accuracy and to avoid classifying legitimate e-mails as spam and vice versa. In this part of the work, for the aforementioned reasons, Enron data set for spam filtering operation were used. It consists of more than 5000 e-mails which are labeled into two classes: spam and ham. First the decision tree was trained with the whole dataset. This found two regions corresponding to spam and legitimate class. But there were some records which occurred very near to the decision boundary. Decision tree may fail to classify such e-mails with much accuracy. The record whose classification confidence is less than the threshold value is classified into a class “x”. Typically, the threshold value is kept in the range of 0.65–0.90. In this BDT-MSVM approach, each class x leaf is replaced by a sub tree with multiclass SVM as the root and 3 leaf nodes (spam, legitimate, likelihood). Since the number of records is less and SVM classifies records more accurately than decision tree, the choice of using SVM here works. Since the application of this work is in classifying records into multiclass, a multi-class SVM approach was used to classify the selected records. Multiclass SVM classifies e-mails into more than two classes efficiently. If the entire e-mail was considered for the SVM classification, it will increase the time complexity as the number of support vectors will be large.

4.3.1. Verification of Correctness

One of the six public, large and well-known Enron Corpus (Klimt, Bryan, and Yiming Yang., 2004) was used in the experiments. Enron data set, consists of more than 5000 attributes (all fields are continuous except one nominal class attribute). The motive of this part of the work is to classify the e-mails into appropriate class by finding the percentage of words in each e-mail that matches the attribute name. The nominal attribute field of the dataset specifies the mail as spam or not spam. As indicated earlier as well, the Enron dataset contains 5701 e-mail messages in the raw format, which contain various sections like to, cc, bcc, and message body. In order to get efficiency in the classification of e-mails, and also to reduce unnecessary processing time, preprocessing was done on the raw dataset by various steps, like removing stop words (is, the, that, etc), stemming. After this, feature selection was done to further reduce the number of features. *(Feature selection is the process of selecting a subset of relevant features for processing in model building.)* After preprocessing and feature selection, the features were reduced to 10000 relevant attributes. In order to analyze the performance difference of SVM and Decision Tree we classified Enron dataset with SVM first then next with Decision Tree. As expected, due to the distinct characteristics of both the classifiers, in Table 4.3, SVM classified the records with more accuracy than Decision Tree but the time taken by SVM was considerably higher than that of Decision Tree.

From the initial analysis of using Decision tree separately it was found that the classifier is not absolutely sure in classifying some records, especially when the confidence level falls below 60–70%. Some records which are classified as spam by the decision tree classifier is not categorized as spam by the other classifier. Precisely, it happens because the generalization error rate of decision tree is more than that of some other classifiers. In the decision tree as the tree was built more complex or big in order to make it fit to the training set model over-fitting occurs. Once model over-fitting happen the tree will satisfy all the training records in classifying, but will fail against previously unseen test record. An important point which had to be considered with respect to SVM is that, SVM is slower than a decision tree in the testing phase. As the number of training data increases, then the number of support vector also increases. If the number of support vector is large SVM takes longer time duration to classify a new data point.

Table 4.3. BDT-MSVM Performance Accuracy Comparison.

TP Rate		**FP Rate**	**Precision**	**Recall**	**F-Measure**
1		1	0.99	1	0.995
0		0	0	0	0
Weighted Average	0.99	0.99	0.98	0.99	0.985

As an attempt to construct an efficient spam filter the proposed work is a hybrid approach of SVM and Decision tree to classify e-mails into multiple classes – Spam, legitimate, Likelihood. Motive of using Likelihood class is to reduce the number of false positive cases. The reason for using a hybrid approach rather than using already existing efficient classifier is mentioned in the background. Here the initial data set, Enron, is partitioned into training and testing set in 7:3 ratios. After that the records are fed into decision tree classifier to create the model (decision tree) and classify the records. From the output of decision tree, it is clear that many records are classified accurately by decision tree with high level of confidence value and some of them were classified with less confidence value. The records whose estimated confidence is less than 70% should be considered for further processing in order to predict the class accurately. For this reason, such records were filtered with low confidence value and stored them separately, which are classified into multiple classes by SVM classifier. The remaining records are classified as Spam or Not Spam by the decision tree.

4.3.2 Comparison with Other Prominent Algorithms of Relevance

The Table 4.3 shows the performance of BDT-MSVM hybrid approach. From the result one can understand that it performed better than what would have happened if SVM or decision tree was used alone. In addition to better accuracy in terms of prediction DT- SVM approach reduced the overall time taken to classify the records. In addition to simple SVM and decision tree different other algorithms were classified using the same test records with K-NN, Bayesian.

Table 4.4 shows their performance. Figure 4.3 and Figure 4.4. show the accuracy and training time comparisons in a graph form. kNN which uses a lazy learner approach to classify the records uses the relative similarity between the attributes of training and test records to classify the test records.

In Bayesian approach classification is done by modeling the probabilistic relationships between attribute set and class variables by using Bayes theorem.

Table 4.4. Comparison of Efficiency, Computational Time of Various Classifiers.

Classifier/Result	K-NN	Bayesian	Decision Tree	SVM	DT-SVM
Correct Classification	88.1%	76.27%	88.84%	91.84%	96.30%
Wrong Classification	19.9%	23.73%	11.16%	8.16%	3.70%
Total E-mails	5171	5171	5171	5171	5171
Training Time (sec)	26 Sec	29 Sec	25 Sec	44 Sec	30 Sec
Total CPU Time (sec)	70 Sec	105 Sec	45 Sec	98 Sec	63 Sec

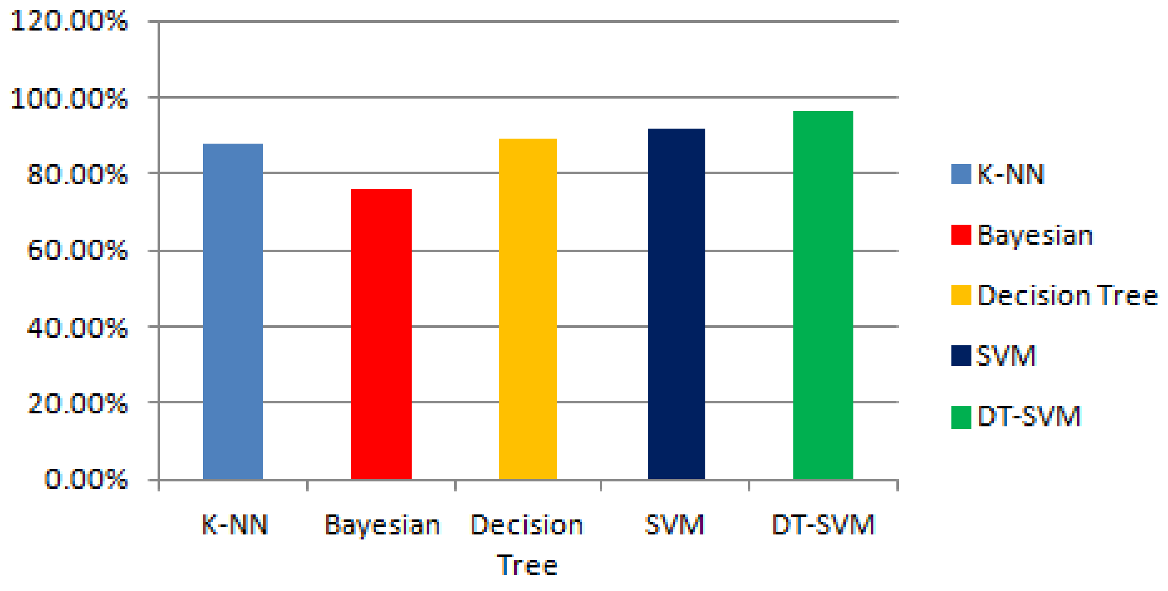


Figure 4.3. Accuracy Comparison with Different Algorithms

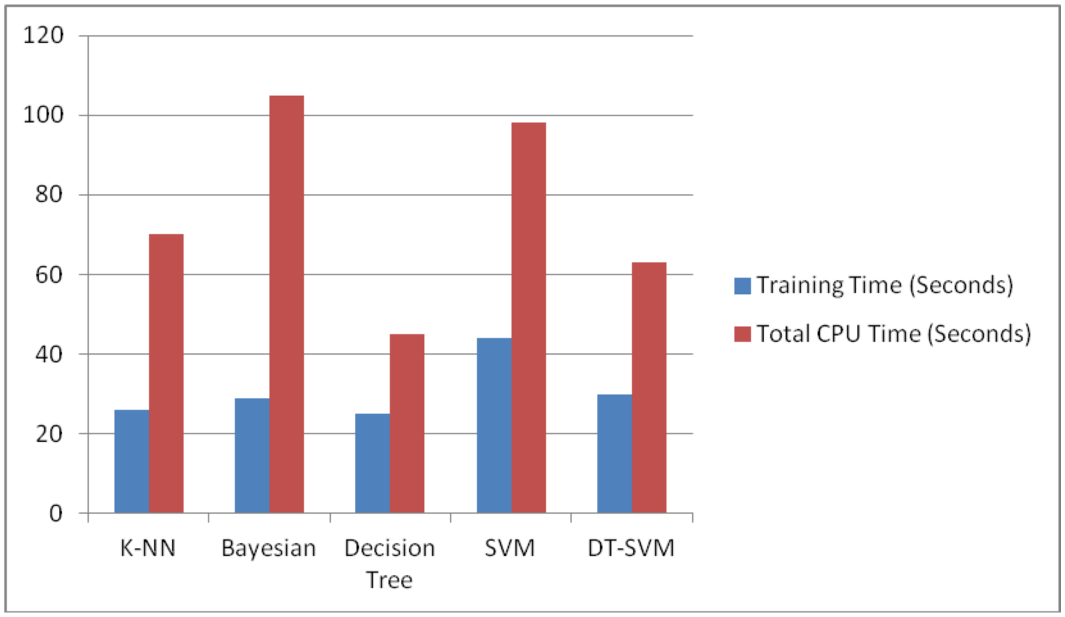


Figure 4.4. Time Comparison with Different Algorithms

4.4.The Ensemble-based SVM Spam-mail Classifier (*The ES^{2}C Technique*)

The Process of Spam Filtering can easily be parallelized using the MapReduce programming framework and by adapting this method SVM training performance can be improved. Apache's MapReduce Implementation Hadoop (Shvachko, Konstantin et. al. 2010) is highly suitable to handle large chunks of data in parallel. A large E-mail dataset can be used in Hadoop and training process for building models will become significantly faster as models will be built in parallel over subsets of original spam dataset. The proposed approach follows an ensemble based method for E-mail filtering using SVM as a classification technique. The parallel programming environment provided by Hadoop proves suitable for this approach where dataset is divided into m data chunks. Each data chunk will lead to the creation of local models which are not globally optimal, but they are locally optimal as per the local data chunk. These local models can further be combined to create a globally optimal set of support vectors. A similar approach of building global model (Caruana, et.al. 2011) is described wherein after the creation of a global model, local models were never utilized. Our proposed work utilizes those local models to form an ensemble and ensemble will accept all local models as well as the global model as input for making prediction.

The ES^{2}C Technique involved a formal analysis of the models that generated with the MapReduce based binary SVM. Figure 4.5 gives the The ES2C Technique architecture. The whole training dataset was distributed over data nodes of cloud computing system. At each node, subset of training dataset is used for training to find out a binary classifier function. The algorithm collects support vectors (SVs) from every node in a cloud computing system, and then merges all SVs to save as global SVs. LIBSVM (Chang, Chih-Chung, and Chih-Jen Lin., 2011) was used to build the algorithm and implemented using the Hadoop implementation of MapReduce.

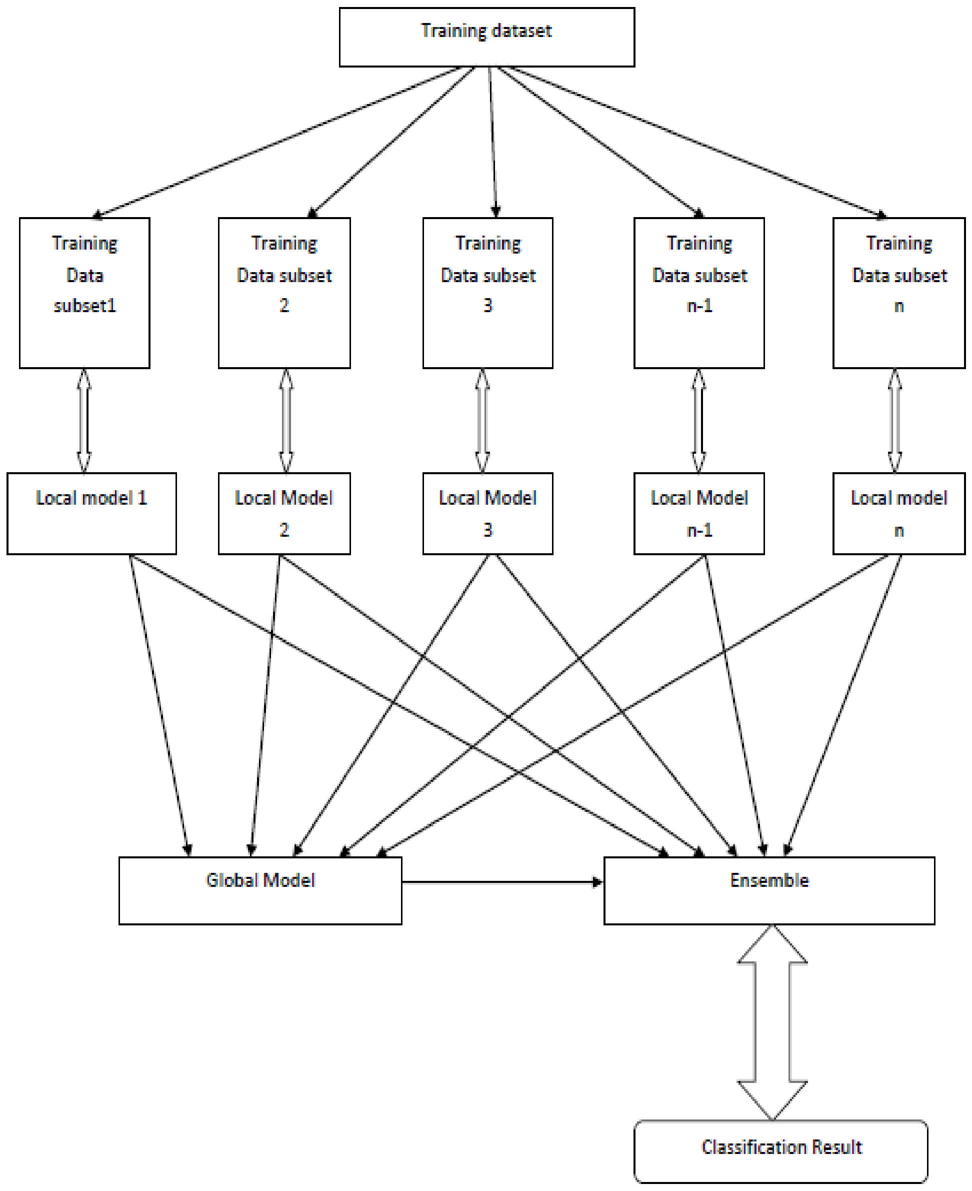


Figure 4.5. The Ensemble-based SVM Spam-mail Classifier architecture (The ES^{2}C Technique)

4.4.1. Verification of Correctness

An Inverted Index approach was used to convert each E-mail into a feature vector. The presence and absence of a particular keyword was not taken into consideration, but converted the frequency of the keyword into a weight in the range [-1, 1]. LIBSVM suggests to keep the attribute value for each dimension in the range [-1,1]. Radial basis function (RBF) was adopted which is a non-linear kernel.

$$k(x_i, x_j) = exp\left(-\|x_i - x_j\|^2/2\sigma^2\right)$$

LIBSVM also requires some runtime parameter which helps in improving the accuracy over the use of default parameters. The reason for parameter's selection in LIBSVM is well explained in Chih-Wei Hsu (2002). In this research work selecting a proper constraint violation cost and gamma parameter to the kernel can produce good and valid accuracy. Such parameter was selected using a very tiny part of our dataset, the reason being the ineffectiveness of size by selecting the best parameters while performing cross validation. The parameters generated through cross-validation were found to be appropriate for the whole dataset. Some of the parameters selected in this case were S (type of SVM), V (number of V-fold cross validation), g (value of gamma in kernel function) etc. A conclusion was drawn while implementing this system that the linear kernel also fits for document classification if the number of features is large in comparison to the number of feature vectors. We have also used a probability estimate defined in LIBSVM which improves the accuracy.

The system divides the complete dataset into a number of subsets using random sub sampling. Each such subset will lead to the creation of a local model which is locally optimal. The local models together create a global model and at the last an ensemble is applied to the predicted outputs. Bagging technique was adopted in this work to construct the ensemble. Each individual SVM is trained independently via a bootstrap method and then they are aggregated via an appropriate combination technique. For a given dataset *TR* $((x_i;y_i)|i=1,2,\ldots,l\}$, k replicated training datasets $\{TR_{bootstrap}|k=1,2,\ldots,K\}$ are first randomly generated by bootstrapping technique with replacement. Next, SVM is applied for each bootstrap datasets. Finally, they are aggregated to make a collective decision by the majority voting. Majority Voting Majority voting is the simplest method for combining several SVMs. Majority voting selects the class label that achieves the highest number of vote from each SVM model. Let $f_k(k = 1, 2, . . .,K)$ be a decision function of the kth SVM in the SVM ensemble and $C_j (j = 1, 2, . . ., C)$ denotes a label of the j-th class. Then, let $N_j = \# \{ k|f_k(x) = C_j \}$, i.e. the number of SVMs whose decisions are known to the jth class. Then, the final decision of the SVM ensemble fmv(x) for a given test vector x due to the majority voting is determined by

$$f_{mv}(x) = argmax_j N_j$$

Zhao et.al. (2012) showed exactly how the bagging technique is suitable for the random subsamples. Breiman (1996) pointed instability as the key element for Bagging and also proved that Bagging can improve accuracy. One efficient change might involve the use of boosting by providing more weight to global model and equal weight to local models. Both parallel as well as a sequential version was implemented in this work. The MapReduce version takes very less training model time in comparison to the single large -machine (centralized) version. The accuracy (with respect to the default parameter) for LibSVM was found to be on lower side. Table 4.5 gives comparison of accuracies of all the local and ensemble models for different datasets. The Figure 4.6 presents all datasets local models and their corresponding accuracy.

Table 4.5 All Datasets Local Models and Corresponding Accuracy

Corpus	**Model 1**	**Model 2**	**Model 3**	**Model 4**	**Model 5**	**Ensemble**
ENRON	97.3214	97.172	95.83	96.42	96.42	96.875
	654/672	653/672	644/672	648/672	648/672	651/672
Lingspam	98.33	98.83	98.67	99	98.67	98.67
	592/602	595/602	594/602	596/602	594/602	594/602

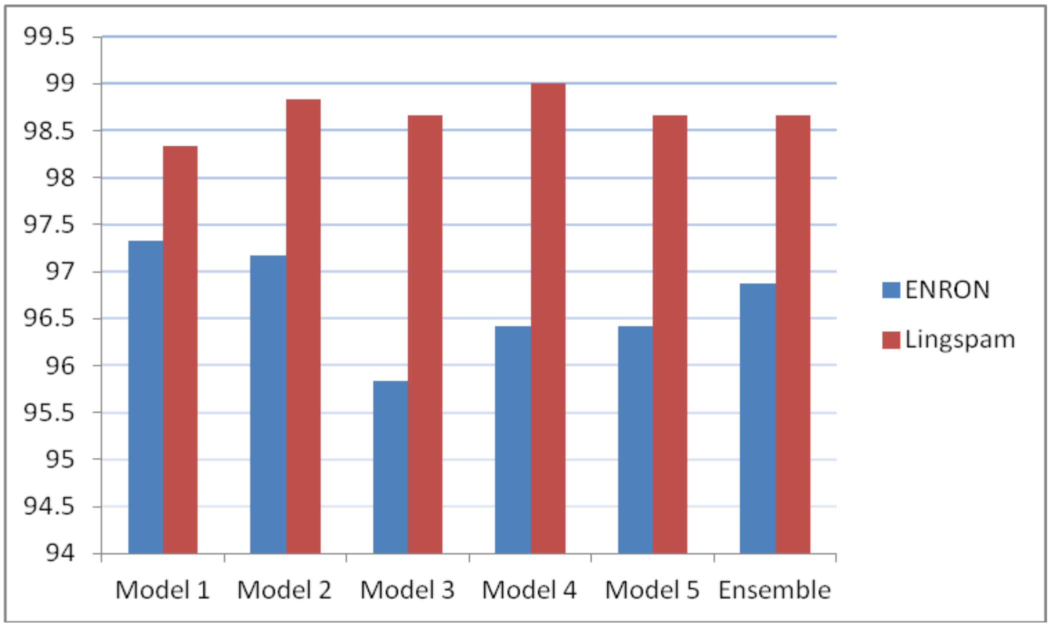


Figure 4.6. All Datasets Local Models and Corresponding Accuracy

4.5. SpamReduce : kNN Join and Classification on Map-Reduce

kNN Join in mapreduce is simple and straightforward method and follows these following steps. Firstly the datasets R and S are partitioned into n equal sized partitions by a linear scan before the map phase. So |R|\n records (|S|\n) were put in one block. According to the number of partitions of both R and S number of reducers will be invoked. In the mapper phase a key is to each object from R and S, in the shuffling phase the objects with the same key are alloted to the same reducer. The reducer performs the block nested loop kNN join over the objects of both R and S within the reducer (Zhang et. al. 2011). So in the reducer, for each object r in R, it finds kNNs of r in S using a nested loop, and the final results from all the reducers will be stored in a file. After collecting the distances a weighted voting scheme is applied Gou et al. (2011). In this voting scheme, for every distance between the elements and its nearest neighbors, it determines the weight of each neighbor. Based on these weights, one can determine the final class of the query element.

The Map-Reduce approach for spam classification have different steps, firstly the testing dataset that consist of the e-mails records (each e-mail record in its text vector form) to be classified is first partitioned into n equal size blocks. This is done by pre-appending a partition number at the beginning of each test e-mail record in testing dataset. All the test e-mail records that are supposed to belong to the same partition is pre-appended with the same partition number. For example, if the testing dataset have 1000 e-mail records and is required to be partitioned into 4 equal size partitions, then e-mail record number 1 to 250 is assigned partition number 1, record number 251 to 500 is assigned partition number 2 and so on. This procedure has to be carried out before the map phase. It, in a way, is a pre-processing step in the whole process, which can be easily done through a linear scan of the testing dataset. The pre-processed data now resides on the distributed file system from where these test e-mail records are sent to different mappers. In the map phase, a map function receives a test e-mail record where record's offset in the testing dataset is treated as key and the corresponding e-mail record along with its partition number is treated as a value.

The map function separates out the partition number from the record and creates a new key-value pair as its output where the new key is the partition number of the record and the new value is the e-mail record (without partition number) itself. This output from the map function is received by a partitioner function on that mapper. On the basis of the key of the key-value pair, the partitioner function assigns a particular reducer to this key-value pair. Hence, all those key-value pairs (which are basically partition number - test e-mail record pairs) that have the same key are assigned to the same reducer by the partitioner function of every mapper. So, all those test e-mail records that belong to the same partition reach the same reducer. Moving onto the reduce phase now, each reducer is initialized with a complete training e-mail dataset. The reducer calls the reduce function for each test e-mail record it received in the form of partition number (key) and record (value) pair. The reduce function discards the partition number, and keeps the e-mail record for processing. In the reduce function, the distance between the received testing record and each training record in the training dataset is calculated. On the basis of these distances, K-nearest neighbors of the test e-mail record in the training e-mail dataset are obtained. Weighted voting scheme is applied for every distance between the elements and its nearest neighbors from the test record, which calculates the weight of each neighbor. Based on these weights, the final class of the E-mail is determined.

This Concludes the kNN Join and Classification process. Figure 4.7 presents the kNN Join MapReduce framework. The output of the reducer for this test e-mail record is a new key-value pair, where the new key is the test e-mail record itself and the new value consists of 1) a list of k-nearest neighbors of this test e-mail record (along with their distance value from this test record) and 2) and the class label assigned to this test e-mail record.

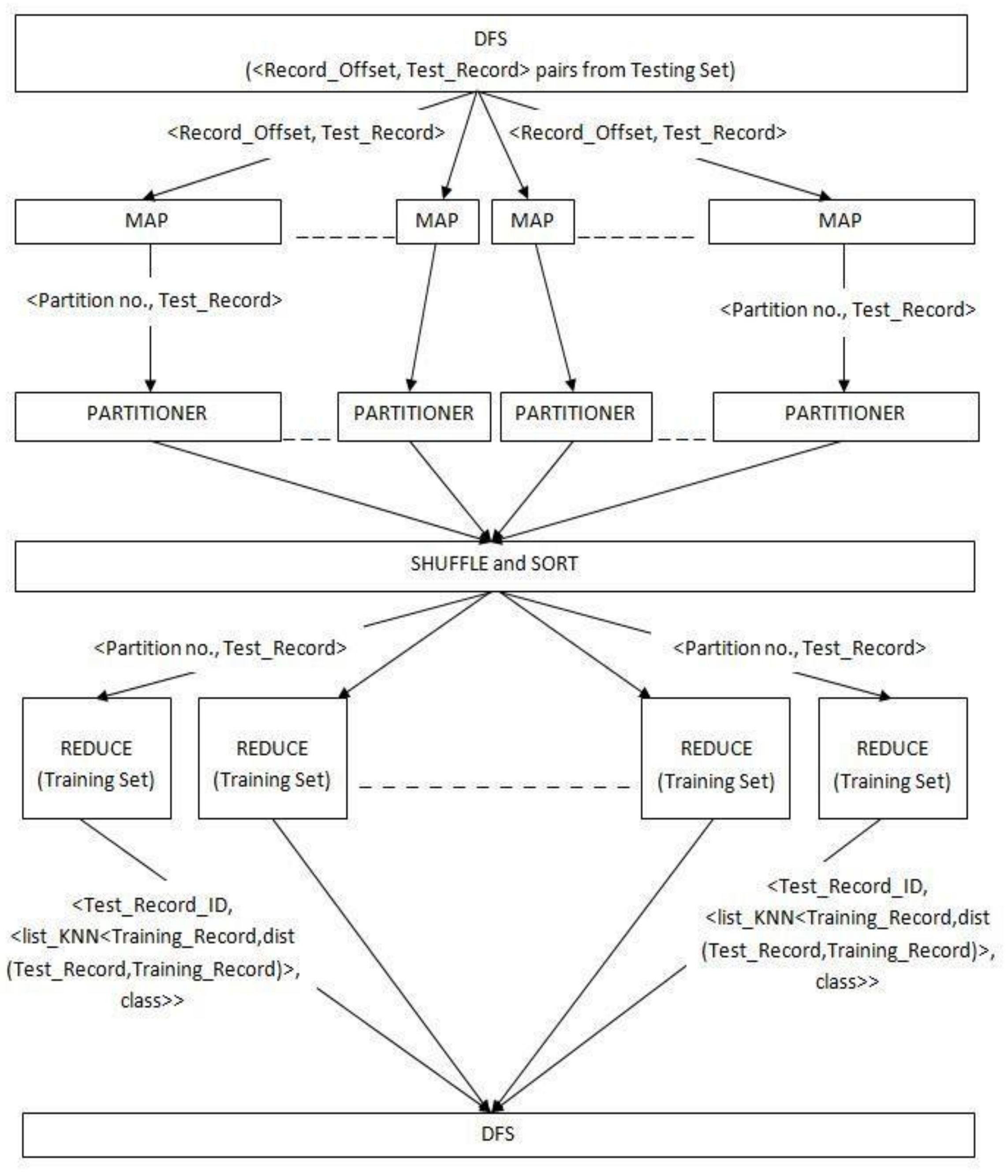

DFS
(<Record_Offset, Test_Record> pairs from Testing Set)
<Record_Offset, Test_Record>
<Record_Offset, Test_Record>
MAP
MAP
MAP
MAP
<Partition no., Test_Record>
<Partition no., Test_Record>
PARTITIONER
PARTITIONER
PARTITIONER
PARTITIONER
SHUFFLE and SORT
<Partition no., Test_Record>
<Partition no., Test_Record>
REDUCE (Training Set)
REDUCE (Training Set)
REDUCE (Training Set)
REDUCE (Training Set)
<Test_Record_ID, <list_KNN<Training_Record,dist (Test_Record,Training_Record)>, class>>
<Test_Record_ID, <list_KNN<Training_Record,dist (Test_Record,Training_Record)>, class>>
DFS

Figure 4.7. kNNJoin MapReduce Framework

All the reducer outputs are then again written back onto the distributed file system from where they can be readily accessed. The key point here is the purpose of partitioning. In this stage, the number of reducers used in the process is governed by the number of partitions of the test dataset defined by the user in the beginning. The more is the number of partitions, more are the number of reducers and hence more is the distributedness achieved in the procedure

4.5.1 Performance Analysis

Enron dataset was used for analysis. This dataset consists of 5172 e-mail records, each e-mail record classified as either a spam or a ham. A self developed program was used to split the given Enron dataset into two separate datasets, one for the training and one for the testing. This program goes through the dataset and picks one record for the test datasets for every three records picked for the training dataset. As a result, 25% of the Enron dataset is used for testing and 75% remaining for the training purpose. Also, to check the accuracy of the classification of the test e-mail records, this program compares the predicted class of each test e-mail record with its actual class and reports the result. This program also reports the number of false positives produced in the results. Table 4.6 and Figure 4.8 gives details of accuracy for different values of *k*.

Table 4.6 Accuracy for Different Values Of K

Result Table		Accuracy For Different Values Of K (in Percent)		
		K=3	K=5	K=7
Number of Features Selected in the Dataset	100	90.09	90.71	90.71
	500	90.78	90.55	90.94
	1000	90.32	90.63	90.63
	5000	90.40	90.86	91.17
	10000	91.09	91.48	91.25
	45750	90.63	90.47	90.78

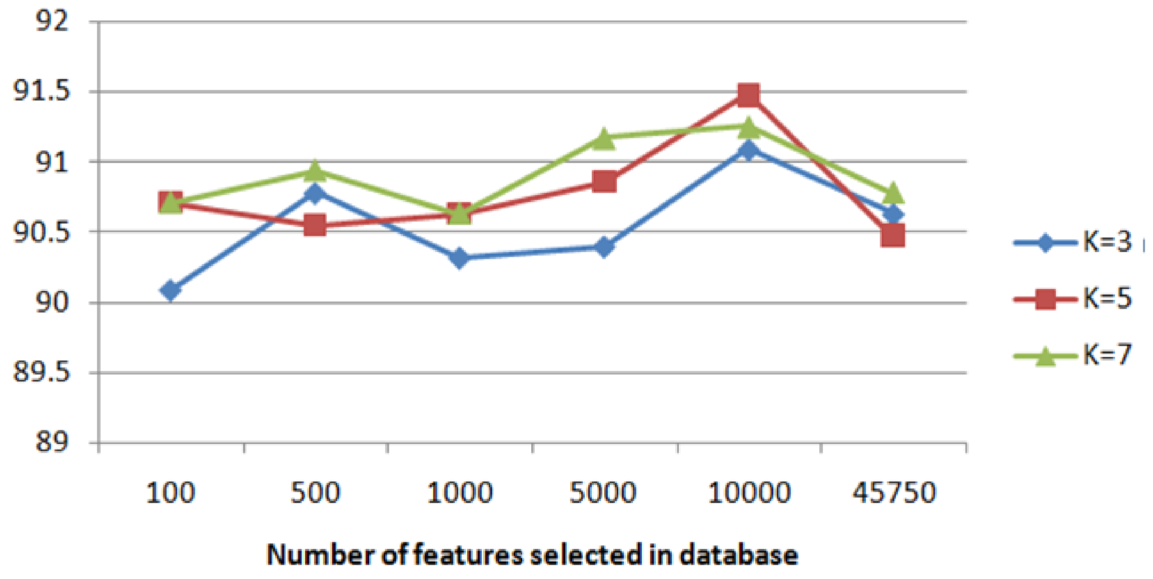


Figure 4.8. Number of Features with Corresponding Accuracy

CHAPTER 5
CONCLUSIONS

5.1 Introduction

Electronic mail is a marvelous tool of business and personal communication. As it is simple, accessible, easy to use, it has become more and more an indispensable part of our professional lives. Spam E-mails have begun to undermine the integrity of E-mail and degrade online experience. Spam is growing rapidly and the financial costs as well as many other factors have become very burdensome to individuals and organizations. Naturally, a solution has to be found to this menace of spam-mail since otherwise this future of the E-mail itself may be at stake.

5.2 Brief Overview of the Work Done Vis-à-vis Original Research Objectives

The objectives of this research was devising new strategies for effective control of E-mail spam, evolving methods that shall allow these strategies to be implemented and building a proof-of-concept software system for evaluation of these strategies in a controlled environment. Both of these research objectives have been met as documented in earlier chapters.

In keeping with the initial literature review and identified research gaps in the PhD Research Proposal stage, this work carried out an extensive literature review, including those related to past, contemporary as well as evolving approaches towards anti-spam classification and control. E-mail spam statistics were collected from multiple verified sources and their growth over the years were duly analysed. Right from the genesis of the first E-mail spam to the multiple types, targets and trends of spam-mails over the last several decades were studied. The negative impact of spam on individuals, organizations, E-mail Service providers, and the Internet Service Providers were also carefully looked at before choosing the best approach towards handling the issue of spam-mails.

It was noted that *Network-based approaches, List-based, Rule-based, Cost-based and Content-based approaches were able to control incoming spam to varying extents.* Clever spammers, over the years, have come up with various tactics to bypass E-mail filters and it was required to have effective ways to counter these strategies. There can be no single strategy or solution that can handle all types of spam or contain all elements of spam in a cost-effective and yet scalable manner. Machine learning algorithms have been applied extensively for spam classification since spam problem can be majorly considered as text classification. These algorithms do go well with various other techniques, including distribution and parallel environment for optimizing the data, improving accuracy, etc. Combining different techniques improves different performance measures and this was precisely what was attempted in this work.

5.3 Principal Contributions of the Work Done

An intelligent E-mail filter 'BNCC' was developed and presented. This filter is the combination of Neural Networks (NN) and Centered Bayesian Networks (CBN) and aims to classify (and separate) spam from legitimate E-mails. This combination yields better results compared to simple classification. In this, both the header and content parts of the E-mail were considered for experimentation, which, by itself, is a complex job. Unlike statistical filters, NNs could quickly identify spam and made the classification process simpler. NNs trained with GA were used next. This helped in reducing the training and testing time. The approach has shown a high efficiency level compared with Bayesian approach. (Table 4.1 showed the False Positive rates, which was improved significantly compared to a simple statistical algorithm.) As complete information about the E-mail is taken into consideration in this approach, error rates have been found to be substantially low.

Another presented approach, 'BDT-MSVM', involved a combination of SVM and Decision Tree for classifying the records more precisely and efficiently. This is a multi-class approach, where a new class called 'likelihood (of) spam' is identified. Normal data points were classified with the help of decision tree, whereas some crucial data points were classified by employing multi-class SVM. When compared with other classifying

algorithms/method (as shown in Table 4.4) this approach demonstrated reasonably higher performance in terms of accuracy without compromising in terms of the time complexity.
As part of yet another effort to find an even better solution, the 'ES^{2}C Technique' as well as 'Spam-Reduce' methods were presented. Both of these are based on the Map-Reduce for large scale E-mail classification using ensemble based SVM and kNN join was, since it has a significant potential for parallelism. The most attractive feature of these strategies lies in its simplicity and flexibility that enable it to be implemented over any parallel/distributed paradigm. These strategies were able to deliver promising results that make these suitable for large scale E-mail classification for significant improvement in E-mail spam reduction.

5.4 Limitations of the Work Done

The principal limitation of the work done is that in view of the scope of the work, the work only targets the textual elements of E-mails and does not consider attached media in multiple formats. Also, scalability of the presented solution has been provisioned but has only been verified on a local scale using multiple computing nodes spread over multiple physical systems and not in a live e-mail environment. However, by involving three public and one private spam mail data-sets, it was possible to state with confidence that the strategies and algorithms presented here would prove useful in live environments as well.

5.5 Future Work

Future work for this study might involve different aspects, like focus towards increasing the efficiency of the identified and developed filters; improving implementation (of the strategies) and exploiting more potential parallelism. Support Vector Machine for spam classification also deserves more attention. The work done in SVM was only to a limited extent, but can be further explored by using the kernel, among other things, for classification. Another possible idea is to extend this work to consider other types of spam like image spam and PDF spam. In addition, issues related to attachments may also be addressed in the subsequent stages, for the sake of overall utility in real-life conditions, even though both of these elements have nothing to do with textual classification.

BIBLIOGRAPHY

A.Back, Hashcash a Denial of Service Countermeasure, 2002, Available online at: http://citeseer.ist.psu.edu/back02hashcash.html last accesed on December 1, 2015.

Abadi, M., Birrell, A., Burrows, M., Dabek, F., & Wobber, T. Bankable postage for network services. In Proceedings of *Advances in Computing Science–ASIAN. Progamming Languages and Distributed Computation Programming Languages and Distributed Computation* pp. 72-90. 2003.
.
Ahmed, Shabbir, and Farzana Mithun. Word Stemming to Enhance Spam Filtering. In Proceedings of First Conference on Email and Anti-Spam (CEAS). 2004.

Alham, N. K., Li, M., Liu, Y., & Qi, M. A MapReduce-based distributed SVM ensemble for scalable image classification and annotation. Computers & Mathematics with Applications, vol.66, no.10, pp.1920-1934. 2013.

Ali, A. B. M., and Yang Xiang. Spam classification using adaptive boosting algorithm. In Proceedings of Computer and Information Science, ICIS. 6th IEEE/ACIS International Conference on, pp. 972-976. IEEE, 2007.

Allman, Eric. Spam, spam, spam, spam, spam, the ftc, and spam. Queue 1,Vol. 6 pp. 62-69. 2003.

Amayri, Ola, and Nizar Bouguila. Online spam filtering using support vector machines. In Proceedings of Computers and Communications, ISCC, IEEE Symposium on, pp. 337-340. IEEE, 2009.

Andreolini, Mauro, Alessandro Bulgarelli, Michele Colajanni, and Francesca Mazzoni. Honeyspam: Honeypots fighting spam at the source. In Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop, pp. 11-18. 2005.

Androutsopoulos I, Paliouras G, Karkaletsis V, Sakkis G, Spyropoulos CD and Stamatopoulos P. Learning to filter spam e-mail: A comparison of a Naive Bayesian and a memory-based approach. In Proceedings of the Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases pp. 1-3, 2000.

Androutsopoulos I, Koutsias J, Chandrinos KV, Paliouras G and Spyropoulos CD, An evaluation of Naive Bayesian anti-spam filtering. In: Proceedings of the Workshop on Machine Learning in the New Information Age, eleventh European Conference on Machine Learning , pp. 9-17, 2000.

Androutsopoulos I, Koutsias J, Chandrinos KV and Spyropoulos CD, An experimental comparison of Naive Bayesian and keyword-based anti-spam filtering with encrypted personal e-mail messages. In: Proceedings of the Annual International ACMSIGIR Conference on Research and Development in Information Retrieval, pp. 160-167. 2000.

Androutsopoulos, I., Koutsias, J.: An Evaluation of Naive Bayesian Networks. In: Machine Learning in the New Information Age. Barcelona Spain 9-17 2000.

ARPANET (2006). Online Encyclopedia, Available online: http://www.webopedia.com/TERM/A/ARPANET.html. Last accessed 1 Dec 2015.

Attenberg, Josh, Kilian Weinberger, Anirban Dasgupta, Alex Smola, and Martin Zinkevich. Collaborative Email-Spam Filtering with the Hashing Trick. In Proceedings of the Sixth Conference on Email and Anti-Spam. CEAS 2009.

B. Schneier and N Ferguson, Practical Cryptography. Wiley Publishing Inc, First Edition, 2004.

BBC News (2005), Microsoft in 7m Dollar Spam Settlement. Available online: http://news. bbc.co.uk/2/hi/business/4137352.stm. Last accessed 1 Dec 2015.

Barracuda Reputation Block List (BRBL); Available online at: http://www.barracudacentral.org/rbl Last accessed: December 1, 2015.

Bishop, J. M., M. J. Bushnell, A. Usher, and S. Westland. Genetic optimisation of neural network architectures for colour recipe prediction. In Artificial Neural Nets and Genetic Algorithms, pp. 719-725. Springer Vienna, 1993.

Böhm, C. and Krebs, F. The k-nearest neighbour join: Turbo charging the KDD process. Knowledge and Information Systems, vol.6 no.6, pp.728-749. 2000.

Böhm, C., B. Braunmüller, M. M. Breunig, and H. P. Kriegel. Fast clustering based on high-dimensional similarity joins. In Proceedings of Information Knowledge Management (CIKM). 2000.

Dwork, Cynthia, Andrew Goldberg, and Moni Naor. On memory-bound functions for fighting spam. In *Advances in Cryptology-Crypto*, pp. 426-444. Springer Berlin Heidelberg, 2003.
.

Caruana, Godwin, Maozhen Li, and Man Qi. A MapReduce based parallel SVM for large scale spam filtering. In Proceedings of Fuzzy Systems and Knowledge Discovery (FSKD), Eighth International Conference on, vol. 4, pp. 2659-2662. IEEE, 2011.

Caruana, Godwin, Maozhen Li, and Hao Qi. SpamCloud: A MapReduce based anti-spam architecture. In Fuzzy Systems and Knowledge Discovery (FSKD), Seventh International Conference on, vol. 6, pp. 3003-3006. IEEE, 2010.

Carreras, Xavier, and Lluis Marquez. Boosting trees for anti-spam email filtering. arXiv preprint cs/0109015, 2001.

Château de Bossey, 2005. Report of the Working Group on Internet Governance, available online at: http://www.wgig.org/docs/WGIGREPORT.pdf last accesed on December 1, 2015.

Cook, Duncan, Jacky Hartnett, Kevin Manderson, and Joel Scanlan. Catching spam before it arrives: domain specific dynamic blacklists. In Proceedings of the Australasian workshops on Grid computing and e-research-vol. 54, pp. 193-202. 2006.

Chuan, Zhan, Lu Xianliang, Hou Mengshu, and Zhou Xu. A LVQ-based neural network anti-spam email approach. ACM SIGOPS Operating Systems Review vol.39, no.1 pp. 34-39. 2005.

Chang, Chih-Chung, and Chih-Jen Lin. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) vol.2, no. 3 p.27. 2011.

Chih-Wei Hsu; Chih-Jen Lin, A comparison of methods for multiclass support vector machines, in Neural Networks, IEEE Transactions on , vol.13, no.2, pp.415-425, Mar 2002.

Clark, James, Irena Koprinska, and Josiah Poon. A neural network based approach to automated e-mail classification. In null, p. 702. IEEE, 2003.

Cortes, Corinna, and Vladimir Vapnik. Support-vector networks. Machine learning vol.20, no.3 pp. 273-297. 1995.

Cristianini, Nello, and John Shawe-Taylor. An introduction to support vector machines and other kernel-based learning methods. Cambridge university press, 2000.

Cunningham, Padraig, and Sarah Jane Delany. k-Nearest neighbour classifiers. Multiple Classifier Systems pp. 1-17. 2007.

Data Engineering and Automated Learning–IDEAL, Springer Berlin Heidelberg, pp.578-585. 2004.

D. A. Turner and D. M. Havey, Controlling Spam through Lightweight Currency. In the Hawaii International Conference on Computer Sciences, 2004.

Damiani, Ernesto, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati. An Open Digest-based Technique for Spam Detection. ISCA PDCS pp. 559-564. 2004.

Delany, Mark. Domain-based email authentication using public keys advertised in the DNS (DomainKeys).RFC 4870. 2007.

Delany, Mark. Method and system for authenticating a message sender using domain keys. U.S. Patent 6,986,049, 2006.

DiBenedetto, Steve, Dan Massey, Christos Papadopoulos, and Patrick J. Walsh. Analyzing the aftermath of the mccolo shutdown. In Proceedings of Applications and the Internet. Ninth Annual International Symposium on, pp. 157-160. IEEE, 2009.

Drake, William J., ed. Reforming internet governance: Perspectives from the working group on internet governance (WGIG). No. 12. United Nations Publications, 2005.

Drucker, Harris, Donghui Wu, and Vladimir N. Vapnik. Support vector machines for spam categorization. Neural Networks, IEEE Transactions on vol.10.no.5 pp. 1048-1054. 1999.

Duan, Zhenhai, Peng Chen, Fernando Sanchez, Yingfei Dong, Mary Stephenson, and James Michael Barker. Detecting spam zombies by monitoring outgoing messages. Dependable and Secure Computing, IEEE Transactions on vol.9, no. 2 pp. 198-210. 2012.

Email sifter, Email Effectiveness: Assessing the Threats of Spam and Viruses and Evaluating Strategies for Managing them, white paper, email sifter, 2005. Available online at: http://www.emailsifter.com/whitepapers. last accessed on December 1st 2015.

ePrivacy Group (2003). Spam by Numbers, ePrivacy Group, a Philadelphia based trust technology company. available online at :
http://www.eprivacygroup.com/pdfs/SpamByTheNumbers.pdf . last accesed on December 1, 2015.

Erickson, David, Martin Casado, and Nick McKeown. The Effectiveness of Whitelisting: a User-Study. In CEAS. 2008.

Garcia, FD, Hoepman, J-H and Van Nieuwenhuizen, J., Spam Filter Analysis, 19th IFIP International Information Security Conference, Toulouse, France, 2004

González-Talaván, Guillermo. A simple, configurable SMTP anti-spam filter: Greylists. computers & security vol.25, no.3 pp. 229-236.2006.

Goldwasser, Shafi, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing 17, no. 2 pp. 281-308. 1988.

Gou, Jianping, Taisong Xiong, and Yin Kuang. "A novel weighted voting for k-nearest neighbor rule." *Journal of Computers* 6, no. 5 pp. 833-840. 2011.

Goweder, Abduelbaset M., Tarik Rashed, and S. Ali. An Anti-Spam System Using Artificial Neural Networks and Genetic Algorithms. In Proceedings of the International Arab Conference on Information Technology, pp. 1-8. 2008.

Graham, Paul. Better bayesian filtering. In *Proceedings of the spam conference*, vol. 11, pp. 15-17. 2003.

Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. ACM SIGKDD explorations newsletter vol.11, no. 1 pp. 10-18. 2009.

Hancock, Peter JB, and Leslie S. Smith. GANNET: Genetic design of a neural net for face recognition. In Parallel Problem Solving from Nature, Springer Berlin Heidelberg, pp. 292-296. 1991.

Harris, E. (2003). The next step in the spam control war: Greylisting. Available online at: http://projects.puremagic.com/greylisting/ last accessed on December 1, 2015.

Haskins, R and Nielsen, D., Slamming Spam, Pearson Education, Inc. 2005.

He, Jingrui, and Bo Thiesson. Asymmetric Gradient Boosting with Application to Spam Filtering. In 4th Conference on Email and Anti-Spam, CEAS. 2007.

Holland, J. Robust algorithms for adaptation set in a general formal framework. In IEEE ninth Symposium on Adaptive Processes Decision and Control Vol. 9, pp. 175. 1970.

Holland, J. H. Adaptation in natural and artificial system: an introduction with application to biology, control and artificial intelligence. Ann Arbor, University of Michigan Press. 1975.

Hsu, Chih-Wei, and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. Neural Networks, IEEE Transactions on vol.13, no.2 pp. 415-425. 2002.

Identified Internet Mail. Available online at: http://www.identifiedmail.com/draftfentonidentifiedmail.txt. last accessed on December 1st 2015.

J. Besag. Spatial Interaction and the Statistical Analysis of Lattice Systems. In Journal of the Royal Statistical Society, Series B, vol. 36, pp. 192–236, 1974.

Jesdanun. As spam filters improve, attention shifts to containment. Available Online at : http://www.technologyreview.com/TR/wtr14284,323,p1.html. Last accessed: December 2015.

Jin, Rong, Yan Liu, Luo Si, Jaime G. Carbonell, and Alexander Hauptmann. A new boosting algorithm using input-dependent regularizer. In Proceedings of Twentieth International Conference on Machine Learning. 2003.

Kakade, Amol G., Prashant K. Kharat, Amit Kumar Gupta, and Tushar Batra. Spam filtering techniques and MapReduce with SVM: A study. In proceedings of *Computer Aided System Engineering (APCASE), 2014 Asia-Pacific Conference on*, pp.59-64. IEEE, 2014.

Khong, W. K. The law and economics of junk emails (spam). EMLE thesis, University of Hamburg. 71 pp. 2001.

Khorsi, A An Overview of Content-Based Spam Filtering Techniques, Informatica (Slovenia), pp. 269-277. 2007.

Klimt, Bryan, and Yiming Yang. The enron corpus: A new dataset for email classification research. In Proceedings of Machine learning: ECML 2004, Springer Berlin Heidelberg, pp. 217-226. 2004.

Kunlun, Li, and Huang Houkuan. An architecture of active learning SVMs for spam. In Signal Processing, 2002 6th International Conference on, vol. 2, pp. 1247-1250. IEEE, 2002.

Lazzari, Lorenzo, Marco Mari, and Agostino Poggi. Cafe-collaborative agents for filtering e-mails. In Enabling Technologies: Infrastructure for Collaborative Enterprise, 2005. 14th IEEE International Workshops on, pp. 356-361. IEEE, 2005.

Lewis, David D., and Marc Ringuette. A comparison of two learning algorithms for text categorization. In Third annual symposium on document analysis and information retrieval, vol. 33, pp. 81-93. 1994.

Leo Breiman. Bagging predictors. Machine Learning, vol. 24 no.2 pp.123-140, 1996.

Li, Xuchun, Lei Wang, and Eric Sung. AdaBoost with SVM-based component classifiers. Engineering Applications of Artificial Intelligence vol.21, no.5 pp. 785-795. 2008.

Lin, Rake& Agrawal King-lp, and Harpreet S. Sawhney Kyuseok Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In Proceeding of the 21th International Conference on Very Large Data Bases, pp. 490-501. 1995.

Lowd, Daniel, and Christopher Meek. Good Word Attacks on Statistical Spam Filters. In Proceedings of Conference on Email and Anti-Spam (CEAS) CEAS. pp 8. 2005.

Lu, Wei, Yanyan Shen, Su Chen, and Beng Chin Ooi. Efficient processing of k nearest neighbor joins using mapreduce. In Proceedings of the VLDB Endowment vol.5, no.10 pp. 1016-1027. 2012.

M. Balakumar and V. Vaidehi, Ontology based classification and categorization of email: In Proceedings of the International Conference on Signal Processing, Communications and Networking, pp. 199-202, 2008.

Mail Abuse.com; Mail Abuse Prevention System. Available online at:
http://www.mailabuse.com. Last accessed: December 1, 2015.

Maniezzo, V. Genetic evolution of the topology and weight distribution of neural networks. Neural Networks, IEEE Transactions on, vol.5 no.1, 39-53. 1994.

Manjusha, K; Shahabas, S; Banerjee, Rahul: An Efficient Method of Spam Classification by Multi-class Support Vector Machine Classifier,In Proceedings of the Seventh International Conference on Data Mining and Warehousing ICDMW, pp. 102-110, 2013.

Manjusha, K., and Rakesh Kumar. Spam Mail Classification Using Combined Approach of Bayesian and Neural Network. In Proceedings of International Conference on Computational Intelligence and Communication Networks CICN. pp. 145-149. IEEE, 2010.

McCallum, Andrew, and Kamal Nigam. A comparison of event models for naive bayes text classification. In AAAI-98 workshop on learning for text categorization, vol. 752, pp.41-48. 1998.

McAfee, Executive Summary—McAfee Threats Report: Fourth Quarter 2012. Available online: www.mcafee.com/us/resources/reports/rp-quarterly-threat-q2-2014.pdf last accesed on December 1, 2015.

McAfee, I. I. The Carbon Footprint of Email Spam Report, 2009. Available online: http://resources. mcafee. com/content/NACarbonFootprintSpam. last accesed on December 1, 2015.

Metsis, Vangelis, Ion Androutsopoulos, and Georgios Paliouras. Spam filtering with naive bayes-which naive bayes?. In CEAS, pp.27-28. 2006.

Meyer, Oliver, Bernd Bischl, and Claus Weihs. Support vector machines on large data sets: Simple parallel approaches. In Proceedings of Data Analysis, Machine Learning and Knowledge Discovery, Springer International Publishing, pp. 87-95. 2014.

Meyer, Tony A., and Brendon Whateley. SpamBayes: Effective open-source, Bayesian based, email classification system. In First Conference on Email and Anti-Spam CEAS. 2004.

Michael Sampson, Spam Control: Problems and Opportunities, Ferris Research Publication, 2003. Available online at: tp://www.ferris.com/view_content.php?o=Spam+Control&id=105 Last accessed 1 Dec 2015.

Michalson The Law Vs The Scourge Of Spam Available online: http://www.michalsons.co.za/blog/the-law-vs-the-scourge-of-spam/1019 Last accessed 1 Dec 2015.

Ming, L., Yunchun, L., & Wei, L. Spam Filtering by Stages, International Conference on Convergence Information Technology pp. 2209-2213. 2007.

Mohammad, Adel Hamdan, and Raed Abu Zitar. Application of genetic optimized artificial immune system and neural networks in spam detection. applied soft computing vol.11, no.4 pp. 3827-3845. 2011.

Neumayer, R. Clustering based ensemble classification for spam filtering. In sixth workshop on data analysis. pp. 11-22. Elfa Academic Press. 2006.

Nosseir, Ann, Khaled Nagati, and Islam Taj-Eddin. Intelligent word-based spam filter detection using multi-neural networks. *IJCSI International Journal of Computer Science Issues* vol. 10, no. 2 pp: 1694-0814. 2013.

O'Brien, Cormac, and Carl Vogel. Spam filters: bayes vs. chi-squared; letters vs. words. In Proceedings of the 1st international symposium on Information and communication technologies, pp. 291-296. Trinity College Dublin, 2003.

Oda, Terri, and Tony White. Increasing the accuracy of a spam-detecting artificial immune system. In Evolutionary Computation. CEC. The 2003 Congress on, vol.1, pp. 390-396. IEEE, 2003.

Orbach, Julian J. Spam whitelisting for recent sites. U.S. Patent 8,095,602, 2012.

Oudot, L. 200). Fighting spammers with honeypots. Securityfocus Available online at: http://www. securityfocus. com/infocus/1747 and http://www. securityfocus. com/infocus/1748. last accesed on December 1, 2015.

Pantel, Patrick, and Dekang Lin. Spamcop: A spam classification & organization program. In Proceedings of AAAI-98 Workshop on Learning for Text Categorization, pp. 95-98. 1998.

Pătraşcu, Alecsandru, Ion Bica, and Victor Valeriu Patriciu. Spam Filtering Using Automated Classifying Services over a Cloud Computing Infrastructure. In *Innovative Security Solutions for Information Technology and Communications*, pp. 226-241. 2015.

Pelletier, L., Jalal Almhana, and Vartan Choulakian. Adaptive filtering of spam. In Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on, pp. 218-224. IEEE, 2004.

Perone, M., (2004). An overview of spam blocking techniques. *Barracuda Networks Corp*. Available online at:
http://www.barracudanetworks.com/ns/downloads/barracuda_spam_blocking_techniques.pdf Last accessed: December 1, 2015.

Pfleeger, SL and Bloom, G., Canning Spam: Proposed Solutions to Unwanted Email , Security & Privacy Magazine, Vol. 3, No. 2, pp. 40-47, IEEE, 2005.

P. G. Capek, B. Leiba, M. N. Wegman, and S. E. Fahlman. Charity Begins at your mail Program. Technical report, IBM Thomas J. Watson Research Center, Hawthorne, NY 10532, USA. 2003.

Phishing activity trends report. Anti-Phishing Working Group, 2004. Available online at: http://docs.apwg.org/reports/apwg_trends_report_q2_2014.pdf last accesed on December 1, 2015.

Platt, John C., Nello Cristianini, and John Shawe-Taylor. Large Margin DAGs for Multiclass Classification. In nips, vol. 12, pp. 547-553. 1999.

Poutanen, Tomi, Heather Hinton, and Michael Stumm. NetCents: A lightweight protocol for secure micropayments. In USENIX Workshop on Electronic Commerce, pp. 25-36. 1998.

Prakash, V. V. Digest-Nilsimsa.2002. Available online at:
http://search.cpan.org/dist/Digest-Nilsimsa/ last accesed on December 1, 2015.

Project Honey Pot; 2004. Available online at:
http://www.projecthoneypot.org/. last accesed on December 1, 2015.

Puniškis, Danius, R. Laurutis, and R. Dirmeikis. An artificial neural nets for spam e-mail recognition. Elektronika ir Elektrotechnika vol.69, no.5 pp. 73-76. 2015.

Ramachandran, A & Feamster, N 'Understanding the network-level behavior of spammers', SIGCOMM Comput. Commun. Rev., vol. 36, no. 4, pp. 291-302. 2006.

Raymond, ES 2005, Bogofilter: A fast open source bayesian spam filters. Available online at: http:/bogofilter.sourceforge.net/ last accesed on December 1, 2015.

Rennie, Jason. ifile: An application of machine learning to e-mail filtering. In Proceedings of. KDD Workshop on Text Mining 6 pp 2000.

Renuka, Karthika D., and P. Visalakshi. Latent Semantic Indexing Based SVM Model for Email Spam Classification. *Journal of Scientific & Industrial Research* vol.73,no.7pp. 437A42. 2014.

R. Rivest and A. Shamir. Payword and Micromint: Two Simple Micropayment Schemes. micropayment schemes. In *Security Protocols* Springer Berlin Heidelberg. pp. 69-87. 1997.

Rhyolite Software 2005 – Distributed Checksum Clearinghouse, Available online at: http://www.rhyolite.com/anti-spam/dcc/ last accesed on December 1, 2015.

Rigoutsos, Isidore, and Tien Huynh. Chung-Kwei: a Pattern-discovery-based System for the Automatic Identification of Unsolicited E-mail Messages (SPAM). In First Conference on Email and Anti-Spam CEAS. 2004.

Saeedian, Mehrnoush Famil, and Hamid Beigy. Dynamic classifier selection using clustering for spam detection. In Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on, pp. 84-88. IEEE, 2009.

Sahami, M, Dumais, S, Heckerman, D and Horvitz, E., A Bayesian Approach to Filtering Junk E-Mail , AAAI-98 Workshop on Learning for Text Categorization, Vol. 62, pp. 98-105 1998.

Sakkis, Georgios, Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Constantine D. Spyropoulos, and Panagiotis Stamatopoulos. A memory-based approach to anti-spam filtering for mailing lists. Information retrieval vol.6, no.1 pp. 49-73. 2003.

Salmi, T. 2005. Foiling Spam with an Email Password System. Available online at: from http://www.uwasa.fi/~ts/info/spamfoil.html. last accesed on December 1, 2015.

Sara Radicati, Email Statistics Report , 2014-2018. available online at: http://www.radicati.com/wp/wp-content/uploads/2014/01/Email-Statistics-Report-2014-2018-Executive-Summary.pdf. last accesed on December 1, 2015.

Schiffmann, W., Joost, M., & Werner,R. Application of genetic algorithms to the construction of topologies for multilayer perceptrons. In Artificial Neural Nets and Genetic Algorithms Springer Vienna. pp.675-682. 1993.

Schneider, Karl-Michael. A comparison of event models for Naive Bayes anti-spam e-mail filtering. In Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics, vol. 1, pp.307-314. 2003.

Sculley, David, Gabriel Wachman, and Carla E. Brodley. Spam Filtering Using Inexact String Matching in Explicit Feature Space with On-Line Linear Classifiers. In TREC. 2006.

Segal, Richard, Jason Crawford, Jeffrey O. Kephart, and Barry Leiba. SpamGuru: An Enterprise Anti-Spam Filtering System. In First Conference on Email and Anti-Spam CEAS. 2004.

Shvachko, Konstantin, Hairong Kuang, Sanjay Radia, and Robert Chansler. The hadoop distributed file system. In Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on, pp. 1-10. IEEE, 2010.

Shrivastava, Jitendra Nath, and Maringanti Hima Bindu. E-mail spam filtering using adaptive genetic algorithm. *International Journal of Intelligent Systems and Applications (IJISA)* vol.6, no. 2 pp. 54 2014.

SpamCop.What is the SpamCop Blocking List (SCBL)? Available online at: www.spamcop.net/bl.html Last accessed: December 1, 2015.

SpamCon Foundation; About SpamCon Foundation – Summary, The SpamCon foundation, San Francisco, California; Available online at: http://spamcon.org/about/spamcon-summary.pdf Last accessed 1 Dec 2015.

SPEW Organisation; Spam Prevention Early Warning System 2005, Available online at: http://www.spews.org/ Last accessed: December 1, 2015.

Stuart, Ian, Sung-Hyuk Cha, and Charles Tappert. A neural network classifier for junk e-mail. In Document Analysis Systems VI, pp. 442-450. Springer Berlin Heidelberg, 2004.

Tao, Yong-Cai, Zheng-Yuan Xue, and Lei Shi. MapReduce-based Bayesian anti-spam filtering mechanism. Jisuanji Yingyong/ Journal of Computer Applications vol.31, no.9 pp. 2412-2416. 2011.

Tang, Fa-ming, Zhong-dong Wang, and Mian-yun Chen. On multiclass classification methods for support vector machines. Control and Decision vol. 20, no. 7 pp: 746. 2005.

The Penny Black Project. Available online at:
http://research.microsoft.com/research/sv/PennyBlack/. Last accessed: December 1,2015.

The spamhaus project: The Definition of Spam. Available online at:
http://www.spamhaus.org. Last accessed 1 Dec 2015.

The Radicati Group, E-mail Statistics Report, 2012-2016. available online at: http://www.radicati.com/wp/wp-content/uploads/2012/04/E-mail-Statistics-Report 2012-2016-Executive-Summary.pdf Last accessed: December 1, 2015.

Tao, Yong-Cai, Zheng-Yuan Xue, and Lei Shi. MapReduce-based Bayesian anti-spam filtering mechanism. Jisuanji Yingyong/ Journal of Computer Applications 31, no. 9 pp. 2412-2416. 2011.

Trend Micro, R. B. L. Service, 2005. Trend Micro Incorporated, available online at:
https://ers.trendmicro.com/reputations/block/64.6.246.4/RBL Last accessed: December 1, 2015.

Trudgian, Dave C. Spam classification using nearest neighbour techniques. In Intelligent Data Engineering and Automated Learning–IDEAL pp. 578-585. Springer Berlin Heidelberg, 2004.

Valentini, Giorgio. Random aggregated and bagged ensembles of SVMs: an empirical bias–variance analysis. In Multiple classifier systems, Springer Berlin Heidelberg, pp. 263-272. 2004.

Vinther, Michael.2002 Intelligent junk mail detection using neural networks. available online at: http://www.logicnet.dk/reports/JunkDetection/JunkDetection.Pdf Last accessed: December 1, 2015.

Vipul's Razor. Available online at:
http://razor.sourceforge.net. Last accessed: December 1, 2015.

Voorhees, Ellen M., and Donna K. Harman, eds. TREC: Experiment and evaluation in information retrieval. Vol. 1. Cambridge: MIT press, 2005.

Wang, J., Gao, K., & Vu, H. Q. SpamCooling: a parallel heterogeneous ensemble spam filtering system based on active learning techniques. Journal of convergence information technology, vol. 5, no. 4, pp. 90-102. 2010.

Wang, Shi-jin, Avin Mathew, Yan Chen, Li-feng Xi, Lin Ma, and Jay Lee. Empirical analysis of support vector machine ensemble classifiers. Expert Systems with Applications vol.36, no. 3 pp. 6466-6476. 2009.

Wang, Zi-Qiang, Xia Sun, Xin Li, and De-Xian Zhang. An efficient SVM-based spam filtering algorithm. In Machine Learning and Cybernetics, 2006 International Conference on, pp. 3682-3686. IEEE, 2006.

Wei, Chih-Ping, Hsueh-Ching Chen, and Tsang-Hsiang Cheng. Effective spam filtering: A single-class learning and ensemble approach. Decision Support Systems vol.45, no.3 pp. 491-503. 2008.

Wenjuan Li; Weizhi Meng, "An empirical study on email classification using supervised machine learning in real environments," in *Communications (ICC), 2015 IEEE International Conference on*, pp.7438-7443, 2015.

Woitaszek, Matthew, Muhammad Shaaban, and Roy Czernikowski. Identifying junk electronic mail in Microsoft outlook with a support vector machine. In null, p. 166. IEEE, 2003.

Wong, Meng, and Wayne Schlitt. Sender policy framework (SPF) for authorizing use of domains in e-mail, version 1. No. RFC 4408. 2006.

Wikipedia, E-mail spam. Available online :
http://en.wikipedia.org/wiki/E-mail_spam, Last accessed 1 Dec 2015.

Wu, Jiansheng, and Tao Deng. Research in anti-spam method based on bayesian filtering. In IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, pp. 887-891, 2008.

Xia, Chenyi, Hongjun Lu, Beng Chin Ooi, and Jing Hu. Gorder: an efficient method for KNN join processing. In Proceedings of the Thirtieth international conference on Very large data bases- vol. 30, pp. 756-767. 2004.

Yang, Zhen, Xiangfei Nie, Weiran Xu, and Jun Guo. An approach to spam detection by naive Bayes ensemble based on decision induction. In Intelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on, vol.2, pp. 861-866. 2006.

Yao, Bin, Feifei Li, and Piyush Kumar. K nearest neighbor queries and knn-joins in large relational databases (almost) for free. In international conference on Data engineering (ICDE), 2010 IEEE 26th, pp. 4-15. 2010.

Yerazunis, William S., Shalendra Chhabra, Christian Siefkes, Fidelis Assis, and Dimitrios Gunopulos. A unified model of spam filtration. In Proceedings of the MIT Spam Conference, Cambridge, MA, USA. 2005.

Ying, K. C., Lin, S. W., Lee, Z. J., & Lin, Y. T.An ensemble approach applied to classify spam e-mails. Expert Systems with Applications, vol.37, no. 3, 2197-2201. 2010.

Yoshida, Kenichi, Fuminori Adachi, Takashi Washio, Hiroshi Motoda, Teruaki Homma, Akihiro Nakashima, Hiromitsu Fujikawa, and Katsuyuki Yamazaki. Density-based spam detector. IEICE transactions on information and systems vol.87, no.12 pp. 2678-2688. 2004.

Youn, Seongwook, and Dennis McLeod. A comparative study for email classification. In Advances and innovations in systems, computing sciences and software engineering, Springer Netherlands, pp.387-391. 2007.

Yu, Cui, Rui Zhang, Yaochun Huang, and Hui Xiong. High-dimensional knn joins with incremental updates. Geoinformatica vol.14, pp. 55-82. 2010.

Yu, Cui, Bin Cui, Shuguang Wang, and Jianwen Su. Efficient index-based KNN join processing for high-dimensional data. Information and Software Technology vol. 49, no.4 pp. 332-344. 2007.

Shi, Zhiwei, H. Hu, and Z. Shi, A bayesian computational cognitive model, in Proceedings of the 3rd International Workshop on Neural-Symbolic Learning and Reasoning, NeSy, 2007.

Zhao, Wenqing, and Zili Zhang. An email classification model based on rough set theory. In proceedings of International Conference on Active Media Technology, pp.403-408.IEEE 2005.

Zhao, Jun, Zhu Liang, and Yong Yang. Parallelized incremental support vector machines based on MapReduce and Bagging technique. In proceedings of International Conference on Information Science and Technology (ICIST), IEEE pp. 297-301. 2012.

Zhang, Chi, Feifei Li, and Jeffrey Jestes. Efficient parallel kNN joins for large data in MapReduce. In Proceedings of the 15th International Conference on Extending Database Technology, ACM, pp. 38-49. 2012.

Zhang, Le, Jingbo Zhu, and Tianshun Yao. An evaluation of statistical spam filtering techniques. ACM Transactions on Asian Language Information Processing (TALIP) vol.3, no.4 pp.243-269. 2004.

List of Publications and Papers Presented in Conferences

Papers Published/Communicated

- Manjusha, K., and Rakesh Kumar. Spam Mail Classification Using Combined Approach of Bayesian and Neural Network. In Computational Intelligence and Communication Networks (CICN), 2010 International Conference on, pp. 145-149. IEEE, 2010.

- Manjusha, K; Shahabas, S; Banerjee, Rahul: An Efficient Method of Spam Classification by Multi-class Support Vector Machine Classifier, Proceedings of the Seventh International Conference on Data Mining and Warehousing ICDMW, pp. 102-110, 2013.

- Manjusha K, Rakesh Kumar, A Two Layered Bayesian Model to Prioritize emails. Indian International Conference of Artficial Intelligence (IICAI 2009), Proceedings, pp, 1618-1625, 2009

- K. Manjusha and N.V.M. Rao, A Cost Analysis of Spam e-Mail Problem in an Educational Organization, Indo- US conference and workshop on & Cyber Security, Cyber Crime and Cyber Forensics, Cochin, India, 2009

- Manjusha K, and Rahul Banerjee,.The Ensemble-based SVM Spam-mail Classifier (The ES^{2}C Technique) Pattern recognition Letters Communicated.

Brief Biography of the Research Supervisor

Dr. Rahul Banerjee is a Professor of Computer Science (Engineering) and the Head of the Department of Computer Science & Information Systems at BITS Pilani. He holds a PhD in Computer Science & Engineering. His interests lie in the areas of Computer Networking (Protocol engineering, QoS, Routing, Mobility, Vehicular Networking), Wearable Computing (involving non-invasive sensing, on-body computing, BAN based routing) and Ubiquitous Computing (involving Internet of Things / Cyber-Physical Systems, Ambient Intelligence).

He has led and participated in several funded research projects including those funded by European Commissionin the area of Next Generation Networking involving IPv6 (in a five-nation project) , Govt. of India (DIT-MCIT) in the area of Technology-enabled Learning, Govt. of France (in a four-nation project) in the area of IPv6-enabled Low-Power Wireless Sensor Networking. In addition, Microsoft Research, IBM, Cisco, DEC and Google have supported his work in a variety of forms, ranging from full research grants/awards to equipment grants and travel grants. He has been a reviewer for IEEE Transactions on Computers, IEEE Internet Computing, IEEE Communications, IEEE Transactions on ITS, IJCNDS and IISc Journal apart from numerous refereed international and national conferences and symposia. He has also served on Technical Committees related to the Bureau of Indian Standards (BIS) and IEEE apart from playing an active role in select work-groups of the Internet Engineering Task-Force (IETF).
He is a Member of the IEEE, ACM, ISTE, IE(I), CSI and ISCA.

Biography of the Candidate

Manjusha K is a Research Scholar in the department of Computer Science and Information System, Birla Institute of Technology and Science, Pilani. She also completed her M.E. (Software Systems) from Birla Institute of Technology & Science, Pilani. She also worked as a faculty member at the department of Computer Science and Information Systems at BITS Pilani. Her areas of interest include Data Mining and Network Security. She is a member of IEEE.