

**An Integrated Approach to Spam filtering and Incremental  
Updation of Spam Corpus using Data Mining Techniques  
with Modified Spell Correction Algorithm**

*Thesis submitted to*

Cochin University of Science and Technology

*in partial fulfillment of the requirements for the award of the degree of*

**DOCTOR OF PHILOSOPHY**

*by*

**LINY VARGHESE**

*Under the guidance of*

**Dr. K. POULOSE JACOB**



**FACULTY OF TECHNOLOGY**

Cochin University of Science and Technology

Kochi - 682 022, Kerala, India

October 2017

# **An Integrated Approach to Spam Filtering and Incremental Updation of Spam Corpus using Data Mining Techniques with Modified Spell Correction Algorithm**

**Ph.D. Thesis in the field of Text Mining**

## **Author**

**LINY VARGHESE**

Information Scientist

University Library

Cochin University of Science and Technology

Kochi- 682 022, India

E-mail: liny@cusat.ac.in

## **Research Advisor**

**Dr. K. POULOSE JACOB** (Supervising Guide)

Former Pro Vice Chancellor

Professor, Department of Computer Science

Cochin University of Science and Technology

Kochi- 682 022, India

E-mail: kpj@cusat.ac.in

**OCTOBER 2017**

*Dedicated to ...*

*My Parents, Husband & Kids*

## Declaration

*I hereby declare that the present work entitled "An Integrated approach to spam filtering and incremental updation of spam corpus using data mining techniques with modified spell correction algorithm" is based on the original work done by me under the guidance of Dr. K. Poulose Jacob, Former Pro-Vice Chancellor and Professor, Department of Computer Science, Cochin University of Science and Technology, Kochi-22 and the work does not form part of any dissertation submitted for the award of any degree, diploma, or any other title or recognition from any University or Institution.*

*Kochi-22*

*30/10/2017*

*Liny Varghese*

*PhD Reg. No. 3553*



FACULTY OF TECHNOLOGY  
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY  
KOCHI- 682022, INDIA

## Certificate

*Certified that the work presented in this thesis entitled “An Integrated approach to spam filtering and incremental updation of spam corpus using data mining techniques with modified spell correction algorithm” is based on the bona fide research work done by Ms. Liny Varghese under my guidance in the Department of Computer Science, Cochin University of Science and Technology, Kochi -22 and has not been included in any other thesis submitted previously for the award of any degree.*

*Kochi-22*

*30/10/2017*

*Dr. K. Poullose Jacob*

## CERTIFICATE

*This is to certify that all the relevant corrections and modifications suggested by the audience during the Pre-synopsis seminar and recommended by the Doctoral Committee of the candidate have been incorporated in the thesis.*

*Kochi-22*

*30/10/2017*

*Dr. K. Poulose Jacob*

## Acknowledgements

*I am deeply indebted and grateful to many people who supported me during the research work and preparation of the thesis.*

*First and foremost, I give special thanks and glory to the God Almighty for giving me the wisdom and health to complete this research work,*

*I express my immense gratitude to my research guide and mentor Dr. K. Poullose Jacob, Former Pro Vice chancellor and Professor, Department of Computer Science, Cochin University of Science and Technology for his exceptional guidance and generous help throughout my Ph.D work. I am extremely fortunate to work under his inspiring guidance.*

*I am very grateful to Dr. Supriya M. H, Dr. Sumam Mary Idicula and Dr. Kannan B for their motivation, orientation, fruitful suggestions, constant encouragement and timely interactions which helped me to fulfill this research.*

*I am thankful to Dr. David Peter S, Registrar, Cochin University of Science and Technology and Dr. Santhosh Kumar G, Head, Department of Computer Science for their administrative support and valuable research guidance during my research. I am also thankful to the library, technical*

*and administrative staff of the Department of Computer Science and my colleagues in University Library for their co-operation and support.*

*I am obliged to Dr. Daleesha M. Viswanathan and Smt. Mini U for their constant encouragement, sharing thoughts and motivational talks which helped me a lot to continue and complete this research work, I am greatly thankful to all CIRM and E-Governance cell staff for their cooperation, support and care which helped me to pursue the research.*

*I owe heartfelt thanks to my parents, Shri. Varghese & Smt. Elsy and my siblings, Lisy, Leo & Linda for their prayers, motivation, encouragement and understanding.*

*Finally, words would not suffice to express my gratitude towards my husband Polly, and my dear little ones, Johan and Ryan for their love, understanding, patience and encouragement that helped me to fulfill my dream.*

**LINY VARGHESE**



## ABSTRACT

Spam is a universal problem, an ongoing issue and one of the most critical problems on Internet. This worldwide issue wastes Internet users' precious time, Internet bandwidth, computers' processing power and storage capacity. Furthermore, there are also some hidden and difficult effects due to spam, such as the loss of legitimate e-mails –namely False Positives (FP) effect– the misleading of Internet consumers, exposure to unethical content for children, electronic frauds, etc. A number of countermeasures have been deployed, which are meant to reduce spam phenomenon. In general, there are three anti-spamming approaches: legal, social and technical. Basically, anti-spam efforts are grouped based on where the filters reside and how the filters react against spammer's techniques. In the first case, anti-spamming efforts are distinguished based on whether they reside either on server side of an e-mail service or at user's computer (user-based or client-based). In the second case, the anti-spamming efforts are complementary approaches to spammers' methodologies. Controlling spam requires an array of complementary techniques and continued efforts to adapt them, as spammers continue to adapt their own methods.

**Spam filtering using Bayesian models:** From the literature it is found that Bayesian models outperform all other spam filtering machine learning algorithms. This research focuses on applying different Bayesian models for spam filtering task. Two models were compared here: Bernoulli model and Multinomial model and promising results were received for both the models.

**Filtering template based spam:** Most of the time, spammers use mail templates for sending spam. To send a particular promotion, they create pre-

---

---

formatted template and merge the template with details of receivers stored in their database. Timely detection of these mails and underlying template features can be used to easily ignore forthcoming spam. Most high-volume spam is sent using such tools which randomizes parts of the message - subject, body, sender address etc. Using the advantage of template phenomena in emails, vector space models are applied to spam filtering. Here two models are applied: Simple vector space model and Rocchio classification model. In simple vector space models, the test mail is checked against each of the mails in the training set. In Rocchio classification each test mail is checked against the template mails only. Both the methods have their own advantages and disadvantages.

**Incremental clustering of training set:** All the methods explained in the literature as well as in the present research depend largely on the training data. The quality of training data depends on how frequently and efficiently the spam training set is updated. The vector space models explained above use template emails stored in the training set. The objective of this research work is to investigate and evaluate the applicability of Genetic algorithm and K-Means algorithm in the process of selection of suitable mail templates.

**Deobfuscation of mails:** To cheat the filtering mechanisms implemented on mail servers and client programs, spammers obfuscate the words in spam mails. Obfuscation can be done in different ways like changing letters, replacing letters with lookalike letters. The text based filters may not be able to find such words and so cannot filter those mails. Hence we require a system to deobfuscate such mails in order to improve the classification accuracy. A modified algorithm for spell correction is also devised for this purpose.

---

**Standard scalable framework solution to spam filtering:** Most of the methods explained in literature are implemented on personal experimental setup. No standard framework or software is used in these analyses. In order to achieve good results and benchmark solutions, we require a scalable solution with a set of standard algorithms to handle this high volume, high velocity and large varieties of spam. In this work, Apache Mahout- an open source machine learning library from Apache - is used to analyze the time and accuracy efficiencies of a big data framework in the context of spam filtering.

## Table of Contents

<b>TABLE OF CONTENTS.....</b>	<b>IV</b>
<b>LIST OF TABLES.....</b>	<b>XII</b>
<b>LIST OF FIGURES.....</b>	<b>XV</b>
<b>LIST OF ABBREVIATIONS.....</b>	<b>XVII</b>
<b>CHAPTER 1 - INTRODUCTION .....</b>	<b>1</b>
<b>1.1 Problems with Spam .....</b>	<b>1</b>
<b>1.2 Definition and General Characteristics of Spam .....</b>	<b>2</b>
<b>1.3 Different Spam Definitions .....</b>	<b>3</b>
<b>1.4 Spammer Methods.....</b>	<b>4</b>
<b>1.5 E-Mail Structure .....</b>	<b>6</b>
<b>1.6 Types of Spam .....</b>	<b>8</b>
1.6.1 Text based Spam.....	8
1.6.2 Image Spam.....	8
1.6.3 Attachment Spam .....	10
<b>1.7 Taxonomy of Spam.....</b>	<b>11</b>

---

1.7.1	Varieties of Spam .....	11
<b>1.8</b>	<b>Different ways to Send Spam .....</b>	<b>13</b>
1.8.1	Email Spam .....	13
1.8.2	Usenet Spam.....	14
1.8.3	Social Networks Spam .....	14
<b>1.9</b>	<b>Anti-Spam Measures.....</b>	<b>14</b>
<b>1.10</b>	<b>Motivation .....</b>	<b>15</b>
<b>1.11</b>	<b>Objectives of Research: .....</b>	<b>16</b>
<b>1.12</b>	<b>Research Method .....</b>	<b>17</b>
<b>1.13</b>	<b>Spam Filtering Approaches .....</b>	<b>17</b>
<b>1.14</b>	<b>Incremental Clustering of Training set .....</b>	<b>18</b>
<b>1.15</b>	<b>Deobfuscation of Mails .....</b>	<b>18</b>
<b>1.16</b>	<b>Scalable and Standard framework for Spam filtering .....</b>	<b>18</b>
<b>1.17</b>	<b>Flowchart of the System .....</b>	<b>19</b>
<b>1.18</b>	<b>Organization of the Thesis .....</b>	<b>20</b>
<b>CHAPTER 2 – LITERATURE REVIEW .....</b>		<b>21</b>
<b>2.1</b>	<b>Introduction.....</b>	<b>21</b>
2.1.1	Spam Filtering as Text Categorization.....	22

---

---

<b>2.2</b>	<b>An Overview of Approaches to Spam Filtering .....</b>	<b>22</b>
2.2.1	Some Prevention Methods from User side .....	23
2.2.2	SMTP Approaches .....	24
2.2.3	Rule based Techniques or Heuristic Filtering .....	24
2.2.4	Anti-spam Legislation Efforts .....	25
2.2.5	Machine Learning & Statistical Spam Filtering Techniques .....	26
2.2.6	Filtering approaches to Image Spam .....	33
2.2.7	Filtering approaches to Attachment Spam .....	34
2.2.8	Filters based on Non-content Features .....	35
<b>2.3</b>	<b>Incremental Updation of Spam Training set .....</b>	<b>36</b>
<b>2.4</b>	<b>Spell Correction Algorithms .....</b>	<b>36</b>
<b>2.5</b>	<b>Deobfuscation of Spam .....</b>	<b>37</b>
<b>2.6</b>	<b>Scalable Spam Filtering solutions using Standard Frameworks .....</b>	<b>38</b>
<b>2.7</b>	<b>Pros and Cons of Rule based versus Learning based approaches .....</b>	<b>39</b>
<b>2.8</b>	<b>Literature Review Summary .....</b>	<b>40</b>
<b>CHAPTER 3 – SPAM FILTERING USING BAYESIAN FILTERS.....</b>		<b>42</b>
<b>3.1</b>	<b>Introduction.....</b>	<b>42</b>
<b>3.2</b>	<b>Models for NB classifier .....</b>	<b>43</b>
3.2.1	Algorithm - Naïve Bayes Classifier .....	43

---

---

---

<b>3.3</b>	<b>Spam Training set .....</b>	<b>44</b>
3.3.1	Tokenization.....	45
3.3.2	Transforming to Term Frequency.....	46
3.3.3	Transforming by Document Frequency .....	46
3.3.4	Feature Selection .....	46
<b>3.4</b>	<b>Implementation .....</b>	<b>53</b>
<b>3.5</b>	<b>Evaluation Measures.....</b>	<b>54</b>
<b>3.6</b>	<b>Results.....</b>	<b>55</b>
3.6.1	Bernoulli model Results .....	55
3.6.2	Multinomial model Results .....	56
<b>3.7</b>	<b>Result Analysis.....</b>	<b>59</b>
3.7.1	Comparison of Results .....	61
<b>3.8</b>	<b>Summary.....</b>	<b>62</b>
<b>CHAPTER 4 - FILTERING TEMPLATE DRIVEN SPAM MAILS.....</b>		<b>63</b>
<b>4.1</b>	<b>Introduction.....</b>	<b>63</b>
<b>4.2</b>	<b>Classification Models.....</b>	<b>64</b>
4.2.1	The Simple Vector Space Model .....	65
4.2.2	Rocchio Classification.....	66
<b>4.3</b>	<b>Methodology.....</b>	<b>67</b>

---

---

---

4.3.1	For Simple VSM .....	68
4.3.2	For VSM using Rocchio Classification .....	68
<b>4.4</b>	<b>Composition of Training and Test Datasets .....</b>	<b>70</b>
<b>4.5</b>	<b>Performance Measures .....</b>	<b>70</b>
<b>4.6</b>	<b>Experimental Results .....</b>	<b>71</b>
<b>4.7</b>	<b>Summary.....</b>	<b>73</b>
<b>CHAPTER 5 – FINDING TEMPLATE MAILS FROM SPAM CORPUS.....</b>		<b>74</b>
<b>5.1</b>	<b>Introduction.....</b>	<b>74</b>
<b>5.2</b>	<b>Related Works.....</b>	<b>75</b>
<b>5.3</b>	<b>Method.....</b>	<b>76</b>
5.3.1	Genetic Algorithm .....	76
5.3.2	Supervised Genetic Learning Algorithm.....	77
5.3.3	K –Means Algorithm .....	79
<b>5.4</b>	<b>Building Representation .....</b>	<b>79</b>
<b>5.5</b>	<b>Experimental Results .....</b>	<b>81</b>
5.5.1	Classification Results after Feature Selection .....	83
5.5.2	Classification Results after Applying Genetic Algorithm .....	84
5.5.3	Classification Results after Applying K –Means Algorithm.....	85

---



---

<b>Case 1: Classification without feature selection .....</b>	<b>87</b>
<b>5.6 Discussion.....</b>	<b>88</b>
<b>5.7 Summary.....</b>	<b>89</b>
<b>CHAPTER 6 – A MODIFIED SPELL CORRECTION ALGORITHM AND DEOBFUSCATION OF EMAILS .....</b>	<b>90</b>
<b>6.1 Introduction.....</b>	<b>90</b>
<b>6.2 Spell Correction Algorithms .....</b>	<b>91</b>
<b>6.3 Related Works.....</b>	<b>92</b>
<b>6.4 Proposed Algorithm.....</b>	<b>94</b>
<b>6.5 Deobfuscation Method .....</b>	<b>95</b>
<b>6.6 Comparison of Algorithms .....</b>	<b>97</b>
<b>6.7 Deobfuscation Method using PSC Algorithm .....</b>	<b>98</b>
<b>6.8 Deobfuscation Experimental Results .....</b>	<b>99</b>
<b>6.9 Discussion .....</b>	<b>101</b>
<b>6.10 Conclusion .....</b>	<b>103</b>
<b>CHAPTER 7 – SCALABLE SPAM FILTERING SOLUTION USING A STANDARD FRAMEWORK.....</b>	<b>104</b>

---

---

<b>7.1</b>	<b>Introduction.....</b>	<b>104</b>
<b>7.2</b>	<b>Apache Mahout .....</b>	<b>105</b>
<b>7.3</b>	<b>Mahout in Classification.....</b>	<b>106</b>
<b>7.4</b>	<b>Methodology.....</b>	<b>107</b>
7.4.1	Extracting Features to Build a Mahout Classifier.....	107
7.4.2	Preprocessing Raw data into Classifiable data .....	108
7.4.3	Transforming Raw data .....	108
7.4.4	Classifying Spam Mails.....	108
7.4.5	Dataset Pre-processing .....	109
7.4.6	Choosing an Algorithm to Train the Classifier.....	110
<b>7.5</b>	<b>Classifying Enron Spam Data with Naive Bayes.....</b>	<b>111</b>
7.5.1	Data Extraction for Naive Bayes .....	111
7.5.2	Training the Naive Bayes Classifier.....	112
7.5.3	Testing with Naive Bayes Model.....	112
<b>7.6</b>	<b>Classifying with Complement Naive Bayes Classifier .....</b>	<b>113</b>
7.6.1	Testing with Complement Naive Bayes classifier.....	113
<b>7.7</b>	<b>Performance Evaluation .....</b>	<b>115</b>
7.7.1	Time Complexity .....	115
7.7.2	Accuracy .....	117
<b>7.8</b>	<b>Summary.....</b>	<b>119</b>

---

---

---

<b>CHAPTER 8.....</b>	<b>121</b>
<b>8.1 Conclusion .....</b>	<b>121</b>
8.1.1 Spam Filtering .....	121
8.1.2 Incremental Updation .....	122
8.1.3 Deobfuscation.....	123
8.1.4 Scalable Solution .....	123
<b>8.2 Contributions.....</b>	<b>124</b>
<b>8.3 Future Work .....</b>	<b>125</b>
8.3.1 LDA for Email spam filtering.....	127
8.3.2 Web Service.....	127
8.3.3 Spam Filter .....	128
<b>8.4 Summary.....</b>	<b>128</b>
<b>LIST OF PUBLICATIONS.....</b>	<b>144</b>

---

---

## **List of Tables**

Table 1-1: Different Spam Definitions.....	3
Table 1-2: Methods used by spammers to send spam .....	5
Table 2-1: Comparison of Rule based and Learning based methods.....	40
Table 3-1: Distribution of spam corpus.....	45
Table 3-2: Features selected from Dataset 1 .....	48
Table 3-3: Features selected from Dataset 2 .....	48
Table 3-4: Features selected from Dataset 3 .....	49
Table 3-5: Features selected from Dataset 4.....	50
Table 3-6: Features selected from Dataset 5 .....	51
Table 3-7: Features selected from Dataset 6.....	52
Table 3-8: classification results using Bayesian Bernoulli model .....	55
Table 3-9: Classification results using Bayesian Multinomial model .....	57
Table 3-10: Comparison with “Which Naive Bayes” Paper .....	60
Table 3-11 : Comparison of results with Which Naive Bayes Paper .....	61
Table 4-1: Data set Composition.....	70
Table 4-2: Experiment results - Simple VSM .....	71
Table 4-3: Experiment results - VSM with Rocchio Classification.....	71
Table 4-4: Performance and correctness measures.....	72
Table 5-1: Supervised genetic algorithm .....	78

---

Table 5-2 : Sample chromosome construction of email.....	80
Table 5-3: Data set Composition.....	81
Table 5-4: Before feature selection and learning algorithms .....	82
Table 5-5: Before feature selection and learning algorithms .....	82
Table 5-6: Performance Measures before feature selection.....	82
Table 5-7: Dataset and Performance Measures after feature selection.....	83
Table 5-8: Confusion Matrix after feature selection.....	84
Table 5-9: Performance Measures after feature selection .....	84
Table 5-10: After feature selection and genetic learning.....	84
Table 5-11: Confusion Matrix After feature selection and genetic learning ....	85
Table 5-12: Performance Measures after feature selection and genetic learning .....	85
Table 5-13: Classification results after feature selection and K-Means.....	86
Table 5-14: Confusion Matrix after feature selection and K-Means.....	86
Table 5-15: Performance Measures after feature selection and K-Means .....	86
Table 5-16: Comparison of algorithms .....	88
Table 6-1: Main Dictionary .....	94
Table 6-2: Generated words dictionary.....	94
Table 6-3: Comparison of algorithms .....	98
Table 6-4: Common rules found in SpamAssassin .....	99

---

---

---

Table 6-5: SpamAssassin scores before and after Deobfuscation .....	100
Table 6-6 : Scores of obfuscated mails with before and after deobfuscation..	103
Table 7-1: Enron Dataset –Distribution of spam and legitimate mails .....	110
Table 7-2: Enron dataset2with 25% split using Naïve Bayes Model .....	112
Table 7-3: Confusion matrix .....	112
Table 7-4: Statistics using Naïve Bayes model.....	113
Table 7-5: Enron dataset 1 with 50% split using Complement Naïve Bayes .	113
Table 7-6: Confusion Matrix.....	114
Table 7-7: Statistics using Complement Naïve Bayes .....	114
Table 7-8: Time taken for Complimentary Naïve Bayes algorithm.....	115
Table 7-9: Time taken for Naïve Bayes algorithm .....	116
Table 7-10: Accuracy of Complimentary Naïve Bayes algorithm .....	117
Table 7-11: Accuracy of Naïve Bayes algorithm.....	118

## List of Figures

Figure 1-1: Spam Everywhere.....	2
Figure 1-2: Example of Image spam.....	9
Figure 1-3: Attachment spam.....	11
Figure 1-4: Research Approach.....	19
Figure 2-1: Spam Filters .....	21
Figure 2-2: Overview of Spam Filtering process.....	22
Figure 2-3: Decision Tree .....	27
Figure 2-4 : Social Networks.....	36
Figure 3-1: Classification using Naïve Bayes models .....	53
Figure 3-2: Bernoulli model Results .....	56
Figure 3-3: Multinomial model Results .....	57
Figure 3-4: ROC Curves of Bernoulli model and Multinomial model .....	59
Figure 4-1: Different steps in classifying emails using vector space models ....	69
Figure 4-2: Performance and correctness measures .....	72
Figure 5-1 : Data reduction Process.....	75
Figure 5-2: Comparison of Performance.....	87
Figure 6-1: Score calculation .....	96
Figure 7-1: Classification systems .....	106
Figure 7-2: Extracting features.....	107

---

Figure 7-3: Time taken for Complimentary Naïve Bayes algorithm .....	116
Figure 7-4: Time taken for Naïve Bayes algorithm .....	117
Figure 7-5: Accuracy of Complimentary Naïve Bayes algorithm .....	118
Figure 7-6: Accuracy of Naïve Bayes algorithm .....	119



## List of Abbreviations

<b>Abbreviation</b>	<b>Expanded Form</b>
DF	Document Frequency
FP	False Positives
FPR	False Positive rate
IDF	Inverse Document Frequency
IG	Information Gain
LDA	Latent Dirichlet allocation
MTA	Message Transfer Agent
ROC	Receiver Operator Characteristics
PCA	Principal Component Analysis
SVM	Support Vector Machine
TC	Text Classification
TF	Term frequency
TPR	True Positive rate
TP	True Positives
VSM	Vector Space Models
UCE/UBE	Unsolicited Commercial Email/Unsolicited Bulk Email

# Chapter 1

## Introduction

---

Spam, also known as “unsolicited commercial e-mail” or “junk e-mail,” pollutes the communication medium of electronic mail [1]. With the proliferation of direct marketers on the Internet and the increased availability of massive email address mailing lists, the volume of junk mail has grown enormously in the past few years. Recipients of spam have to waste their time for deleting such annoying and possibly disgusting messages. When a user is troubled with a large amount of spam, the chances of overlooking a legitimate message increases. Also spam creates overload on mail servers and Internet traffic. Legislative efforts to curb spam have been ineffective or counter-productive as spam accounts for more than two thirds of the mails received in a year. So dealing with spam is one of the problems that all email users share in common [2].

---

### 1.1 Problems with Spam

For a user, spam is annoying and it is a waste of time and mostly contains spyware, malware and even pornography. The main reasons why unsolicited commercial e-mail is a problem are: Cost shifting, theft, fraud, Consumer perception and Global implications [3]. Each and every resource requires money and spammers misuse the resources like Internet bandwidth, computers’ processing power and Storage capacity.

By disguising the origin of messages and headers of the messages, spammers trick the Internet Service Providers also. The annoyance and frustration caused

by such situations made the Internet users to complain in many discussion forums, that their e-mail addresses are harvested and added to junk mail lists. Such distrust threatens the acceptance and growth of e-commerce among online communities.



Figure 1-1: Spam Everywhere

## 1.2 Definition and General Characteristics of Spam

The actual origin of the word “spam” is derived from the words “spiced ham” (first entry in 1937) for describing a tinned meat made mainly from ham by Hormel Foods Corporation [1], [4]. In the electronic world, “spam” has not an official definition, thus some people consider advertisement via e-mails as “spam”, and others consider “spam” as just all unwanted e-mails in their mailbox.

Actually, there are two official definitions for “spam” [5]: Unsolicited Commercial E-mail (UCE) and Unsolicited Bulk E-mail (UBE) [6]. UCE is explicitly used for commercial messages or in other words advertisements, such as messages with pornographic content, marketing of illegal software, etc. UBE is more broad definition of spam covering messages sent to a large number of recipients who have not opt-in or have opt-out.

### 1.3 Different Spam Definitions

Author(s)	Definition [7]
Vapnic et al. [8]	An e-mail message that is unwanted: Basically it is the electronic version of junk mail that is delivered by the postal service.
Oda and White [9]	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. [10]	Electronic messages posted blindly to thousands of recipients, and represent one of the most serious and urgent information overload problems
Zhao and Zhang [11]	Spam or junk mail, is an unauthorized intrusion into a virtual space - the E-mail box.
Youn and McLeod [12]	Spam as bulk e-mail - e-mail that was not asked for which is sent to multiple recipients
Wu and Deng [13]	Spam e-mails are unsolicited ones sent in bulk unsolicited bulk E-mail with hidden or forged identity of the sender, address, and header information.
Spamhaus [6]	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

Table 1-1: Different Spam Definitions.

#### 1.4 Spammer Methods

In order to send spam, spammers first obtain e-mail addresses by harvesting addresses through the Internet using specialized software. This software systematically gathers e-mail addresses from discussion groups or websites [15]. Other than that spammer is also able to purchase or rent collections of e-mail addresses from other spammers or services providers. Table 1-2 indicates many tricks used by the spammers to avoid detection [7][16].

Methods	Descriptions
Hiding Text	Hide the part of the message that makes it spam by Splitting Words, JavaScript Messages, Pattern Recognition, Dyslexia, Tiny Nonsense, URL One-Liner, One Big Image, Encodings, ASCII Art, Vertical Horizon, See Attached, Bait And Switch
URL Hiding	URL Encoding, Faking It, Copy And Paste
Zombies or Botnets	Compromised PCs on the Internet that sent vast amount of spam, viruses, and malware
Bayesian sneaking and poisoning	Writing spam message so it does not contain any words that are normally used in spam messages, or "poison" the Bayesian filter's database
Social Engineering	Spammers try to trick them into reading spam messages that made it past the filter or into not reporting the spam to the proper authorities. For example: Faking Legitimacy, Sender Unknown
IP address	Borrowing or using an IP address that has a good or neutral reputation

Offshore ISPs	Usage of offshore ISPs that lack in security measures
Third-party mailback software	Use improperly-secured mailback applications on innocent websites
Falsified header information	Add bogus header information to the spam message
Obfuscation	Obscuring the words in spam messages by splitting words or messages using nonsense HTML tags or other 'creative' symbols
Vertical slicing	Writing the spam messages vertically
HTML manipulation	Manipulation of HTML format to avoid detection
HTML encoding	Usage of encoding scheme such as Base64 to turn a binary attachment into plain text characters
JavaScript messages	Placing entire contents of the spam message inside a JavaScript snippet that is activated when the message is opened
ASCII art	Usage of letter glyphs of standard letters to write spam messages Image based Using image to send textual information
URL address or redirect URL	Only add URL address to bypass detection / use expendable "portals" to point to their actual websites
Encrypted messages	Encrypting message where it only decrypted once it reaches the mail box

Table 1-2: Methods used by spammers to send spam

## 1.5 E-Mail Structure

Before beginning any sort of data analysis on the emails, it is important to pre-process all text in the emails first. E-mail messages are divided into two parts: Header information and message body.

Header information or the header field consists of information about the message's transportation which generally shows the following information:

**From:** displays sender's detail such as e-mail address

**To:** displays receiver's detail such as e-mail address

**Date:** displays the date the e-mail was sent to the recipient;

**Received:** intermediary server's information and the date the e-mail message is processed

**Reply to:** reply address

**Subject:** the subject of message specified by the sender

**MessageId:** unique id of the message and others.

The message body contains the message of the e-mail. E-mail messages are presented in plain text or HTML. An e-mail may also have attachments such as graphics, video or other format type and to facilitate these attachments MIME (multipurpose internet mail extension) is used. An example of raw email text is given below.

```
Delivered-To: XXX@gmail.com
Received: by 10.31.73.66 with SMTP id w63csp1188476vka; Mon, 3 Oct 2016
11:30:18 -0700 (PDT)
X-Received: by 10.237.47.5 with SMTP id 15mr21307338qtd.39.1475519418379;
Mon, 03 Oct 2016 11:30:18 -0700 (PDT)
Return-Path: <office_fill006@iol.pt>
Received: from zmail.goiania.go.gov.br (correio.goiania.go.gov.br. [200.199.226.131])
by mx.google.com with ESMTP id 123si282488qkm.316.2016.10.03.11.30.17
for<XXX@gmail.com>; Mon, 03 Oct 2016 11:30:18 -0700 (PDT)
```

Received-SPF: fail (google.com: domain of office\_fill006@iol.pt does not designate 200.199.226.131 as permitted sender) client-ip=200.199.226.131;  
Authentication-Results: mx.google.com;  
spf=fail (google.com: domain of office\_fill006@iol.pt does not designate 200.199.226.131 as permitted sender) smtp.mailfrom=office\_fill006@iol.pt  
Received: from localhost (localhost.localdomain [127.0.0.1]) by zmail.goiania.go.gov.br (Postfix) with ESMTP id 163375F6D4E; Mon, 3 Oct 2016 15:30:17 -0300 (BRT)  
X-Virus-Scanned: amavisd-new at zmail.goiania.go.gov.br  
Received: from zmail.goiania.go.gov.br ([127.0.0.1]) by localhost (zmail.goiania.go.gov.br [127.0.0.1]) (amavisd-new, port 10024) with ESMTP id B5AYbCDCBev1; Mon, 3 Oct 2016 15:30:16 -0300 (BRT)  
Received: from [100.67.149.222] (unknown [162.219.176.101]) by zmail.goiania.go.gov.br (Postfix) with ESMTPSA id 41D705F6D3D; Mon, 3 Oct 2016 15:29:50 -0300 (BRT)  
Content-Type: text/plain; charset="iso-8859-1"  
MIME-Version: 1.0  
Content-Transfer-Encoding: quoted-printable  
Content-Description: Mail message body  
Subject: Business Proposal  
To: Recipients <office\_fill006@iol.pt>  
From: Eric Scott <office\_fill006@iol.pt>  
Date: Mon, 03 Oct 2016 23:59:32 +0530  
Reply-To: ericscott102@gmail.com  
Message-Id: <20161003182951.41D705F6D3D@zmail.goiania.go.gov.br>  
Dear Friend  
I have a very serious and GENUINE business proposal for you in my company (U.S.A Genesis Pharmaceutical Company we need a correspondent of a reliable partner a citizen of India who can help us do this and this will be of a great benefit to all of us. Get back to me if you are interested please contact me through this email address  
Thank You very much.  
DR ERIC SCOTT



As you can see a lot of preprocessing is required to extract the useful information from these raw emails.

## 1.6 Types of Spam

### 1.6.1 Text based Spam

Earlier spam was in the form of text- or html-based emails. Spammers use personalized template emails to deliver their messages and then make use of bulk mailing software for distribution. To block spam, keywords which distinguish the spam from legitimate emails are drawn up and used for the detection of spam.

In order to get rid of this type of filtering, spammers use another method called *Obfuscation*, which is to replace common spam keywords such as 'viagra' to 'vlagra'.

The spam filters also make use of blacklists that contain a list of IP addresses of known spammers or compromised hosts. But this list constantly gets updated because spammers change their IP addresses rapidly.

Botnets (a network of compromised networks) and zombies (the nodes in botnets) are used by spammers to send huge volume of spam mails. The anti-virus industry noticed correlations between the spam industry and botnets. The malware writers also write malicious code to suit their needs.

### 1.6.2 Image Spam

Spam had been mainly text based, but spammers began making use of images to bypass text-based content filtering. Simply by putting text in the image files, spammers were attacking the defenses of most anti-spam solutions because the

text based filters could see only the pixels. OCR can be used to extract keywords from such mails but it is very costly and time consuming task. One example of image spam which exhibits loan details to customers is shown above [17].



Figure 1-2: Example of Image spam

To escape from OCR based email anti-spam solutions, spammers applied CAPTCHA[18] (Completely Automated Public Turing test to tell Computers and Humans Apart), an anti-spam solution that is used on web forums which is a type of challenge-response test used in computing to determine whether or not the user is human. CAPTCHAs are made, by fusing noise and distortions to images to make it even harder for the OCR machine to recognize text. Although it is possible for the machine to read this text, the process is very

CPU intensive – especially when it is handling multitudes of images every few seconds.

Image filters wanted a solution to avoid image spams by isolating the spammers from the source itself. This approach provided positive result and considerably decreased the number of image spam and thus gave users some relaxation from these annoying activities.

### **1.6.3 Attachment Spam**

With the improvements in spam filtering, spammers then started with attachment spam. That is instead of embedding the image within the email itself; package the images within an attachment using one of the most common file formats in use: PDF, Excel or ZIP files [19].

The reasons behind this move are:

- ) Email users expect spam to be an image or text within the body of the email and not an attachment.
- ) Most business communications use PDF format, Excel for spreadsheets, databases, presentations and so on, email users will have to check and open such documents otherwise they may lose important documentation.
- ) Most anti-spam software products in the market aim to filter text based or image based spams; not attachment based spams. This gave spammers an opportunity to fake users.



Figure 1-3: Attachment spam

## 1.7 Taxonomy of Spam

Today people use the word "spam" to mean almost any kind of unwanted email message or news article they receive. A *spammer* is someone who posts or sends spam, and *spamming* is the act of posting or sending spam. Spam does not have any language barriers: Although spam written in English is the most common, it comes in all languages including Chinese, Korean and other Asian languages.

### 1.7.1 Varieties of Spam

In most cases spam is advertising. Spammers are interested to sell or promote some goods or services. Computer users choose some products because he is likely to be interested on those products. Sometimes the items they offer are likely to be fraud or illegal. Hence we can say that spam is illegal because they are using our means to advertise, but also the goods and services being offered

are themselves illegal. Some mass mailings are outright fraud. For example, a recipient is asked to provide their bank account details. If the recipient provides these details, their bank account may be emptied without their consent. This type of spam is usually called 'scam'. Nigerian letters are best example of such scams.

Spammers constantly and periodically extend the range of their offers and are always searching for new ways of attracting innocent users. According to [20], most of spam falls into the following categories:

- ) Fake online Health and Medicine advertisements - This category includes advertisements for weight loss, hair fall, skin care, beauty tips etc.
- ) Computers and the Internet -This category includes offers for low-priced gadgets, mobiles, hardware and software, services for website owners such as hosting, domain registration, website optimization and so on.
- ) Personal finance - Spam which falls into this category offers insurance, mutual fund investments, stock market updates etc.
- ) Adult content - This category of spam includes links and advertisements to porn sites, offers for products designed to sexual potency, etc
- ) Education -This category includes offers for courses, seminars, training and online degree programmes.
- ) Fashion: This category includes news and offers of fashion news, trends, furnishings, shopping etc.
- ) Political spam - This category includes mudslinging or political threats from extremists and possible terrorists. Security and law enforcement officials need to be aware of such mailings, since they can provide clues to genuine potential threats, or may be actual communication between terrorists.

- J Anti-spam solutions - Spammers advertise supposed anti-spam solutions in an effort to cash in on the negative publicity generated by spam itself. which often lead the user to sites where a Trojan will be downloaded to the victim machine, which will then be used for future mass mailings.

## 1.8 Different ways to Send Spam

It's important to distinguish between the different kinds of unwanted messages on the Internet today. The following are the different platforms by which spammers used to send spam emails.

### 1.8.1 Email Spam

*Unsolicited commercial email* (UCE) is just what it sounds like: an email message that you receive without asking for it advertising a product or service. This is also called *junk email*.

*Unsolicited bulk email* (UBE) refers to email messages that are sent in bulk to thousands (or millions) of recipients. UBE may be commercial in nature, in which case it is also UCE. But it may be sent for other purposes as well, such as political lobbying or harassment.

*Make money fast* (MMF) messages [21], often in the form of chain letters or multi-level marketing schemes, are messages that suggest you can get rich by sending money to the top name on a list, removing that name, adding your name to the bottom of the list, and forwarding the message to other people.

*Reputation attacks* are messages that appear to be sent from one person or organization, but are actually sent from another. Reputation attacks constitute wire fraud, since they use forged addresses, and are illegal.

### 1.8.2 Usenet Spam

*Excessive multi-posting* (EMP)[22] refers to an identical news article posted individually to many newsgroups.

*Excessive cross-posting* (ECP) refers to news articles cross-posted to many newsgroups.

*Registration spam* - Spammers use a variety of ways, some manual and some automated, to ask users to register into various forums with valid email address so that they can post their spam.

### 1.8.3 Social Networks Spam

As social networks such as MySpace, Facebook, WhatsApp and Twitter became increasingly popular, spam quickly found a new home. Spammers have a variety of ways to spread spam on social media. Applications that promise special features like revealing the number of users that have seen your Facebook profile but are actually spam apps.

## 1.9 Anti-Spam Measures

Spam is a universal problem, an ongoing issue and one of the most critical problems on Internet. This worldwide issue wastes Internet users' precious time, the misuse of Internet bandwidth, computers' processing power and storage capacity. Furthermore, there are also some hidden and difficult effects due to spam, such as the loss of legitimate e-mails –namely False Positives (FP) effect– the misleading of Internet consumers, exposure to unethical content for children, electronic frauds, etc.

As a matter of fact, a number of countermeasures have been deployed, which are meant to reduce spam phenomenon. In general, there are three anti-spamming approaches: legal, social and technical. Basically, anti-spam efforts are grouped based on where the filters reside and how the filters react against spammer's techniques. In the first case, anti-spamming efforts are distinguished based on whether they reside either on server side of an e-mail service or at user's computer. In the second case, the anti-spamming efforts are complementary to spammers' methodologies. Controlling spam requires an array of complementary techniques and continued efforts to adapt them, as spammers continue to adapt their own methods.

In literature we could find that a lot of works had been carried out to handle spam using different algorithms and techniques. Still spam is an ongoing problem because spam filtering depends on many factors like filtering algorithms used, quality and quantity of spam training set, methods used for deobfuscation, ways to scale or parallelize the filtering task and observing current trends in spam. Although spam filtering and the associated algorithms are much covered in many works, other areas like incremental updation of spam corpus to improve quality of training set, scalable solutions to filtering spam, and algorithms for deobfuscation, are not much discussed. This gap in research and technology motivated us to carry out this research work to provide an integrated solution to spam filtering and incremental updation of spam corpus with a modified spell correction algorithm for deobfuscation.

### **1.10 Motivation**

As explained, spam is a universal problem. Spammers and filters are fighting each other to win the market. For email user's perspective, spam causes several



problems and it is a nuisance. It steals the resources of users such as time, space, bandwidth and sometimes money also. Hence we should be equipped with latest tools and best practices to combat spam.

Previous research results claimed that significant improvements in performance can be achieved through pre-processing. Different processes like tokenization, stop words removal, normalization, Feature Dimensionality Reduction are done in the preprocessing stage. Heuristics knowledge can also be applied by observing current trends in spam. From the literature review done, it is found that Bayesian methods outperform all other spam filtering machine learning algorithms.

Most high-volume spam is sent using some tools which randomizes parts of the message - subject, body, sender address etc. Timely detection of mails using various templates or patterns can be used to easily ignore forthcoming spam. Templates of such mails can be included in the training set to minimize the search volume rather than using every mail in the corpora. A scalable solution is also required to handle this high volume spam received by an email server.

Hence the main research question is:

*How to develop an integrated scalable solution to spam filtering and incremental updation of spam corpus?*

### **1.11 Objectives of Research:**

Based on the main research question and sub-questions arisen, the objectives of research work are enumerated below:

- ) Improve preprocessing using Information Retrieval techniques to make filtering more effective

- ) Filter spam using alternative ways of Bayesian models
- ) Filter spam using vector space models with clustering approach
- ) Study the problem of obfuscation using spell correction algorithms. A modified algorithm is also devised for spell correction
- ) Study the applicability of data mining algorithms in Incremental Clustering of spam dataset
- ) Explore the possibilities of Apache Mahout to classify spam for make it as a scalable solution.

### **1.12 Research Method**

The following are the set of machine learning approaches proposed in this research work for classifying spam and legitimate mails. Based on these algorithms, the efficiency of spam filtering methods were modeled and tested.

### **1.13 Spam Filtering Approaches**

Many machine learning models exist in the world to filter spam and it is evident that Bayesian filters outperform all other spam filtering machine learning algorithms. This research focuses on applying different alternate Bayesian models for spam filtering task. Two models are discussed here; Bernoulli model and Multinomial model. Most of the time, spammers use mail templates for sending spam. Most high-volume spam is sent using such tools which randomizes parts of the message - subject, body, sender address etc. Templates of such mails are only included in the training set and using the advantage of template phenomena in emails vector space models are applied here to filter spam.

### **1.14 Incremental Clustering of Training set**

All the methods explained in the literature and current research depend upon the spam training data. The quality of training data depends on how frequently and efficiently the spam training set is updated. Two approaches can be used for incremental updation of training set:

- ) Adding all incoming emails to spam corpus. But this makes the spam corpus big and filtering process slow.
- ) Cluster the training data and store only the template mails in the spam training set. Check new mails against these centroids and update the training set incrementally.

Genetic algorithm and K-Means algorithm are used here to find an optimum training set to use along with spam filtering task.

### **1.15 Deobfuscation of Mails**

To cheat the filtering mechanisms implemented on mail servers and client programs, spammers obfuscate the words in spam mails. Obfuscation can be done in different ways like changing letters, replacing letters with lookalike letters. The text based filters may not be able to find such words and so cannot filter those mails. Hence we require a system to deobfuscate such mails in order to improve the classification efficiency.

### **1.16 Scalable and Standard framework for Spam filtering**

Most of the methods are implemented on personal experimental setup. No standard framework or software is used in these analyses. In order to achieve good results and benchmark solutions, we require a set of standard algorithms

and to handle this high volume, high velocity and large varieties of spam, a scalable solution is required.

In order to be successful in spam filtering, the spam filters have to embrace multiplicity. Multiplicity refers to: multiple skills in the decision-making, multiple algorithms, and multiple tools, to handle multiple types of spam.

### 1.17 Flowchart of the System

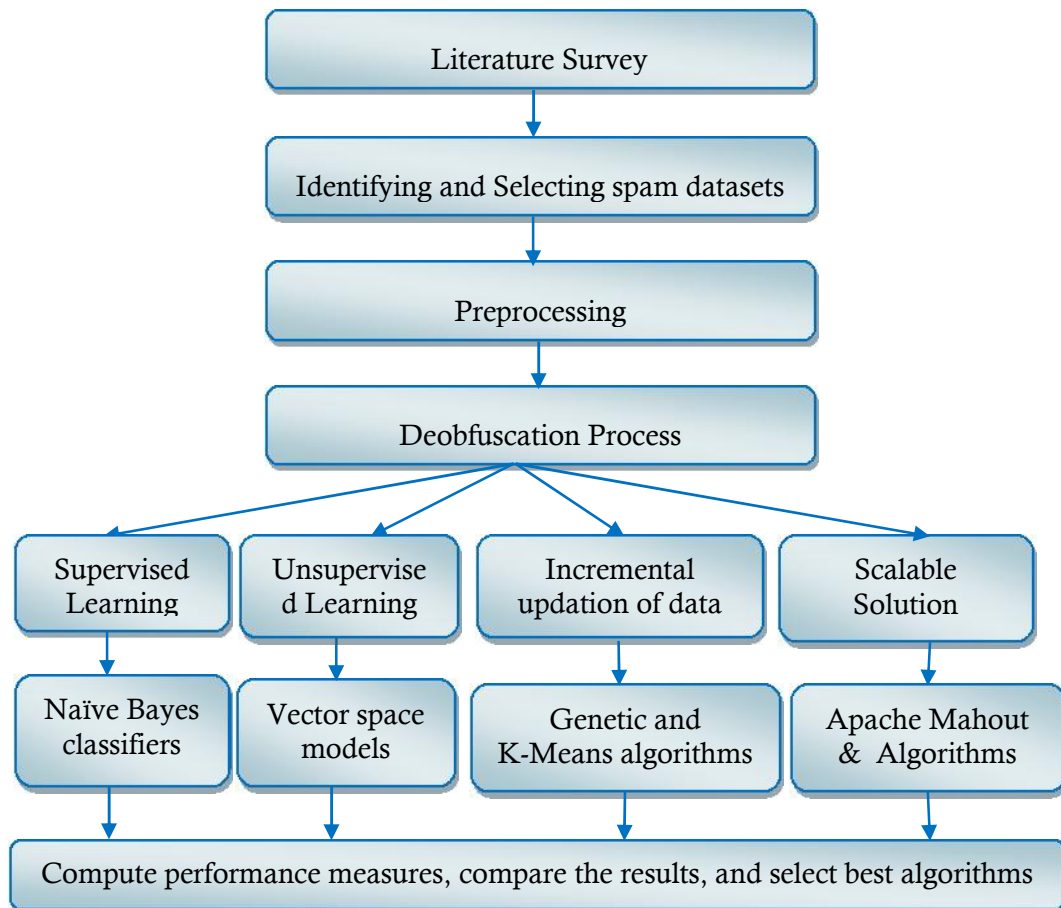


Figure 1-4: Research Approach

### **1.18 Organization of the Thesis**

Chapter 1 introduces spam and its associated problems, spammer's tricks, taxonomy of spam, different types of spams, importance of spam filtering, gaps along with motivation and research problem

Chapter 2 contains literature review of the works in spam filtering. This review outlines an overview of various spam filtering approaches based on SMTP protocols, machine learning algorithms, legal efforts in text, image and attachment spam. Literature review of incremental updation and standard frameworks are also included in this chapter.

Chapter 3 describes alternate Bayesian models in spam filtering and evaluation results are shown

Chapter 4 describes the vector space models in template based spam filtering

Chapter 5 discusses and evaluates two learning algorithms in the task of incremental updation of spam training set to improve its quality.

Chapter 6 describes deobfuscation of mails to protect the spam filter from misguiding.

Chapter 7 deals with Apache Mahout Framework in the context of spam filtering. Time and accuracy were computed and results are shown in this chapter.

Chapter 8 includes the summary of the research work carried out, important contributions and details of possible future directions of work in this field.

## Chapter 2

### Literature Review

This chapter provides a literature review of existing works using data mining and other techniques for the process of spam filtering. This chapter also gives a general description of the algorithms used for the classification of spams, spell correction algorithms and big data frameworks for spam filtering. The drawbacks found in existing methods and a summary is also provided in this chapter.

#### 2.1 Introduction

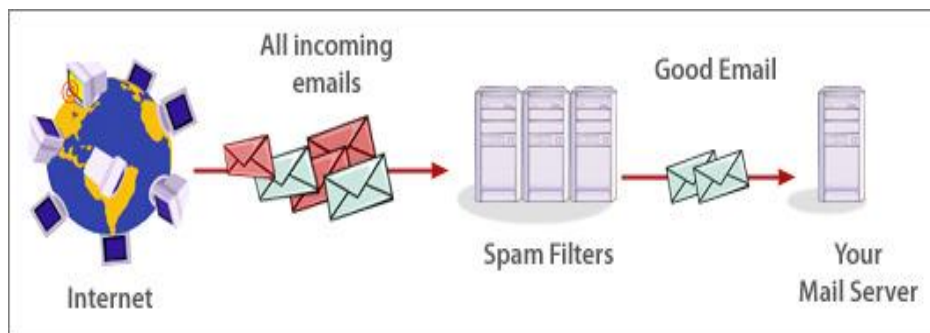


Figure 2-1: Spam Filters

Spam filtering is a real-world application of automated text classification task [23], a field that has undergone intensive research in recent years. The early approaches to text classification were to manually construct document classifiers with rules compiled by domain experts. This is appropriate when few

machine-readable texts were available and the computational power was expensive. An overview of spam filtering is shown in the following figure [24].

### 2.1.1 Spam Filtering as Text Categorization

Recent trends in the Text Categorization community have shifted to building classifiers automatically by applying some machine-learning algorithms to a set of pre-classified documents (training data). This is also called the statistical approach, in the sense that differences among documents are usually expressed statistically as the likelihood of certain events, rather than some heuristic rules written by human. This trend is reflected in the goal of statistical spam filtering, which aims at building effective spam filters automatically from email corpus.

## 2.2 An Overview of Approaches to Spam Filtering

Different approaches are practiced by mail servers and end users to prevent spam or stopping spam at some levels. These prevention methods are explained below [25].

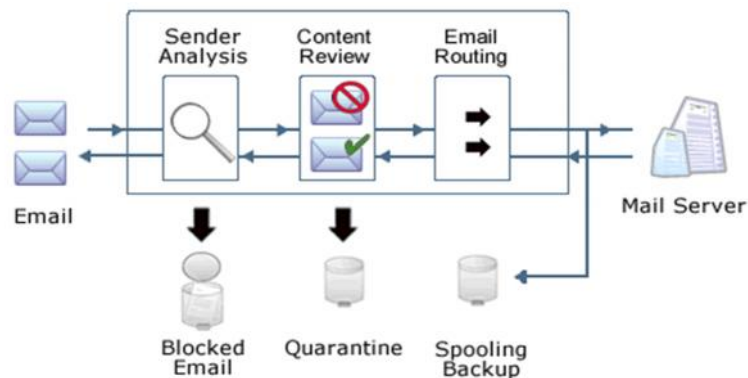


Figure 2-2: Overview of Spam Filtering process

### 2.2.1 Some Prevention Methods from User side

**a. Protect e-mail addresses:** Spammers are trying to harvest e-mail addresses by using a number of methods, such as dictionary attacks to mailer and collect them from web sites. Robots and crawlers can quickly gather thousands of emails at a time from websites where the email addresses are made public. Also, sometimes humans actually grab e-mails from websites to use them for sign-up offers. Preventing an e-mail address to be listed from spammers is mainly a set of directives and techniques that users can adopt. For instance, Make your email address unscannable by masquerading of e-mail addresses by replacing the “at” symbol (@) or the “dot” (.) or using image picture of your e-mail address or using JavaScript to dynamically construct the display of your email.

**b. Prevent spam from being sent:** It is very common for spammers to use compromised/hijacked computers (also called as zombie) all around the world in order to send unsolicited messages. Preventing spam from being sent involves not only the regular checks of computers’ security holes, but also blocking of SMTP and proxy relays.

**c. Block spam to be delivered:** To refuse delivery of spam messages, Internet Services Providers adopt different techniques. Some of them are checking sender authentication such as Sender Policy Framework (SPF) [26] and blocking exploited sources of known spammer based on IP or DNS [27]–[30].

**d. Identify and separate spam after delivery:** Two methods have been found more effective: i) the analysis that is based on the targeted link analysis and ii) Bayesian filters that are relying on adaptive filtering algorithms.



**e. Report Spam:** Before delete spam, report that mail as spam to the mail server. This will block further mails from that sender and will report spammers to ISPs and government agencies.

As already mentioned, the technology-based techniques as well as heuristics based techniques are used to combat spam from user side.

### **2.2.2 SMTP Approaches**

SMTP based approaches to identify and filter spam can be classified into two categories: those based on characterizing the properties of the sending SMTP server, and those based on analyzing the contents of the email [31]. They are also called pre-acceptance and post-acceptance tests, respectively. Pre-acceptance tests can be further classified into two categories: those based on the reputation of IP address, i.e., IP reputation list filters, and those based on the characteristics of individual SMTP transactions, e.g., envelope from addresses, recipient addresses, and HELO/EHLO messages.

Examples of the former approach are DNS Blacklists (DNSBL) [30], DNS Whitelists (DNSWL) [29], and other commercial IP reputation services such as [6] and [32]. Examples of the latter approach are grey listing [28], sender authentication [26], DNS validation [33], domain validation [30] and protocol defects [34].

### **2.2.3 Rule based Techniques or Heuristic Filtering**

A rule-based approach expresses the domain knowledge in terms of a set of heuristic rules. A set of rules is applied to a message and a score which represents the possibility of being spam mail accumulates based on these rules.

The message is categorized as spam or legitimate, according to the specific score threshold.

A content based heuristic filter is a set of hand coded rules that analyze the contents of a message and classify it as spam or legitimate [35]. However, as new types of spam emerge and spammers alter content and behavior to avoid detection, the filters need hundreds of rules and these rules need to be updated regularly [36]

Association rule mining algorithms like Apriori [37], [38] are used to generate rules by analyzing the association of keywords with the spam and legitimate mails.

#### **2.2.4 Anti-spam Legislation Efforts**

Fighting spam requires uniform international laws, as the Internet is a global network and only uniform global legislation can combat spam. Legal Methods like Prohibition, Enforcement of Anti-Spam policies, Opt-out clause, other Statutory Provisions and Enforcement Mechanisms can be implemented. A number of nations have implemented legal measures against spam.

Spam legislation is non-existent in India [39]. The much-awaited Information Technology Act of 2000 does not discuss the issue of spamming at all. It does not have any bearing on violation of individual's privacy in Cyberspace. The illegality of spamming is not considered. However, in the absence of stringent laws and technical advancements, the proliferation of spam seems inevitable.

A trustable spammers' hall of fame is maintained by The SpamHaus Project, and it is known as the Register of Known Spam Operations (ROKSO)[40].

### **2.2.5 Machine Learning & Statistical Spam Filtering Techniques**

A statistical-based approach expresses the differences among messages in terms of the likelihood of certain events. In general, a spam filter is an application which implements a function:

$$f(m, \theta) = \text{Spam, if the message } m \text{ is considered spam}$$

Legitimate, if the message  $m$  is considered legitimate mail

where  $m$  is a message to be classified,  $\theta$  is a vector of parameters, and ‘Spam’ and ‘Legitimate’ are labels assigned to the messages.

Numerous machine-learning algorithms exist, including Decision Trees, Bayesian classifiers, k Nearest Neighbor (kNN), Artificial Neural Networks (ANN), and SVM.

#### **2.2.5.1 Decision Trees**

The decision tree is one of the most famous tools of decision-making theory [41]. When classifying an unknown instance, the unknown instance is routed down the tree according to the values of the attributes in the successive nodes. C4.5 is one of the most popular decision trees algorithms. Some of the experiments done on spam filtering using decision tree are explained in [42]–[44].

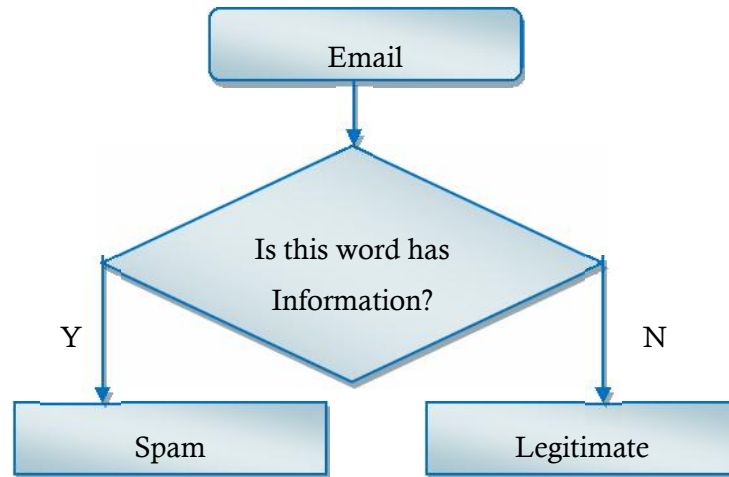


Figure 2-3: Decision Tree

### 2.2.5.2 Maximum Entropy Model

Maximum Entropy (ME) models have been successfully applied to various Natural Language Processing tasks including sentence boundary detection, part-of-speech tagging, prepositional phrase attachment and adaptive statistical language modeling with the state-of-the-art accuracies [45],[46], [47].

### 2.2.5.3 Memory-Based Learning

Sakkis, G et.al [48] investigated different attribute and distance-weighting schemes, and studied on the effect of the neighborhood size, the size of the attribute set, and the size of the training corpus. Three different cost scenarios are identified, and suitable cost-sensitive evaluation functions are employed. They concluded that memory based anti-spam filtering for mailing lists is practically feasible, especially when combined with additional safety nets.

Androutsopoulos et. al [49] investigate thoroughly the performance of the Naive Bayesian filter on a publicly available corpus, contributing towards standard benchmarks. They compared the performance of the Naive Bayesian filter to an alternative memory-based learning approach, after introducing suitable cost-sensitive evaluation measures. Both methods achieved very accurate spam filtering and keyword-based filter outperformed clearly.

El-Sayed M et. al [50] discuss about various learning algorithms that have been applied to this problem and survey the related work. They presented a case study to compare the performance of a number of these learning methods on one of the publicly available datasets.

#### **2.2.5.4 Artificial Neural Networks**

According to Neural Network theory, for static pattern classification the best performance shows the layered feed forward networks, called Multilayer Perceptrons (MLPs), typically trained with static back propagation. Their main advantage is that they are easy to use, and that they can approximate any input/output map. The key disadvantages are that they train slowly, and require lots of training data.

Puniškis, D et.al [51] applied a neural network (NN) approach to the classification of spam in this paper. The method employs attributes comprised from descriptive characteristics of the evasive patterns that spammers employ rather than the context or frequency of keywords in the messages. They find out which ANN configuration will have the best performance and least error to desired output. Khorsi, A [52] gives an overview of different Content-Based Spam Filtering Techniques especially on ANN.

### **2.2.5.5 Support Vector Machine**

SVMs have out-performed other learning algorithms with good generalization, global solution, number of tuning parameters, and its solid theoretical background. The state of the art of SVMs evolved mapping the learning data from input space into higher dimensional feature space where the classification performance is increased. This has been developed by applying several kernels each with individual characteristics.

Drucker, H et al. [8] studied the use of support vector machines (SVM's) in classifying e-mail as spam or non-spam by comparing it to three other classification algorithms: Ripper, Rocchio, and boosting decision trees. These four algorithms were tested on two different data sets: one data set where the number of features were constrained to the 1000 best features and another data set where the dimensionality was over 7000. SVM's performed best when using binary features. For both data sets, boosting trees and SVM's had acceptable test performance in terms of accuracy and speed. However, SVM's had significantly less training time.

Joachims, T. [53] uses Transductive Support Vector Machines (TSVMs) for text classification. While regular Support Vector Machines (SVMs) try to induce a general decision function for a learning task, Transductive Support Vector Machines take into account a particular test set and try to minimize misclassifications of just those particular examples. The paper presents an analysis of why TSVMs are well suited for text classification. These theoretical findings are supported by experiments on three test collections. The experiments show substantial improvements over inductive methods, especially for small training sets, cutting the number of labeled training

examples down to a twentieth on some tasks. This work also proposes an algorithm for training TSVMs efficiently.

Amayri, O et. al [54] detail feature mapping variant in text classification (TC) that yields improved performance for the standard SVM in filtering task. Furthermore, they propose an online active framework for spam filtering. They described the use of string kernels in order to improve spam filter performance.

#### **2.2.5.6 Bayesian Classifiers**

Sahami, M et. al [49], [55] in their classic paper examined the methods for the automated construction of filters to eliminate spam mails. The authors have found that it is possible to automatically learn effective filters to eliminate a large portion of such junk from a user's mail stream. The efficacy of such filters can also be greatly enhanced by considering not only the full text of the E-mail messages to be filtered, but also a set of hand-crafted features which are specific for the task at hand.

Vangelis Metsis et. al [49] discussed five different versions of Naive Bayes, and compared them on six new, non-encoded datasets, that contain legitimate messages of particular Enron users and fresh spam messages. They adopted an experimental procedure that emulates the incremental training of personalized spam filters, and roc curves are plotted to compare the different versions of NB over the entire tradeoff between true positives and true negatives.

Androutsopoulos et.al [56] conducted a thorough evaluation on a publicly available corpus and investigated the effect of attribute-set size, training-corpus size, lemmatization, and stop-lists on the filter's performance, issues that had not been previously explored. After introducing appropriate cost-sensitive

evaluation measures, the authors concluded that additional safety nets are needed for the Naive Bayesian anti-spam filter to be viable in practice.

Chen, C et. al.[57] reported their work on spam filtering with three novel Bayesian classification methods: aggregating one-dependence estimators (AODE), hidden Naive Bayes (HNB), locally weighted learning with Naive Bayes (LWNB). Other four traditional classifiers: Naive Bayes, k nearest neighbor (kNN), support vector machine (SVM), C4.5 are also performed for comparison. Four feature selection methods: gain ratio, information gain, symmetrical uncertainty and Relief, are used to select relevant words for spam filtering. Results of experiments on two corpora show the promising capabilities of Bayesian classifiers for spam filtering, especial for that of AODE.

#### **2.2.5.7 Boosting**

Several variants of the AdaBoost algorithm with confidence-rated predictions have been applied in Carreras, X et. al.[58], which differ in the complexity of the base learners considered. Two main conclusions drawn from their experiments are the boosting-based methods clearly outperform the baseline learning algorithms and increasing the complexity of the base learners allows to obtain better “high-precision” classifiers.

Zhang, L et. al. [59] evaluates five supervised learning methods in the context of statistical spam filtering. They found support vector machine, AdaBoost, and maximum entropy model are top performers in this evaluation, sharing similar characteristics: not sensitive to feature selection strategy, easily scalable to very high feature dimension, and good performances across different datasets.



#### **2.2.5.8 Ensemble Methods**

Stacking or Ensemble is an approach for constructing classifier ensembles. A classifier ensemble, or committee, is a set of classifiers whose individual decisions are combined in some way to classify new instances. Stacking combines multiple classifiers to induce a higher-level classifier with improved performance. The latter can be thought of as the president of a committee with the ground-level classifiers as members. Each unseen incoming message is first given to the members; the president then decides on the category of the message by considering the opinions of the members and the message itself.

Sakkis et. al [48] examined a combined memory-based and a Naïve Bayes classifier in a two-member committee, in which another memory-based classifier presided. The classifiers have been evaluated individually on the same data as, i.e. the Ling-Spam corpus.

Delany, S. J et. al. [60] compared the ensemble approach to an alternative lazy learning approach to concept drift whereby a single case-based classifier for spam filtering keeps itself up-to-date through a case-base maintenance protocol. The case-base maintenance approach offers a more straightforward strategy for handling concept drift than updating ensembles with new classifiers. The results shows that the ensemble approaches can have very good performance but this comes at considerable cost to the overall accuracy.

#### **2.2.5.9 Artificial Immune System Inspired Behavior-based Anti-spam Filter**

Yue, X et. al. [61] proposes a novel behavior-based anti-spam technology for email service based on an artificial immune-inspired clustering algorithm. The

suggested method is capable of continuously delivering the most relevant spam emails from the collection of all spam emails that are reported by the members of the network. The work discusses on behavior-based characteristics of spam and then identifying similar groups of spam based on immune-inspired clustering algorithm. From the experiment results, the new approach could be used in conjunction with other filtering systems.

Bezerra, G. B et. al [62] presented a model in which antibody network is generated automatically from the training dataset and evaluated on unseen messages. The authors validated this approach using a public corpus, called PU1, which has a large collection of encrypted personal e-mail messages containing legitimate messages and spam.

Oda, T et. al [9], [63] undertakes an extended examination of the spam-detecting artificial immune system focusing on comparison of scoring schemes, the effect of population size, and the libraries used to create the detectors.

### **2.2.6 Filtering approaches to Image Spam**

Image spam consists of embedding the spam message into attached images to defeat techniques based on the analysis of e-mails' body text, and in using content obscuring techniques to defeat OCR tools. This suggests that computer vision and pattern recognition techniques will play a prominent role in the development of the next generation spam filters.

Image spam poses a great threat to email communications due to high volumes, bigger bandwidth requirements, and higher processing requirements for filtering.

Biggio, B [64], [65] propose an approach to recognize image spam based on detecting the presence of content obscuring techniques, and describe a possible implementation based on two low-level image features.

Lee, M. G [66] devises a method of detecting spam images in emails by determining if the compressed forms of the extracted images are identical to the compressed form of any known spam image from a corpus of known spam images.

A study by Wakade, S. V [67] attempts to understand the techniques used in spamming and identifying a set of features that can help in classification of image spam from photographs. A set of eight features were identified based on observations and existing research in this area are Luminance of image, Number of colors, Color saturation, White pixel concentration, Standard deviation of colors and Hue of the image.

The characteristics of image spam, which uses the visual features for classification, are used as features for classifiers like SVM and for near duplication detection in images involves clustering of image GMMs (Gaussian Mixture Models) based on the Agglomerative Information Bottleneck (AIB) principle, using Jensen-Shannon divergence (JS) as the distance measure are studied in [68][65].

### **2.2.7 Filtering approaches to Attachment Spam**

As mentioned in the Chapter 1, instead of embedding the image within the email itself, spammers repackage spam messages within an attachment using one of the most common file formats in use: PDF, Excel, ZIP, etc. A detailed study on this aspect is not done widely, since this is upcoming way of spams.

## **2.2.8 Filters based on Non-content Features**

### **2.2.8.1 Analyzing SMTP path**

Most systems of domain authentication suggest combining domain authentication with reputation services. Algorithms are devised for learning the reputation of email domains and IP addresses based on analyzing the paths used to transmit known spam and known good mail. The algorithms provide the reputation information needed to combine with domain authentication to make filtering decisions effective [69].

SMTP Path Analysis works by learning about the spamminess or goodness of IP addresses by analyzing the past history of e-mail sent using that IP address.

Ramachandran, A et. al [70] studies the network-level behavior of spammers, including: IP address ranges that send the most spam, common spamming modes (e.g. , BGP route hijacking, bots), how persistent across time each spamming host is, and characteristics of spamming botnets. The trends suggest that developing algorithms to identify botnet membership, filtering email messages based on network-level properties (which are less variable than email content), and improving the security of the Internet routing infrastructure, may prove to be extremely effective for combating spam.

### **2.2.8.2 Analyzing the User's Social Network Behaviors**

Social networks are useful for judging the trustworthiness of outsiders. An automated anti-spam tool exploits the properties of social networks to distinguish between spam and legitimate messages associated with people the user knows [71]. Social networks can be used to create White lists, blacklists and grey lists. Global social email networks possess several properties that can

be exploited using recent advances in complex networks theory to provide an efficient collaborative spam filter [72][73].



Figure 2-4 : Social Networks

### 2.3 Incremental Updation of Spam Training set

Accuracy and recall of prediction algorithms depend mainly on the training data set. As velocity of spam is tremendous, anti-spam measures require much attention in timely updation of spam corpus. Although this updation is done in many filters by default, it is not mentioned in scholarly literature.

### 2.4 Spell Correction Algorithms

From the literature reviewed, it is found that three algorithms are mostly used in spell correction, they are:

1. Damerau-Levenshtein distance [74], [75] is the distance between two strings by counting the minimum number of operations needed to transform one string into the other, where an operation is defined as an insertion, deletion, or substitution of a single character, or a transposition of two adjacent characters. Smaller the distance, similar the words are.
2. Peter Norvig Algorithm [76] generates all the possible combinations of the input term with an edit distance  $\leq 2$  and then search each term with the dictionary. It is better than the first method, but still expensive and language dependent.
3. Symmetric Delete Algorithm [77], [78] generates terms with an edit distance  $\leq 2$  (deletes only operation) from each dictionary term and add them together with the original term to the dictionary. This has to be done only once during a pre-calculation step. Then it generate terms with an edit distance  $\leq 2$  (deletes only operation) from the input term and search them in the dictionary. This algorithm claims that it is 1000 times faster than the second algorithm. Here the algorithm is language independent.

## **2.5 Deobfuscation of Spam**

Emails involve some sort of fraud which is centered on obfuscation techniques to hide real words in the message from spam filters but giving the readers same visual look of the real words. In response to these fraud problems, researchers have developed many methods to fight against obfuscated spam emails.

Changwei Liu and Sid Stamm [79] demonstrate how Unicode polymorphism can be used to circumvent spam filters such as SpamAssassin and describe a deobfuscation technique that can be used to catch messages that have been obfuscated. Freschi, Vet. al [80] proposed a new technique for filtering obfuscated email spam that performs approximate pattern matching both on the original message and on its phonetic transcription.

Obfuscation was used in a different way by Eggendorfer, T. et.al [81] to obfuscate email addresses in the www and presented experimental results that indicate the usefulness of obfuscation. Prabaharan Poornachandran et. al [82] evaluated the effectiveness of different techniques to obfuscate an email address and analyze the frequency at which spam mails arrive for each obfuscation technique.

CRAIN, J et.al [83] developed an email client plugin to aid in the prevention of phishing by combining automatic and transparent email signing. They assert that this plugin can detect unsigned spoofed messages and the user is prevented from visiting malicious web sites.

## **2.6 Scalable Spam Filtering solutions using Standard Frameworks**

Apache – Mahout is a set of scalable algorithms to carry out the clustering and classification in big data arena problem free [84][85]. Mahout is used as a machine learning tool when the collection of data to be processed is very large, or too large for a single machine [86]. Mahout algorithms are written in Java, and some portions are built upon Apache's Hadoop distributed computation project [87]. It doesn't provide a user interface; but a framework of tools intended to be used and adapted by developers [88].

In the book [89], the authors explain how mahout can be used to build and personalize effective classifiers. Different data mining and machine learning models are explained with examples. The book discusses classification and its applications and what algorithms and classifier evaluation techniques are supported by Mahout.

The paper [90] compares k-means and fuzzy c-means for clustering a noisy realistic and big dataset. They made the comparison using a free cloud computing solution Apache Mahout/ Hadoop and Wikipedia's latest articles. The authors claim that in a noisy dataset, fuzzy c-means can lead to worse cluster quality than k-means. They concluded that Mahout is a promise clustering technology but is premature.

The study [91] uses Apache Mahout for Collaborative Filtering and conclude that it is a mature framework for building recommenders, still a lot of room for improvements and extensions. An ideal situation to evaluate an e-commerce recommender systems, the study [92] suggests to find an open-source platform with many active contributors that provides a rich and varied set of recommender system functions that meets all or most of the baseline development requirements.

## **2.7 Pros and Cons of Rule based versus Learning based approaches**

Rule based and Machine learning based methods are the two major approaches for filtering spam mails from mails in the form of natural language texts. The pros and cons of these approaches are listed in the following table.



Sl.No	Characteristic	Approaches	
		Rule based approaches	Learning based approaches
1.	Algorithm Approach	Rule-Driven	Data-driven
2.	Scaling	Management of rules become complicated	Attribute wise properties are stored
3.	Generalization	Very poor	Good
4.	Performance	Excellent performance with narrow domain	Good performance with large datasets
5.	Overfitting	Largely unavoidable	The large problem that machine learning aims to avoid
6.	Implementation overhead	Extremely labor/system intensive to create rules, test rules before accepting	Not trivial to implement, but a lot of powerful tools like scikit-learn, Weka, etc. are available.

Table 2-1: Comparison of Rule based and Learning based methods

## 2.8 Literature Review Summary

From the literature study done, the following points are revealed:

- ) Some of the techniques explained in the literature, like Bayesian filters, are well suited for filtering spam.

- ) Bayesian filters can be improved – there is scope to improve the output by improving the tasks in pre-processing step.
- ) For effective filtering, training data needs to be up to-date with most dissimilar features. Incremental updation of data is required to cope with new spam phenomenon. The training data needs to be updated with templates of new spams.
- ) Since the volume, variety and velocity of mails coming to servers and user inboxes are BIG, a scalable and standard framework is needed to handle spam. Analysis and studies in big data context of spam filtering are very few in literature.

## Chapter 3

### Spam Filtering using Bayesian models

---

From the literature reviewed, it is found that Bayesian models are performing well in the process of classification. In this chapter, implementation of spam filter by utilizing the publically available data set of spam and legitimate email is addressed. Dimensionality reduction is implemented by using Principal component Analysis and Information gain methods. Two Bayesian models are applied to the reduced data set, the system is tested with 10-fold cross-validation and overall accuracy of the system is found to be 100% for Bernoulli model.

---

#### 3.1 Introduction

Treating e-mail filtering as a binary text classification problem, researchers have applied several statistical learning algorithms to email corpora with promising results. This study examines the performance of two variants of Naive Bayes classifier with two different feature selection approaches and tokenization on different corpus sizes.

Naive Bayes has several advantageous properties than other algorithms due to their simplicity, linear computational complexity, and their accuracy. This classifier can be constructed by a single scan through the training data and classification requires just a single table lookup per token, plus a final product or sum over each token. Most of the other approaches require iterated

evaluation. Storage requirements are small in Naive Bayes because we need to store only the token counts, rather than whole messages. The classifier can be updated incrementally as new messages arrive. This study examines two variants of Naive Bayes text classification algorithm. Each method makes the independence assumption that the probability of tokens occurring in a message is independent.

### 3.2 Models for NB classifier

Among different ways to setup an NB classifier, Multinomial[93] and Bernoulli models are analysed in this study. Multinomial model generates one term from vocabulary in each position of the document, and assumes generative model. Bernoulli model generates an indicator for each term of the vocabulary, 1 for the presence of term and 0 for the absence. The Bernoulli model has the same time complexity as Multinomial model.

#### 3.2.1 Algorithm - Naïve Bayes Classifier

Applying Bayes theorem in spam filtering context,

$$P(\text{class}|\text{email}) = \prod P(c_i | t_{i-1})$$

For each token find:

$$P(c_i | t_{i-1}) = \frac{P(t_i | c_i) \times p(c_i)}{P(t_i)}$$

where

*Class* = {Spam, legitimate}    *Email* = {set of tokens}

$P(\text{Class} \mid \text{Email})$  = Probability of the email to be spam/legitimate given the tokens in an email

$P(\text{token} \mid \text{Class})$  = Probability of the tokens to be spam/legitimate in emails (computed from training set)

$P(\text{Class})$  = Probability of the spam and legitimate emails

$P(\text{token})$  = Probability of the tokens.

*We can ignore this value since it is same for all classes.*

Our goal is to find a class with highest posterior probability

$$a \quad c \quad P(\text{class}|\text{email}) = a \quad c_a \prod P(c_i | t_{o_i})$$

Spam filtering is a two class problem and the classes are spam and legitimate. The Naïve Bayes assumption is that tokens are conditionally independent of one another, given the class.

### **3.3 Spam Training set**

The Enron data set is used in the Bayesian classification methods in this research work. This dataset was collected and prepared by the CALO Project (A Cognitive Assistant that Learns and Organizes). It contains data from about 150 users, mostly senior management of Enron, organized into folders. The corpus contains a total of about 0.5M messages. This data was originally made public, and posted to the web, by the Federal Energy Regulatory Commission during its investigation.

The original raw Enron corpus contains 619,446 messages belonging to 158 users. The preprocessing of Enron raw dataset is done by V. Metsis, I.

Androutopoulos and G. Paliouras for their research work and it is publicly available at <http://www.aueb.gr/users/ion/data/enron-spam/>.

The "preprocessed" subdirectory contains the messages in the preprocessed format that was used in the experiments of the paper. Each message is in a separate text file. The number at the beginning of each filename is the "order of arrival". During the preprocessing, legitimate and/or spam messages were randomly sub sampled to obtain the desired legitimate-spam ratios.

<b>Corpus</b>	<b>Number of messages</b>	<b>Distribution</b>
Spam	17171	51%
Legitimate	16545	49%
Total	33716	100%

Table 3-1: Distribution of spam corpus

Since the Enron corpus has become the standard legitimate email corpus amongst researchers, the results we present in this thesis were obtained using the cleaned version of this corpus.

### **3.3.1 Tokenization**

While tokenising the following factors are considered:

- a) Extract header information: - From, time, subject and body of the message. The body is unrestricted in its content.
- b) Attachments are ignored
- c) HTML tags are stripped off from the message.
- d) Stop words removed, No stemming applied

- e) Hyphens are replaced with space character, Special characters are (@,\$,!) are retained. Punctuation marks are ignored.
- f) Finally 'space' is used as the delimiting character to tokenize <sup>[5]</sup>.

### **3.3.2 Transforming to Term Frequency**

Each mail is considered as a single document. In the experiments, each message is represented as a vector  $(t_1, t_2, \dots, t_m)$ , where  $t_1, \dots, t_m$  are the values of attributes  $T_1, \dots, T_m$  and  $m$  is the number of tokens. In the Bernoulli model, the values for all attributes are Boolean:  $X_i = 1$  if the message contains that token; otherwise  $X_i = 0$ . In multinomial model, attribute values are term frequencies (TF), showing the token frequency. Attributes with TF values carry more information than Boolean ones. A third alternative is called normalized TF, is to divide term frequencies by the total number of token occurrences in the message.

### **3.3.3 Transforming by Document Frequency**

The document frequencies /collection frequency are computed and for feature selection (to create the final term-frequency matrix), only the tokens with document frequency greater than 10(for Bernoulli) and collection frequency greater than 10(for multinomial) are considered for training. This number is selected by heuristics, for initial reduction of dimensionality.

### **3.3.4 Feature Selection**

Initially 227 attributes were chosen from dataset 1 using the frequency model because of its simplicity. Frequency is defined as the document frequency for Bernoulli model and collection frequency for the multinomial model. Then

two commonly available methods were considered for feature selection/dimensionality reduction and they are: Principal component analysis (PCA)[94]–[96] and Information Gain.

IG is one of the important feature selection techniques. It measures the importance of features globally based on the decrease in entropy after a dataset is split on an attribute and top ranked features can be selected based on reduction in the uncertainty. This reduced feature set is used for better classification results. PCA is a feature reduction technique that transforms high dimensional feature vector into lower dimensional such that maximum variance is extracted from the data. PCA is accomplished by projecting the data onto the largest eigenvectors of its covariance matrix. Before eliminating features, complete feature space is transformed such that the underlying uncorrelated components are obtained. For reducing the size of the feature vector top k principal components are selected. This exercise was done on all the six datasets and the results are shown below.

Data sets	Dimensionality reduction methods		Attributes selected
	PCA	Information Gain	
Data set 1	Account, better, Charset, Cialis, Online, overnight, paliourg, pharmacy, prescription, prices, quality, shipping, software, valium, viagra, vicodin, windows	Vicodin, Valium, Viagra, overnight, Paliourg, cialis, Better, quality, Pharmacy, charset, Shipping, account, Software, prescription, Windows, prices, computer	Account, better, Charset, Cialis, Online, overnight, Paliourg, pharmacy, prescription, prices, quality, shipping, software, valium, Viagra, vicodin, windows



Number of attributes in dataset after freq. based dimensionality reduction : **227**  
 Number of records: **5172**  
 Number of attributes selected after using the above models : **17**

Table 3-2: Features selected from Dataset 1

Data sets	Dimensionality reduction methods		Attributes selected after dimensionality reduction
	PCA	Information Gain	
Data set 2	Better, charset, delivery, Failed, getting, Graphics, guaranteed, Identity, languages, Localized, mortgage, perfect, returned, Sender, transcript, Viagra	Better, charset, failed, Getting, graphics, Guaranteed, identity, Languages, localized, Mortgage, perfect, Returned, sender, Transcript, viagra	Better, charset, Delivery, failed, Getting, graphics, Guaranteed, identity, languages, Localized, mortgage, perfect, returned, Sender, transcript, viagra
Total number of attributes in dataset after frequency based dimensionality reduction : <b>286</b> Total number of records: <b>5857</b> Total number of attributes selected after feature selection using the above models : <b>16</b>			

Table 3-3: Features selected from Dataset 2

Data sets	Dimensionality reduction methods		Attributes selected after dimensionality reduction
	PCA	Information Gain	
Data set 3	Account, charset, Cialis, Excelled, graphics, greatest, inexpensive, medication, medicine, prescription, tablets, Viagra, winning	Account, charset, Cialis, excelled, generic, graphics, Greatest, inexpensive, medication, medicine, prescription, quality, Software, tablets, Unable, Viagra, winning	Account, charset Cialis, excelled, Graphics, greatest, Inexpensive, medication, medicine, prescription, quality Software, tablets, Viagra, winning
<p>Total number of attributes in dataset after frequency based dimensionality reduction : 296</p> <p>Total number of records: 5511</p> <p>Total number of attributes selected after feature selection using the above models : 15</p>			

Table 3-4: Features selected from Dataset 3

Data sets	Dimensionality reduction methods		Attributes selected after dimensionality reduction
	PCA	Information Gain	
Data set 4	Abazis, account, complimentary, discount, medications, movies, Natural, overnight, pharmacy, rescription, priced, product, quality, shipping, smoking, valium, Viagra, weightloss, winning	Abazis, account, complimentary, discount, medications, movies, natural, overnight, pharmacy, prescription, priced, Product, quality, shipping, smoking, valium, Viagra, weightloss, winning	Abazis, account, complimentary, discount, medications, movies. Natural, Overnight, pharmacy, prescription, priced, product  Quality, shipping, smoking, valium, Viagra, Weightloss, winning
<p>Total number of attributes in dataset after frequency based dimensionality reduction : 170</p> <p>Total number of records: 5998</p> <p>Total number of attributes selected after feature selection using the above models : 20</p>			

Table 3-5: Features selected from Dataset 4

Data sets	Dimensionality reduction methods		Attributes selected after dimensionality reduction
	PCA	Information Gain	
Data set 5	Better, charset, failed Getting, graphics, Guaranteed, identity, impotence, interest, insurance, languages, localized, Message, Mortgage, Perfect, Sender, Transcript, Viagra,	Message, movies, mortgage, complimentary, million, charset, Download, assistance, prescription, attract Benefits, quality, Better, prices, product, Viagra	Better, charset, Failed, graphics, Guaranteed, Identity, impotence, Insurance, languages, localized, message, mortgage, perfect, sender, Transcript, viagra
<p>Total number of attributes in dataset after frequency based dimensionality reduction : 199</p> <p>Total number of records: 5175</p> <p>Total number of attributes selected after feature selection using the above models : 16</p>			

Table 3-6: Features selected from Dataset 5

Data sets	Dimensionality reduction methods		Attributes selected after dimensionality reduction
	PCA	Information Gain	
Data set 6	Appointment assistance cheapest delivery discount graphics medication shipping viagra winning	medicine chemist assistance replica medication cheapest remedy shares shipping medical appointment	appointment assistance cheapest discount medication shipping viagra winning remedy
<p>Total number of attributes in dataset after frequency based dimensionality reduction : 242</p> <p>Total number of records: 6000</p> <p>Total number of attributes selected after feature selection using the above models : 10</p>			

Table 3-7: Features selected from Dataset 6

### 3.4 Implementation

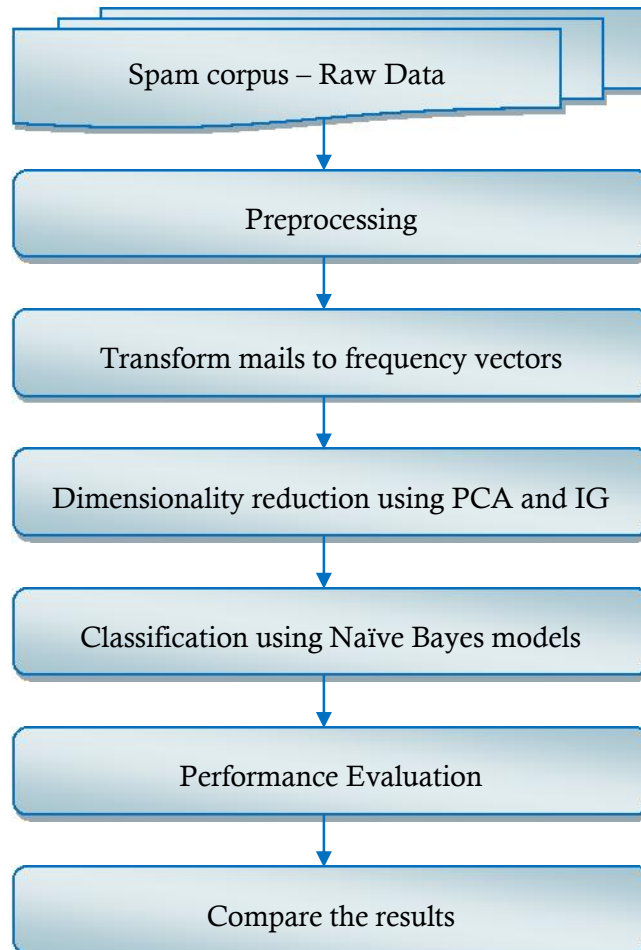


Figure 3-1: Classification using Naïve Bayes models

This work evaluates the implementation of two versions of **Naïve Bayes** models on six non-encoded datasets [6]. The approach consisted of practical work involving several experiments which is supported by theoretical background.

To carry out the experiments test environments in Perl and WEKA were established. The attributes selected from each dataset are tested against Bernoulli NB (Naïve Bayes) model and Multinomial NB model. Both the Naïve Bayes models are run with 10 fold cross-validation.

### 3.5 Evaluation Measures

The following Performance and correctness measures are considered while evaluating the experiment results.

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

$$Specificity = \frac{TN}{TN + FP} = 1 - F$$

ROC (Receiver Operating Characteristic) - FPR and TPR in  $x$  and  $y$  axes respectively.

In the experiment, spam recall (TP/ (TP+FN)) and legitimate recall (TN/ (TN+FP)) are used for evaluation. Spam recall is the proportion of spam messages blocked by the filter, whereas legitimate recall is the proportion of legitimate messages that passed the filter.

The results are shown in the following pages.

#### ROC graphs

Besides confusion matrices ROC graphs are another way to examine the performance of classifiers. A ROC graph is a plot with the false positive rate on the  $x$ -axis and the true positive rate on the  $y$ -axis. The point (0,1) is the perfect classifier: it classifies all positive cases and negative cases correctly because the false positive rate is 0 (none), and the true positive rate is 1 (all). The point (0,

0) represents a classifier that predicts all cases to be negative, while the point (1, 1) corresponds to a classifier that predicts every case to be positive. Point (1, 0) is the classifier that is incorrect for all classifications.

### 3.6 Results

#### 3.6.1 Bernoulli model Results

<b>Datasets</b>	<b>Spam Recall (Sensitivity)</b>	<b>Legitimate Recall (Specificity)</b>	<b>ROC Value</b>
Dataset 1(enron1) Legitimate : 3672   Spam : 1500	1	1	1
Dataset 2(enron2) Legitimate: 4361   Spam: 1496	1	1	1
Dataset 3(enron3) Legitimate: 4012   Spam: 1500	1	1	1
Dataset 4(enron4) Legitimate: 1500   Spam: 4500	1	1	1
Dataset 5(enron5) Legitimate: 1500   Spam: 3675	1	1	1
Dataset 6(enron6) Legitimate: 500   Spam: 4500	1	1	1

Table 3-8: classification results using Bayesian Bernoulli model



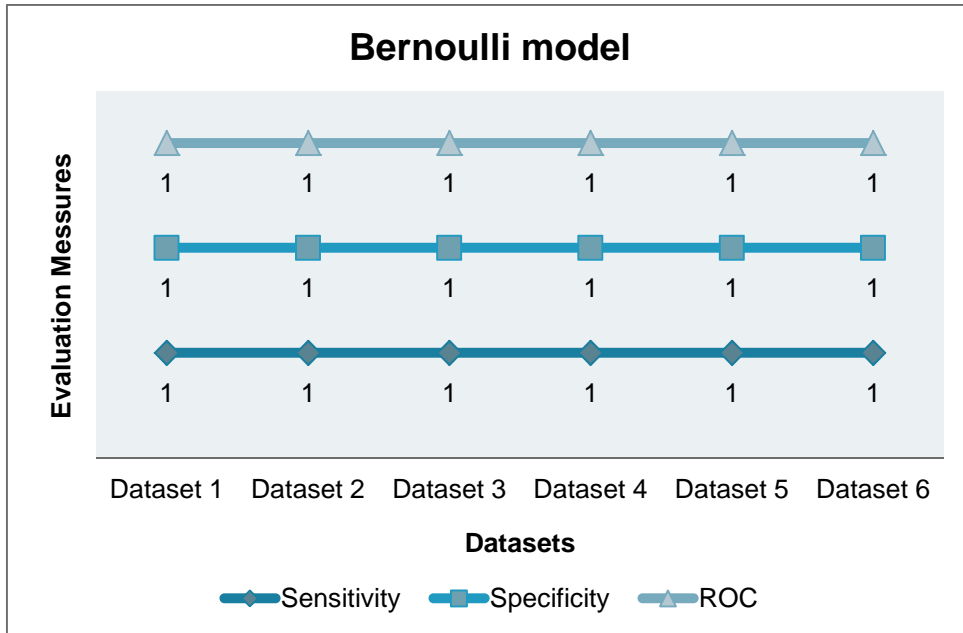


Figure 3-2: Bernoulli model Results

### 3.6.2 Multinomial model Results

Datasets	Spam Recall (Sensitivity)	Legitimate recall (Specificity)	ROC
Dataset 1(enron1) Legitimate : 3672   Spam : 1500	.242	.966	.889
Dataset 2(enron2) Legitimate: 4361   Spam: 1496	.392	.973	.891
Dataset 3(enron3) Legitimate: 4012   Spam: 1500	.304	.441	.862

Dataset 4(enron4) Legitimate: 1500   Spam: 4500	.996	.734	.91
Dataset 5(enron5) Legitimate: 1500   Spam: 3675	.97	.616	.889
Dataset 6(enron6) Legitimate: 1500   Spam: 4500	.979	.629	.887

Table 3-9: Classification results using Bayesian Multinomial model

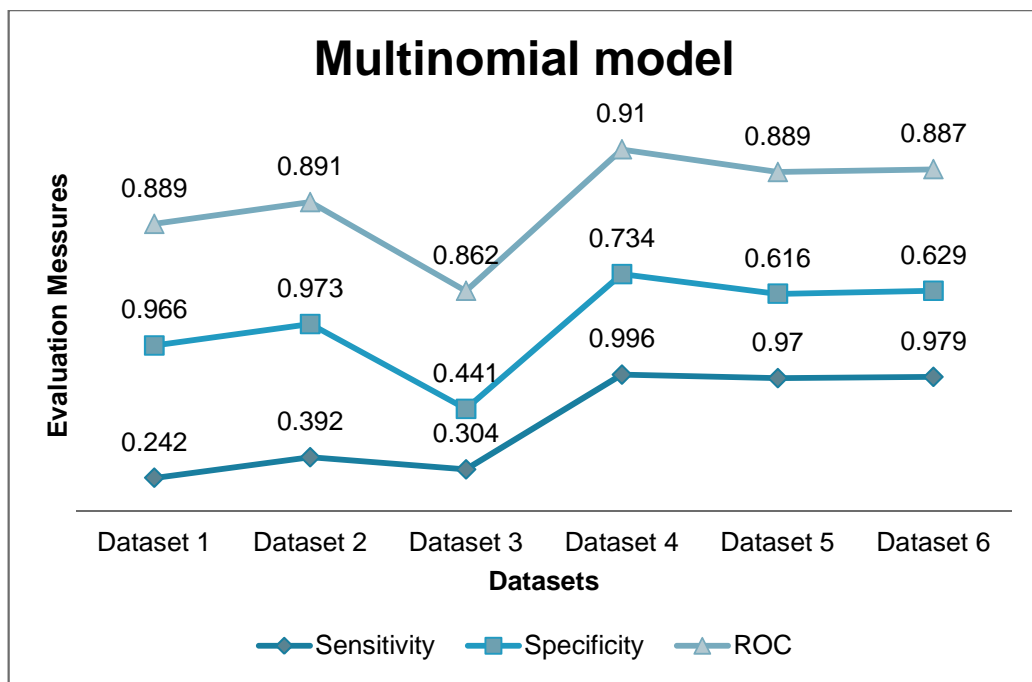
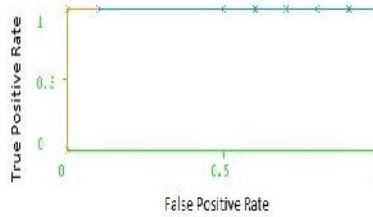
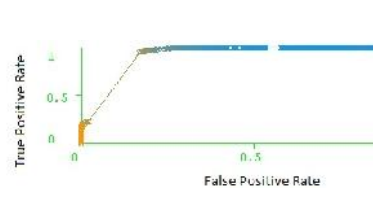
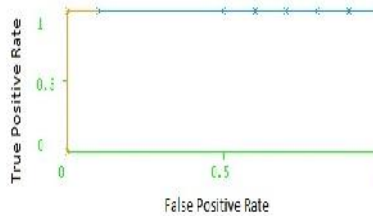
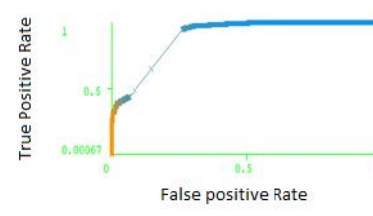
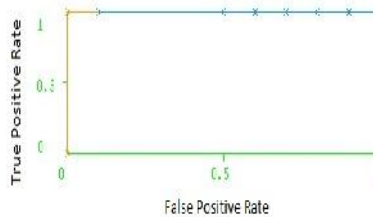
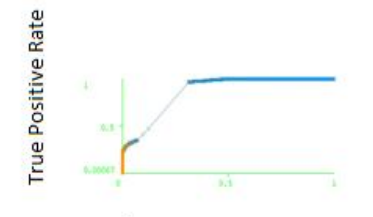
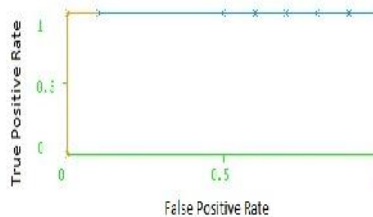
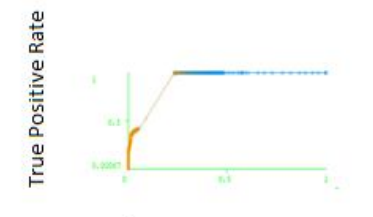


Figure 3-3: Multinomial model Results

ROC Curves

Data set	Bernoulli Model	Multinomial Model
Dataset 1		
Dataset 2		
Dataset 3		
Dataset 4		

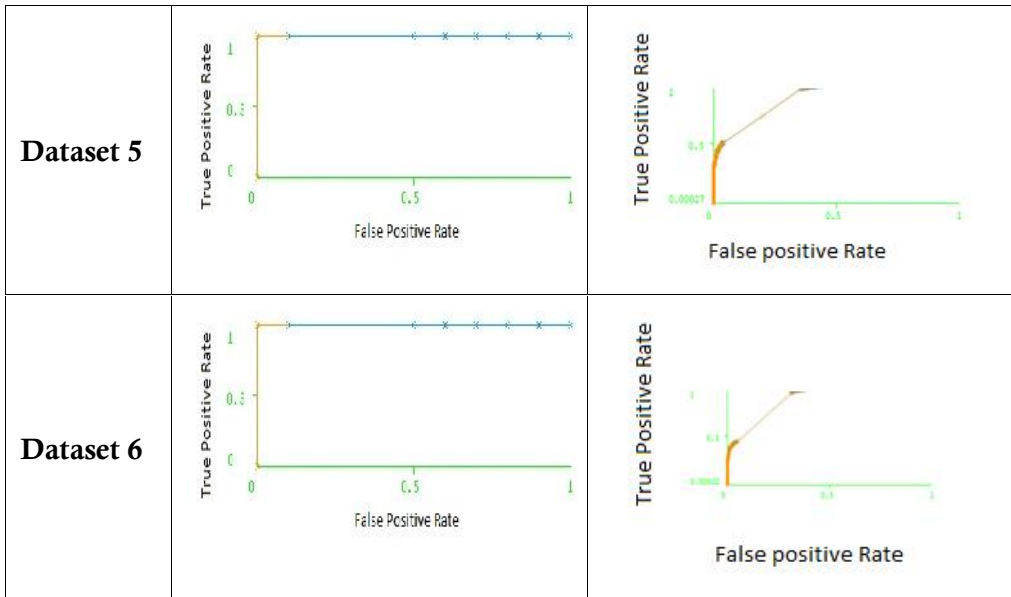


Figure 3-4: ROC Curves of Bernoulli model and Multinomial model

### 3.7 Result Analysis

In Enron datasets 4, 5 and 6 spam recall is much higher than the legitimate recall, for multinomial method. The legitimate recall can be increased by changing the feature selection cutoff values and the equation to incorporate the difference between collection frequencies of spam and legitimate. In datasets 1 and 2, legitimate recall is greater than spam recall. But for Bernoulli model, legitimate recall=spam recall=1 for all the six datasets. Based on the results from these six datasets it is found that Bernoulli model outperforms multinomial model in the context of spam filtering. The results show that, the Bernoulli model performs very well than multinomial model. Boolean values are used for Bernoulli model. This indicates that the number of times a word

repeats in a message is not important, its presence or absence is important to detect its class.

Spam Filtering with Naive Bayes – Which Naive Bayes[97] is a very relevant and related paper to the method explained in this chapter and this work discussed about five different versions of Naive Bayes, and the experiments are done on the same six Enron datasets. Comparison of this work with the proposed model is given below.

Sl. No	Feature	“Which Naive Bayes Paper” Methods	Proposed Method
1	Algorithm used	Naive Bayes(NB)	Naive Bayes(NB)
2	NB variants used	<ul style="list-style-type: none"> <li>) Multinomial NB, TF attributes(MTF)</li> <li>) Multi-variate Bernoulli NB(MBN)</li> <li>) Multinomial NB, Boolean attributes</li> <li>) Multi-variate Gauss NB</li> <li>) Flexible Bayes</li> </ul>	<ul style="list-style-type: none"> <li>) Multinomial NB, TF attributes</li> <li>) Bernoulli NB</li> </ul>
3	Datasets	6 encoded Enron datasets	6 encoded Enron datasets

Table 3-10: Comparison with “Which Naive Bayes” Paper

### 3.7.1 Comparison of Results

Comparison of papers	"Which Naive Bayes Paper" Methods							Proposed Method						
	NB variant	E 1	E 2	E 3	E 4	E 5	E 6	NB variant	E 1	E 2	E 3	E 4	E 5	E 6
Spam recall	M T F	95.66	96.81	95.04	97.79	99.42	98.08	M T F	24.2	39.2	30.4	99.6	97.0	97.9
	M B N	97.08	91.05	97.42	97.7	97.95	97.92	M B N	1	1	1	1	1	1
Ham recall	M T F	94.0	96.78	98.83	98.3	95.65	95.12	M T F	96.6	97.3	44.1	73.4	61.6	62.9
	M B N	93.19	97.22	75.41	95.86	90.08	82.52	M B N	1	1	1	1	1	1

Table 3-11 : Comparison of results with Which Naive Bayes Paper

### **3.8 Summary**

Spam filtering with two different versions of the Naive Bayes (NB) classifier are discussed and evaluated experimentally. The Bernoulli and multinomial models are included in this analysis. To accommodate the current patterns and future trends in spam filtering, periodic updation of corpora is required. The system can perform very well on the problem of spam by improving preprocessing of stop words and selection of document frequency.

## Chapter 4

### Filtering Template-Driven Spam emails

---

Spam e-mail messages tend to have several elements or features in common, which are usually not present in legitimate e-mail. Sometimes spammers tend to send promotional, campaigns mails to a group of users within a community. Mostly these emails will have a common content except the 'to' address and such emails are called template emails. The template emails are collected and stored in training set and each test email is checked against this training set. The main objective of this study is to investigate and evaluate spam filtering by using two information retrieval techniques, Simple Vector Space Models (VSM) and VSM using Rocchio Classification utilizing the cosine similarities.

---

#### 4.1 Introduction

By the late 1990's proactive communication was the hot trend in marketing. The arrival of newer and more effective marketing automation solutions accelerated the trend. The term marketing automation [98] refers to software platforms designed to automate repetitive tasks in campaign management. In the case of emails, the process of communication execute in real time with marketing automation. That meant that messages directed to the consumers could be richer, more timely, more relevant and more engaging. The word



"Spam" as applied to Email means "Unsolicited Bulk Email" [6]. Unsolicited means that the Recipient has not granted verifiable permission for the message to be sent [6]. Bulk means that the message is sent as part of a larger collection of messages, all having substantively identical content. A message is Spam only if it is both Unsolicited and Bulk. As Spam e-mail messages having identical content, they tend to have several elements or features in common, which are usually not present in legitimate e-mail. Sometimes spammers tend to send promotional, campaigns mails to a group of users within a community using some software or App. Mostly these emails will have a common content except the 'to' address and such emails are called template emails. To send a particular promotion, they create pre-formatted template and merge the template with details of receivers stored in their database. Timely detection of these mails and underlying template features can be used to easily ignore forthcoming spam. Most high-volume spam is sent using such tools which randomizes parts of the message - subject, body, sender address etc. Templates of mails are only needed to include in the training set which will minimize the training set i.e. search volume rather than using every mail in the corpora.

The template emails are collected and stored in training set and each test email is checked against this training set. The main objective of this work is to investigate and evaluate spam filtering by using two information retrieval techniques, Simple Vector Space Models (VSM) and VSM using Rocchio Classification utilizing the cosine similarities.

## **4.2 Classification Models**

The standard VSM [99] using cosine similarity with Euclidean distance are used in this work. Simple VSM and VSM using Rocchio Classification [74]

methods and their application to the task of spam filtering are applied for classification task.

#### **4.2.1 The Simple Vector Space Model**

Vector Space Model is an algebraic model which represents text documents as vectors of terms. In information retrieval, a vector space model (VSM) is a widely used model for representing information. Documents and queries are represented as points in a potentially very high dimensional, metric vector space. The distance (or similarity) between a query vector and the document vectors is the basis for the information retrieval process.

Documents and queries are represented as vectors.

$$D_j \times \{w_{1,j}, w_{2,j}, w_{3,j}, \dots, w_{t,j}\}$$

$$q \times \{w_{1,q}, w_{2,q}, w_{3,q}, \dots, w_{t,q}\}$$

Each dimension corresponds to a separate term. If a term occurs in the document, its value in the vector is non-zero. We can use term frequency or tf-idf weight as the value in the vector. The terms are distinct words in the vocabulary/corpus and the dimensionality of the vector is the number of distinct words in the vocabulary/corpus.

Vector operations can be used to compute the distance between documents and queries by comparing the deviation of angles between each document vector and the query vector where the query is represented as same kind of vector as the documents. In practice, we calculate the cosine of the angle between the vectors; instead of the angle itself. The documents are similar if the angle has small value.

$$\cos w = \frac{D_2 \cdot q}{\|D_2\| \|q\|}$$

Where  $D_2 \cdot q$  is the intersection (i.e dot product) of the document and the query vectors,  $\|D_2\|$  is the norm of vector  $D_2$ , and  $\|q\|$  is the norm of vector  $q$ . The norm of a vector is calculated as such:

$$\|v\| = \sqrt{\sum_{i=1}^n v_i^2}$$

$$\text{similarity} = \cos(w) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \sqrt{\sum_{i=1}^n (B_i)^2}}$$

A cosine value of zero means that the query and document vector are orthogonal and have no match and one means they are identical.

#### 4.2.2 Rocchio Classification

In basic vector space model discussed above, we compute the cosine similarity of new incoming email with each training mails and assign the class of email with maximum  $\cos(\cdot)$ . This is the decision boundary, which is chosen to separate the two classes. Another way to determine the decision boundary is Rocchio Classification. This method uses the centroids of each class to determine the boundaries. The centroid of a class is computed as the vector average or center of mass of its members.

$$\vec{\mu}(c) = \frac{1}{|D_c|} \sum_{d \in D_c} \vec{v}(d)$$

where  $|D_c|$  is the set of documents in  $D$  whose class is  $c$ :

$D_c = \{d: (d, c) \in D\}$ . The normalized vector of  $d$  is denoted by  $\vec{v}(d)$ .

The boundary between two classes in Rocchio Classification is the set of points with equal distance from the two centroids and the new email is classified into class with closest centroid  $\mu(c)$  from the new email.

### 4.3 Methodology

The working of the method is:

1. Consider emails as documents and words as terms
2. Tokenize the mails and store all the tokens in the feature vector
3. For finding the exact template, stemming and stop word removals are not done
4. Assign each training mail into two given classes (spam or legitimate)
5. Find Document Frequency,  $d_f$ ,
6. Find term frequency-inverse document frequency, tf-idf.

$$\tilde{S}_{t,d} = \text{tf}_{t,d} \log \frac{|D|}{\sum_{d \in D} \mathbb{1}_{d \in D} \mathbb{1}_{t \in d}}$$

Where  $|D|$  is the total number of documents in the document set

$\sum_{d \in D} \mathbb{1}_{d \in D} \mathbb{1}_{t \in d}$  is the number of documents containing the term  $t$ .

$tf_{t,d}$  is term frequency of term  $t$  in document  $d$  (a local parameter),

7. Normalize feature vector by dividing with Euclidean distance to make it unit vector. Convert each mail in training data set into unit vector
8. Likewise convert each email in the test corpora into unit vector. Store query vector also in the same format.

#### **4.3.1 For Simple VSM**

Find the cosine similarities between each training email vectors and the query email vector. Then select the training email vector with maximum cosine value; assign new query email to the class of selected training email vector.

#### **4.3.2 For VSM using Rocchio Classification**

Find the centroid ( $\mu(c)$ ,  $c = \{\text{spam, legitimate}\}$ ) of each class by applying Rocchio Classification. Then calculate the cosine similarity of test email from the centroids  $\cos(q, \mu(c))$  where  $c = \{\text{spam, legitimate}\}$  and then assign test email to class with  $\text{Max}(\cos(q, \mu(c)))$ .

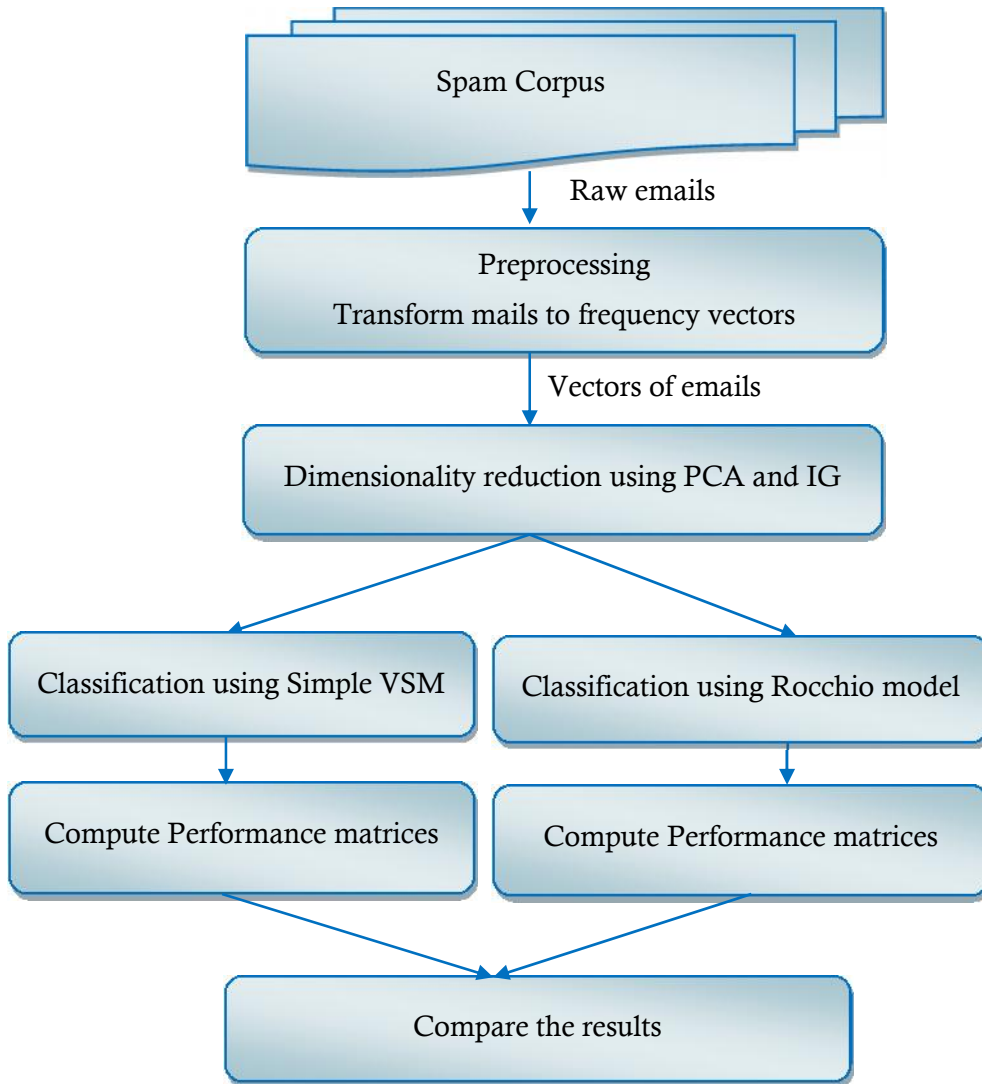


Figure 4-1: Different steps in classifying emails using vector space models

#### 4.4 Composition of Training and Test Datasets

The dataset is prepared using the mails received in 2 days for the testing. If the mail server is not capable to handle spam, large numbers of spams are received every day. The testing is done using only this miniature dataset. Large datasets are available online, but when we go for large datasets, the computational time will be increased and this will delay the mail delivery. Also template based spams are time dependent, earlier templates may not be helpful to detect spam. Only unique templates are included in the training set. To generate spam training set of template mails, experiments were done. It is discussed in the next chapter.

Dataset	No. of spam Mails	No. of Legitimate mails
Training set	42	59
Test set	42	59

Table 4-1: Data set Composition

#### 4.5 Performance Measures

The following Performance and correctness measures are considered while evaluating the experiment results.

$$\begin{aligned}
 \text{Precision (P)} &= \frac{T}{T+F} \\
 \text{Recall (R)} &= \frac{T}{T+N} = 1 - F \\
 \text{F1-Measure (F)} &= 2 * \frac{(P * R)}{(P + R)}
 \end{aligned}$$

The F measure (F1 score or F score) is a measure of a test's accuracy and is defined as the weighted harmonic mean of the precision and recall of the test.

#### 4.6 Experimental Results

The results of experiments run on the given dataset are given below:

Confusion Matrix		Predicted	
		Spam	Legitimate
Actual	Spam	28	14
	Legitimate	8	51
TP=28		FN=14	
FP=8		TN=51	

Table 4-2: Experiment results - Simple VSM

Confusion Matrix		Predicted	
		Spam	Legitimate
Actual	Spam	37	5
	Legitimate	14	45
TP=37		FN=5	
FP=14		TN=45	

Table 4-3: Experiment results - VSM with Rocchio Classification



Performance and correctness measures are given in the following table.

Performance and correctness measures	Simple VSM	VSM using Rocchio Classification
Sensitivity(Recall)	66.66%	88%
Specificity	86.44%	76.27%
Positive predictive value (precision)	77.78%	72.54%
F-measure( F)	71.78%	79.38%

Table 4-4: Performance and correctness measures

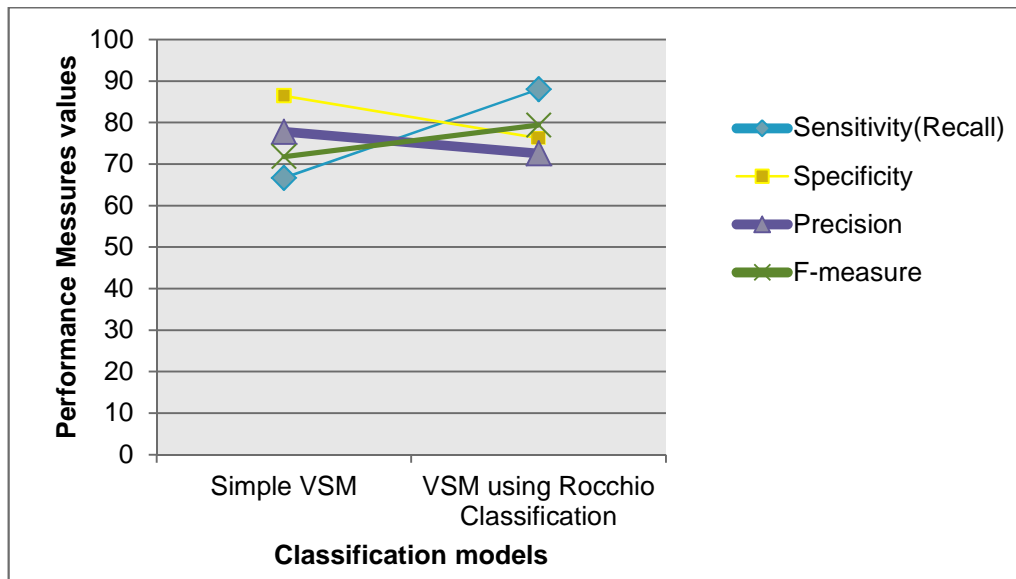


Figure 4-2: Performance and correctness measures

#### **4.7 Summary**

Present study considered the problem of spam filtering. In literature most of the spam filters are either rule based models or Bayesian models. This study considered another idea focused on two schemes based on vector space models followed in classic Information Retrieval. To find semantic distance, cosine similarity is used in both methods. This study has been carried out on 101 real datasets with attributes of td-idf values. First method used all the mails in the training set to test against the spam, while in the second method, only the centroids of each class (only two vectors) are used to find the similarity. VSM using Rocchio Classification is much faster than simple VSM because the number of iterations required is less. The results showing that VSM using Rocchio Classification scheme performs better than Simple VSM scheme. Since templates are changing with time and promotional activities, the training data need to be changed periodically in order to incorporate new templates.

The simple VSM model is efficient to find out the exact spam template. But when the test training set becomes large, time to find similarity is also increasing ( $O(n)$ ). Hence we have to update the training corpus by deleting the templates that are not used by spammers and by adding new mail templates. The training data size can be further reduced by storing only unique mail templates. In that way simple VSM can perform better than Rocchio Classification. The optimum size of the training set has to be studied. The method presented here can be enhanced with semantic distance between mails.

## Chapter 5

### Finding Template Emails from Spam Corpus

---

Spam has become a headache for users on the Internet over the last few decades. Many solutions are developed and applied on this problem, still spam continues to be major nuisance and we are still away from a satisfactory and long lasting solution. This is due to the fact that many heuristics are applied to proposed and developed methods and these heuristics are temporal and pertaining only to that particular corpus. We need to update the spam training set in order to handle forthcoming spam. Thus updating the corpus itself is a research problem that is to be handled with apt domain knowledge with the help of suitable algorithms. The main objective of this work is to investigate and evaluate the applicability of Genetic algorithm and K-Means algorithm in the process of selection of suitable mails to store in the training set.

---

#### 5.1 Introduction

Most of the time, spammers use mail templates for sending spam. To send a particular promotion, they create pre-formatted template and merge the template with details of receivers stored in their database. Timely detection of these mails and underlying template features can be used to easily ignore forthcoming spam. Most high-volume spam is sent using such tools which

randomizes parts of the message - subject, body, sender address etc. Templates of mails are only needed to include in the training set which will minimize the size of training set (or search volume) rather than storing each and every mail. The main objective of this work is to investigate and evaluate the applicability of Genetic algorithm and K-Means algorithm in the process of selection of suitable mail templates[100].

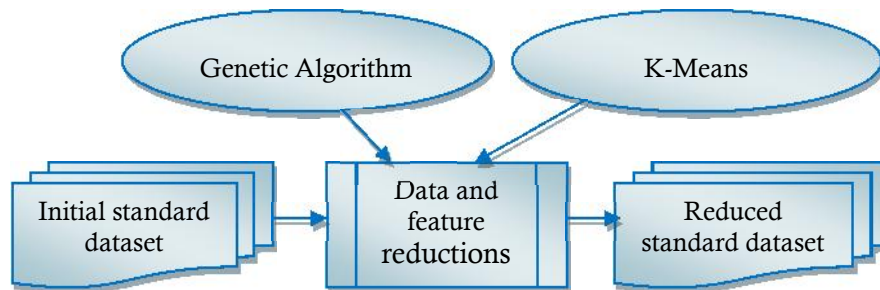


Figure 5-1 : Data reduction Process

## 5.2 Related Works

A related work in this research area explains a method called I-match which uses collection statistics to select the best terms to represent the document [101]. An extension of I-Match algorithm suggest a setup where a suitable lexicon is found and then K different perturbations of the original lexicon are derived by randomly eliminating a fraction p of terms from the original. The extended I-Match signature is defined as a (K+1)-tuple, consisting original lexicon and its K perturbations. Any two documents are considered to be near duplicates if their extended signatures overlap on at least one of the K +1 coordinates. They also suggested term ranking induced by standard feature

selection can be used as an alternative to traditional idf -based method in document classification tasks [102]. In this method it is required to create k perturbations; it is not a trivial task.

The proposed approach tries to find out alternative ways for updating training set. The method compares the mails within the training set and creates the best population (genetic algorithm) or vectors of centroids (K-Means). All the incoming mails are tested against the population/centroids. No perturbations are being done on training set. If an incoming mail matches with any of the tuples within a threshold it is considered as a near duplicate and discarded. Otherwise we need to update the training set and for that the population or centroids are recalculated. Since the methods described here are totally different with previously mentioned works, it is very difficult to compare the methods.

### **5.3 Method**

The spam corpus contains both spam and legitimate mails. Our aim is to find out a small subset of these mails which best represent the corpus. Firstly, the attributes have to be analyzed using Information gain algorithm and important attributes from the corpus are selected. Secondly, spam mails and legitimate mails are clustered separately using Genetic Algorithm and K-Means algorithm. Finally the experimental results are compared and best method is chosen.

#### **5.3.1 Genetic Algorithm**

Genetic Algorithms (GA) apply an evolutionary approach to inductive learning. GA's were introduced as a computational analogy of adaptive

systems. They are modeled loosely on the principles of the evolution via natural selection, employing a population of individuals that undergo selection in the presence of variation-inducing operators such as mutation and recombination (crossover). A fitness function is used to evaluate individuals, and reproductive success varies with fitness. Crossover function forms new elements for the population by combining parts of two elements currently in the population. Mutation is applied to elements chosen for elimination by randomly flipping bits within a single element. Selection is to replace the elements to be deleted by copies of elements that pass the fitness test with high scores. With selection, the overall fitness of the population is guaranteed to increase.

**Fitness Function:** Let  $N$  be the number of matches of the input attribute values of  $E$  with training instances from its own class. Let  $M$  be the number of input attribute value matches to all training instances from the competing classes. Add 1 to  $M$  and divide  $N$  by  $M$ . Higher the fitness score, smaller will be the error rate of solution.

### **5.3.2 Supervised Genetic Learning Algorithm**

**Step 1:** This step initializes a set of email feature vectors (population of elements) referred as  $P$ .

**Step 2:** A fitness function to evaluate each element currently in the population and elements not satisfying the fitness criteria are eliminated from the population. This results a set of population elements that best represents the training data. Then it adds new elements to the population in place of eliminated elements if any. New elements are formed from previously deleted elements by applying crossover and mutation.

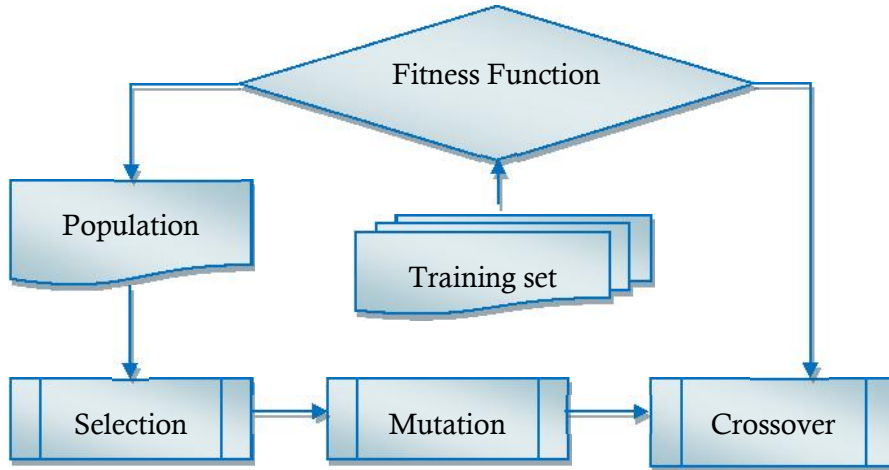


Table 5-1: Supervised genetic algorithm

The fitness function used is

$$\text{Score} = N / (M+1); \text{where}$$

N - the number of matches of the input attribute values with training instances from its own class

M - the number of input attribute values that matches to all training instances from the competing class.

The instances with high scores are selected while low scored instances are eliminated. For eliminated instances, 50% single entry cross-over and 28 bits mutation are done and which will be the second generation population. This process is continued until the population is converged which means feature vectors are not changing anymore from the previous population. To implement the method, customized Perl scripts were used.

### 5.3.3 K –Means Algorithm

K-Means algorithm is used in cluster analysis [11][103][104] and it works as follows. The method is used here to find representatives of mails for the training set. The main idea is to define k centroids, one for each cluster[105]. When new feature vector arrives, it will be compared with each centroid in existing dataset and the new mail is assigned to the nearest cluster. At this point, re-calculate k new centroids as centers of the clusters resulting from the previous step. The procedure is repeated until the centroids do not move to a new point or we can fix the number of iterations (cut-off). The objective of K-Means algorithm is to minimize the *objective function*, or a squared error function within clusters.

The objective function given below is used while computing new centroids.

$$J = \sum_{j=1}^k \sum_{i=1}^n (x_i^{(j)} - c_j)^2,$$

where  $(x_i^{(j)} - c_j)^2$  is a chosen distance measure between a data point  $x_i^{(j)}$  and the cluster centre  $c_j$ . It is an indicator of the distance of the  $n$  data points from their respective cluster centers.

### 5.4 Building Representation

We have considered emails as documents and terms as features. Each mail is tokenized (space as the delimiter) and stop words are removed. Term frequency and document frequency (tf and idf) measures are used to select



initial feature vectors. The attributes with document frequency  $\geq 10$  and term frequency  $\geq 4$  are only selected.

Then Information gain method is applied for dimensionality reduction and a subset of reduced feature vector of 26 attributes is selected. Thus the Chromosome - Blueprint of a mail consisted of 26 attributes. All the mails are encoded into their frequency representation.

A sample chromosome representation of email is given below

Features	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	$C_6$
Weights	$W_1$	$W_2$	$W_3$	$W_4$	$W_5$	$W_6$
Sample values	1	0	0	2	0	1

Table 5-2 : Sample chromosome construction of email

As discussed  $W_i$  represents the frequencies of each selected feature.

For genetic algorithm, we need two sets of data, the initial population and the training set. The initial population is chosen from the training set according to some rules. It was selected by the percentage of attribute contribution. The fitness score of each element in the population was computed and the elements with fitness score above the threshold (cut-off) value were selected. Here we selected 50 elements each for spam and legitimate classes. For training set, entire data set is used.

For K-Means algorithm, we need a training set and a value for K. Mostly K is chosen by applying heuristics which depends on the problem and in this case K

is chosen as 50 for each class; spam and legitimate. The cluster centroids are added to the training set for future spam classification.

### **Composition of Initial population and Training set**

Mails are considered as documents [106] and tokenization and other preprocessing techniques are applied to the dataset. A personalized dataset is used in this work.

Dataset	No. of spam Mails	No. of Legitimate mails
Initial Population (for genetic algorithm)	50	50
Training set	200	200

Table 5-3: Data set Composition

The training dataset is prepared using the mails received in one month for the testing. Since the server is not capable to handle spam, we receive large numbers of spam mails every day. The initial population (N=50) is taken from this training set. Large datasets are available online, but when go for large datasets, the computational time increases and this will delay the mail delivery.

### **5.5 Experimental Results**

The experiments were setup and evaluated in four scenarios as explained below. For classification process, Simple Naïve Bayes was used.

**Scenario 1:** Before applying attribute selection and genetic algorithm (GA), the Simple Naïve Bayes classification produced the following results:

Parameters	Values
Correctly Classified Instances	1583(97.2359 %)
Incorrectly Classified Instances	45(2.7641 %)
Root mean squared error	0.1507
Total Number of Instances	1628
Number of Attributes	310

Table 5-4: Before feature selection and learning algorithms

### Confusion Matrix

Spam	legitimate	classified as
193	9	spam
36	1390	legitimate

Table 5-5: Before feature selection and learning algorithms

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.95	0.025	0.84	0.955	0.896	0.992	1
0.97	0.045	0.99	0.975	0.984	0.992	2
0.97	0.042	0.975	0.972	0.973	0.992	avg

Table 5-6: Performance Measures before feature selection

When Naïve Bayes Simple classification algorithm with 10-fold cross-validation was applied on whole dataset, the RMSE reported was 15%. We

applied Information Gain algorithm to select high valued 50 attributes out of 310 attributes. The high valued attributes obtained by Information gain in their rank order are as follows:

*your, cialis, software, attached, Viagra, cheap, soft, paliourg, file, xanax, meds, valium, tabs, prices, actuals, online, forwarded, quality, here, free, best, nomination, prescription , etc.*

### 5.5.1 Classification Results after Feature Selection

An initial population of 99 mails is chosen for supervised genetic algorithm. (50 – Legitimate mails and 49 spam mails). The performance of the online filtering strongly depends on the attributes and the training set selected. By applying supervised genetic algorithm, 99 best candidate instances from the large data set are selected for the final filtering of spam mails.

**Scenario 2:** The results of classification using Simple Naïve Bayes algorithm with 10 fold cross-validation after the dimensionality reduction

Parameters	Values
Correctly Classified Instances	89 (89.899 %)
Incorrectly Classified Instances	10(10.101 %)
Root mean squared error	0.2971
Total Number of Instances	99
Number of Attributes	78

Table 5-7: Dataset and Performance Measures after feature selection

<b>Spam</b>	<b>legitimate</b>	<b>classified as</b>
49	0	<b>spam</b>
10	40	<b>legitimate</b>

Table 5-8: Confusion Matrix after feature selection

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
1	0.2	0.831	1	0.907	0.955	1
.8	0	1	0.8	0.889	0.95	2
0.8	0.09	0.916	0.89	0.898	0.95	avg

Table 5-9: Performance Measures after feature selection

### 5.5.2 Classification Results after Applying Genetic Algorithm

**Scenario 3:** After implementing Genetic algorithm, the Simple Naïve Bayes classification with 10 fold cross-validation produced the following results:

Parameters	Values
Correctly Classified Instances	94 (94.949%)
Incorrectly Classified Instances	5(5.0505 %)
Root mean squared error	0.2238
Total Number of Instances	99
Number of Attributes	78

Table 5-10: After feature selection and genetic learning

**Confusion Matrix**

<b>Spam</b>	<b>legitimate</b>	<b>classified as</b>
49	0	<b>spam</b>
5	45	<b>legitimate</b>

Table 5-11: Confusion Matrix After feature selection and genetic learning

<b>TP Rate</b>	<b>FP Rate</b>	<b>Precision</b>	<b>Recall</b>	<b>F-Measure</b>	<b>ROC Area</b>	<b>Class</b>
1	0.2	0.90	1	0.951	0.99	1
0.9	0	1	0.9	0.947	0.99	2
0.95	0.1	0.95	0.95	0.949	0.99	avg

Table 5-12: Performance Measures after feature selection and genetic learning

**5.5.3 Classification Results after Applying K –Means Algorithm**

K-Means algorithm was applied on the reduced attribute-set to find out the 100 centroids, 50 clusters each from spam and legitimate mails. R package was used for implementing the algorithm. These cluster centroids-templates- are used for classifying mails either as spam or legitimate.

**Scenario 4:** After executing K-Means algorithm, the Simple Naïve Bayes classification with 10 fold cross-validation is applied on the centroids and it produced the following results:

Parameters	Values
Correctly Classified Instances	91 (91%)
Incorrectly Classified Instances	9(9 %)
Root mean squared error	0.2818
Total Number of Instances	100
Number of Attributes	78

Table 5-13: Classification results after feature selection and K-Means

### Confusion Matrix

Spam	legitimate	classified as
49	1	spam
8	42	legitimate

Table 5-14: Confusion Matrix after feature selection and K-Means

### Detailed Accuracy by Class

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.98	0.16	0.86	0.98	0.916	0.955	1
0.84	0.02	0.977	0.84	0.903	0.955	2
0.91	0.09	0.918	0.91	0.91	0.955	avg

Table 5-15: Performance Measures after feature selection and K-Means

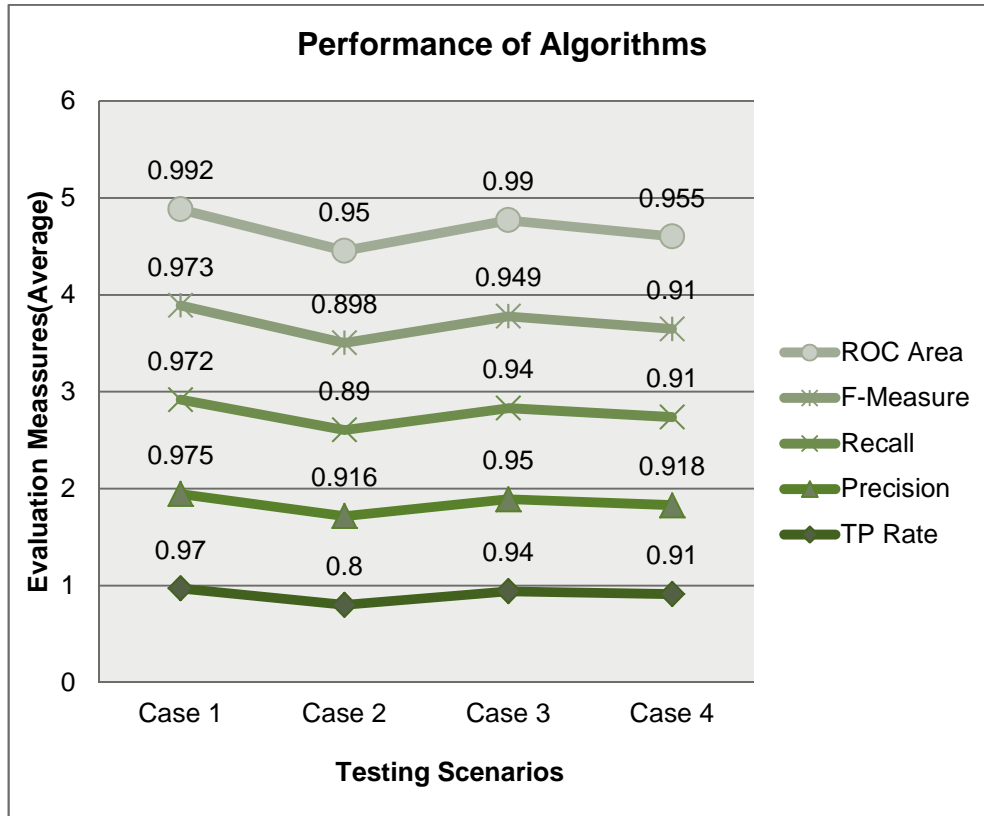


Figure 5-2: Comparison of Performance

Case 1: Classification without feature selection

Case 2: Classification after feature selection

Case 3: Classification with feature selection and Genetic Algorithm

Case 4: Classification with feature selection and K-Means Algorithm



## 5.6 Discussion

Although Genetic algorithm and K-Means algorithms are used in different scenarios of data mining, we have utilised these algorithms here to find best candidate mails for the dataset. Our aim is to get an optimized training dataset that will reduce the learning time and space complexities in spam filtering task. Both the algorithms are experimented with feature selection and without feature selection. Naïve Bayes algorithm is used here to test the generated dataset to compare the efficiency of Genetic and K-Means algorithms. The results are shown in above sections.

The comparison of algorithms is given below.

Scenario	Root mean squared error	Precision
Before feature selection	0.1507	97
After feature selection	0.2971	89
Genetic algorithm with feature selection	0.2238	94
K-Means algorithm with feature selection	0.2818	91

Table 5-16: Comparison of algorithms

The results depict that Genetic algorithm performs better than K-Means algorithm. This is due to the fact that Genetic algorithm can be tuned with parameters specific to a particular problem and manual adjustments are possible in Genetic algorithms during crossover and mutation operations.

Some earlier works in this research area [107] used near-duplicate algorithms to applications like document classification, such as spam filtering, term ranking. They suggest the techniques with standard feature selection can be used an alternative to traditional idf-based method. This method is used to classify mails after checking near duplicates with I-Match algorithm. Sampling techniques may also be applied without affecting spam: legitimate ratio.

### **5.7 Summary**

The experimented template mail selection process which uses supervised genetic algorithm and K-Means algorithm are found good for generating template mails for future spam filtering. The genetic algorithm allows manual adjustments in the threshold value for fitness function, percentage of crossover operations, for appropriate level of filtering. The overall quality of template mail instances is increased by applying genetic algorithm. In case of K-Means algorithm the new cluster centroids are selected for the training set. Hence the new dataset will be small compared to original dataset. This allows easy maintenance of data sets and less requirement of storage.

## Chapter 6

# A Modified Spell Correction Algorithm and Deobfuscation of Emails

---

Emails involve some sort of fraud which is centered on obfuscation techniques to hide real words in the message from spam filters but giving the readers same visual look of the real words. In response to these fraud problems, researchers have developed many methods to fight against obfuscated spam emails. This study focuses on educating the spam filter about obfuscation and deobfuscation process and thereby increasing the spamminess of an obfuscated email using an improved symmetric delete spelling correction algorithm. A modified spell correction algorithm is also devised for deobfuscation task.

---

### 6.1 Introduction

To circumvent the methods of blocking spam, spammers have developed clever techniques to fool anti-spam tools. With the wide-spread application of excellent spam filters, spammers have developed “image-based” spam emails, word obfuscation, and other methods to get around the filters. Some of the techniques they used to hide spam words are: misplaced spaces, purposeful misspelling, repeated vowels, embedded special characters, transliteration and HTML redrawing. Combining these techniques together, spammers generate

large number of words which would have the same visual features of real words that are normally unnoticeable by human beings and at the same time make the spam filters inefficient to handle such emails. For example, some of these words are generated by repeating vowel characters (like 'i'), replacing a character with visually similar character (like letter I with numeric 1) etc. Such words may be bypassed by the spam filters; since mostly there will not be any such word in spam training corpus.

This work focuses on applying spell checking algorithms to substitute obfuscated words with real words. Repeated characters, misspelled words, substituted special characters and letters in emails are being checked and corrected with real words in dictionary. The methodology of this work is to find the spam score of obfuscated as well as deobfuscated emails using the new modified spell correction algorithm and compare the results to find how much cheating can be eliminated or reduced.

## **6.2 Spell Correction Algorithms**

Spell correction algorithms provide a way to correct misspelled words. In Spell Checking algorithms, every term is checked against a dictionary and if the term is not found in the dictionary, then most similar terms to that word from dictionary are shown as spelling suggestions. This research work also proposes a new algorithm for spelling correction.

From the literature reviewed, it is found that the Peter Norvig Algorithm for spell correction generates all possible combinations of query term dynamically, i.e at the time of spell correction and then it needs to search each generated

word against the dictionary. The time complexity of this algorithm depends on the number of characters in the query term and the alphabets in the language.

The number of generated words =  $2n + 2an + a - 1$

where 'n' is the word size, 'a' is the alphabet size, with edit distance  $d=1$

For eg: if word size=8, alphabet size = 30 and edit distance = 1, then there will be 525 words in the search term list.

The Symmetric Delete Algorithm considers only delete operation in edit distance algorithm. Other operations like insert, transpose and substitute are skipped. The algorithm generates edit distance  $\leq 2$  words of both dictionary and query string. Although generating edit-distance words of dictionary is a pre-calculation step, generating edit distance words of query string is a run-time process as in the case of Peter Norvig Algorithm. During the search time, the number of words generated is:

No. of words generated =  $nCd$ , where  $n$ =word size and  $d$ =edit distance.

For example: if word size=8, alphabet size is 30(alphabet size is not important since only delete operation is used) and edit distance is 1, then there will be 8 new words in the search term list.

### **6.3 Related Works**

In a related work, ZHONG [108] proposed a simple backtrack algorithm based on SEDA(String edit distance algorithm) to handle the problem of inserting and substituting non-alphabetical and bogus segmentation characters in an obfuscated word. SEDA firstly calculates the distance score, then backtrack algorithm is applied to remove the influence of non-alphabetical and bogus

segmentation characters on edit operations. The algorithm is applied to each string which cannot be found in a dictionary in an email in order to calculate the distance score between  $s$  and other pre-defined bad words ( $B$ ). The proposed method claims efficient recognition of obfuscated words with inserting or substituting the non-alphabetical character although its time complexity is slightly larger than SEDA. Sergio Rojas [109] called the same concept of bogus character as homoglyph anomaly. Homoglyphs are a pair of symbols whose graphical depiction is almost identical although their computer encodings are different. They propose a solution inspired by sequence alignment techniques used in the field of molecular biology to prevent the anomaly. Their aim was to obstruct obscenity appearance in the comment rather than perform the actual alignment and they remove or replace obfuscated sequence with a censoring mask. They used a penalty function to find out mutations caused by homoglyph substitutions and bogus segmentations. They also proposed a new version aimed at searching and tracing the locations of the potential obfuscations.

Christian [110] discuss about a web forum which is free of profanity disguises. They adapted the idea of phylogenetic tree diversification to the profanity disguise anomaly. They assumed that the guises of a profanity grow down in a similarity tree from a common ancestor to the variants obtained by recurring application of edits or corrections made on the predecessors. The idea is to trace back the disguised variant up to its common ancestor via classical sequence alignment algorithms or string matching algorithms. They suggested their tool as content pre-processor, to reconstruct corrupted comments that can be then used as input for other information extraction and machine learning techniques for content classification

#### 6.4 Proposed Algorithm

The proposed algorithm “Pre-calculated Spelling Correction algorithm (PSC)” [111] uses only the pre-calculated dictionary with terms of Edit distance  $\leq 2$ . The difference between Symmetric Delete Algorithm and this algorithm is Delete and Transpose operations are considered for calculating the edit distance. Two hash tables are used to store original dictionary words and newly generated words (with edit distance between 1 and 2). In hash table 1, each row entry records the original word and its frequency, which is computed using a language model. The words with edit distance  $\leq 2$  and links to the original words are stored in the hash table 2 (See Table 6-1 and Table 6-2).

Original word	Frequency
Term <sub>1</sub>	f <sub>1</sub>
Term <sub>2</sub>	f <sub>2</sub>
...	..
Term <sub>n</sub>	f <sub>n</sub>

Table 6-1: Main Dictionary

Generated Words	Original words		
New Term <sub>1</sub>	Term <sub>11</sub>	Term <sub>12</sub>	Term <sub>13</sub> ...
New Term <sub>2</sub>	Term <sub>21</sub>	Term <sub>22</sub>	Term <sub>23</sub>
...	....	...	...
New Term <sub>m</sub>	Term <sub>m3</sub>	Term <sub>m2</sub>	Term <sub>m3</sub> ...

Table 6-2: Generated words dictionary

The query term is first checked against the main dictionary. If there are no matching words, the search continues to the generated words dictionary. If a record is found, then correct word is retrieved using the link to main dictionary. If more than one match is there, all matched words in main dictionary are being retrieved and the word with highest frequency is outputted for spell correction. The number of words in the search term list is only one; since no other words are generated dynamically for the search term as in the case of other two algorithms.

The main dictionary is the same dictionary used in Symmetric Delete Algorithm. But the generated word dictionary is much larger than Symmetric Delete Algorithm because transposition operation is also used along with delete operation to generate new words in the new algorithm. For each original word in the main dictionary, with an edit distance=1 and a word length= $n$ , there will be  $n$  words (due to deletions) and  $n-1$  words (due to transpositions) and a total of  $n+n-1=2n-1$  new words in the new dictionary for each original word.

For example take a word 'exit'(n=4)

Deletions: xit,eit,ait, exi , Transpositions: xeit,eixt,exit

Main dictionary: exit

Generated words dictionary: xit,eit,ait, exi, xeit,eixt,exit

## **6.5 Deobfuscation Method**

All the incoming messages propagate through a message-transformation pipe composed of three tasks: calculation of spamminess score of obfuscated mail,



deobfuscation process, and recalculation of spamminess score of deobfuscated mail.

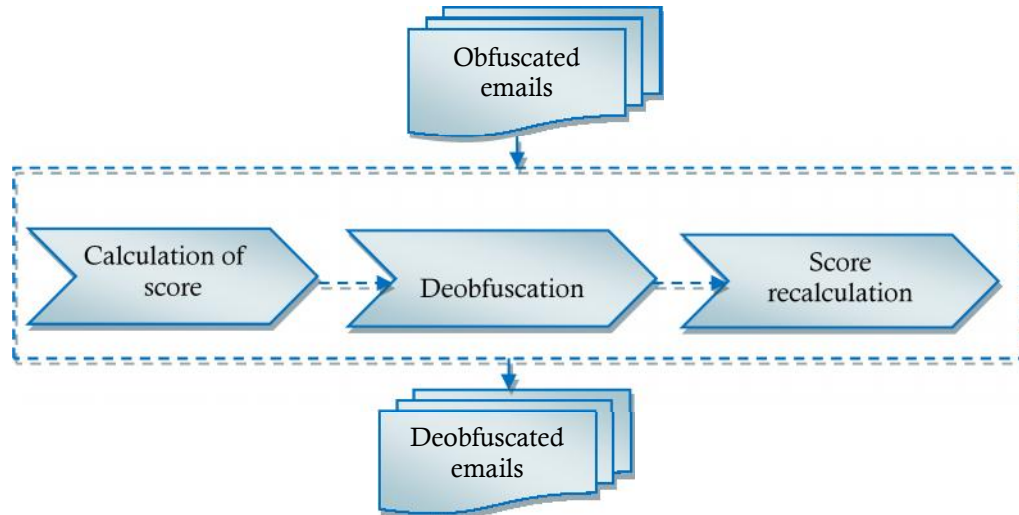


Figure 6-1: Score calculation

Since getting obfuscated mail for training data for a supervised classifier is difficult, a program to misspell the word by using some language-based heuristics is used. These messages are then passed into Apache SpamAssassin, a largely used open source spam filter, which assigns a score based on the spamminess features of the message. SpamAssassin uses a robust scoring framework and plug-ins to integrate a wide range of advanced heuristic and statistical analysis tests on email headers and body text including text analysis, Bayesian filtering, DNS block lists, and collaborative filtering databases.

Pre-calculated Spelling Correction algorithm [111] is used in the process of deobfuscation. This algorithm generates terms with an edit distance  $\leq 2$  (deletes and transpositions only) for each dictionary term and create a new

dictionary with generated terms and pointers to original terms. This has to be done only once during the initial pre-calculation step. The input term was searched in the dictionary to get the original word. The Pre-calculated spelling correction algorithm uses Damerau-Levenshtein Algorithm during the pre-calculation step. Levenshtein Edit Distance (LD) is a measure of the similarity between two strings. The greater the Levenshtein distance, the more different the strings are. The dictionary used to check the misspelled words was GNU ASpell English Dictionary[112]. The dictionaries are stored in hash tables.

### 6.6 Comparison of Algorithms

Search time complexity is the time required to find a matching word from the dictionary for the misspelled word. From table 6-3, we can say that last two algorithms are better than the second one, but the space complexities are much more than the second one because we need to store more derived words.

Edit distance : 1				
No. of words in dictionary : m				
No of letters in each word : n				
<b>Algorithm</b>	Naive Levenshtein algorithm [75]	Peter Norvig Algorithm [78]	Symmetric Delete Algorithm [78]	Pre-calculated Spelling Correction algorithm
<b>Operations</b>	insertion, deletion, substitution, transposition	insertion, deletion, substitution, transposition	Delete only. edit distance d=1	Delete and transpositions
<b>Alphabet size</b>	NA	$2n(a+1)+a-1$	$nC_d$	$2n-1$

<b>Dictionary creation</b>	NA	No changes in dictionary, all possible terms from the query term are generated at run time	Pre-calculated Dictionary, Delete only terms are derived from search term at run time	Pre-calculated Dictionary, no changes in search term
<b>Increase in Dictionary size</b>	No change	No change	$\sum_{i=1}^m n_i C_d$	$\sum_{i=1}^m 2n_i - 1$

Table 6-3: Comparison of algorithms

### 6.7 Deobfuscation Method using PSC Algorithm

**Prerequisite:**

- ) Main dictionary(hash table 1) , and frequency using language models
- ) Create Hash table 2 with original word and its generated words using PSC algorithm

**Pseudo Code:**

Search keyword in Hash table 1

If *found* then

*Search term is correct, no spelling correction is required*

else search in hash table2

If not found then No matching words in dictionary

If not found and single record match, retrieve the original word from Hash table 2

Else Retrieve all original words; Output the word with highest frequency

## 6.8 Deobfuscation Experimental Results

Points	Rule name	Description
0.0	NO_RELAYS	Informational: message was not relayed via SMTP
1.2	MISSING_HEADERS	Missing To: header
0.1	MISSING_MID	Missing Message-Id: header
1.0	MISSING_FROM	Missing From: header
1.8	LONGWORDS	Long string of long words
0.0	NO_RECEIVED	Informational: message has no Received headers
1.4	MISSING_DATE	Missing Date: header
0.0	NO_HEADERS_MESSAGE	Message appears to be missing most RFC-822 headers

Table 6-4: Common rules found in SpamAssassin

Some of the common rules set up in SpamAssassin's score checking system are given in the table given below. It depicts the rules, its descriptions and points associated with each rule. Most of these rules happened to apply on every mail in the corpus because emails in these dataset contain no information about 'from' and 'To' addresses, dates, IP addresses due the confidentiality and privacy of the emails.

The proposed system was implemented and tested on a dataset of 24 spam messages; randomly chosen from Enron dataset 1. The analysis was done on 'subject' and 'body' parts of the email. Spell correction algorithm is used to find the original word. A threshold value, i.e., edit distance=1, is set to deobfuscate

the mails. Each mail is inputted into SpamAssassin and the scores are calculated. The scores are calculated on the basis of rules violated and points of each rule violated are summed up.

The following table shows the SpamAssassin score of 24 randomly selected mails from the dataset.

#id	obfuscated score	Deobfuscated score	#id	obfuscated score	deobfuscated score
1	3.7	6.5	13	6.2	7.7
2	3.7	3.7	14	3.7	7.8
3	6.2	8.5	15	3.7	6
4	3.7	7.7	16	3.7	3.7
5	3.9	7.6	17	3.7	3.7
6	3.7	3.7	18	3.7	7.1
7	7.6	16.4	19	3.7	3.7
8	3.7	6.1	20	3.7	6.2
9	3.7	5.6	21	3.7	3.7
10	3.7	5.6	22	3.8	7.3
11	5.2	10.1	23	3.7	3.7
12	3.7	5.6	24	3.7	3.7

Table 6-5: SpamAssassin scores before and after Deobfuscation

## **6.9 Discussion**

The methods used in Symmetric Delete Spelling Correction algorithm is modified in the proposed algorithm (PSA) to get better results. The dictionary used in the proposed algorithm is augmented with new words from dictionary by applying Levenshtein algorithm (only delete and insert operations). Also the process of dynamic creation of words from search terms is eliminated. Since we are considering only delete and insert operations, the search operation would consume less time and space consumption. Hence instant search comes at no extra cost.

The two major differences between earlier algorithms and proposed algorithm are the size of dictionary generated and dynamic generation of new terms from the search term. The slight increase in dictionary size (due to transpositions operation) is a hitch compared to Symmetric Delete Spelling Correction algorithm but it is ignorable. In contrast, the removal of dynamic generation of new terms from search term will definitely reduce the search time required for spell correction.

The aim of deobfuscation process is to increase the spamminess score by finding and correcting obfuscation and thereby protecting the spam filter from cheating. As shown in results, the spamminess scores are increased for the obfuscated mails after the deobfuscation process. For example, in the case of mail #1 and mail #7, the scores are increased from 3.7 to 6.5 and 7.6 to 16.4 respectively.

The following table shows some examples:

<b>Sl. No</b>	<b>Obfuscated Email</b>	<b>Score before deobfuscation</b>	<b>Score after deobfuscation</b>
1	<p>Subject: with hgh my energy level has gone up ! stuknntroducingd0ctor – formulated hgh human growth horm0ne - also called hgh is referred to in medical science as the master hormone . it is very plentiful when we are young , but near the age of twenty - one our bodies begin to produce less of it . by the time we are forty nearly everyone is deficient in hgh , and at eighty our production has normally diminished at least 90 - 95 % . advantages of hgh : - increased muscle strength - loss in body fat - increased bone density - lower blood pressure</p>	3.7	6.5

	- quickens wound healing - reduces cellulite....		
2	Subject: re : patches work better then pillz worlds first dermal p ; atch technology for p * nis enlarg ; ment a ; dd 3 + in ; ches today - 100 % doc ; tor approved the viriitiy p ; atch r . x . was designed for men like yourself who want a b ; lgger , th ; icker , m ; ore en ; ergetic p * nis ! imagine sky will also super _ charge	7.6	16.4

Table 6-6 : Scores of obfuscated mails with before and after deobfuscation.

### 6.10 Conclusion

The use of deobfuscation method significantly improved the performance of SpamAssassin for spam detection. Since obfuscation is a common method spammers make use of, a slight improvement in filtering will improve overall performance of the spam filtering system. The pre-calculated algorithm can be used in spell checkers, word processors and search engines etc.



## Chapter 7

# Scalable Spam Filtering Solution using a Standard Framework

---

Spam consists of varieties of contents like text, image, embedded HTML, MIME attachments and also the volume of spam mails sent per day is massive. To handle this high volume, high velocity and large varieties of spam, a scalable spam filtering solution is required. Scalable solutions available for machine learning and statistical studies can be used to implement a scalable solution for spam filtering also. Comparing traditional analytics to big data analytics is different. The differences in speed, scale and complexity are tremendous. From Big data Analytics domain, Mahout is an open source library from Apache for building scalable solutions in machine learning. This paper uses Apache Mahout Framework to analyse the time and accuracy efficiencies of two Naïve Bayes classification algorithms.

---

### 7.1 Introduction

Spam has arrived in unprecedented ways. It consists of varieties of contents like text, image, embedded HTML, MIME attachments and also the volume of spam mails sent per day is massive. The perfect storm of the three V's (volume, velocity and variety) makes it extremely complex and cumbersome with current spam filtering solutions and small scale implementation of spam filters [113]. The mails that become large enough cannot be processed using conventional methods. To handle this high volume, high velocity and large

varieties of spam, a scalable spam filtering solution is required [114]. Comparing traditional analytics to Big Data analytics is different. The differences in speed, scale and complexity are tremendous. Big data refers to datasets whose size is beyond the ability of typical database software tools to capture, store, manage and analyse [98]. As technology improved, emails adopted HTML and color imagery, which lead directly to a spike in the use of email marketing. An email can facilitate a call to action, because somebody clicked on a link in an email, it would take him to a website where they'd be able to purchase a product.

Scalable solutions available for machine learning and statistical studies can be used to implement a scalable solution for spam filtering also. There are many Big Data technologies for handling Big Data like Big data stack from Google [115], NoSQL, Apache Hadoop. Apache Hadoop is an open-source platform for storage and processing of diverse data types that rapidly derives complete value from all other data. From Big data Analytics domain, Apache Mahout is an open source library from Apache for building scalable solutions in machine learning. This research uses Mahout Framework to analyse the time and accuracy efficiencies of two Naïve Bayes classification algorithms in varying sizes of dataset.

## **7.2 Apache Mahout**

Apache –Mahout is a set of scalable algorithms to carry out the clustering and classification in big data arena problem free [84], [85]. Mahout is used as a machine learning tool when the collection of data to be processed is very large, or too large for a single machine [86]. Mahout algorithms are written in Java,

and some portions are built upon Apache's Hadoop distributed computation project [87]. It doesn't provide a user interface; but a framework of tools intended to be used and adapted by developers [88].

### 7.3 Mahout in Classification

In the book [89], the authors explain how Mahout can be used to build and personalize effective classifiers. Different data mining and machine learning models are explained with examples. The book discusses classification and its applications and what algorithms and classifier evaluation techniques are supported by Mahout.

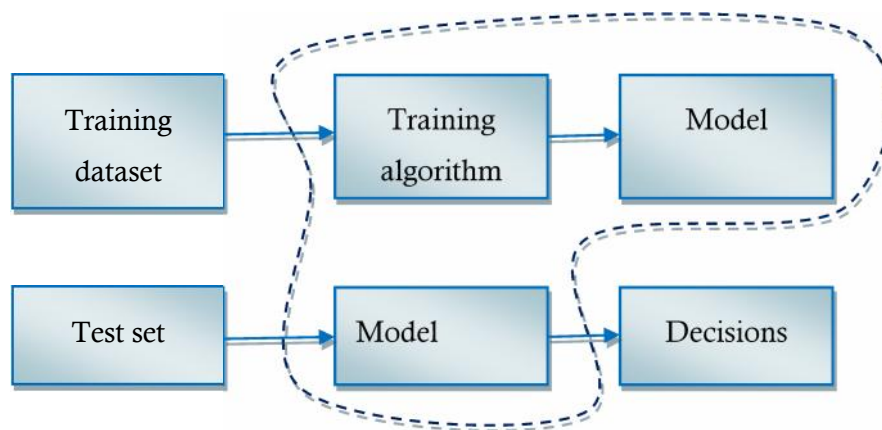


Figure 7-1: Classification systems

The paper [90] compares k-means and fuzzy c-means for clustering a noisy realistic and big dataset. They made the comparison using a free cloud computing solution Apache Mahout/ Hadoop and Wikipedia's latest articles. And the authors claim that in a noisy dataset, fuzzy c-means can lead to worse

cluster quality than k-means. They concluded that Mahout is a promise clustering technology but is premature. The study [91] uses Apache Mahout for Collaborative Filtering and conclude that it is a mature framework for building recommenders, still a lot of room for improvements and extensions. An ideal situation to evaluate an e-commerce recommender systems, the study [92] suggests to find an open-source platform with many active contributors that provides a rich and varied set of recommender system functions that meets all or most of the baseline development requirements

#### **7.4 Methodology**

This study discusses on how to choose and extract features effectively to build a Mahout classifier, how these extracted features are used for creating a model to test the new incoming mails. The steps in methodology are explained below.

##### **7.4.1 Extracting Features to Build a Mahout Classifier**

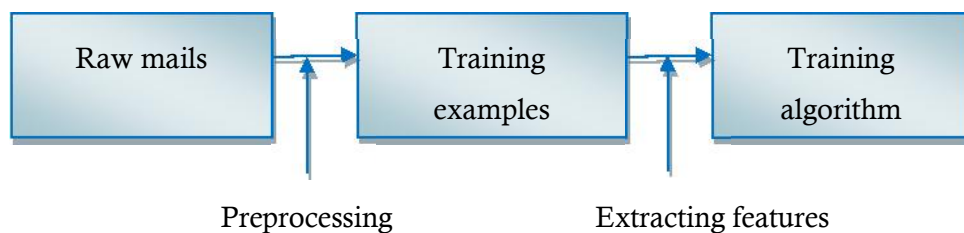


Figure 7-2: Extracting features

Getting data into a form usable by a classifier is a complex and often time-consuming step [116]. Preparing data for the training algorithm consists of two main steps:

1. Preprocessing raw data: Raw data is rearranged into records with identical fields. In spam filtering context, the data are words.
2. Converting data to vectors: Classifiable data is parsed and vectorized using custom code or tools such as Lucene analyzers and Mahout Vector encoders. Some Mahout classifiers also include vectorization code.

#### **7.4.2 Preprocessing Raw data into Classifiable data**

The first phase of feature extraction involves rethinking the data and identifying features in mails to use as predictor variables. Here the header and body parts of the mails are used to extract the features in preprocessing task.

#### **7.4.3 Transforming Raw data**

Once the features are identified, they must be converted into a format that's classifiable. This involves rearranging the data into a single location and transforming it into an appropriate and consistent form. Each record contains the fully de-normalized description of one training example.

#### **7.4.4 Classifying Spam Mails**

Mahout currently has two Naive Bayes implementations. The first is standard Multinomial Naive Bayes. The second is an implementation of Transformed Weight-normalized Complement Naive Bayes (CBayes) as introduced by [117]

where CBayes is an extension of Bayes that performs particularly well on datasets with skewed classes and has been shown to be competitive with algorithms of higher complexity such as Support Vector Machines. Both Bayes and CBayes are currently trained via MapReduce Jobs. Testing and classification can be done via a MapReduce Job or sequentially

Classification models for the spam using the learning algorithms Naïve bias and complement Naïve bias are built based on the spam data set. These models are applied to new set of test data and the efficiency is computed and compared.

#### **7.4.5 Dataset Pre-processing**

The first step in preparing a data set is to examine the data and decide which features might be useful in classifying spam.

To begin, we downloaded Enron data set from this URL:

[http://nlp.cs.aueb.gr/software\\_and\\_datasets/Enron-Spam/index.html](http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html)

The Enron data set consists of one mail per file. Each file begins with header lines that specify things such as: who sent the message, how long it is, what kind of software was used, and the subject. The predictor features in this kind of data are either in the headers or in the message body. A natural step when first examining this kind of data is to count the number of times different header fields are used across all emails. This helps determine which ones are most common and thus are likely to affect our classification of emails.

Dataset	Number of mails in each dataset	
	spam	legitimate
Enron 1	1500	3672
Enron 2	1496	4361
Enron 3	1500	4012
Enron 4	4500	1500
Enron 5	3675	1500
Enron 6	4500	1500
Total	33716	

Table 7-1: Enron Dataset –Distribution of spam and legitimate mails

#### 7.4.6 Choosing an Algorithm to Train the Classifier

The main advantage of Mahout is its robust handling of extremely large and growing data sets. The algorithms in Mahout all share scalability, but they differ from each other in other characteristics. This study uses Naive Bayes and complement Naive Bayes as the classifiers.

For complement Normal Naïve Bayes, instead of calculating the likelihood of a word occurring in a class, calculate the likelihood that it occurs in other classes. For normal Naïve Bayes, we would do the calculation and find the class with the maximum argument viz:

$$a \quad p(y) * \prod p(w|y)^f$$

Where  $f_i$  is the frequency count of word  $i$  in email  $d$

But for complement Naïve Bayes, we see the one minimum argument.

$$a \quad p(y) * \prod \frac{1}{p(w|\hat{y})^{f_i}}$$

The Naive Bayes and complementary Naive Bayes algorithms in Mahout are parallelized algorithms that can be applied to larger data sets because they can work effectively on multiple machines at once.

The Mahout implementation of naive Bayes, however, is restricted to classification based on a single text-like variable which is apt for spam problem since spam contains only words or text.

## **7.5 Classifying Enron Spam Data with Naive Bayes**

The data extraction step is applied to get the data ready for the training and then the model is trained. Once that's done, the process of evaluating initial model is started to determine whether it is performing well or changes need to be made.

### **7.5.1 Data Extraction for Naive Bayes**

First get the spam and legitimate mails into a classifiable form and convert it to a file format for use with the Naïve Bayes algorithm. The Naïve Bayes classifier's parser creates a file with each line contains the value of the target variable followed by space-delimited features, where a 1 indicates the presence of the feature name and 0 indicates absence. Each directory is scanned and each file is transformed into a single line of text that starts with the directory name and then contains all the words in the email.



### 7.5.2 Training the Naive Bayes Classifier

In this step, the Naïve Bayes classification model is trained with the training and test data converted in right format. The resulted model is stored in a directory and the model consisted of several files that contain the components of the model. These files were in binary format and used to classify the test data.

### 7.5.3 Testing with Naive Bayes Model

To evaluate the performance of newly trained model, naive Bayes model is run on the test data. The test program produced the following table shows that output generated for Enron dataset 2with 50% split using Naïve Bayes Model. The summary has raw counts of how many emails were classified correctly or incorrectly.

Correctly Classified Instances	1425	99.234%
Incorrectly Classified Instances	11	0.766%
Total Classified Instances	1436	

Table 7-2: Enron dataset2with 25% split using Naïve Bayes Model

### Confusion Matrix

a	b	←Classified as
1040	6	a = legitimate
5	385	b = spam

Table 7-3: Confusion matrix

In this testing, the naive Bayes model is performing well, with a score of nearly 93% correct. The program also produced the following confusion matrix.

**Statistics**

Kappa	0.9739
Accuracy	99.234%
Reliability	66.041%
Reliability	0.572

Table 7-4: Statistics using Naïve Bayes model

**7.6 Classifying with Complement Naïve Bayes Classifier**

In this step, the Complement Naïve Bayes classification model is trained with the training dataset as in the case of Naïve Bayes. A new model with complement Naïve Bayes algorithm is generated and this model is used to classify the test data.

**7.6.1 Testing with Complement Naïve Bayes classifier**

To classify the mails in test dataset, the newly trained is model is run with the test data. The test program produced the following. The summary has raw counts of how many emails were classified correctly or incorrectly

Correctly Classified Instances	2394	93.0431%
Incorrectly Classified Instances	179	6.9569%
Total Classified Instances	2573	

Table 7-5: Enron dataset 1 with 50% split using Complement Naïve Bayes

In this testing, the Complimentary Naive Bayes model is performing well, with a score of nearly 93% correct. The program also produced the following confusion matrix.

**Confusion Matrix**

a	b	←Classified as
132	2	a = legitimate
177	562	b = spam

Table 7-6: Confusion Matrix

**Statistics**

Parameter	Value
Kappa	0.143
Accuracy	93.0431%
Reliability	5.6466%
Reliability (standard deviation)	0.5217

Table 7-7: Statistics using Complement Naïve Bayes

## 7.7 Performance Evaluation

### 7.7.1 Time Complexity

The time complexity of Naive Bayes is  $O(Nd)$ , to compute the frequency of every feature for each class where  $N$  is number of mails and  $d$  is number of features. In Mahout Implementations this process is done via a MapReduce Job [116] and therefore the time complexity depends on the degree of parallelism we can grant.

The time taken for testing of different partitions of training set and test set are given in the following table. The algorithms took almost same amount of time even in different sample sizes of training and test data set.

Dataset	Time taken for testing at different Splits of dataset ( time in milli seconds)			
	25%	50%	75%	99%
Enron 1	2014	2083	3049	3057
Enron 2	2020	3086	3128	3140
Enron 3	2059	3091	3087	3143
Enron 4	2019	3091	3116	3184
Enron 5	2042	2068	3120	3151
Enron 6	1995	3050	3135	3129

Table 7-8: Time taken for Complimentary Naïve Bayes algorithm

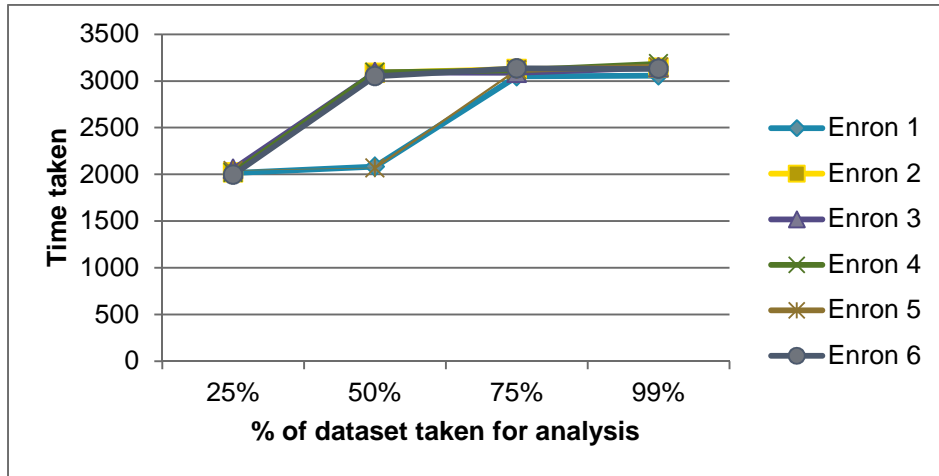


Figure 7-3: Time taken for Complimentary Naïve Bayes algorithm

Classifying mails with Mahout shows that increasing the number of mails in training set and test set do not increase the time complexity even linearly as shown in the table. The results of Naïve Bayes and complementary Naïve Bayes prove this statement.

Dataset	Time taken for testing at different Splits of dataset ( time in milli seconds)			
	25%	50%	75%	99%
Enron 1	92255	92255	93210	93350
Enron 2	92194	93219	93284	93235
Enron 3	92191	93181	93297	94338
Enron 4	92198	93248	93237	93302
Enron 5	92161	92406	93291	93194
Enron 6	92142	93365	93298	93289

Table 7-9: Time taken for Naïve Bayes algorithm

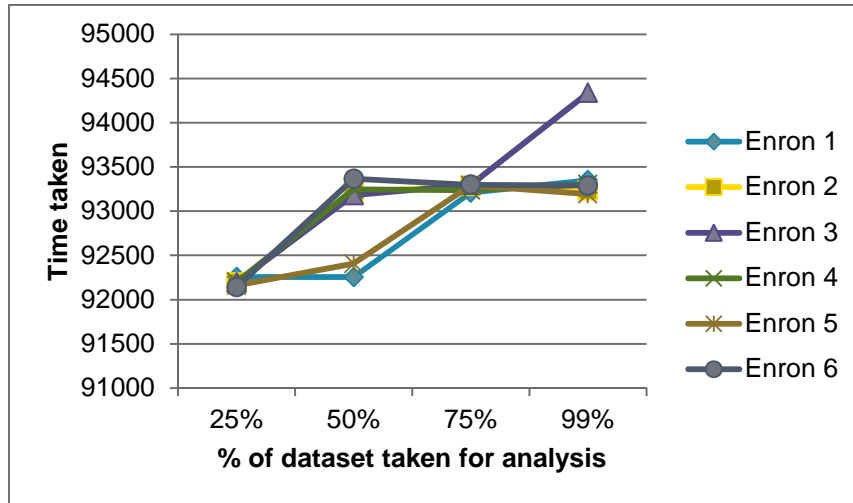


Figure 7-4: Time taken for Naïve Bayes algorithm

### 7.7.2 Accuracy

Dataset	Accuracy for testing at different Splits of dataset			
	25%	50%	75%	99%
Enron 1	95.02%	93.04%	95.76%	87.5%
Enron 2	99.24%	98.9%	98.47%	82.49%
Enron 3	98.41%	98.73%	96.55%	77.96%
Enron 4	81.50%	89.22%	95.71%	86.2%
Enron 5	96.59%	97.55%	98.76%	92.06%
Enron 6	81.39%	87.03%	93.44%	80.29%

Table 7-10: Accuracy of Complimentary Naïve Bayes algorithm

As presented in Table 7-10 the accuracy of classification is almost same and high for all the split-ups. This shows that the algorithms in Mahout are designed to work robustly and reliable in any size of datasets.

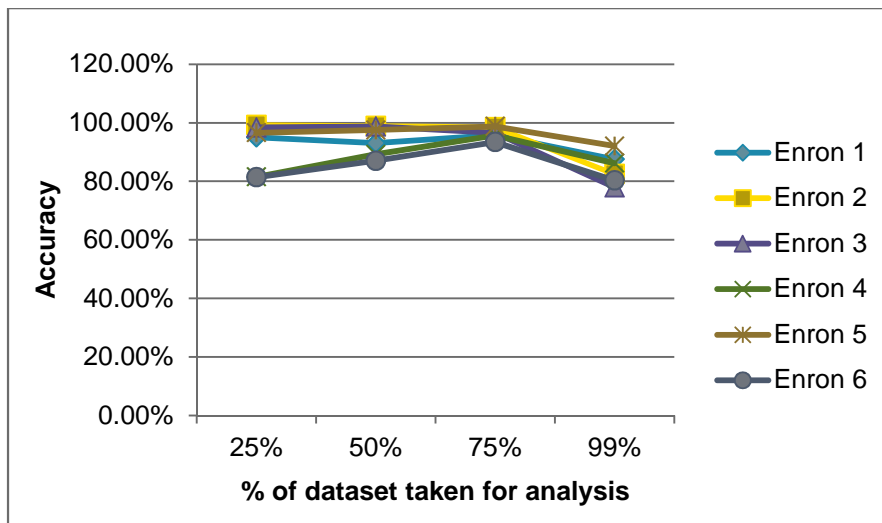


Figure 7-5: Accuracy of Complimentary Naïve Bayes algorithm

Dataset	Accuracy for testing at different Splits of dataset			
	25%	50%	75%	99%
Enron 1	98.1395	97.920	97.0226	88.2099
Enron 2	99.234	99.1456	98.6725	72.913
Enron 3	99.0566	99.0345	98.4672	74.642
Enron 4	98.6622	98.0451	96.0071	27.729
Enron 5	99.3039	99.243	98.7394	62.139
Enron 6	98.2456	97.1765	92.7236	46.54

Table 7-11: Accuracy of Naïve Bayes algorithm

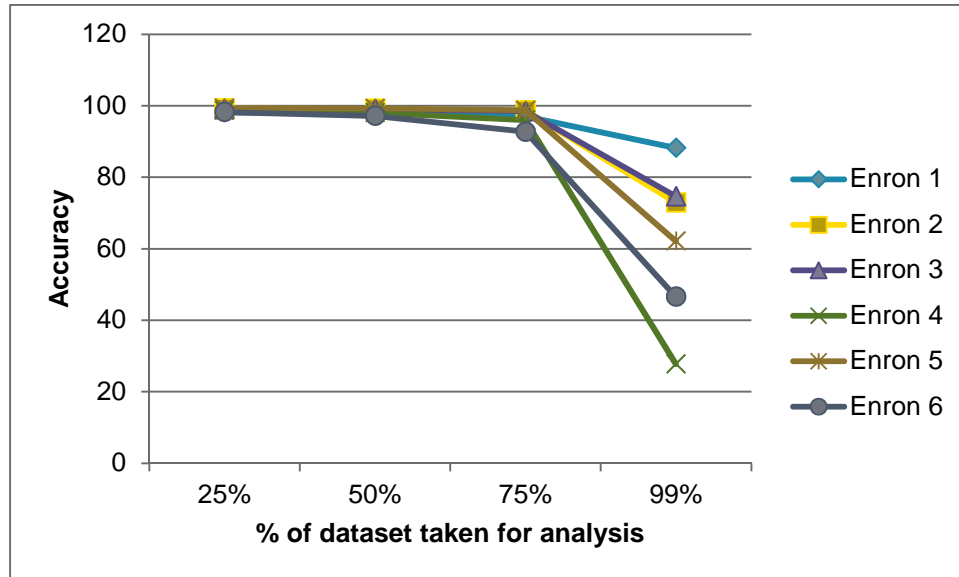


Figure 7-6: Accuracy of Naïve Bayes algorithm

## 7.8 Summary

The reason Mahout has an advantage with larger data sets is that as input data increases; the time or memory requirements for training may not increase linearly as in a non-scalable system. The classification algorithms in Mahout require resources that increase not faster than the number of training or test examples, and in most cases the computing resources required can be parallelized. This allows to trade off the number of computers used against the time the problem takes to solve.

If the training samples are more than ten million and the predictor variable is a single, text-like value, naive Bayes or complement naive Bayes may be the best choice of algorithm. Naive Bayes algorithms are best choice for data with more



than 100,000 training examples. The amount of time taken for classification does not linearly depend on the number of training data.

Apache-Mahout and Naïve Bayes algorithms are only used in this work. There are many other big data frameworks and data mining algorithms available on big data domain. This work can be extended in future to find the suitability of other algorithms and frameworks.

## Chapter 8

### Conclusion and Future Work

---

Spam filtering is a difficult task for the mail servers and it seems to be an ongoing problem. This research work put an effort to minimize the number of spams coming to user mailboxes by filtering it in mail servers by integrating different ways of filtering techniques. The contributions of this research work are elaborated and future directions to spam filtering are incorporated in this chapter.

---

#### **8.1 Conclusion**

##### **8.1.1 Spam Filtering**

Spam filtering with two different versions of the Naive Bayes (NB) classifier are discussed and evaluated experimentally. The Bernoulli model and multinomial model were included in the analysis. To accommodate the future trends and to filter spam efficiently, periodic updation of corpora is required. The number of features was reduced by applying some dimensionality reduction methods like PCA and information gain methods. Improving on stop words and selecting good cut-off document frequency, the system can perform very well on the problem of spam.

In literature, most of the spam filters are either rule based models or Bayesian models. Another idea focused on two schemes based on vector space models followed in classic Information Retrieval was explained in Chapter 2. To find semantic distance, cosine similarity was used in both methods. This study has been carried out on 101 real datasets with attributes of td-idf values. First method used all the mails in the training set to test against the spam, while in the second method, only the centroids of each class (only two vectors) were used to find the similarity.

VSM using Rocchio Classification was much faster than simple VSM because the number of iterations required is less. The results show that VSM using Rocchio Classification scheme performs better than Simple VSM scheme. Since templates are changing with time and promotional activities, the training data need to be changed periodically in order to incorporate new templates. The simple VSM model is efficient to find out the exact spam template. But when the test training set becomes large, time to find similarity is also increasing ( $O(n)$ ). Hence we have to update the training corpus by deleting the templates that are not used by spammers and by adding new mail templates. The training data size can be further reduced by storing only unique mail templates. The optimum size of the training set has to be studied more. The method presented here can be enhanced to find semantic distance between mails.

### **8.1.2 Incremental Updation**

In this study, the proposed template mail selection which uses supervised genetic algorithm and its operations, i.e., crossover and mutation, to create best templates in the training set for future spam filtering. The experiments show

that proposed template mail selection performs efficiently and give better results. In addition, the system allows manual adjustments in the threshold value for fitness function, percentage of crossover operations, to the appropriate level of filtering. The overall quality of template mail instances is increased by applying genetic algorithm.

In K-Means algorithm the cluster centroids forms the training set for the final spam filtering task. This method can be applied and tested on big data to get an optimum training set. Other works in related area also discussed in this chapter.

### **8.1.3 Deobfuscation**

The use of deobfuscation method significantly improved the performance of SpamAssassin for spam detection. Since obfuscation is a common method spammers make use of, slight improvement in filtering task will improve overall performance of the mail system. A new algorithm was also devised to implement deobfuscation method. SpamAssassin is used to compute the spam score of each mail. Bogus words, Homoglyphs can be removed using this method.

### **8.1.4 Scalable Solution**

The reason Mahout has an advantage with larger data sets is that as input data increases; the time or memory requirements for training may not increase linearly as in a non-scalable system. The classification algorithms in Mahout require resources that increase not faster than the number of training or test examples, and in most cases the computing resources required can be

parallelized. Since Mahout is implemented on MapReduce paradigm, scalable, cost effective, flexible and speedy solutions can be obtained.

If the training samples are more than ten million and the predictor variable is a single, text-like value, naive Bayes or complement naive Bayes may be the best choice of algorithm. Naive Bayes algorithms are best choice for data with more than 100,000 training examples. The amount of time taken for classification does not linearly depend on the size of training data.

Apache-Mahout and Naïve Bayes algorithms are only used in this work. There are many other big data frameworks and data mining algorithms available on big data domain. This work can be extended in future to find the suitability of other algorithms and frameworks.

## **8.2 Contributions**

The main contribution of this research work includes:

- ) A spam filter system with Bayesian models and Vector space models was developed and tested. Experiments were carried out with transformations of raw emails with various dimensionality reduction methods and improved preprocessing tasks. We achieved accurate classification results using Bernoulli model than Bayesian models. When template driven mails are filtered using vector space models we obtained good accuracy filter with simple vector space model, but space and complexity were high compared to Rocchio classification. Hence a trade-off based on these complexities is to be considered while choosing a filter.
- ) Two models were identified for incremental updation of spam training set and its accuracy and time required were tested. The models chosen -

Genetic algorithm and K-Means algorithm- have performed equally well with the given dataset.

- ) A deobfuscation system with modified spell correction algorithm is devised and tested with spam training set. This way the rate of cheating a spam filter is decreased. The modified algorithm can also be used with spell suggestion, word processors, search engines etc.
- ) Moreover, an experimental work was conducted using a big data framework-Apache Mahout- to implement a scalable and robust solution to spam filtering. With experiments on different dataset sizes, it is found that the accuracy and time requirements to filter spam are not changing much.

### **8.3 Future Work**

The concerns and solutions to some of aspects of spam filtering are discussed in this thesis. But this work has substantial scope for further research to improve accuracies and performance. The testing and performance evaluation can be done using other databases available in this domain to learn new patterns in spam emails. Deployment of the software on large-scale real-life spam filtering environment with the consideration of all the associated algorithmic or computational issues, time complexities, accuracy and concurrency is also appealing.

We have carried out dimensionality reduction methods using PCA, information gain, document frequency tf-idf values. Alternative methods like  $\chi^2$  analysis, I-Match algorithm (to find clusters of near duplicate records) etc. can also be tested. There are a lot of machine learning algorithms in practice, but we tried only a few. Algorithms like SVM, Random Forest, Decision trees

are a good choice to make good classifiers. Although some studies based on these algorithms are already in scholarly literature, its variants may be tested to produce better results. There are many public datasets available; we only used Enron dataset and some personal datasets. The experiments listed in this thesis can also be extended to those other datasets and the results may be compared to check the consistency of methods.

Alternate Bayesian models would be useful than Simple Naïve Bayes learning to improve the overall performance of the system. There are methods which combine Boosting and Bayesian learning. Boosting is a general method of improving the predictive accuracy of any two-class learning algorithm, which works in successive stages. In each stage, examples that are misclassified by the previous stage classifier are up-weighted and a new classifier is learnt. This process is repeated for as many stages as desired. This aspect can be incorporated in future studies. The idea of ensemble methodology [118], [119] is to build a predictive model by integrating multiple models. The main idea behind the ensemble methodology is to weigh several individual classifiers, and combine them in order to obtain a classifier that outperforms every one of them. AdaBoost [118] like algorithms may be included in spam filtering context to improve the classification accuracy in future works.

In the area of deobfuscation, enhanced homoglyph based replacement algorithms for email may be implemented to detect obfuscation. As mentioned in Chapter 6, I-Match, DNA sequencing and pattern matching algorithms can be incorporated with filtering template driven mails to obtain better results. One can use the same I-match or its variants to find out best template mails for training set by eliminating near duplicates. Ontology based extended datasets

and algorithms can be implemented along with spam filtering to improve its accuracy and recall.

Currently only Naïve Bayes and Complimentary Naïve Bayes are implemented in Apache Mahout. One can add more machine learning algorithms to the toolkit as a MapReduce Job in order to reduce the time taken for preprocessing, learning, classification and testing tasks without compromising on accuracy.

### **8.3.1 LDA for Email spam filtering**

Latent Dirichlet allocation (LDA)[120] is a fully generative statistical language model on the content and topics of a corpus of documents. It takes a group of documents and returns a number of topics that are most relevant to these documents. Given a set of unknown mails, LDA will tell whether it is spam or not. Simple LDA, Linked LDA methods are applied in many web spam filtering tools/studies. This can be extended to Email spam also. LDA and its variants in the context of Email spam filtering needs to be further explored.

Some other factors like weightage to attributes can also be considered while computing performance measures. One such case is that misclassifying a legitimate mail as spam is much more severe than misclassifying a spam mail as legitimate mail.[59] New weighted measures to take care of such cases can be introduced as a future work.

### **8.3.2 Web Service**

As Spam is a global problem, a web service may be provided internationally so that anyone who wants to check spamminess of a mail can use that server. It would help administrators and normal users around the world to have a



standard solution for their mail server or user client programs to be free from spams. As of now no such solution is available globally.

Once a user submits an email to this web service, it should return the spamminess value ( $0 \leq \leq 1$ ) of that mail. Crowdsourcing can be used to develop a solution in this direction.

### **8.3.3 Spam Filter**

A provision may be provided to get the spamminess score and to filter mails according to some threshold using a spam filter plugin with mail clients. Some of the cases, it is found that the solutions to spams itself is spam generating tools. The problem is that such spam filters should come from genuine vendors; otherwise such solutions make the problem more severe.

## **8.4 Summary**

While all these approaches discussed in this thesis and other related studies seem good, it is very difficult to make a fair comparison between their performances based on the results. This is because most of the works used different corpora, different pre-processing tasks. Moreover most of the methods were implemented in different toolkits, environments and configurations. Feature selection methods, optimum number of attributes and changes in spam phenomenon were also varying. The performance of all these methods may vary due to these factors. Hence evaluating a particular method and analysing with previous methods and make a fair comparison is very difficult. Therefore spam filtering task still remains as an open question.

An attempt has been made in this chapter to bring out the contributions of the thesis and a general conclusion and enlisted the scope for future research in this area. When the training data set is up-to-date with recent changes in spam patterns and the learnt classification models perform well, spams can be reduced at a rate user expects.

## REFERENCES

- [1] A. A., C. Pallas, and Z. Patrikakis, “An Overview of Spam Phenomenon; and the Key Findings of a Survey for Spam in Greece’,” in *1st International Scientific Conference eRA, Supported by TEI of Piraeus (GR) & University of Paisley (UK), Tripolis*, 2006.
- [2] S. Dutta, “How to Stop Spam Emails – 6 Most Effective Ways to Filter Junk Emails.” [Online]. Available: <http://techchai.com/2011/05/23/how-to-stop-spam-emails-6-most-effective-ways-to-filter-junk-emails/>. [Accessed: 10-May-2013].
- [3] S. Hasib, M. Motwani, A. Saxena, and others, “Anti-Spam Methodologies: A Comparative Study,” *International Journal of Computer Science and Information Technologies*, vol. 3, no. 6, pp. 5341–5345, 2012.
- [4] V. V Arutyunov, “Spam: Its past, present, and future,” *Scientific and Technical Information Processing*, vol. 40, no. 4, pp. 205–211, 2013.
- [5] S.-A. Kelin, “State Regulation of Unsolicited Commercial E-Mail,” *Berkeley Technology Law Journal*, pp. 435–459, 2001.
- [6] “Definition of Spam,” *The Spamhaus Project Ltd.*, 2010. [Online]. Available: <https://www.spamhaus.org/consumer/definition/>. [Accessed: 10-May-2013].
- [7] T. Subramaniam, H. A. Jalab, and A. Y. Taqa, “Overview of textual anti-spam filtering techniques,” *International Journal of Physical Sciences*, vol. 5, no. 12, pp. 1869–1882, 2010.
- [8] H. Drucker, D. Wu, and V. N. Vapnik, “Support vector machines for

- 
- spam categorization,” *IEEE Transactions on Neural networks*, vol. 10, no. 5, pp. 1048–1054, 1999.
- [9] T. Oda and T. White, “Increasing the accuracy of a spam-detecting artificial immune system,” in *The 2003 Congress on Evolutionary Computation*, 2003, vol. 1, pp. 390–396.
- [10] L. Lazzari, M. Mari, and A. Poggi, “A collaborative and multi-agent approach to e-mail filtering,” in *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, 2005, pp. 238–241.
- [11] W. Zhao and Z. Zhang, “An email classification model based on rough set theory,” in *Proceedings of the International Conference on Active Media Technology*, 2005, pp. 403–408.
- [12] S. Youn and D. McLeod, “Efficient spam email filtering using adaptive ontology,” in *Fourth International Conference on Information Technology*, 2007, pp. 249–254.
- [13] J. Wu and T. Deng, “Research in anti-spam method based on bayesian filtering,” in *Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, 2008, vol. 2, pp. 887–891.
- [14] Spamhaus, “Definition of Spam,” 2010. [Online]. Available: <https://www.spamhaus.org/consumer/definition/>. [Accessed: 10-May-2011].
- [15] M. Y. Schaub, “Unsolicited email: Does Europe allow spam? The state of the art of the European legislation with regard to unsolicited commercial communications,” *Computer Law & Security Review*, vol. 18, no. 2, pp. 99–105, 2002.

- 
- [16] Process Software, “Common Spammer Tricks White Paper.” [Online]. Available:  
[http://www.process.com/psc/fileadmin/user\\_upload/whitepapers/pm as/common\\_spammer\\_tricks.pdf](http://www.process.com/psc/fileadmin/user_upload/whitepapers/pm%20as/common_spammer_tricks.pdf). [Accessed: 11-Sep-2015].
- [17] “Spam Emails.” [Online]. Available:  
<https://www.ictlounge.com/html/spam.htm>. [Accessed: 12-Oct-2013].
- [18] T. Kumaresan, “Image Spam Filtering using Support Vector Machine and Particle Swarm Optimization,” *International Journal of Computer Applications*, pp. 17–21, 2015.
- [19] “Attachment spam – the latest trend,” *GFi Whitepaper*, 2009. [Online]. Available: <https://www.gfi.com/whitepapers/attachment-spam.pdf>.
- [20] “Types of spam.” [Online]. Available:  
<https://securelist.com/threats/types-of-spam/>. [Accessed: 12-Sep-2013].
- [21] A. Leung, “SPAM The Current State,” *Telus Corporation*, 2003. [Online]. Available: <http://www.homer.com.au/webdoc/spam.pdf>. [Accessed: 13-Nov-2015].
- [22] C. F. Endorf, *Secured Computing: A SSCP Study Guide*. Trafford Publishing, 2002.
- [23] F. Sebastiani, “Machine learning in automated text categorization,” *ACM computing surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.
- [24] “Junk Email Filter.” [Online]. Available:  
<http://www.junkemailfilter.com/spam/>. [Accessed: 14-Mar-2014].

- 
- [25] “How Spam & Virus Filtering Works.” [Online]. Available: [http://www.cordhosting.com/spam\\_virus\\_filtering/how\\_it\\_works.php](http://www.cordhosting.com/spam_virus_filtering/how_it_works.php). [Accessed: 11-Apr-2014].
- [26] “Sender Policy Framework.” [Online]. Available: <http://www.openspf.org/>. [Accessed: 10-Feb-2014].
- [27] J. Levine, “DNS blacklists and whitelists, IRTF anti-spam research group,” *Internet Research Task Force (IRTF)*, 2010. [Online]. Available: <https://tools.ietf.org/html/rfc5782>. [Accessed: 12-Oct-2013].
- [28] “Greylisting.” [Online]. Available: <http://www.greylisting.org/>. [Accessed: 20-Oct-2011].
- [29] “DNS Whitelist - Protect against false positives.” [Online]. Available: <http://www.dnswl.org/>. [Accessed: 21-Sep-2012].
- [30] “Spam domain blacklist.” [Online]. Available: <http://www.joewein.de/sw/blacklist.htm>. [Accessed: 14-Oct-2012].
- [31] H. Esquivel, A. Akella, and T. Mori, “On the effectiveness of IP reputation for spam filtering,” in *Second International Conference on Communication Systems and Networks (COMSNETS)*, 2010, pp. 1–10.
- [32] “SenderBase.” [Online]. Available: <http://www.senderbase.org/>. [Accessed: 07-Feb-2013].
- [33] “IRTF The Anti Spam Research Group wiki, DNS validation.” [Online]. Available: <http://wiki.asrg.sp.am/wiki/DNSvalidation>. [Accessed: 02-Feb-2013].
- [34] R. Clayton, “Stopping Outgoing Spam by Examining Incoming Server

- 
- Logs,” in *CEAS 2005 - Second Conference on Email and Anti-Spam, Stanford University, California, USA, 2005*.
- [35] E. P. Sanz, J. M. G. Hidalgo, and J. C. C. Pérez, “Email spam filtering,” *Advances in computers*, vol. 74, pp. 45–114, 2008.
- [36] Q. Luo, B. Liu, J. Yan, and Z. He, “Design and Implement a Rule-Based Spam Filtering System Using Neural Network,” in *International Conference on Computational and Information Sciences (ICCIS)*, 2011, pp. 398–401.
- [37] G. K. Gupta, *Introduction to data mining with case studies*. PHI Learning Pvt. Ltd., 2014.
- [38] S. M. Ali, W. A B, and S. Ahmed, *Data mining: methods and techniques*. South Melbourne, Vic.: Thomson Learning Australia, 2007.
- [39] R. Donde, “Spam: Is it time to legislate?” [Online]. Available: <http://www.legalservicesindia.com/articles/spamli.htm>. [Accessed: 04-Sep-2011].
- [40] “Register of Known Spam Operations database.” [Online]. Available: <https://www.spamhaus.org/rokso/>. [Accessed: 09-Sep-2012].
- [41] A. K. Pujari, *Data mining techniques*, Illustrate. Universities press, 2001.
- [42] L. Shi, Q. Wang, X. Ma, M. Weng, and H. Qiao, “Spam email classification using decision tree ensemble,” *Journal of Computational Information Systems*, vol. 8, no. 3, pp. 949–956, 2012.
- [43] S. Chakraborty and B. Mondal, “Spam mail filtering technique using different decision tree classifiers through data mining approach-A

- 
- comparative performance analysis,” *International Journal of Computer Applications*, vol. 47, no. 16, 2012.
- [44] R. K. Kumar, G. Poonkuzhali, and P. Sudhakar, “Comparative study on email spam classifier using data mining techniques,” in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2012, vol. 1, pp. 14–16.
- [45] L. Zhang and T. Yao, “Filtering junk mail with a maximum entropy model,” in *Proceeding of 20th international conference on computer processing of oriental languages (ICCPOL03)*, 2003, pp. 446–453.
- [46] Y. Li, B.-X. Fang, and L. Guo, “A Novel Online Spam Filter Based on URLs and Maximum Entropy Model,” in *International Conference on Computational Intelligence and Security*, 2006, vol. 2, pp. 1575–1578.
- [47] S. Krasser, Y. Tang, J. Gould, D. Alperovitch, and P. Judge, “Identifying image spam based on header and file properties using C4. 5 decision trees and support vector machine learning,” in *Information Assurance and Security Workshop, 2007. IAW’07. IEEE SMC*, 2007, pp. 255–261.
- [48] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. D. Spyropoulos, and P. Stamatopoulos, “Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach,” *arXiv preprint cs/0009009*, 2000.
- [49] V. Metsis, I. Androutsopoulos, and G. Paliouras, “Spam filtering with naive bayes-which naive bayes?,” in *CEAS*, 2006, vol. 17, pp. 28–69.
- [50] E.-S. M. El-Alfy, “Learning methods for spam filtering,” *International*



- 
- Journal of Computer Research*, vol. 16, no. 4, 2008.
- [51] D. Puniškis, 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.
- [52] A. Khorsi, “An overview of content-based spam filtering techniques,” *Informatika (Slovenia)*, vol. 31, no. 3, pp. 269–277, 2007.
- [53] T. Joachims, “Transductive inference for text classification using support vector machines,” in *ICML*, 1999, vol. 99, pp. 200–209.
- [54] O. Amayri and N. Bouguila, “Online spam filtering using support vector machines,” in *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*, 2009, pp. 337–340.
- [55] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, “A Bayesian approach to filtering junk e-mail,” in *Learning for Text Categorization: Papers from the 1998 workshop*, 1998, vol. 62, pp. 98–105.
- [56] I. Androutsopoulos, J. Koutsias, K. V Chandrinos, G. Paliouras, and C. D. Spyropoulos, “An evaluation of naive bayesian anti-spam filtering,” *arXiv preprint cs/0006013*, 2000.
- [57] C. Chen, Y. Tian, and C. Zhang, “Spam filtering with several novel bayesian classifiers,” in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008, pp. 1–4.
- [58] X. Carreras and L. Marquez, “Boosting trees for anti-spam email filtering,” *arXiv preprint cs/0109015*, 2001.
- [59] L. Zhang, J. Zhu, and T. Yao, “An evaluation of statistical spam

- 
- filtering techniques,” *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 3, no. 4, pp. 243–269, 2004.
- [60] S. J. Delany, P. Cunningham, and A. Tsymbal, “A Comparison of Ensemble and Case-Base Maintenance Techniques for Handling Concept Drift in Spam Filtering.,” in *FLAIRS Conference*, 2006, pp. 340–345.
- [61] X. Yue, A. Abraham, Z.-X. Chi, Y.-Y. Hao, and H. Mo, “Artificial immune system inspired behavior-based anti-spam filter,” *Soft Computing*, vol. 11, no. 8, pp. 729–740, 2007.
- [62] G. B. Bezerra, T. V Barra, H. M. Ferreira, H. Knidel, L. N. de Castro, and F. J. Von Zuben, “An immunological filter for spam,” in *International conference on artificial immune systems*, 2006, pp. 446–458.
- [63] T. Oda and T. White, “Immunity from spam: An analysis of an artificial immune system for junk email detection,” in *International conference on artificial immune systems*, 2005, pp. 276–289.
- [64] B. Biggio, G. Fumera, I. Pillai, and F. Roli, “Improving Image Spam Filtering Using Image Text Features,” *Fifth Conference on Email and Anti-Spam (CEAS)*, pp. 1–3, 2008.
- [65] B. Biggio, G. Fumera, I. Pillai, and F. Roli, “Image spam filtering using visual information,” in *ICIAP 2007. 14th International Conference on Image Analysis and Processing*, 2007, pp. 105–110.
- [66] M. G. Lee, “U.S. Patent No. 7,817,861. Washington, DC: U.S. Patent and Trademark Office.” 2010.
- [67] S. V. Wakade, “Classification of Image Spam,” University of Akron,

- 2011.
- [68] B. Mehta, S. Nangia, M. Gupta, and W. Nejdl, “Detecting image spam using visual features and near duplicate detection,” *Proceeding of the 17th international conference on World Wide Web WWW 08*, vol. 6, no. 2, pp. 497–506, 2008.
- [69] B. Leiba, J. Ossher, V. T. Rajan, R. Segal, and M. N. Wegman, “SMTP Path Analysis,” in *CEAS*, 2005.
- [70] A. Ramachandran and N. Feamster, “Understanding the network-level behavior of spammers,” in *ACM SIGCOMM Computer Communication Review*, 2006, vol. 36, no. 4, pp. 291–302.
- [71] P. O. Boykin and V. P. Roychowdhury, “Leveraging social networks to fight spam,” *Computer*, vol. 38, no. 4, pp. 61–68, 2005.
- [72] J. S. Kong, P. O. Boykin, B. A. Rezaei, N. Sarshar, and V. P. Roychowdhury, “Scalable and Reliable Collaborative Spam Filters: Harnessing the Global Social Email Networks,” in *CEAS*, 2005.
- [73] R. Kumar, “How to increase visibility of your social media?” [Online]. Available: <http://www.techncom.net/2015/03/how-to-increase-visibility-of-your-social-media/>. [Accessed: 09-Sep-2016].
- [74] C. D. Manning, P. Raghavan, H. Schütze, and others, *Introduction to information retrieval*. Cambridge university press Cambridge, 2008.
- [75] “Damerau–Levenshtein distance.” [Online]. Available: [https://en.wikipedia.org/wiki/Damerau–Levenshtein\\_distance](https://en.wikipedia.org/wiki/Damerau–Levenshtein_distance). [Accessed: 12-Dec-2012].

- 
- [76] “How to Write a Spelling Corrector.” [Online]. Available: <http://norvig.com/spell-correct.html>. [Accessed: 10-Oct-2015].
- [77] “Fast approximate string matching with large edit distances in Big Data.” [Online]. Available: <http://blog.faroo.com/2015/03/24/fast-approximate-string-matching-with-large-edit-distances/>. [Accessed: 10-Oct-2015].
- [78] “1000X Faster Spelling Correction Algorithm.” [Online]. Available: [blog.faroo.com](http://blog.faroo.com). [Accessed: 10-Oct-2015].
- [79] C. Liu and S. Stamm, “Fighting Unicode-Obfuscated Spam,” *Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit*, pp. 45–59, 2007.
- [80] V. Freschi, A. Seraghiti, and A. Bogliolo, “Filtering obfuscated email spam by means of phonetic string matching,” in *European Conference on Information Retrieval*, 2006, pp. 505–509.
- [81] T. Eggendorfer, J. Keller, and I. Informatikzentrum, “Preventing spam by dynamically obfuscating email-addresses,” *Proceedings of CNIS*, 2005.
- [82] P. Poornachandran, D. Raj, S. Pal, and A. Ashok, “Effectiveness of Email Address Obfuscation on Internet,” in *Innovations in Computer Science and Engineering*, Springer, 2016, pp. 181–191.
- [83] J. Crain, L. Opyrchal, and A. Prakash, “Fighting phishing with trusted email,” in *Availability, Reliability, and Security, 2010. ARES’10 International Conference on*, 2010, pp. 462–467.
- [84] G. Ingersoll, “Introducing apache mahout,” *IBM developerWorks Technical Library*, 2009. [Online]. Available:
-

- 
- <https://www.ibm.com/developerworks/library/j-mahout/index.html>.  
[Accessed: 10-Nov-2016].
- [85] P. Giacomelli, *Apache mahout cookbook*. Packt Publishing Ltd, 2013.
- [86] G. Ingersoll, “Introducing Apache Mahout Scalable, commercial-friendly machine learning for building intelligent applications,” *IBM Corporation*. 2009.
- [87] A. Mahout, “Scalable machine learning and data mining,” 2012. [Online]. Available: <http://mahout.apache.org/>. [Accessed: 11-Jul-2015].
- [88] A. Mahout, “Scalable machine-learning and data-mining library,” 2008. [Online]. Available: [mahout. apache. org](http://mahout.apache.org/). [Accessed: 06-Jun-2015].
- [89] A. Gupta, *Learning Apache Mahout Classification*. Packt Publishing Ltd, 2015.
- [90] C. Rong and others, “Using mahout for clustering wikipedia’s latest articles: A comparison between k-means and fuzzy c-means in the cloud,” in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, 2011, pp. 565–569.
- [91] S. Schelter and S. Owen, “Collaborative filtering with Apache Mahout,” *Proc. of ACM RecSys Challenge*, vol. i, no. September 2012, pp. 1–13, 2012.
- [92] S. G. Walunj and K. Sadafale, “An online recommendation system for e-commerce based on apache mahout framework,” in *Proceedings of the 2013 annual conference on Computers and people research*, 2013, pp. 153–158.

- 
- [93] B. Liu, *Web data mining: exploring hyperlinks, contents, and usage data*. Springer Science & Business Media, 2007.
- [94] D. T. Larose, *Data mining methods & models*. John Wiley & Sons, 2006.
- [95] D. T. Larose, *Discovering knowledge in data: an introduction to data mining*. John Wiley & Sons, 2014.
- [96] K. J. Cios, R. W. Swiniarski, W. Pedrycz, and L. A. Kurgan, “Data mining The Knowledge Discovery Process,” in *Data Mining*, Boston, MA: Springer US, 2007, pp. 9–24.
- [97] V. Metsis, I. Androutsopoulos, and G. Paliouras, “Spam filtering with naive bayes-which naive bayes?,” *Ceas*, p. 9, 2006.
- [98] M. Minelli, M. Chambers, and A. Dhiraj, *Big Data Analytics - Emerging BI and Analytics trends for today’s businesses*. wiley, 2013.
- [99] R. Baeza-Yates and B. Ribeiro-Neto, “Modern information retrieval,” *New York*, vol. 9, p. 513, 1999.
- [100] S. M. Weiss and N. Indurkha, *Predictive data mining: a practical guide*. Morgan Kaufmann Publishers, 1998.
- [101] A. Chowdhury, O. Frieder, D. Grossman, and M. C. McCabe, “Collection statistics for fast duplicate document detection,” *ACM Transactions on Information Systems*, vol. 20, no. 2, pp. 171–191, 2002.
- [102] A. Kolcz, A. Chowdhury, and J. Alspector, “The Impact of Feature Selection on Signature-Driven Spam Detection,” in *First Conference on Email and Anti-Spam(CEAS-2004)*, 2004.
- [103] A. K. Jain, “Data clustering: 50 years beyond K-means,” *Pattern*

- 
- recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.
- [104] H.-H. Bock, “Clustering methods: a history of k-means algorithms,” *Selected contributions in data analysis and classification*, pp. 161–172, 2007.
- [105] P. S. Bradley and U. M. Fayyad, “Refining Initial Points for K-Means Clustering,” in *ICML*, 1998, vol. 98, pp. 91–99.
- [106] M. Steinbach, G. Karypis, V. Kumar, and others, “A comparison of document clustering techniques,” in *KDD workshop on text mining*, 2000, vol. 400, no. 1, pp. 525–526.
- [107] A. Kořcz, A. Chowdhury, and J. Alspector, “The Impact of Feature Selection on Signature-Driven Spam Detection.”
- [108] X. Zhong, “Deobfuscation based on edit distance algorithm for spam filtering,” in *2014 International Conference on Machine Learning and Cybernetics*, 2014, vol. 1, pp. 109–114.
- [109] S. Rojas-galeano and U. D. Fjc, “On Obstructing Obscenity Obfuscation,” *ACM Transactions on the Web (TWEB)*, vol. 11, no. 2, pp. 1–24, 2017.
- [110] C. M. Pinzón and S. Rojas-galeano, “A Web-Forum Free of Disguised Profanity by Means of Sequence Alignment 1 Un foro web libre de obscenidades enmascaradas utilizando,” *Ingeniería y Universidad*, vol. 20, no. 2, pp. 239–265, 2016.
- [111] L. Varghese, M. H Supriya, and K. Poullose Jacob, “Pre-calculated Spelling Correction Algorithm (PSC),” *International Journal of Computer Science and Technology*, vol. 8, no. 1, 2017.

- 
- [112] “Aspell.” [Online]. Available: <ftp://ftp.gnu.org/gnu/aspell/dict/index.html>. [Accessed: 12-May-2015].
- [113] B. Baesens, “Analytics in a Big Data World: The Essential Guide to Data Science and its Applications,” pp. 1–11, 2014.
- [114] N. Marz and J. Warren, *Big Data: Principles and best practices of scalable realtime data systems*. Manning Publications Co., 2015.
- [115] J. Tigani and S. Naidu, *Google BigQuery Analytics*. John Wiley & Sons, 2014.
- [116] S. OWEN, R. ANIL, T. DUNNING, and E. FRIEDMAN, *Mahout in Action*. Manning Publications Co. Greenwich, CT, USA, 2011.
- [117] J. D. M. Rennie, L. Shih, J. Teevan, and D. R. Karger, “Tackling the Poor Assumptions of Naive Bayes Text Classifiers,” *Proceedings of the Twentieth International Conference on Machine Learning (ICML)-2003*, vol. 20, no. 1973, pp. 616–623, 2003.
- [118] T. G. Dietterich, “Ensemble Methods in Machine Learning,” *Multiple Classifier Systems*, vol. 1857, pp. 1–15, 2000.
- [119] L. Rokach, “Ensemble-based classifiers,” *Artificial Intelligence Review*, vol. 33, no. 1–2, pp. 1–39, 2010.
- [120] I. Bíró, D. Siklósi, J. Szabó, and A. A. Benczúr, “Linked Latent Dirichlet Allocation in Web Spam Filtering \*.”



## List of Publications

- Liny Varghese, Supriya M.H. and Poullose Jacob, K. (2017). Improved spam Filter for obfuscated emails. *International Journal of Data Mining and Emerging Technologies*, Vol.7, No.1, pp. 33-35, May 2017
- Liny Varghese, Supriya M.H. and Poullose Jacob, K. (2017). Spam: A Big Data Challenge. *International Journal of Advanced Research in Computer Science (IJARS)*, vol.8, No.1, pp. 195–198, April 2017.
- Liny Varghese, Supriya M.H. and Poullose Jacob, K. (2017). Pre-calculated spelling correction algorithm, *International Journal of Computer Science and Technology (IJCST)*, Vol. 8, No. 1, January-March 2017
- Liny Varghese, Supriya M.H. and Poullose Jacob, K. (2015). Finding Template Mails from Spam Corpus Using Genetic Algorithm and K-Means Algorithm. *International Journal of Computer Science and Information Technologies (IJCSIT)*, Vol. 6, No. 4, pp. 3548–3551.
- Liny Varghese, Supriya M.H. and Poullose Jacob, K (2012). Feature Selection and Comparison of Two Naïve Bayes Classification Methods in the Context of Spam Filtering. *International Journal of Computer Science & Communication (IJCSC)*, Vol3, No.1, pp.81-84, January-June 2012.

- Liny Varghese, Supriya M.H. and Poulose Jacob, K. (2012). Filtering Template driven spam mails using Vector Space models. International Journal of Computer Applications (IJCA). Vol. 39, No. 14, pp. 33-35 February 2012, DOI: 10.5120/4891-7383.
- Liny Varghese, K Poulose Jacob (2018), A Review On Open Data Repositories For Email Spam And Pre-Processing Methods, Library Herald, Vol .56, Issue.1,pp. 3-10. . ISSN : 0976-2469. DOI : [10.5958/0976-2469.2018.00001.5](https://doi.org/10.5958/0976-2469.2018.00001.5)