A CONTENT-BASED SPAM FILTERING APPROACH USING ARTIFICIAL NEURAL NETWORKS

A THESIS
SUBMITTED TO THE
COLLEGE OF SCIENCE, BAGHDAD UNIVERSITY
IN PARTIAL FULLFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER IN COMPUTER SCIENCE

BY
NOOR ALHUDA JASIM MOHAMMED ALMADHKOR
(B.Sc. 2010)

SUPERVISED BY
DR. SARAB M. HAMEED

August 2013
Acknowledgement

I'd like to express my deep gratitude to my supervisor, Dr. Sarab M. Hameed for the continuous support of my research and for her patience, motivation, enthusiasm, and immense knowledge. I have learned many things since I became Dr. Sarab student. She spent very much time instructing me how to search, collect information and put it down in the suitable places. Her guidance helped me in all the time of research work and writing. I am sure it would have not been possible without her help.

Special thanks to the department of computer science in college science in university of Baghdad. Most of my theoretical foundations are built in the department of computer science.

During the preparations phase and search, many friends are helpful to color my life. I have to acknowledge all my friends for their assistance in many aspects that I cannot list all them.

I owe more than thanks to my family member, my parents, my brothers and my husband for their financial support and encouragement through my life.
Dedicate

To my mom, dad, my husband, my brothers, my dear friends and all those who supported and encouraged me along the way.
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

Abstract
As the popularity of the Internet increased, electronic mails (emails) have become a very common and convenient medium for daily communications. The spam defined as unsolicited commercial bulk emails, or uninteresting emails has threatened the internet security and email services. Since the spammers constantly improve their techniques to compromise the spam filters, building a spam filter that can be incrementally learned and adapted became an active research field.

This thesis proposes a spam filtering approach using two “Artificial Neural Networks” ANNs algorithms “Back Propagation” BP and “Optical Back Propagation” OBP to identify whether a message is spam or legitimate email based on the content of the message. These two neural networks should be trained with group of trained samples to distinguish whether a message is spam or legitimate email. These samples are drawn from Spam-based dataset. The samples of this dataset should be preprocessed to be in a suitable form that could be understood by neural networks. The preprocessing operations are features extraction using “Principle Component Analysis” PCA and normalization.

Several experiments are conducted to show the effectiveness of the proposed spam filtering approach and a comparison is made among these experiments with different evaluation measurements. The results of the tested spam-based dataset feature set size of 100% show that BP and OBP are comparable in terms of accuracy (100%), recall (100%), precision (100%), false positive (0%), and false negative (0%) but OBP takes the least execution time (1.222 seconds). Also the results show that OBP with 75%, 50%, and 25% feature set size are better than the corresponding BPs in all evaluation measurements.
TABLE OF CONTENTS

List of Figures 7
List of Tables 9
List of Abbreviations 11
List of Symbols 13

CHAPTER ONE: OVERVIEW

1.1 Introduction 14
1.2 The Evolution of Spam 15
1.3 Related Work 16
1.4 Thesis Objective 20
1.5 Thesis Organization 20

CHAPTER TWO: SPAM AND NEURAL NETWORKS

2.1 Introduction 22
2.2 Electronic Mail 23
2.3 Email Threats 23
  2.3.1 Out-Bound E-Mail Threat 24
  2.3.2 In-Bound E-Mail Threat 25
2.4 The Spam 25
  2.4.1 The Advertisement Spam 26
  2.4.2 Financial Spam 26
  2.4.3 Phishing 27
2.5 Non-Technological Solution for Limiting Spam 28
2.6 Technological Solution for Limiting Spam 30
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

2.6.1 Origin-Based Filtering 30
2.6.2 Content-Based Filtering 33
2.7 Challenges of Filtering Techniques 34
2.8 Artificial Neural Network 36
2.9 Types of Artificial Neural Network 39
   2.9.1 Feed-Forward Artificial Neural Network 40
   2.9.2 Recurrent Artificial Neural Network 41
2.10 Learning the Neural Network 42
2.11 Back Propagation 43
2.12 Optical Back Propagation 47
2.13 Preprocessing 47

CHAPTER THREE: NEURAL NETWORKS FOR SPAM FILTERING

3.1 Introduction 49
3.2 The Proposed Approach for Spam Filtering 49
3.3 Spam-Based Dataset 51
3.4 Spam-Based Dataset Preprocessing 52
   3.4.1 Feature Extraction 52
   3.4.2 Normalization 54
3.5 Training Phase 56
   3.5.1 BP Training 57
   3.5.2 OBP Training 62
3.6 Testing Phase 65
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

### CHAPTER FOUR: EXPERIMENTS AND RESULTS EVALUATION

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Introduction</td>
<td>66</td>
</tr>
<tr>
<td>4.2 Evaluation Measure</td>
<td>66</td>
</tr>
<tr>
<td>4.3 Training and Testing Dataset</td>
<td>67</td>
</tr>
<tr>
<td>4.4 Experimental Results</td>
<td>69</td>
</tr>
<tr>
<td>4.4.1 Impact of Size of Training Dataset vs. Size of Testing</td>
<td>71</td>
</tr>
<tr>
<td>4.4.2 Impact of Feature Set Size</td>
<td>76</td>
</tr>
<tr>
<td>4.4.3 Error Signal Evaluation</td>
<td>85</td>
</tr>
<tr>
<td>4.4.4 Execution Time</td>
<td>87</td>
</tr>
</tbody>
</table>

### CHAPTER FIVE: CONCLUSIONS AND FUTURE WORK

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Introduction</td>
<td>88</td>
</tr>
<tr>
<td>5.2 Conclusions</td>
<td>88</td>
</tr>
<tr>
<td>5.3 Future Works</td>
<td>89</td>
</tr>
</tbody>
</table>

### REFERENCES

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>REFERENCES</td>
<td>90</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF PUBLICATION</td>
<td>97</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure (2.1): Email Threats Types 24
Figure (2.2) Feed-Forward (FNN) and Recurrent (RNN) Topology of an Artificial Neural Network. 40
Figure (2.3): Output Calculation 44
Figure (3.1): Basic Components of the Spam Filtering Approach Based on Neural Networks 50
Figure (3.2): Input Layer, Hidden Layers and Output Layer of BP-1 and OBP-1 63
Figure (3.3): Input Layer, Hidden Layer and Output Layer of BP-2 and OBP-2 63
Figure (3.4) Input Layer, Hidden Layer and Output Layer of BP-3 and OBP-3 64
Figure (3.5): Input Layer, Hidden Layer and Output Layer of BP-4 and OBP-4 64
Figure (4.1) Accuracy Results of BP-1, BP-2, BP-3, and BP-4 80
Figure (4.2) Recall Results of BP-1, BP-2, BP-3, and BP-4 81
Figure (4.3) Precision Results BP-1, BP-2, BP-3, and BP-4 81
Figure (4.4) False Positive Results of BP-1, BP-2, BP-3, and BP-4 82
Figure (4.5) False Negative Results of BP-1, BP-2, BP-3, and BP-4 82
Figure (4.6) Accuracy Results of OBP-1, OBP-2, OBP-3, and OBP-4 83
Figure (4.7) Recall Result of OBP-1, OBP-2, OBP-3, and OBP-4 83
Figure (4.8) Precision Results of OBP-1, OBP-2, OBP-3, and OBP-4 84
Figure (4.9) OBP-1, OBP-2, OBP-3, and OBP-4 False Positive Result 84
Figure (4.10) False Negative Result of OBP-1, OBP-2, OBP-3, and OBP-4 85
Figure (4.11) Depicts the Performance in Terms of Error Signal for Four BP Structure

Figure (4.1) Depicts the Performance in Terms of Error Signal for Four BP Structure
LIST OF TABLES

Table (4.1): Training Dataset Groups Details 68
Table (4.2): Testing Dataset Groups Details 68
Table (4.3): Quantitative Structure of BP and OBP Neural Networks 70
Table (4.4): Setting Parameters of BP and OBP Structures 70
Table (4.5) BP-1 Accuracy Results 71
Table (4.6) BP-1 Recall Results 72
Table (4.7) BP-1 Precision Result 72
Table (4.8) BP-1 False Positive Results 73
Table (4.9) BP-1 False Negative Results 73
Table (4.10) OBP-1 Accuracy Results 74
Table (4.11) OBP-1 Recall Results 74
Table (4.12) OBP-1 Precision Results 75
Table (4.13) OBP-1 False Positive Results 75
Table (4.14) OBP-1 False Negative Results 76
Table (4.15): BP-2, BP-3 and BP-4 Accuracy Results 77
Table (4.16): BP-2, BP-3 and BP-4 Recall Results 77
Table (4.17): BP-2, BP-3 and BP-4 Precision Results 77
Table (4.18): BP-2, BP-3 and BP-4 False Positive Result 78
Table (4.19): BP-2, BP-3 and BP-4 False Negative Results 78
Table (4.20): OBP-2, OBP-3 and OBP-4 Accuracy Results 78
Table (4.21): OBP-2, OBP-3 and OBP-4 Recall Result 79
Table (4.22): OBP-2, OBP-3 and OBP-4 Precision Results 79
Table (4.23): OBP-2, OBP-3 and OBP-4 False Positive Results 79
Table (4.25): OBP-2, OBP-3 and OBP-4 False Negative Results 80
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

Table (4.26) Execution Time of BP and OBP Structures  

87
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

LIST OF ABBREVIATIONS

AC       "Accuracy"
AIS      "Artificial Immune System"
ANN      "Artificial Neural Network"
BP       "Back Propagation"
BPNN     "back-propagation neural network"
CBART    "Classifier based on Bayes Additive Regression Tree"
CLA_ANN  "Continuous Learning Approach Artificial Neural Network"
DHA      "Directory harvesting"
DoS      "Denial of Service"
Email    "Electronic Mail"
FFNN     "Feed-Forward Artificial Neural Network"
FN       "False Negative"
FP       "False Positive"
HMMs     "Hidden Markov Models"
IP       "Internet Protocol"
ISPs     "Internet Service Providers"
K-NN     "K-Nearest Neighbor"
NB       "Naïve Bayes"
NN       "Neural Network"
OBP      "Optical Back Propagation"
OCR      "Optical Character Reader"
P        "Precision"
PCA      "Principal Component Analysis"
R        "Recall"
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBNN</td>
<td>&quot;Radial Basis Neural Network&quot;</td>
</tr>
<tr>
<td>RNN</td>
<td>&quot;recurrent artificial neural network&quot;</td>
</tr>
<tr>
<td>S</td>
<td>&quot;Spam&quot;</td>
</tr>
<tr>
<td>SIR</td>
<td>&quot;Security Intelligence Report&quot;</td>
</tr>
<tr>
<td>SP</td>
<td>&quot;Spam Precision&quot;</td>
</tr>
<tr>
<td>SR</td>
<td>&quot;Spam Recall&quot;</td>
</tr>
<tr>
<td>SVM</td>
<td>&quot;Support Vector Machine&quot;</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>&quot;Transmission Control Protocol/Internet Protocol&quot;</td>
</tr>
<tr>
<td>TP</td>
<td>&quot;True Positive&quot;</td>
</tr>
<tr>
<td>UBE</td>
<td>&quot;Unsolicited Bulk Email&quot;</td>
</tr>
<tr>
<td>UCE</td>
<td>&quot;Unsolicited Commercial Email&quot;</td>
</tr>
<tr>
<td>UCI</td>
<td>&quot;University of California, Irvine&quot;</td>
</tr>
</tbody>
</table>
LIST OF SYMBOLS

δ  Error Signal
η  Learning Rate
λ  Steepness Parameter
∞  Momentum
Δ  Changeable Parameter
 ε  Stopping Criteria
Chapter One
Overview

1.1 Introduction
Today, computer has replaced all means of traditional communication significantly. Many distant communication tools claim to be interactive, but few can offer two-way communication. (Email) “Electronic Mail” is the most popular means of communication medium nowadays. Email and Internet are effective tools for interaction as well as to make a bridge of communication between people [Rez08].

Email security is a priority concern for many organizations. There are various threats to email security. Email security is threatened by a range of issues. One of the most publicized and high risk of all issues is spamming [Coc04].

Internet users differ widely in what they perceive to be spam. Some users consider all advertisements, jokes and chain letters or even all unwanted messages to be spam, while others try to define it in terms of existing acceptable used policies or network etiquette rules. Leung defines spam as unsolicited email messages or news articles sent in bulk to recipients without their permission. The Center for Democracy and Technology uses a broader definition and refers to spam merely as junk mail. Junk mail can consist of jokes and chain letters from business colleagues, friends and family. Solkin identifies the two most common definitions of spam as being (UCE) “unsolicited commercial email” and (UBE) “unsolicited bulk email” [Chi05].
Spam emails are flooding networks simply as a spammer can make a profit or achieve its selfish purposes by sending spam emails. However spam emails are doing harm to others, both networks and human society. Spam emails cost people time and money, cause the consumptions of computing and network resources, degrade the network performance, and lead to a lot of security problems from the networks [Hao11].

1.2 The Evolution of Spam
The era of spamming began in 1978 and lasted until the middle of the 90’s. At the first years, spam was sent manually. Spamming needed huge amount of human resource and so it did not reach millions of users. Spammers used to send the messages one by one and used inner address lists of small communities [Gul06].

Spams were initially distributed over newsgroups. Although the very first incident is disputed, two events are said to mark the beginning of unsolicited bulk mailings. The first took place in January 1994 when a young employee at a Michigan Adventist college sent a message to nearly 5000 newsgroups announcing the coming of Christ. The second followed on its heels when two lawyers, Laurence Canter and Martha Seigal, sent a message to some 7000 groups offering help obtaining a green card. The offer involved the completion and mailing of forms available to anyone free of charge from the United State government [Ree04].

Spamming slowly became a business. The first known email list was offered for sale in 1995 with more than 2 million addresses. After the first weak attempts in 1997 the first real spam-filter software was made.
Nevertheless spam turned to be totally out of control, by the end of 1997 statistics showed an exponential growth. The early spam filters were rule-based software. All the new tricks of the spammers indicated to settle up new rules into the filters. Unfortunately, the spammers were able to try the filters with their messages, so they had the ability to change the letter to get through the filters [Gul06].

The big breakthrough was in 2002, when Paul Graham introduced the use of machine learning and statistical classification techniques on the field of spam filtering with Bayesian networks [Gul06].

From this time spammers could only hope that they found the right method to pass the filters, but they could not be sure they could reach a defined number of users [Gul06].

1.3 Related Work

Several research publications have been proposed in the literature for filtering spam, some of these publications are:

- In [Dav04], they investigate the impact of applying more sophistication to lower layers in the filtering process like impact of pair tokens on the classification process. Several types of obfuscation they discussed, and the results obtained by removing certain types of obfuscation show improvements in the classification process. Two classifiers were used (K-NN) "K-Nearest Neighbor" with three neighbor (k=3) so it called (3-NN), and “Bayesian”. A combination of Obfuscation 1, 4 and 2(i) could decrease error further. In obfuscation 1, the results are as follows: dataset1 (3-NN: (FP) “False Positive” =0.0175, (FN) “False
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

Negative” = 0.1445), dataset2: (3-NN: FP=0.115278, FN=0.062778), and dataset3: (3-NN: FP=0.2226, FN=0.0171), (Bayesian: FP=0.0494, FN=0.0777)). In Obfuscation 4, the results are as follows: dataset1 (3-NN: FP=0.03825, FN=0.1015), (Bayesian: FP=0.01625, FN=0.3105), dataset2 (3-NN: FP=0.185, FN=0.034167), (Bayesian: FP=0.033333, FN=0.027778), and dataset3 (Bayesian: FP=0.2599, FN=0.007), (Bayesian: FP=0.0983, FN=0.0545)). In Obfuscation 2(i), the results are follows: dataset1 (3-NN: FP=0.0145, FN=0.18725), (Bayesian: FP=0.02475, FN=0.0535), dataset2 (3-NN: FP=0.211944, FN=0.073611), (Bayesian: FP=0.028333, FN=0.0875), and dataset3 are: (3-NN: FP=0.2295, FN=0.0249), (Bayesian: FP=0.0707, FN=0.0364).

- In [Pun06], a neural network (NN) approach is applied to the classification of spam. The method employs attributes comprised of descriptive characteristics of the evasive patterns that spammers employ rather than the context or frequency of keywords in the messages. They found out which NN configuration will have the best performance and least error to the desired output. Spam/email dataset used was created by variety of text attributes consisting of most common words and characters for spam emails. The NN trained using 57 email parameters, producing the lowest number of misclassifications, giving the lowest FP = 5.18% and FN = 9.43% values.

- In [You07], an adaptive ontology is used to find an efficient spam email filtering method. Four classification methods: NN, (SVM)
“Support Vector Machine” classifier, (NB) “Naïve Bayes” classifier, and J48 classifier evaluated the effects based on different datasets and different features. The best classification method was obtained from a training datasets of 4500 emails, and 55 features. These results are used as in experiment of ontology spam filter and produced the following results: for spam: (TP "True Positive" =0.952, FP=0.015, precision =0.976, and recall=0.952), and for email: (TP=0.985, FP=0.048, precision=0.969, recall=0.985).

In [Tar09], the techniques involved in the design of the spam filters include NB, SVM, NN, and (CBART) “Classifier based on Bayes Additive Regression Tree”. They used real life dataset created by themselves from email received in ten different email accounts set up on different mail servers. The dataset used is consisting of 8000 emails and 50 features. The results are as follow: (NB: FP=3.95, FN=4.09), (SVM: FP=10.21, FN=6.73), (NN: FP=12.08, FN=7.39), (CBART: FP=3.78, FN= : FP = 3.78 , FN = 3.64).

In [Tah10], a modification on (ANN) in the input layers is applied which allows the input layers to be changed over time and to replace useless layers with new promising layers which give promising results. They use a developed perceptron learning algorithm approach. Testing is done by subjecting ANN to input layers that were not used in training without adjusting the weights. Their modifications give promising results that could be used in the process of fighting against spam. They have an accepted FP value when the number of the input layers is 300 (FP= 0.534%) and (FN= 3.668%). The best FP value is
found when the number of input layers is 700 (FP= 0.121%) and (FN= 5.465%).

- In [Awa11], the most popular machine learning methods: Bayesian, K-NN, ANNs, SVMs, Artificial immune system AIS and Rough sets RS classification. Spam Assassin spam corpus is used. In addition to the body of the email message they use some fields of the header. The experiment is performed with the most frequent words in spam email; they select 100 of them as features. The results are as follow: (NB: SR=98.46, SP=99.66, Accuracy=99.46), (SVM: SR=95.00, SP=93.12, Accuracy=96.90), (KNN: SR=97.14, SP=87.00, Accuracy=96.20), (NN: SR=96.92, SP96.02, Accuracy=96.83), (AIS: SR=93.68, SP=97.75, Accuracy=96.23), (RS: SR=92.26, SP=98.70, Accuracy=97.42).

- In [Kuf12], a new technique for filtering spam is presented. The technique consists of a single perceptron that is designed to learn and distinguish email and spam sending server parameter values and messages. The perceptron algorithm due to the incorporation of a continuous learning feature also produces favorable detection rates. They used spam and ham messages from the Spam Assassin. The best FP value is found when the number of iterations is 900 (FP = 0.097 and FN = 5.247).

- In [Ndu13] a spam filtering system using (HMMs) "Hidden Markov Models" and artificial neural networks to filter out spam where word obfuscation on the keyword is conducted to evade detection. The results showed that this model was able to detect over 90% of spam
with a FP of less than 13%. The experiment used a dataset of 75419 email messages with 172 email keywords from TREC 2007 Spam Corpus with five thresholds such as 60%, 65%, 70%, 75% and 80% meant to evaluate the best performance for the spam filter. Each group of email keyword was classified based on each of the five thresholds. The results show that spam filter recorded the highest spam recall of 95.87% at threshold 60% while the lowest spam recall of 71.90% at threshold 80% was recorded. The filter achieved the highest spam precision of 97.75% at threshold 80% with the lowest spam precision of 85.93% at threshold 60%. The highest accuracy decision of 88.37% at threshold 70% was successfully achieved.

1.4 Thesis Objective
The aim of this thesis is to present an automated tool to identify whether a message is spam or legitimate email using two neural network algorithms namely (BP) “back propagation” and (OBP) “optical back propagation” and show the applicability of these algorithms to filter spam in terms of the accuracy, recall, precision, false positive, false negative, and complexity.

1.5 Thesis Organization
The rest of this thesis is organized as follows:

- Chapter Two "SPAM AND NEURAL NETWORKS" the first part of this chapter presents an introduction to email, types of email threats and clarifies spam threats in some details. It also, presents the types of spam and technological and non-technological solutions for eliminating the
spam. The chapter presents, in the second part, the basic background of neural networks, its benefits, learning paradigms including back propagation and optical back propagation neural networks.

- **Chapter Three** "NEURAL NETWORKS FOR SPAM FILTERING" presents a general layout of the proposed neural network approach for spam filtering. Then, it explains the components of this approach including preprocessing, training, and testing in details. There is also an explanation of the dataset used in this research, samples and details of these samples.

- **Chapter Four** "EXPERIMENTS AND RESULTS EVALUATION" presents the results of the proposed spam filtering approach and evaluates the performance of this proposal.

- **Chapter Five** "CONCLUSIONS AND FUTURE WORK" presents conclusion remarks and some future work suggestions.
Chapter Two
Spam and Neural Networks

2.1 Introduction
Email is an efficient form of communication that has become widely adopted by both individuals and organizations. Today, more and more people are relying on email to connect them with their friends, family, colleagues, customers and business partners. Unfortunately, as email usage has evolved, so too has its threats, in particular spam, which is also known as unsolicited bulk email or junk mail, has become an increasingly difficult threat to detect and is being delivered in incredibly high volumes.

There are several approaches which try to stop or reduce the huge amount of spam on individuals. These approaches include legislative measures such as anti-spam laws world-wide. Other techniques are known such as Origin-Based filters which are based on using network information and IP addresses in order to detect whether a message is spam or not. The most common techniques are the filtering techniques which attempt to identify whether a message is spam or not based on the content and other characteristics of the message.

First part of this chapter describes the types of email threats and clarifies the spam threats in some details. Also, it presents the types of spam and technological and non-technological solutions for eliminating them. The second part of this chapter presents the basic background of neural networks, its benefits, learning paradigms and back propagation and optical back propagation neural networks.
2.2 Electronic Mail

Email is a computer-based communication system where messages can be written by a sender on a computer. These messages are then transmitted via computers to the addressee’s mail server where they can be opened and read by the receiver. Originally these messages could contain only text, but nowadays anything that is storable on a computer can be transmitted via email messages. Messages that contain other information than text are considered as email messages with *attachments*. These attachments are normally files created with other programs (e.g. Frame Maker, Word, and Excel) than the email program.

Email messages consist of two parts: A list of *headers* and a *body*. The body is used for the actual message. On the other hand, the headers consist of a *tag* and a *content* that define e.g. who the message should be sent to (the To-header with to: as the tag and the addressee(s) as the content) and the topic of the message (the Subject-header). Some of these headers are mandatory, but most are optional.

One mandatory header is the Subject, which is normally shown when the content of a mail folder is displayed on the screen. It is thereby often used for identifying the message. There are other headers in each message that are also used for identification by users, such as the name or the user id of the sender, and arrival time for the message [Oll98].

2.3 Threats of Email

Email security has become a hot topic in information technology circles as new exploits and vulnerabilities affecting the most popular email clients and
operating systems continue to make headline news on a regular basis. Email security is threatened by a range of issues such as spam, virus, phishing, and so on [Pam04]. The identified email threats are classified as In-Bound and Out-Bound Email threats as show in figure (2.1) [Dha12].

![Diagram of Email Threats Types](image)

**Figure (2.1): Email Threats Types**

### 2.3.1 Out-Bound Email Threats

Out-Bound email threats are those that originate from someone sitting inside the corporate network. Any information that is being transmitted over the Internet must be considered at risk from being seen or in some way tampered with. In an organization, Email policies should be carefully designed and implemented that any confidential data is masqueraded and sent outside in any form. For example, .exe files should ensure that it is not sent by changing it files type to .pdf or .doc which is allowable [Dha12].
2.3.2 In-Bound Email Threats

On the other hand, inbound email threats are those that originate from outside the corporate network, for example, from an attacker on the Internet who intends to penetrate the corporation’s perimeter defenses. There are several types of inbound threats such as: (spam, phishing, virus, executable attachment, denial of service, directory harvesting, relay hijacking) [Dha12].

2.4 The Spam

It is all email the user does not want to receive and has not asked to receive. Thus, spam is in the eye of the beholder, and is therefore all unwanted emails that the user cannot easily stop from receiving. Some of the recent literature has correctly arrived at the same definition, but also include UCE, which only includes a small portion of the overall spam problem.

Spam is exist because email is a very cost effective method of marketing legitimate products or services to millions of users. Physical bulk mail per recipient costs are substantially higher (about 100 times higher) than email advertisements. At the same time, email can also be used to conduct scams and confidence schemes to steal information or user identities. Although only a minute percentage of email users respond to spam messages, given the low cost of distribution, it is enough to fuel the popularity and existence of spammers and spam messages.

It is important to note that spam is not only annoying to the individual user, but also represents a security risk and resource drain on the system. By weeding out spam from the email stream, the user is once again empowered to use their email for what it was mean to be, a personal communication tool.
2.4.1 Advertisement Spam

Most spam is commercial advertisement, often a direct product offer. Spam costs the sender very little to send, compared to other advertisement methods. The most common subcategories of the advertisement spam are:

- **Online Pharmacy spam**: Spam promoting different versions of Viagra, Cialis, -depressant pills that can be purchased online.
- **Penny Stock spam**: Stock-encouraging spam, encouraging people to buy cheap stocks.
- **Porn or (sex-) dating spam**: Porn-sites and (sex-) dating sites were often marketed via spam (nowadays its rate out of all spam is getting less and less, fortunately).
- **Pirate Software spam**: spam offering pirate software, usually much cheaper than the official prices.
- **Online Casino spam**: Spam promoting gambling in online casinos.
- **Fake Degrees spam**: Spammers often try to sell fake Degrees and Diplomas.
- **Mule job spam**: Promoting jobs ‘working from home’ (which are typically scams, or mule jobs, like laundering money) [Glu06].

2.4.2 Financial Spam

While advertisement spam have at least a little probability, that the responder could get something for the sent money, the financial spam only...
tries to fool people and get their money somehow, without the chance to buy anything. The most common financial spam kinds are the following:

- **419 scams**: Usually a plea for help to recover millions of dollars from a bank account in a foreign country (typically Nigeria).
- **Lottery spam**: Similar to the 419 scam, these spam are telling, ‘You have already won X Million’ in order to try to extract transfer fees etc. [Glu06].

### 2.4.3 Phishing

Phishing spam is fake alert from banks (mostly Citibank), PayPal, eBay etc., and it asks for confirmation, validation or monitoring of details in order to defraud people of their personal information. Phishing spam are usually linked to fake login sites, which can be used to capture user details (e.g. passwords) in order to use this information to steal money or goods. The term phishing was coined because the fraudsters are “fishing” for personal information.

Fraudulent emails harm their victims through loss of funds and identity theft. They also make a draw back in on-line business, since people lose their trust in Internet transactions. Phishing emails use mostly the listed methods:

- **Using the company’s Image**: the fraudulent emails often contain the company’s logo and use similar fonts and color schemes.
- **Links to the real company’s site**: The main link in a fraudulent email sends the recipient to the fraudulent phishing web site, but many
fraudulent emails include other links that send the recipient to sections of the real company’s web site.

- *Email appears to be from the spoofed company:* To further convince the recipient that the email originated from the company, the spammers use an email address that appears to be from the company (e.g., @ebay.com, @paypal.com) [Glu06].

2.5 Non Technological Solutions for Limiting Spam

The basic nature of these solutions is that they do not use any technological tools to address the problem rather they demand the users and companies to take actions which will terrify people from sending spam. If appropriate awareness and devotion is created on the side of email users then the following solutions can have very good results [Nou07].

- **Recipient Revolt:** this solution suggests that on reception of any spam the user will react with anger in emails and in physical world. This solution helped significantly to scare more legitimate companies to keep themselves away from using junk email and forced the "Internet Service Providers" ISPs to change policies. Some of the advantages from this solution are: Forcing ISPs to change policies, legit companies will be afraid to spam resulting in removal of email ids from their contacts and if it gains momentum then it will be having a nice positive feedback. The fewer spams the more effort can be spent on punishing them.

- **Customer Revolt:** most of the spams contain advertisements of different sorts from companies. To deal with it this solution suggests
that companies to which the users submit their data should be forced to disclose what they will do with that data and should stick to whatever they claim. There should be proper publishing of policies on the web pages, mentioning the purpose of data gathering.

- **Vigilante Attack**: this solution suggests that spam addresses should be deal with anger and should be treated with mail bombs and denial of service attacks. Though it will make spammers to think before sending spam but sometimes an innocent might be a victim claiming that he is spammer.

- **Hide email Address**: this solution includes using two emails addresses. One email address is used to receive all of the emails. The user then scans the emails and those found valid are forwarded to the second email address. The second email address is disclosed only to known persons and is never publicized on the internet.

- **Contract-Law and Limiting Trial Accounts**: this solution requires an agreement between the user and the organization which provide the email facility. The user should sign a proper agreement before get the registration. Sufficient information should be gathered regarding the user to know his identity. The account should be on trial basis. After passing the trail successfully i.e. without being reported to have send spam, his account will get registered fully. If found violating the laws at any stage, his account will be abundant and should be punished. While this solution looks quite attractive but the big hurdle in its implementation is the disclosure of people’s identity without their will
to the organizations which might not be acceptable to many users [Nou07].

2.6 Technological Solutions for Limiting Spam

Spam filtering method is not only used for technical purposes such as overspreading of network bandwidth and email storage, but also related to social issues such as child safety, and phishing email. Many of these methods has done lot of good job for ISPs by helping them to safe at least a ton of money and also filtering of spam mails into junk, allowing legitimate mails to be delivered successfully, but still they cannot be relied on because they are not so effective [Ola11]

The following subsections present different filtering methods which are Origin-based filtering and content-based filtering.

2.6.1 Origin-Based Filtering

Origin or address based filters are methods which is based on using network information such as "Internet Protocol" IP and email address in order to detect whether a message is spam or not. That is, the domain which the message (spam) is been listed, either blacklist or Whitelist verification list because spam and trusted mails come from the domain of some IP address. The information found on network header or domain name and TCP/IP address can be used for spam detection [Ola11]. Origin based filter can be divided into several types such as Blacklist filters, Whitelist filters and Challenge/Response filters

- **Blacklist filter**: Blacklists are the oldest approach to filter spam messages. A central list, usually managed and altered by an operator,
is used to store information about the addresses of mail servers sending spam. Especially ISPs, hosting mail accounts for many users, used to apply this filter technique and simply blocked all messages coming from a server listed on such a blacklist. There are two levels of blacklisting: the network-level and the address-level blacklisting [Nic04].

1. The network-level blacklisting is based on creating intentional network outages. The method has the ability to detect spam letters based on its origin rather than its content. Unfortunately new spam hosts can pop up instantly and the propagation time could be a significant weakness. Moreover if a legitimate user was accidentally blacklisted, there is no way to get off the blacklist, hence all mails were rejected from the blacklisted part of the network.

2. The address-level blacklist is an updated list of known spam sender addresses. There are on-line accessible blacklists and the user can administrate personal blacklist as well. By receiving a letter, a simple search engine tries to find the address of its sender in the list. If it matches, the letter should be surely marked as spam, or with more strict rules, it could be deleted immediately. The “answer” of the spammers for this solution is to try sending the letters from addresses, containing random parts within, to avoid the match with the blacklist entries. Both blacklisting solutions implicated only a small change from the spammers’ side and made no big change of the amount of spam,
but it is still a very good technique to combine with other filter methods [Glu06].

- **White list:** A white list is a collection of reliable contacts. If email comes from the members of this list, it should be marked automatically as legitimate letter what is also called *ham*. Just as the blacklisting, the white list also needs a continuous upgrade and refreshment. White listing is used only for classify letters as legitimate mails, and has nothing to do when the sender is unknown. If the blacklist and white list methods are used together, further filtering is only required for letters that does not match any of the entries in the two lists [Glu06]. Further, there are two versions of white list as follows [Ola11]:

1. **Challenge-Response filtering:** is another version of Whitelist filter where senders look for possible ways that their messages can be delivered to the recipient inbox and be added to the Whitelist contact in order to appear as legitimate. Challenge-response method operate by sending a message (challenge) to sender whenever a new message is received from unknown sender and regard all incoming mails as spam except the address on the white list will be permitted to be delivered safely, this can be given by number of ISPs, proxy services and client-side filters. The challenge message usually come inform of web link asking the sender to click on the link and fill in some of the senders personal information or some codes in order for the ISP to know whether the sender is trusted or not. So for senders identifying their identity by responding to the challenge will make
their mails be delivered safely to the inbox. One of the shortcomings of this method is the high rate of false positive and the stress it gives to so many trusted email senders another disadvantage of using this tool is that it exults both the email users and senders' time, especially those that do not check their inbox always and read an email messages. Another shortcoming of using this tool is that it does not allow senders with text based-system to see the picture in their challenge message.

2. **Greylisting**: another version of Whitelist filtering is the greylisting method which similar to challenge response method. Greylisting method operate by allowing a network server to challenge another network server whenever a mail is received and the other network server re-send the mail unlike the challenge-response method where only the sender will be challenged, but with this idea it means that the mail is intentionally sent.

**2.6.2. Content-Based Filtering**

Each email message consists of two sections, the header and the body. The header section contains the vital information about the message including *origination date, sender, recipient(s), delivery path, subject, and format information* [Tra07].

A message body is what an average user typically refers to as “the email”. It might contain just a plain US-ASCII text, an HTML message with embedded images or it could be a recursively defined entity with rich tree structure. Previously, a message body was just a plain text, while since the
codification of this standard; it became possible to exchange more structured information [Kun09].

Spam content filters identify spam mails by the nature of the content of each email. Content filtering is commonly implemented by many email end users. It is popularly used to reduce UBE, which is most like to contain some predictive keywords. These predictive keywords are used to identify spam emails in content filters. The information, which could be used to detect spam emails, is contained in the mail bodies or on the mail headers (like "subject"). The techniques applied in content filters, are Bayesian Classifier, memory-based approach, maximum entropy, neural networks, genetic programming, and so on [Hao11].

This thesis adopts the neural networks to filter a spam based on its content. Hence, section 2.8 illustrates the neural network in more details and chapter three gives how this research utilizes neural network to filter a spam.

2.7 Challenges of Filtering Techniques
Since approximately 75 percent of emails now contain spam, spam filtering is a huge necessity. However, irrespective of the spam filtering techniques being implemented today, persistent problems still abounds. Challenges of spam filtering techniques include spam bots, image spamming, false positives, and blacklist evasion [Ola11].

- **Spam Bots**: is written computer software that scans for hyperlinks and email addresses and are capable of automatically sending spam messages to collect email addresses. Email addresses are usually collected from websites, and more sophisticated spam bots are
capable of recognizing differences in email address. Today, these spam bots are responsible for sending billions of spam messages on a daily basis. Estimation by Symantec shows that 90% of spam generated daily is caused by five to six million spam bots infested computers. Furthermore, “Security Intelligence Report” (SIR) the large amount of spam generated by these spam bots leading to three out of every four email being spam means every message need to go through the filter before it is allocated to the inbox. Due to the fact that these organize spammers are illegal, they do not pay taxes hence they have more capability of building more sophisticated spam bot [Ola11].

- **Image Spamming**: Image spam is a variant of email spam where the spammers actually embed the spam message in an image instead of directly placing it as mail content to evade spam filters. Spam filters look for certain key words like Viagra, cash, money which are commonly related to spam emails. However when message is inside an image the spam filters cannot effectively filter these messages. As stronger filter developed to track these messages spammers came up with newer techniques like image spam, using "Portable Document Format" (PDF) documents to send spam etc. With the use of “Optical Character Reader”’ (OCR) filters it was possible to extract the contents of the images and then check if the image had spam content. However, this process is expensive and spammers came up with new ways to evade the OCR filter. Some of the ways include: By rotating
images or making them look wavy, adding noise to the images and slice the image and rotate each component [Wak11].

- **False Positives**: The effect of false positives caused by email spam filters is indispensable. However, after several measures and financial resources being exhausted to stop false positive, a 100 percent success seems impossible. Spam fighters are only left with an option of trying to mitigate false positives to the barest minimum. Although anti-spam vendors try to conceal the false positive errors made by their products, spam experts are already aware of these errors, which mark genuine mails as spam. Genuine and important messages which are filtered into spam folder causes confusion and can alter business transactions, missed deadlines and frustration. The greatest culprits of false positive errors are banking and mortgage sector because, most spam mails involve financial or mortgage issues. Granted, most anti-spam vendors guarantee 99 percent success rate with their email filter, organizations with finances relying on email services cannot endure a 1 percent failure rate [Ola11].

**2.8 Artificial Neural Networks**

There are different ways to defining what the ANN are, from short and generic definitions to the once that try to explain in detailed way what means a neural network. Teuvo kohonen proposed this definition [Pra13].

*Artificial neural networks are massively interconnected networks in parallel of simple elements (usually adaptable), with hierarchic*
organization, which try to interact with the objects of the real world in the same way that biological nervous does.

As a simple element describe the artificial equivalent of a neuron that is known as computational neuron or node. These are organized hierarchically by layers and are interconnected between them just as in biological nervous system. Upon the presence of an external stimulus the artificial neural network generate an answer, which is confronted with the reality to determine the degree of adjustment is known as learning network or training, after which the network is ready to answer to the external stimulus in an optimum way [Pra13]. The external stimulus or sometimes called network inputs, patterns or samples might need to be preprocessed, simplified explanation of the preprocessing in section 2.13.

There are many advantages of ANN that get it the importance and make it one of the best ways of problems' solutions. These advantages or benefits are listed as follow:

- **Its structure massively distributed in parallel:** The information processing takes place through the iteration of great amount of computational neurons, each one of theme send exciting or inhibiting signals to other nodes in the network. Differing from other classic artificial intelligence methods where information processing can be considered sequential – this is step by step even when there is not a predetermined order - , in the artificial neural network this process is essentially in parallel, which is the origin of its flexibility. Because the calculations are divided in many nodes, if any of them gets astray
from the expected behavior it does not affect the behavior of the network.

- **Its ability to learn and generalize:** The ANNs have the capability to acquire knowledge from its surroundings by the adaptation of its internal parameters, which is produced as a response to the presence of an external stimulus. The network learns from examples which are presented to it, and generalize knowledge from them. The generalization can be interpreted as the property of artificial neural network to produce an adequate response to unknown stimulus which is related to the acquired knowledge.

These two characteristics for information processing make an ANN able to give solution to complex problems normally difficult to manage by the traditional ways of approximation. Additionally, using theme gives the following benefits [Pra13].

- **No linearity:** the answer from the computational neuron can be linear or not. A neural network formed by the interconnection of non-linear neurons, is in itself non-linear, a trail which is distributed to the entire network. No linearity is important over all in the cases where the task to develop present a behavior removed from linearity, which is presented in most of real situations.

- **Adaptive learning:** the ANN is capable of determine the relationship between the different examples which are presented to it, or to identify the kind to belong, without requiring a previous model.
• **Self – organization:** this property allows the ANN to distribute the knowledge in the entire network structure, there is no element with specific information.

• **Fault tolerance:** this characteristic is shown in two senses: *the first* is related to the samples shown to the network, in which case it answers correctly even when the examples exhibit variability or noise; *the second*, appears when in any of the elements of the network occurs a failure, which does not impossibilities its functioning due to the way in which it stores information.

### 2.9 Types of Artificial Neural Networks

The way that individual artificial neurons are interconnected is called topology, architecture or graph of an artificial neural network. The fact that interconnection can be done in numerous ways results in numerous possible topologies that are divided into two basic classes. Figure (2.1) shows these two topologies; the left side of the figure represents simple feed forward topology (acyclic graph) where information flows from inputs to outputs in only one direction and the right side of the figure represents simple recurrent topology (semi cyclic graph) where some of the information flows not only in one direction from input to output but also in opposite direction. For easier handling and mathematical describing of an artificial neural network the individual neurons is grouped in layers [Kre11]. The following subsections clarify these two basic classes in some details.
2.9.1 Feed-Forward Artificial Neural Networks

Artificial neural network with feed-forward topology is called *Feed-Forward* artificial neural network and as such has only one condition: information must flow from input to output in only one direction with no back-loops [Kre11].

The advantages of using "*Feed-Forward Artificial Neural Network*" (FFNN) are as follows [Moh12]:

1. Generalizing system prediction at any input or extrapolating off-grid training space. After the network is trained, it will be able to predict any new input, even those out of the training limits.

2. Working well for many applications, especially curve fitting of the time series data (i.e., data that come in different times and values).

FFNN, however, has some limitations that constrain using it for some applications. These limitations include the following [Moh12]:

---

**Figure (2.2) Feed-Forward (FNN) and Recurrent (RNN) Topology of an Artificial Neural Network.**
1. It could be highly inaccurate because of local minima solution that comes from optimization. Usually, FFNN has more neurons in its hidden layer than other types of ANN. So, a local optimization solution is more likely to occur in FFNN.

2. It experiences training time and memory issues during the training process because it has more neurons to be optimized. Therefore, these limitations exclude FFNN as an option in some applications when the number of the training cases and/or inputs and outputs are large. It is also excluded when high accuracy is required for system performance.

2.9.2 Recurrent Artificial Neural Networks

Artificial neural network with the recurrent topology is called recurrent artificial neural network also it can be called [Ioa99], "Radial Basis Neural Network" (RBNN) [Moh12]. It is similar to feed-forward neural network with no limitations regarding back loops. In these cases information is no longer transmitted only in one direction but it is also transmitted backwards. This creates an internal state of the network which allows it to exhibit dynamic temporal behavior. Recurrent artificial neural networks can use their internal memory to process any sequence of inputs [Ola99]. The advantages of using RBNN [Moh12]:

1. It provides highly accurate results within the limits of the training space (i.e., inside the domain of the training values).
2. There are no local minima problems. The network does not optimize to local minimum solutions because the number of hidden neurons is
optimized automatically in the training process. Thus, the optimal solution is obtained in terms of the number of neurons and the network weight matrix $W$.

3. There are no computational time and computer memory problems, especially when there are a large number of input/output training sets, because the network does not have a large number of neurons and weights. The weight values to be optimized exist only on the output side of the hidden layer, while FFNN has weights in both sides.

4. It was found by experience that RBNN is the best type of ANN for high dimensional regression models.

Although RBNN has powerful prediction capabilities, it has some expected limitations, as follows [Moh12]:

1. The network parameter (Gaussian width) is determined heuristically, which could produce poor results.

2. It cannot predict points that are out of training grid space. The network cannot provide accurate outputs when the input is outside the range of training data (i.e., no extrapolation).

### 2.10 Learning the Neural Networks

There are three major learning paradigms; supervised learning, unsupervised learning and reinforcement learning. Usually they can be employed by any given type of artificial neural network architecture. Each learning paradigm has many training algorithms.

1. **Supervised Learning:** Supervised training involves a mechanism of providing the network with the desired output either by manually
"grading" the network's performance or by providing the desired outputs with the inputs [And92].

2. **Unsupervised Learning**: In unsupervised learning, the network is provided with inputs but not with desired outputs. The system itself must then decide what features it will use to group the input data. This is often referred to as self-organization or adaption [Dav92].

3. **Reinforcement Learning**: In this learning process, only critical information is available, not the exact information. The learning based on this critic information is called reinforcement learning and the feedback sent is called reinforcement signal [Pra13].

### 2.11 Back Propagation

The "Back-Propagation Neural Network" (BPNN) is the most novel and oldest supervised learning ANN algorithm proposed in 1986 by Rumelhart, Hinton and Williams. BPNN learns by calculating the errors of the output layer to find the error in the hidden layers. Due to this ability of back-propagation it is highly suitable for problem in which no relationship is found between the output and the inputs. The gradient descent method is utilized to calculate the weights and adjustments are made to the network to minimize the output error [Reh11].

Back propagation is a general purpose learning algorithm. It is powerful but also expensive in terms of computational requirements for training [San06]. The basic back propagation algorithm consists of three steps [San06].
The input pattern is presented to the input layer of the network. These inputs are propagated through the network until they reach the output units. This forward pass produces the actual or predicted output pattern as formulated in equations (2.1) and (2.2) respectively (see figure (2.3)).

\[ net = \sum_{i=1}^{\text{no. of attribute}} a_i w_i \]  

(2.1)

Where \( a_i \): is \( i^{\text{th}} \) attribute of input sample \( A \)

And

\( w_i \): is the weight of the input

\[ f(net) = \frac{1}{1+\exp(-net)} \]  

(2.2)

\[ f(net) \]

\( a_1 \)

\( w_1 \)

\( a_2 \)

\( w_2 \)

\( a_i \)

\( w_i \)

\( y_i \)

Because back propagation is a supervised learning algorithm, the desired outputs are given as part of the training vector. The actual network outputs are subtracted from the desired outputs and an error signal is produced as formulated in equation (2.3)
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

\[ E = (b_i - y_i) \]  
\[ (2.3) \]

Where \( b_i \) is the desired output, and \( y_i \) is the actual output of the sample \( i \)

- This error signal is then the basis for the back propagation step, whereby the errors are passed back through the neural network by computing the contribution of each hidden processing unit and deriving the corresponding adjustment needed to produce the correct output. The connection weights are then adjusted and the neural network has just “learned” from an experience. The error signal terms of the output layer is as shown in equation (2.4)

\[ \delta_y = (b_i - y_i)y_i(1 - y_i) \]  
\[ (2.4) \]

The error signal terms of the hidden layer is formulated in equation (2.5).

\[ \delta h_k = h_k(1 - h_k) \sum \delta y_i v_k \]  
\[ (2.5) \]
\[ \forall k \in \{1, \ldots, \text{no. of hidden nodes}\} \]

Output layer weights and hidden layer weights are adjusted as shown in equations (2.6) and (2.7) respectively:

\[ v_k \leftarrow v_k + \eta \delta y h_k \]  
\[ (2.6) \]

\[ w_{jk} \leftarrow w_{jk} + \eta \delta h_k a_i \]  
\[ (2.7) \]
\[ \forall k \in \{1, \ldots, \text{no. of hidden nodes}\} \]
\[ \forall j \in \{1, \ldots, \text{no. of input nodes}\} \]

Where \( \eta \): is the learning rate.
There are several parameters that could be added to the back propagation to improve its performance. These are:

- Steepness parameter or \((\lambda)\) Where \(\lambda > 0\) is proportional to the neuron gain determining the steepness of the continuous function \(f(\text{net})\) near \(\text{net} = 0\), and \(\lambda \rightarrow \infty[\text{Zur92}]\), it use with the activation function as in equation (2.8):

\[
f(\text{net}) = \frac{1}{1 + \exp^{\lambda(-\text{net})}} \quad (2.8)
\]

- Momentum\((\alpha)\) which is generally used to accelerate the convergence and to avoid local minima [Dia11]. It uses in weight adaptation equation as in equations (2.9) and (2.10).

\[
v_{kj} \leftarrow v_{kj} + \eta \delta_{yk} h_j + \alpha \Delta v_{kj} \quad (2.9)
\]

\[
w_{ji} \leftarrow w_{ji} + \eta \delta_{hj} a_i + \alpha \Delta w_{ji} \quad (2.10)
\]

- The bias \((\text{bias})\) works as a fine adjustment by which the product of weight and output from the preceding layer is added. This means that information is stored and distributed within a neural network and even minor destruction of some of the weights and biases will have large effect on the recall of learned information [Che97]. It use as in equation (2.11):

\[
\text{net} = \sum_{i=1}^{\text{no.of attribute}} a_i w_i + \text{bias}_i \quad (2.11)
\]
2.12 Optical Back Propagation

The “Optical Back-Propagation” (OBP) algorithm is designed to overcome some of the problems associated with standard BP training using nonlinear function. One of the important properties of this algorithm is that it can escape from local minima with high speed of convergence during the training period. The convergence speed of the learning process can be improved by adjusting the error, which will be transmitted backward from the output layer to each unit in the intermediate layer [Pri11]. In BP, the error at a single output unit is defined in equation (2.3) while the error at a single output unit in adjusted OBP will be as in equations (2.12).

\[
\delta_{y_{new}} = \begin{cases} 
(1 + e^{(b_i - y_i)}) & \text{if } (b_i - y_i) \geq 0 \\
-(1 + e^{(b_i - y_i)}) & \text{if } (b_i - y_i) < 0 
\end{cases} \tag{2.12}
\]

2.13 Preprocessing

Preprocessing is type of processing performed on raw data to prepare if for another processing procedure. It uses to transform the data into a format that will be more easily and effectively processed. Preprocessing includes sampling, dimensionality reduction, feature extraction, feature selection, transformation, normalization, and so forth [Tao11].

Feature extraction is used in this thesis as preprocessing for the input samples of the BP and OBP. Principal component analysis (PCA) Principal component analysis is a statistical tool used to analyze data sets. The central idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of large number of interrelated variables, while retaining as much as possible of the variation present in the
data set. The mathematics behind principle component analysis is statistics and is hinged behind standard deviation, eigenvalues and eigenvectors [Kha11]. The steps of PCA are [Smi02, Lak10]:

- **Get some data:** Suppose \( x_1, x_2, \ldots, x_M \) are \( N \times 1 \) vectors
- **Calculate the mean for each vector.**
  \[
  \bar{x} = \frac{1}{M} \sum_{i=1}^{M} x_i
  \]
- **Subtract the mean**
  Each vector have to be subtracted from its mean. This produces a data set whose mean is zero.
  \[
  \phi_i = x_i - \bar{x}
  \]
- **Calculate the covariance matrix.**
  From the matrix \( A = [\phi_1, \phi_2, \ldots, \phi_M] \) find the covariance matrix
  \[
  C = \frac{1}{m} \sum_{n=1}^{M} \phi_n \phi_n = AA^T
  \]
  Where:
  \( C \) is \( N \times N \) matrix.
- **Calculate the eigenvectors and eigenvalues of the covariance matrix.**
  Compute the eigenvectors and eigenvalues for covariance matrix (C).
- **Choosing components and forming a feature vector.**
  Here is where the notion of data compression and reduced dimensionality comes into it.
Chapter Three

Neural Network for Spam Filtering

3.1 Introduction

The basic principal used in any spam filtering technique, whether heuristic or keyword-based, is identical: spam messages generally look different from good messages and detecting these differences is a good way to identify and stop spam. The BP and OBP approaches are used in this thesis to accomplish this task.

This chapter presents the workflow process of the proposed NNs for spam filtering, describes the basic architecture in block diagram, and then gives details of each part.

3.2 The Proposed Approach for Spam Filtering

The proposed spam filtering approach classifies emails into two categories: spams and legitimate emails with the help of neural networks. The approach consists of three phases: preprocessing phase, training phase, and testing phase. Figure (3.1) depicts the general layout of the proposed approach.
Figure (3.1): Basic Components of the Spam Filtering Approach Based on Neural Networks

The preprocessing phase is the first phase which carries out operations on the spam-based dataset to make it suitable for the NN to be trained in the next phases. These operations are feature extraction and normalization.

The training phase is the second and one of the important phases in this approach because it has the decision of making the NN able or not able to distinguish between spams and emails. Two algorithms namely BP and OBP are used in this thesis to train the neural network.
The final phase is the *testing phase* in which an examination on the trained NN is done to see if it is able to distinguish between those coming spams and emails.

### 3.3 Spam-based Dataset

The information source that feeds the proposed spam filtering approach is the spam-based dataset which is one of the publicly available datasets in the (UCI) machine learning repository [Spa99].

Spam-based dataset consists of 4601 samples (1813 spam samples and 2788 email samples). Each sample is represented by 58 attributes, 57 attributes are continuous and 1 nominal class label. Most of the attributes indicate whether a particular word or character is frequently occurring in the email [Spa99].

The first 48 continuous real attributes include percentage of words in the email that match the following words (*make, address, all, 3d, our, over, remove, internet, order, mail, receive, will, people, report, addresses, free, business, email, you, credit, your, font, 000, money, hp, hpl, George, 650, lab, labs, telnet, 857, data, 415, 85, technology, 1999, parts, pm, direct, cs, meeting, original, project, re, edu, table, conference*) [Spa99].

The next 6 continuous real attributes include percentage of characters in the email that match the following characters (\(;,(,!,\$,\#\)).

The next three attributes (55-57) measure the length of sequences of consecutive capital letters as follow: 1 continuous real attribute includes average length of uninterrupted sequences of capital letters, 1 continuous integer attribute includes length of longest uninterrupted sequence of capital
letters, and the last continuous integer attribute includes summation lengths of uninterrupted sequence of capital letters.

Finally, the last 1 nominal class attribute denotes whether the email was considered spam (1) or legitimate email (0).

### 3.4 Spam-based Dataset Preprocessing

The purpose of preprocessing is to transform the email messages into a uniform format that can be understood by NN. The preprocessing includes two parts: the first optional part is *features extraction process* and the second default part is *normalization process*. The following subsections illustrate each part in details.

#### 3.4.1 Features Extraction

Extract specific features from the spam-based dataset is another critical operation because the dataset contains a lot of redundancy, and extracting specific features from a lot of 57 features eliminates this redundancy and only the distinguished features are kept. PCA is used in this thesis to extract the distinguished features from the spam-based dataset. Algorithm (3.1) outlines how PCA is applied to extract distinguished features.
### Algorithm (3.1): Features-Extraction by PCA

**Input:** Number of Samples (N), Spam-based Dataset Samples $a_{ij} : 1 \leq i \leq N, 1 \leq j \leq 57$

**Output:** Proportion Vector of the attributes $P$

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1.   | For $j \leftarrow 1$ to $57$ /*Calculate the mean value for each attribute*/ $\text{mean}_j \leftarrow \frac{\sum_{i=1}^{N} a_{ij}}{N}$  
End for |
| 2.   | For $i \leftarrow 1$ to $N$ /* Subtract each attribute from its mean to normalize the data*/  
For $j \leftarrow 1$ to $57$  
$new a_{ij} \leftarrow a_{ij} - \text{mean}_j$  
End for  
End for |
| 3.   | For $j \leftarrow 1$ to $57$ /* Calculate the covariance matrix */  
For $k \leftarrow 1$ to $57$  
$\text{cov}_{j,k} \leftarrow \frac{1}{N} \sum_{i=1}^{N} (a_{ij} - \text{mean}_j)(a_{ik} - \text{mean}_k)$  
End for |
| 4.   | /*Calculate the eigenvalues $V$ and eigenvector $D$ from the calculated covariance.*/  
$V, D \leftarrow \text{eig}(\text{cov})$ |
| 5.   | /* Determine the proportion of each attribute from the diagonal of the eigenvalues matrix. The resulted values $P = \{p_i, 1 \leq i \leq 57\}$ where $P$ }
3.4.2 Normalization

The values of spam-based dataset's attributes have different ranges. Therefore the normalization process is applied on the values of these attributes to set them in a uniform range. Normalization can be done using formula (3.1).

\[ f_{x_{\text{new}}} = \frac{f_{x_{\text{old}}}-\text{min}_{\text{old}}}{\text{max}_{\text{old}}-\text{min}_{\text{old}}} \]  

(3.1)

Where:
- \( f_{x_{\text{old}}} \) : old value of attribute.
- \( \text{min}_{\text{old}} \) : minimum value that attribute \( f_{x_{\text{old}}} \) can get.
- \( \text{max}_{\text{old}} \) : maximum value that the attribute \( f_{x_{\text{old}}} \) can get.

To normalize the attributes of spam-based dataset, the dataset first divided into two parts: spam dataset which contains spams and email dataset which contains legitimate emails. Then equation (3.1) is applied on spam samples and email samples independently.

The following example clarifies how the normalization process is applied on a subset of samples of 4 mails and 4 spams from the spam-based dataset. The last value of each sample in bold indicates the label of spam or email.

**Sample1:** 0,0,0,0,0,0,3.19,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1.06,0,0,0,0,0.207,0.207,0.207,0
The minimum and maximum attribute's value of emails and spams samples must first be found. So there are 57 minima and 57 maxima values.

Let $min_{spam_j}$ and $max_{spam_j}$ be the minimum and maximum values of the $j^{th}$ attribute of spam, and $min_{email_j}$ and $max_{email_j}$ be the minimum and maximum values of the $j^{th}$ attribute of email.
For example, to apply the normalization process on the attribute number 57 (the underlined one), the new value of attribute number 57 for each sample is as follow:

\[
\begin{align*}
\text{Sample1} : & \quad f_{57_{new}}^1 = \frac{79-63}{79-63}, f_{57_{new}} = 1 \\
\text{Sample2} : & \quad f_{57_{new}}^2 = \frac{63-63}{79-63}, f_{57_{new}} = 0 \\
\text{Sample3} : & \quad f_{57_{new}}^3 = \frac{54-63}{79-63}, f_{57_{new}} = -0.5625 \\
\text{Sample4} : & \quad f_{57_{new}}^4 = \frac{78-63}{79-63}, f_{57_{new}} = 0.9375 \\
\text{Sample5} : & \quad f_{57_{new}}^5 = \frac{150-1}{150-1}, f_{57_{new}} = 1 \\
\text{Sample6} : & \quad f_{57_{new}}^6 = \frac{1-1}{150-1}, f_{57_{new}} = 0 \\
\text{Sample7} : & \quad f_{57_{new}}^7 = \frac{28-1}{150-1}, f_{57_{new}} = 0.1812 \\
\text{Sample8} : & \quad f_{57_{new}}^8 = \frac{16-1}{150-1}, f_{57_{new}} = 0.1006
\end{align*}
\]

### 3.5 Training Phase

Once preprocessing phase is completed, the spam-based dataset is ready to be trained with NN. The aim of this phase is to train NN on amount of samples of spam and email messages to be able to distinguish between them.

To train the NN, there is a need to give it samples of what the user wants (A) and the output label (B) for a particular input. For each sample there is one output label equal to (0 or 1). The samples are selected randomly from spam-based dataset. The weights of all input nodes are initializes and given to the network.
Two types of training algorithms are used BP algorithm and OBP algorithm. The following subsections explain how these algorithms are utilized for spam filtering.

3.5.1 BP Training

BP is a form of supervised learning for multi-layer nets. It is most often used as training algorithm. Four different neural network structures for BP are used in this thesis depends on whether the feature extraction process is applied or not.

The first structure coined as **BP-1**, consists of input layer with 57 nodes according to number of features in spam-based dataset. The hidden layer in this structure consists of multiple layers (experimentally four hidden layers are used) and there are 40 nodes at the first hidden layer, 30 nodes at the second hidden layer, 20 nodes at the third hidden layer, and 10 nodes at the fourth hidden layer. There is only one node at the output layer because there are two classes (email or spam). Figure (3.2) demonstrates BP-1 structure and algorithm (3.2) clarifies how the BP-1 operates.

The second structure named **BP-2** consists of input layer with 42 nodes according to selecting 75% of features of spam-based dataset. There is one hidden layer with 20 nodes. The output layer consists of one node. Figure (3.3) depicts BP-2 structure.

The third structure **BP-3**, there is 28 nodes in the input layer according to selecting 50% of features of spam-based dataset. There are only one hidden layer with 24 nodes and one node at output layer. Figure (3.4) shows BP-3 structure. The last structure **BP-4** includes 14 nodes in the input layer
according to selecting 25% of features of spam-based dataset, one hidden layer with 9 nodes, and the output layer has one node. Figure (3.5) shows BP-4 structure.
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

Algorithm 3.2: BP-Spam Filtering

| Input: Spam-based Dataset Samples: each sample is a pair of input $A$ and output label $B$  
Learning rate($\eta$), steepness parameter ($\lambda$), momentum ($\alpha$), stopping criteria ($\epsilon$) and initializes all weights $W_1, W_2, W_3, W_4, W_5$ with small random numbers.  
Output: Final Weights ($W_1, W_2, W_3, W_4, W_5$)  

2: Repeat  
3: /* For every sample in the spam-based dataset propagated the sample input forward the network or every node in the layer calculate the weight sum of the inputs to the node, add the bias value to the sum and calculate the activation function for the node */  
For $i \leftarrow 1$ to no. of samples  
4: For $k_1 \leftarrow 1$ to no. of hidden1 nodes  
\hspace{0.5cm} $h_{1i_{k1}} \leftarrow \sum_{j=1}^{\text{no. of attributes}} a_{ij} w_{1jk1}$,  
\hspace{0.5cm} $h_{1i_{k1}} \leftarrow h_{1i_{k1}} + \text{bias}1_{i_{k1}}$  
\hspace{0.5cm} $f(h_{1i_{k1}}) \leftarrow \frac{1}{1+exp(-h_{1i_{k1}})}$,  
End for  
5: For $k_2 \leftarrow 1$ to no. of hidden2 nodes  
\hspace{0.5cm} $h_{2i_{k2}} \leftarrow \sum_{k_1=1}^{\text{no. of hidden1 node}} h_{1i_{k1}} w_{2k1k2}$,  
\hspace{0.5cm} $h_{2i_{k2}} \leftarrow h_{2i_{k2}} + \text{bias}2_{i_{k2}}$  
\hspace{0.5cm} $f(h_{2i_{k2}}) \leftarrow \frac{1}{1+exp(-h_{2i_{k2}})}$,  
End for  
6: For $k_3 \leftarrow 1$ to no. of hidden3 nodes  
\hspace{0.5cm} $h_{3i_{k3}} \leftarrow \sum_{k_2=1}^{\text{no. of hidden2 node}} h_{2i_{k2}} w_{3k2k3}$,  
\hspace{0.5cm} $h_{3i_{k3}} \leftarrow h_{3i_{k3}} + \text{bias}3_{i_{k3}}$  
\hspace{0.5cm} $f(h_{3i_{k3}}) \leftarrow \frac{1}{1+exp(-h_{3i_{k3}})}$,  
End for  
7: For $k_4 \leftarrow 1$ to no. of hidden4 nodes  
\hspace{0.5cm} $h_{4i_{k4}} \leftarrow \sum_{k_3=1}^{\text{no. of hidden3 node}} h_{3i_{k3}} w_{4k3k4}$,  
\hspace{0.5cm} $h_{4i_{k4}} \leftarrow h_{4i_{k4}} + \text{bias}4_{i_{k4}}$  
\hspace{0.5cm} $f(h_{4i_{k4}}) \leftarrow \frac{1}{1+exp(-h_{4i_{k4}})}$,  
End for  
8: $y_i \leftarrow \sum_{k_4=1}^{\text{no. of hidden4 nodes}} h_{4i_{k4}} w_{5k4}$,  
\hspace{0.5cm} $y_i \leftarrow y_i + \text{bias}5_i$  
\hspace{0.5cm} $f(y_i) \leftarrow \frac{1}{1+exp^{\lambda(-y_i)}}$  

59
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

9: \[ E \leftarrow E + \frac{1}{2} (b_i - y_i)^2 /\text{Calculate mean square error} */ \]
End For /*this is the end of the samples loop*/

10: /*For every sample in the spam-based dataset propagate the errors backward through the network, calculate the error signal for each node in the output layer and hidden layers, and then adjust all node's weights */
For \( i \leftarrow 1 \) to no. of samples

11: /*Calculate the error in the output node*/
\[
\delta_5 \leftarrow (b_i - f(y_i))f(y_i)(1 - f(y_i))
\]
For \( k4 \leftarrow 1 \) to no. of hidden4 nodes
\[
w5_{k4} \leftarrow w5_{k4} + \eta \delta_5 h4_{ik4} + \alpha \Delta w5_{k4}
\]
\[
bias5_i \leftarrow bias5_i + \eta \delta_5 h4_{ik4} + \alpha \Delta bias5_i
\]
End for

12: For \( k4 \leftarrow 1 \) to no. of hidden4 nodes
\[
\delta_4 \leftarrow h4_{ik4} (1 - h4_{ik4}) + \delta_5 w5_{k4}
\]
End For
For \( k3 \leftarrow 1 \) to no. of hidden3 nodes
For \( k4 \leftarrow 1 \) to no. of hidden4 nodes
\[
w4_{k3k4} \leftarrow w4_{k3k4} + \eta \delta_4 h3_{ik3} + \alpha \Delta w4_{k3k4}
\]
\[
bias4_{ik4} \leftarrow bias4_{ik4} + \eta \delta_4 h3_{ik3} + \alpha \Delta bias4_{ik4}
\]
End for
End for

13: For \( k3 \leftarrow 1 \) to no. of hidden3 nodes
\[
\delta_3 \leftarrow h3_{ik3} (1 - h3_{ik3}) \sum_{k4=1}^{\text{no. of hidden4 nodes}} \delta_4 w4_{k3k4}
\]
End For
For \( k2 \leftarrow 1 \) to no. of hidden2 nodes
For \( k3 \leftarrow 1 \) to no. of hidden3 nodes
\[
w3_{k2k3} \leftarrow w3_{k2k3} + \eta \delta_3 h2_{ik2} + \alpha \Delta w3_{k2k3}
\]
\[
bias3_{ik3} \leftarrow bias3_{ik3} + \eta \delta_3 h2_{ik2} + \alpha \Delta bias3_{ik3}
\]
End for
End for

14: For \( k2 \leftarrow 1 \) to no. of hidden2 nodes
\[
\delta_2 \leftarrow h2_{ik2} (1 - h2_{ik2}) \sum_{k3=1}^{\text{no. of hidden3 nodes}} \delta_3 w3_{k2k3}
\]
End for
For \( k1 \leftarrow 1 \) to no. of hidden1 nodes
For \( k2 \leftarrow 1 \) to no. of hidden2 nodes
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

Algorithm (3.2)
can be applied for BP-2, BP-3 and BP-4 but with slight modifications. For BP-2, BP-3 and BP-4 there is one hidden layer, thus there are two weights($W_1, W_5$), there is no need for steps 5, 6, 7, 12, 13, and 14 and the steps 8, 11, and 15 will be modified as follows.

Step 8:

\[ y_i \leftarrow \sum_{k1=1}^{\text{no.of hidden1 nodes}} h1_{ik1}w5_{k1} \]

Step 11:

For $k1 \leftarrow 1 \text{ to no. of hidden1 nodes}$

\[ w5_{k1} \leftarrow w5_{k1} + \eta \delta_5 h1_{ik1} + \alpha \Delta w5_{k1} \]

\[ bias5_i \leftarrow bias5_i + \eta \delta_5 h1_{ik1} + \alpha \Delta bias5_i \]

Step 15:

\[ \delta_1 \leftarrow h1_{ik1} (1 - h1_{ik1}) \sum \delta_5 w5_{k1} \]

3.5.2 OBP Training
OPB is a form of back propagation algorithm without the problem of local minima and the slow rate of convergence. This method has been applied to the multilayer NN to improve the efficiency in terms of convergence speed. OBP resolves the shortcoming by slightly modifying the error signal function of BP algorithm, thereby greatly accelerating the convergence rate of the training process. The convergence speed of the training process can be improved significantly by OBP through maximizing the error signal, which will be transmitted backward from the output layer to each output unit in the intermediate layer [Pat10].

Four structures for OBP are used in this thesis depend on whether a feature extraction process is applied or not. The four structures coined as OBP-1, OBP-2, OBP-3 and OBP-4 respectively. The number of layers and the number of nodes in each layer for four OBP structures are similar to BP structures. Figure (3.2), (3.3), (3.4), and (3.5) depict the four structures of OBP.

The algorithm (3.2) is applied on these four structures with some modifications. This modification occurs in step 11 in the equation of error signal calculation in the output layer is changed as in equation (2.12).
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

Figure (3.2): Input Layer, Hidden Layers and Output Layer of BP-1 and OBP-1

Figure (3.3): Input Layer, Hidden Layer and Output Layer of BP-2 and OBP-2
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

Figure (3.4) Input Layer, Hidden Layer and Output Layer of BP-3 and OBP-3

Figure (3.5) Input Layer, Hidden Layer and Output Layer of BP-4 and OBP-4
3.6 Testing Phase

When the training phase is done, the network becomes ready to determine if the coming message is a legitimate email or a spam. The testing phase uses BP or OBP to identify the type of a message as a spam or an email.

In this thesis the testing samples drawn randomly from spam-based dataset. The test samples should be preprocessed as discussed in section 3.4. When preprocessing is completed, the test sample propagates forward the network on every node in the layer. Then the weight sum of the inputs to the node is calculated, and the bias value is added to the sum, finally the activation function for the node (i.e. the desired output) is calculated. The network decides if the test sample is a legitimate email or a spam depends on the desired output (result).
Chapter Four
Experiments and Results Evaluation

4.1 Introduction
This chapter evaluates the performance of the proposed NN-based spam filtering approach for solving spam filtering problem. First, it determines the criteria that are used to measure the propose approach. Then, it presents the training and testing dataset used in the conducted experiment. Finally, it presents the experiments that are conducted to show the results of the proposed spam filtering approach.

4.2 Evaluation Measures
The evaluation measures used in this thesis to evaluate the performance of the proposed neural network spam filtering approach are: accuracy (AC), precision (P), recall (R), false positive (FP), false negative (FN), and the number of epochs verse error signal.

The accuracy is the proportion of the total number of predictions that were correct. It is determined using the equation (4.1) [Con08]:

$$AC = \frac{(S \rightarrow S) + (E \rightarrow E)}{(S \rightarrow S) + (E \rightarrow E) + (S \rightarrow E) + (E \rightarrow S)}$$

(4.1)

Where:
$S$: spam
$E$: email
$S \rightarrow S$: total number of spams that are classified as spam
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

\[ E \rightarrow E: \text{total number of emails that are classified as email} \]
\[ S \rightarrow E: \text{total number of spams that are classified as email} \]
\[ E \rightarrow S: \text{total number of emails that are classified as spam} \]

The *Precision* is the proportion of the predicted positive cases that were correct, as calculated using equation (4.2) [Con08]:

\[ P = \frac{S \rightarrow S}{(S \rightarrow S) + (E \rightarrow S)} \quad (4.2) \]

The *recall* is the proportion of positive cases that were correctly identified, as calculated using the equation (4.3) [Con08]:

\[ R = \frac{S \rightarrow S}{(S \rightarrow S) + (S \rightarrow E)} \quad (4.3) \]

The *false positive* is the proportion of negatives cases that were incorrectly classified as positive, as calculated using the equation (4.4) [Con08]:

\[ FP = \frac{E \rightarrow S}{(E \rightarrow S) + (E \rightarrow E)} \quad (4.4) \]

The *false negative* is the proportion of positives cases that were incorrectly classified as negative, as calculated using the equation (4.5) [Con08]:

\[ FN = \frac{S \rightarrow E}{(S \rightarrow E) + (S \rightarrow S)} \quad (4.5) \]

### 4.3 Training and Testing Datasets

The spam-based dataset is adopted to conduct the experiments. The dataset is divided into two parts: training dataset and testing dataset.

The training dataset is used in the training phase and it is divided into nine groups and each group contains samples selected randomly from
spam-based dataset. The number of samples for each group is tabulated in table (4.1).

The testing dataset is used to evaluate the performance of the proposed spam filtering approach. The testing dataset is divided into three groups and each group contains samples selected randomly from spam-based dataset. The first group (testing DS1) contains 250 samples, the second group (testing DS2) has 500 samples and third one (testing DS3) contains 750 samples. Table 4.2 shows details of testing dataset groups.

Table (4.1) Training Dataset Groups Details

<table>
<thead>
<tr>
<th>Training Datasets</th>
<th>Number of Sample</th>
<th>Email/Spam Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training DS1</td>
<td>500</td>
<td>250/250</td>
</tr>
<tr>
<td>Training DS2</td>
<td>750</td>
<td>375/375</td>
</tr>
<tr>
<td>Training DS3</td>
<td>1000</td>
<td>500/500</td>
</tr>
<tr>
<td>Training DS4</td>
<td>1250</td>
<td>625/625</td>
</tr>
<tr>
<td>Training DS5</td>
<td>1500</td>
<td>750/750</td>
</tr>
<tr>
<td>Training DS6</td>
<td>1750</td>
<td>875/875</td>
</tr>
<tr>
<td>Training DS7</td>
<td>2000</td>
<td>1000/1000</td>
</tr>
<tr>
<td>Training DS8</td>
<td>2250</td>
<td>1125/1125</td>
</tr>
<tr>
<td>Training DS9</td>
<td>2500</td>
<td>1250/1250</td>
</tr>
</tbody>
</table>

Table (4.2) Testing Dataset Groups Details

<table>
<thead>
<tr>
<th>Testing Datasets</th>
<th>Number of Sample</th>
<th>Email/Spam Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing DS1</td>
<td>250</td>
<td>125/125</td>
</tr>
<tr>
<td>Testing DS2</td>
<td>500</td>
<td>250/250</td>
</tr>
<tr>
<td>Testing DS3</td>
<td>750</td>
<td>375/375</td>
</tr>
</tbody>
</table>
Training and testing datasets should be preprocessed before passing to the neural network. The preprocessing includes two parts: either feature normalization or make feature extraction and normalization.

In normalization process, first, the spam-based dataset is divided into two set: spam dataset and email dataset. Then the normalization process is applied on each feature for all samples in spam dataset and then it is applied on email dataset.

The feature extraction process is carried out by PCA technique. The result obtained is proportions of each feature. Three different proportions are used namely 75%, 50% and 25%. The first means that the dataset contains the highest 75% of values features and this represents 42 features at each sample. The second represents dataset contains the highest 50% of values features (i.e. 28 features at each sample). The last one means that the dataset includes highest 25% of values features and this represents 14 features at each sample.

4.4 Experimental Results
The proposed spam filtering approach was written in Microsoft visual C++ 6.0 programming language. The experiments run under Windows 7 ultimate service pack 1 operating system, Intel(R) Core(TM) i3-2328M CPU @ 2.20GHz, 4 GB random access memory and 64-bit system type.

Several experiments in this thesis are conducted to show the effectiveness of the proposed spam filtering approach. However, several parameters should be set before the experiments are conducted. These are:
First, the number of input nodes, the number of hidden layers and the number of hidden nodes at each hidden layer for BP-1, BP-2, BP-3, BP-4 OBP-1, OBP-2, OBP-3, and OBP-4 are set as demonstrated in table (4.3).

Second, the (λ), (η) and (α) parameters of BP and OBP neural networks are set experimentally as shown in table (4.4).

Finally, the weights that connect input layer nodes, intermediate hidden layers nodes and output layer node of BP and OBP neural networks are initialized randomly to value in range [-0.5, 0.5].

Table (4.3): Quantitative Structure of BP and OBP Neural Networks

<table>
<thead>
<tr>
<th>Structures</th>
<th>No. of Input Nodes</th>
<th>No. of Hidden Layers</th>
<th>No. of Hidden Nodes at each Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP-1, OBP-1</td>
<td>57</td>
<td>4</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>BP-2, OBP-2</td>
<td>42</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>BP-3, OBP-3</td>
<td>28</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>BP-4, OBP-4</td>
<td>14</td>
<td>1</td>
<td>9</td>
</tr>
</tbody>
</table>

Table (4.4): Setting Parameters of BP and OBP Structures

<table>
<thead>
<tr>
<th>Structures</th>
<th>λ</th>
<th>η</th>
<th>α</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP-1, OBP-1</td>
<td>1</td>
<td>0.2</td>
<td>1</td>
</tr>
<tr>
<td>BP-2, OBP-2</td>
<td>0.02</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>BP-3, OBP-3</td>
<td>0.02</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>BP-4, OBP-4</td>
<td>0.06</td>
<td>0.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>
4.4.1 Impact of Size of Training Dataset vs. Size of Testing Dataset

In this experiment, the proposed spam filtering approach was conducted on training datasets that contain all features (i.e., 57 features). BP-1 and OBP-1 are trained using nine groups of training datasets and tested with three groups of testing datasets to show the impact of training dataset size verses size of testing dataset. The results of AC, P, R, FP and FN of BP-1 and OBP-1 for three testing dataset groups are shown in tables (4.5) - (4.14) respectively.

**Table (4.5) BP-1 Accuracy Results**

<table>
<thead>
<tr>
<th>Training Datasets</th>
<th>AC</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing DS1</td>
<td>Testing DS2</td>
<td>Testing DS3</td>
</tr>
<tr>
<td>Training DS1</td>
<td>100</td>
<td>0.99</td>
<td>0.832</td>
</tr>
<tr>
<td>Training DS2</td>
<td>100</td>
<td>100</td>
<td>0.993</td>
</tr>
<tr>
<td>Training DS3</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Training DS4</td>
<td>100</td>
<td>0.994</td>
<td>0.992</td>
</tr>
<tr>
<td>Training DS5</td>
<td>100</td>
<td>100</td>
<td>0.998</td>
</tr>
<tr>
<td>Training DS6</td>
<td>0.992</td>
<td>0.996</td>
<td>0.996</td>
</tr>
<tr>
<td>Training DS7</td>
<td>100</td>
<td>100</td>
<td>0.997</td>
</tr>
<tr>
<td>Training DS8</td>
<td>100</td>
<td>100</td>
<td>0.998</td>
</tr>
<tr>
<td>Training DS9</td>
<td>100</td>
<td>100</td>
<td>0.998</td>
</tr>
</tbody>
</table>
Table (4.6) BP-1 Recall Results

<table>
<thead>
<tr>
<th>Training Datasets</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing</td>
</tr>
<tr>
<td></td>
<td>DS1</td>
</tr>
<tr>
<td>Training DS1</td>
<td>100</td>
</tr>
<tr>
<td>Training DS2</td>
<td>100</td>
</tr>
<tr>
<td>Training DS3</td>
<td>100</td>
</tr>
<tr>
<td>Training DS4</td>
<td>100</td>
</tr>
<tr>
<td>Training DS5</td>
<td>100</td>
</tr>
<tr>
<td>Training DS6</td>
<td>0.984</td>
</tr>
<tr>
<td>Training DS7</td>
<td>100</td>
</tr>
<tr>
<td>Training DS8</td>
<td>100</td>
</tr>
<tr>
<td>Training DS9</td>
<td>100</td>
</tr>
</tbody>
</table>

Table (4.7) BP-1 Precision Result

<table>
<thead>
<tr>
<th>Training Datasets</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing</td>
</tr>
<tr>
<td></td>
<td>DS1</td>
</tr>
<tr>
<td>Training DS1</td>
<td>100</td>
</tr>
<tr>
<td>Training DS2</td>
<td>100</td>
</tr>
<tr>
<td>Training DS3</td>
<td>100</td>
</tr>
<tr>
<td>Training DS4</td>
<td>100</td>
</tr>
<tr>
<td>Training DS5</td>
<td>100</td>
</tr>
<tr>
<td>Training DS6</td>
<td>100</td>
</tr>
<tr>
<td>Training DS7</td>
<td>100</td>
</tr>
<tr>
<td>Training DS8</td>
<td>100</td>
</tr>
<tr>
<td>Training DS9</td>
<td>100</td>
</tr>
</tbody>
</table>
### Table (4.8) BP-1 False Positive Results

<table>
<thead>
<tr>
<th>Training Datasets</th>
<th>FP</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing DS1</td>
<td>Testing DS2</td>
</tr>
<tr>
<td>Training DS1</td>
<td>0</td>
<td>0.02</td>
</tr>
<tr>
<td>Training DS2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS9</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table (4.9) BP-1 False Negative Results

<table>
<thead>
<tr>
<th>Training Datasets</th>
<th>FN</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing DS1</td>
<td>Testing DS2</td>
</tr>
<tr>
<td>Training DS1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS4</td>
<td>0</td>
<td>0.012</td>
</tr>
<tr>
<td>Training DS5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS6</td>
<td>0.016</td>
<td>0.008</td>
</tr>
<tr>
<td>Training DS7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS9</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Table (4.10) OBP-1 Accuracy Results

<table>
<thead>
<tr>
<th>Training Datasets</th>
<th>AC</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing DS1</td>
<td>Testing DS2</td>
<td>Testing DS3</td>
</tr>
<tr>
<td>Training DS1</td>
<td>100</td>
<td>0.994</td>
<td>0.834</td>
</tr>
<tr>
<td>Training DS2</td>
<td>100</td>
<td>100</td>
<td>0.996</td>
</tr>
<tr>
<td>Training DS3</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Training DS4</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Training DS5</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Training DS6</td>
<td>0.984</td>
<td>0.98</td>
<td>0.974</td>
</tr>
<tr>
<td>Training DS7</td>
<td>0.955</td>
<td>0.962</td>
<td>0.960</td>
</tr>
<tr>
<td>Training DS8</td>
<td>0.996</td>
<td>0.998</td>
<td>0.998</td>
</tr>
<tr>
<td>Training DS9</td>
<td>0.996</td>
<td>0.994</td>
<td>0.994</td>
</tr>
</tbody>
</table>

### Table (4.11) OBP-1 Recall Results

<table>
<thead>
<tr>
<th>Training Datasets</th>
<th>R</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing DS1</td>
<td>Testing DS2</td>
<td>Testing DS3</td>
</tr>
<tr>
<td>Training DS1</td>
<td>100</td>
<td>100</td>
<td>0.728</td>
</tr>
<tr>
<td>Training DS2</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Training DS3</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Training DS4</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Training DS5</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Training DS6</td>
<td>0.968</td>
<td>0.96</td>
<td>0.949</td>
</tr>
<tr>
<td>Training DS7</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Training DS8</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Training DS9</td>
<td>0.992</td>
<td>0.988</td>
<td>0.989</td>
</tr>
</tbody>
</table>
### Table (4.12) OBP-1 Precision Results

<table>
<thead>
<tr>
<th>Training Datasets</th>
<th>P</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100</td>
<td>0.988</td>
<td>0.925</td>
</tr>
<tr>
<td>Training DS1</td>
<td>100</td>
<td>100</td>
<td>0.992</td>
</tr>
<tr>
<td>Training DS2</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Training DS3</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Training DS4</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Training DS5</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Training DS6</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Training DS7</td>
<td>0.917</td>
<td>0.929</td>
<td>0.925</td>
</tr>
<tr>
<td>Training DS8</td>
<td>0.992</td>
<td>0.996</td>
<td>0.997</td>
</tr>
<tr>
<td>Training DS9</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

### Table (4.13) OBP-1 False Positive Results

<table>
<thead>
<tr>
<th>Training Datasets</th>
<th>FP</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0.012</td>
<td>0.058</td>
</tr>
<tr>
<td>Training DS1</td>
<td>0</td>
<td>0</td>
<td>0.008</td>
</tr>
<tr>
<td>Training DS2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training DS7</td>
<td>0.09</td>
<td>0.076</td>
<td>0.08</td>
</tr>
<tr>
<td>Training DS8</td>
<td>0.008</td>
<td>0.004</td>
<td>0.002</td>
</tr>
<tr>
<td>Training DS9</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Table (4.14) OBP-1 False Negative Results

<table>
<thead>
<tr>
<th>Training Datasets</th>
<th>FN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing DS1</td>
</tr>
<tr>
<td>Training DS1</td>
<td>0</td>
</tr>
<tr>
<td>Training DS2</td>
<td>0</td>
</tr>
<tr>
<td>Training DS3</td>
<td>0</td>
</tr>
<tr>
<td>Training DS4</td>
<td>0</td>
</tr>
<tr>
<td>Training DS5</td>
<td>0</td>
</tr>
<tr>
<td>Training DS6</td>
<td>0.032</td>
</tr>
<tr>
<td>Training DS7</td>
<td>0</td>
</tr>
<tr>
<td>Training DS8</td>
<td>0</td>
</tr>
<tr>
<td>Training DS9</td>
<td>0.008</td>
</tr>
</tbody>
</table>

4.4.2 Impact of Feature Set Size

In this experiment, the spam filtering approach was conducted on training datasets with samples that include the extracted features from PCA (i.e., 42, 28, and 14 features). The results of BP-1 and OBP-1 show that the training dataset “Training DS3” gives the best results in terms of accuracy, recall, precision, false positive, and false negative. Therefore, the training dataset with 1000 samples (i.e., Training DS3) is used to train BP-2, BP-3, BP-4, OBP-2, OBP-3 and OBP4. The results of AC, P, R, FP and FN for testing dataset groups are shown in tables (4.15) - (4.24) respectively.
Table (4.15): BP-2, BP-3 and BP-4 Accuracy Results

<table>
<thead>
<tr>
<th>Structures</th>
<th>AC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing DS1</td>
</tr>
<tr>
<td>BP-1</td>
<td>100</td>
</tr>
<tr>
<td>BP-2</td>
<td>0.952</td>
</tr>
<tr>
<td>BP-3</td>
<td>0.952</td>
</tr>
<tr>
<td>BP-4</td>
<td>0.916</td>
</tr>
</tbody>
</table>

Table (4.16): BP-2, BP-3 and BP-4 Recall Results

<table>
<thead>
<tr>
<th>Structures</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing DS1</td>
</tr>
<tr>
<td>BP-1</td>
<td>100</td>
</tr>
<tr>
<td>BP-2</td>
<td>0.952</td>
</tr>
<tr>
<td>BP-3</td>
<td>0.912</td>
</tr>
<tr>
<td>BP-4</td>
<td>0.84</td>
</tr>
</tbody>
</table>

Table (4.17): BP-2, BP-3 and BP-4 Precision Results

<table>
<thead>
<tr>
<th>Structures</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing DS1</td>
</tr>
<tr>
<td>BP-1</td>
<td>100</td>
</tr>
<tr>
<td>BP-2</td>
<td>0.952</td>
</tr>
<tr>
<td>BP-3</td>
<td>0.971</td>
</tr>
<tr>
<td>BP-4</td>
<td>0.99</td>
</tr>
</tbody>
</table>
### Table (4.18): BP-2, BP-3 and BP-4 False Positive Result

<table>
<thead>
<tr>
<th>Structures</th>
<th>FP</th>
<th>Testing DS1</th>
<th>Testing DS2</th>
<th>Testing DS3</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BP-2</td>
<td>0.048</td>
<td>0.064</td>
<td>0.074</td>
<td></td>
</tr>
<tr>
<td>BP-3</td>
<td>0.028</td>
<td>0.052</td>
<td>0.047</td>
<td></td>
</tr>
<tr>
<td>BP-4</td>
<td>0.008</td>
<td>0.048</td>
<td>0.061</td>
<td></td>
</tr>
</tbody>
</table>

### Table (4.19): BP-2, BP-3 and BP-4 False Negative Results

<table>
<thead>
<tr>
<th>Structures</th>
<th>FN</th>
<th>Testing DS1</th>
<th>Testing DS2</th>
<th>Testing DS3</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BP-2</td>
<td>0.048</td>
<td>0.04</td>
<td>0.053</td>
<td></td>
</tr>
<tr>
<td>BP-3</td>
<td>0.088</td>
<td>0.088</td>
<td>0.12</td>
<td></td>
</tr>
<tr>
<td>BP-4</td>
<td>0.16</td>
<td>0.128</td>
<td>0.152</td>
<td></td>
</tr>
</tbody>
</table>

### Table (4.20): OBP-2, OBP-3 and OBP-4 Accuracy Results

<table>
<thead>
<tr>
<th>Structures</th>
<th>AC</th>
<th>Testing DS1</th>
<th>Testing DS2</th>
<th>Testing DS3</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBP-1</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>OBP-2</td>
<td>0.976</td>
<td>0.954</td>
<td>0.944</td>
<td></td>
</tr>
<tr>
<td>OBP-3</td>
<td>0.96</td>
<td>0.94</td>
<td>0.926</td>
<td></td>
</tr>
<tr>
<td>OBP-4</td>
<td>0.964</td>
<td>0.94</td>
<td>0.925</td>
<td></td>
</tr>
</tbody>
</table>
### Table (4.21): OBP-2, OBP-3 and OBP-4 Recall Result

<table>
<thead>
<tr>
<th>Structures</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing DS1</td>
</tr>
<tr>
<td>OBP-1</td>
<td>100</td>
</tr>
<tr>
<td>OBP-2</td>
<td>0.96</td>
</tr>
<tr>
<td>OBP-3</td>
<td>0.936</td>
</tr>
<tr>
<td>OBP-4</td>
<td>0.928</td>
</tr>
</tbody>
</table>

### Table (4.22): OBP-2, OBP-3 and OBP-4 Precision Results

<table>
<thead>
<tr>
<th>Structures</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing DS1</td>
</tr>
<tr>
<td>OBP-1</td>
<td>100</td>
</tr>
<tr>
<td>OBP-2</td>
<td>0.991</td>
</tr>
<tr>
<td>OBP-3</td>
<td>0.983</td>
</tr>
<tr>
<td>OBP-4</td>
<td>100</td>
</tr>
</tbody>
</table>

### Table (4.23): OBP-2, OBP-3 and OBP-4 False Positive Results

<table>
<thead>
<tr>
<th>Structures</th>
<th>FP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing DS1</td>
</tr>
<tr>
<td>OBP-1</td>
<td>0</td>
</tr>
<tr>
<td>OBP-2</td>
<td>0.008</td>
</tr>
<tr>
<td>OBP-3</td>
<td>0.016</td>
</tr>
<tr>
<td>OBP-4</td>
<td>0.044</td>
</tr>
</tbody>
</table>
Table (4.24): OBP-2, OBP-3 and OBP-4 False Negative Results

<table>
<thead>
<tr>
<th>Structures</th>
<th>FN</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing DS1</td>
<td>Testing DS2</td>
<td>Testing DS3</td>
</tr>
<tr>
<td>OBP-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OBP-2</td>
<td>0.04</td>
<td>0.044</td>
<td>0.056</td>
</tr>
<tr>
<td>OBP-3</td>
<td>0.064</td>
<td>0.072</td>
<td>0.098</td>
</tr>
<tr>
<td>OBP-4</td>
<td>0.072</td>
<td>0.076</td>
<td>0.104</td>
</tr>
</tbody>
</table>

When a comparison is made between the structures of BP (BP1, BP-2, BP-3, and BP-4), the results show that the results of BP-1 for all three testing dataset groups give the best results in terms of accuracy, recall, precision, false positive, and false negative as shown in figures (4.1)-(4.5) respectively.

Figure (4.1): Accuracy Results of BP-1, BP-2, BP-3, and BP-4
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

Figure (4.2): Recall Results of BP-1, BP-2, BP-3, and BP-4

Figure (4.3): Precision Results BP-1, BP-2, BP-3, and BP-4
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

Figure (4.4) False Positive Results of BP-1, BP-2, BP-3, and BP-4

Figure (4.5) False Negative Results of BP-1, BP-2, BP-3, and BP-4
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

when a comparison is made between the structures of OBP (i.e., OBP1, OBP-2, OBP-3, and OBP-4), the results show that the result of OBP-1 provide the best results in terms of accuracy, recall, precision, false positive, and false negative as demonstrated in figures (4.6)- (4.10) respectively.

Figure (4.6) Accuracy Results of OBP-1, OBP-2, OBP-3, and OBP-4

Figure (4.7) Recall Result of OBP-1, OBP-2, OBP-3, and OBP-4
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

Figure (4.8): Precision Results of OBP-1, OBP-2, OBP-3, and OBP-4

Figure (4.9) OBP-1, OBP-2, OBP-3, and OBP-4 False Positive Result
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

Figure (4.10) False Negative Result of OBP-1, OBP-2, OBP-3, and OBP-4

As shown from the above result, the results of first structure of BP and OBP are better than the results of rest structures, because the first structure uses all 57 features of each sample and this provides more information about the incoming message that makes the training of NN more accurate.

The results of OBP structures (OBP-2, OBP-3, and OBP-3) are better than the corresponding BPs in all evaluation measurements (i.e., accuracy, precision, recall, false positive and false negative).

4.4.3 Error Signal Evaluation

The performance of the BP and OBP structures could be evaluated depending on the error signal against number of epochs. Figure (4.11) depicts the performance in terms of error signal of four BP Structure. BP-1 takes 10 epochs to obtain the appropriate error signal (1.94886), BP-2 takes 19 epochs to get appropriate error signal (2.9632), BP-3 takes 25 epochs
with (4.06592) error signal, and BP-4 takes 39 epochs with (4.41309) error signal.

Figure (4.12) depicts the error signal against number of epochs for OBP-1, OBP-2, OBP-3, and OBP-4 structures. OBP-1 take 10 epochs to obtain the appropriate error signal (4.36613), BP-2 take 17 epochs to take appropriate error signal (4.38528), OBP-3 take 8 epochs with (5.80074) error signal, and OBP-4 take 39 epochs with (5.9372) error signal.

![Figure (4.11): Error Signal vs. Number of Epochs for BP Structures](image-url)
Figure (4.12): Error Signal vs. Number of Epochs for OBP Structures

4.4.4 Execution Time

The performance of the BP and OBP structures could be evaluated depending on execution time of training phase. The execution time of BP and OBP are calculated in seconds as shown in table (4.26).

Table (4.26) Execution Time of BP and OBP Structures

<table>
<thead>
<tr>
<th>Structure</th>
<th>Execution Time in Second</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP-1</td>
<td>1.253</td>
</tr>
<tr>
<td>OBP-1</td>
<td>1.222</td>
</tr>
<tr>
<td>BP-2</td>
<td>0.603</td>
</tr>
<tr>
<td>OBP-2</td>
<td>0.582</td>
</tr>
<tr>
<td>BP-3</td>
<td>0.582</td>
</tr>
<tr>
<td>OBP-3</td>
<td>0.553</td>
</tr>
<tr>
<td>BP-4</td>
<td>0.313</td>
</tr>
<tr>
<td>OBP-4</td>
<td>0.311</td>
</tr>
</tbody>
</table>
Chapter Five
Conclusions and Future Works

5.1 Introduction
This chapter summarizes the techniques that are used for solving spam filtering problem and presents the conclusions of the work. Furthermore, some important points are listed and described in a simplified form to be as future works.

5.2 Conclusions
The work described in this thesis concerns the application of two different techniques of neural network (BP and OBP) to filter incoming email messages. The filter distinguishes the spams from legitimate emails. The neural network should be trained to be able to distinguish between spams and legitimate emails. Nine groups of training dataset are used in training phase and three testing dataset groups are used to evaluate the performance of the proposed spam filtering. Also, four different feature set sizes (57, 42, 28, and 14) are used in BP and OBP to show the effect of feature set size on the performance of proposed spam filtering. In general, the results of the proposed spam filtering approach based on OBP are better than the results of BP.

The training dataset size and testing dataset size play a significant role on performance on the proposed spam filtering. Increasing the training dataset size increases the ability of neural network to differentiate between
legitimate email and spam. In this thesis, the training dataset with 1000 samples give the best results for all testing dataset groups.

The features set size has a significant impact on performance of the proposed spam filtering approach. Increasing features set size increases the performance of the proposed spam filtering. Consequently, the results of BP and OBP with 57 feature set size are better than the results of BP and OBP with 42, 28 and 14 feature set size in terms of accuracy, recall, precision, false positive, false negative, and execution time because they use 100% of features of each sample which provide more information about the incoming message.

Also, the result of OBP with 42, 28 and 14 features set size are better than the result of BP with 42, 28 and 14 features set size in all evaluation measurements because the ability of OBP to escape from local minima with high speed convergence.

5.3 Future Works

Several points have not been implemented which could be addressed in future researches and experimental works.

1. The current work could be traced on a real client-server system. This should consider real emails and how to extract proper feature set for training and testing. Moreover, it should consider computation time overhead needed to filter out spams from legitimate emails.

2. One of the future works is how to contract the setting of parameters in order to build out a proper NN structure instead of trail error setting.
References


[Cof05] M. Cofield, "Symantec Norton Anti-Virus 9.0.1for Macintosh Download, Installation, and Basic Use Tutorial", Based on information found at:
A Content-Based Spam Filtering Approach Using Artificial Neural Networks


A Content-Based Spam Filtering Approach Using Artificial Neural Networks


A Content-Based Spam Filtering Approach Using Artificial Neural Networks


A Content-Based Spam Filtering Approach Using Artificial Neural Networks


[Smi02] L. I Smith, A tutorial on Principal Components Analysis,


A Content-Based Spam Filtering Approach Using Artificial Neural Networks

February 26, 2002

A Content-Based Spam Filtering Approach Using Artificial Neural Networks

List of Publication

المستخلص

نتيجة ازدياد انتشار الإنترنت، أصبح البريد الإلكتروني وسيلة شائعة جداً وملائمة للاتصالات اليومية. هدفت الرسائل المزعجة والتي تعرف برسائل البريد الإلكتروني التجاري غير المطلوب، أو رسائل البريد الإلكتروني غير المرغوب فيها، إلى إثارة التهديد وحماية البريد الإلكتروني. بما أن مرسل الرسائل المزعجة يعملون باستمرار على تحسين أساليبهم لخرق مرشحات الرسائل غير المرغوب فيها، فإن هذا ومنشأ وتحسين تصفية الرسائل غير المرغوب فيها الذي يعمل باستمرار ويتكييف أصبح مجالاً للبحث الفعال.

يقدم هذا البحث أسلوب لتصفية الرسائل غير المرغوب فيها باستخدام اثنين من الشبكات العصبية الاصطناعية (ANN: الانتشار الخلفي (BP) و الانتشار الخلفي البصري (OBP)) للتعلم على الرسائل غير المرغوب فيها والتعرف على الرسائل المشروعة. يجب تدريب هذه الشبكات باستخدام مجموعة من العينات لتميز فيما إذا كانت الرسالة هي رسالة غير مرغوبة أو رسالة مشروعة. يجب عمل معالجة أولية لعينات مجموعة البيانات لتكون في شكل يمكن فهمها من قبل الشبكات العصبية. عمليات المعالجة الأولى تتضمن استخلاص الخواص باستخدام "مبدأ تحليل المكونات" PCA والتسويق. 

أجرت العديد من التجارب لبيان كيف تؤثر تلك الأساليب المقترحة لتصفية البريد المزعج، وتمت المقارنة بين هذه التجارب باستخدام مقياس مختلف للتقييم. أظهرت النتائج أن OBP و BP متماثل من حيث الدقة (100%), الاستدعاء (100%), الضبط (100%)، الإيجابية الكاذبة (0%), والسلبية الكاذبة (0%) في حالة استخدام 100% من خواص مجموعة البيانات المستندة إلى البريد المزعج تحتاج وقت تنفيذ (1,222 ثانية) أقل. كذلك أظهرت النتائج أن OBP استخدام 75%, 50% و 25% من حجم مجموعة الخواص أفضل من BP المقابله لها في جميع مقاييس التقييم.
A Content-Based Spam Filtering Approach Using Artificial Neural Networks

A Content-Based Spam Filtering Approach Using Artificial Neural Networks