

Classifying spam with generalized additive neural networks

P Labuschagne
21269777

Dissertation submitted in partial fulfilment of the requirements for the degree *Magister Scientiae* in *Computer Science* at the Potchefstroom Campus of the North-West University

Supervisor: Dr JV du Toit

May 2017



Dedicated to my family and friends.

“The question of whether computers can think is just like
the question of whether submarines can swim.”

Edsger W. Dijkstra

**LETTER OF CONFIRMATION:
PROOFREADING AND EDITING OF DISSERTATION**

7 November 2016

To whom it may concern

I, Stefanus van Zyl, proofread the MA dissertation (titled: *Classifying spam with generalized additive neural networks*) submitted by **Mr P Labuschagne (21269777)** and edited it to the best of my abilities. The final version of the document remains the responsibility of the student.

Regards

A handwritten signature in black ink, appearing to read 'Stefanus van Zyl', with a stylized flourish at the end.

Stefanus van Zyl

BA, BA Hons (Literature), BA Hons (Psychology), MA (Literature)

Acknowledgements

The completion of this degree has been a challenging, but worthwhile experience. It was no trivial endeavour while working fulltime and there were many highs and lows along the journey to completion. After years of hard work and determination I finally finished the degree and proved to myself and others that anything is possible. I am proud of this achievement and know that the research community will benefit from my contributions.

All honour and gratitude to my Heavenly Father for blessing me with many gifts and giving me the strength to finish.

I thank my supervisor, Dr Tiny du Toit, for all his contributions, support and encouragement. I am grateful for the opportunities to present my work at national conferences and the research skills acquired under your guidance. You are an exceptional supervisor who is always willing to help.

I especially appreciate the considerable support from my family, friends and colleagues. Thank you for showing interest even if you did not also understand the subject matter.

I also want to express my gratitude to SAS[®] Institute Inc. for providing the SAS[®] Enterprise Miner[™] software used to compute the results presented in this dissertation.

Thank you to Mr Stefan van Zyl for the language editing and everyone who contributed to this dissertation.

Abstract

E-mail is an important and convenient communication tool used by many people on a daily basis. For individuals it is an inexpensive way to stay in contact with family and friends located around the world. An e-mail address serves as an online identity when signing up for different online services like social media (Facebook) and social networking (LinkedIn). Companies use e-mails to facilitate communication between employees and to communicate with their clients by sending information such as newsletters, invoice statements and promotional content. E-mails are also used for core business marketing. Unfortunately, some of the benefits provided by the e-mail application like sending out mass e-mails with little effort at a minimal cost to the sender, are abused by some e-mail users known as spammers. A spammer's incentive for sending unsolicited e-mails in large quantities to an indiscriminate set of recipients is mostly driven by revenue generation. Most spam messages sent contain content related to promotional products and services, which might be a scam or phishing attempt to steal sensitive user information like banking details and passwords. Currently, more than 55.00% of all e-mail network traffic comprises unsolicited spam e-mails which clutters users' inboxes. Traditional spam-filtering approaches have thus far been unsuccessful in solving the spam problem. This is partly due to spammers who generate new spam message content on a regular basis making it difficult for spam filters to classify spam according to a fixed pattern.

The main purpose of this study is to determine the feasibility of employing a Generalized additive neural network (GANN) to filter spam e-mail messages with a specific automated construction algorithm. The GANN is a relatively new supervised machine learning technique capable of recognising complex patterns in data and able to adapt to changes over time. The use of GANN models is suggested for classification problems where it might be important to understand the relationship between input attributes and the expected target value. In this study the definition of spam, consequences of unmanaged spam and current spam-filtering techniques are investigated. The current state of the spam problem is summarised followed by a discussion on artificial neural networks that have pattern recognition capabilities. Literature related to the GANN is reviewed with a discussion on both the interactive and automated construction methodologies for the GANN. The latter will be considered as a possible spam filter to try and mitigate the spam problem. A number of spam filtering experiments are conducted on five publicly available spam corpora (Enron, GenSpam, PU1, SpamAssassin and TREC2005) each with different pre-processing techniques and evaluation measures. The Bagging and Boosting ensemble techniques which may improve on the GANN's results are also considered. The GANN and ensembles are then compared to other spam filtering techniques applied to the five corpora before being compared to each other. Results show that the GANN is a feasible spam filter able to mitigate spam e-mails. It compares well to other spam filter techniques found in the literature. In addition, both ensemble methods are able to improve on the GANN's results in most cases.

Keywords: artificial neural networks, AutoGANN, bagging, boosting, classification, e-mail, ensemble, GANN, Generalized additive neural networks, MLP, multilayer perceptron, spam.

Opsomming

E-pos is 'n belangrike en gerieflike kommunikasiesistelsel wat deur baie mense op 'n daaglikse basis gebruik word. Vir individue is dit 'n goedkoop manier om in kontak te bly met familie en vriende regoor die wêreld. 'n E-pos adres dien ook as aanlyn identiteit wanneer aangesluit word by verskillende aanlyndienste soos sosiale media (Facebook) en sosiale netwerke (LinkedIn). Maatskappye gebruik e-pos om kommunikasie tussen werknemers te fasiliteer en met kliënte te kommunikeer deur inligting soos nuusbriewe, fakture en promosie-inhoud te stuur. E-posse word ook gebruik vir kernbesigheidbemarking. Ongelukkig word sommige voordele wat die e-posstelsel bied, soos die uitstuur van massa e-posse met min moeite teen 'n minimale koste aan die sender, misbruik deur sommige e-posgebruikers bekend as spammers. Spammers se aansporing vir die stuur van ongevraagde e-posse in groot hoeveelhede na 'n willekeurige stel ontvangers is meestal gedryf deur inkomstegenerering. Die meerderheid gemorsposboodskappe wat gestuur word, bevat inhoud wat verband hou met die bevordering van produkte en dienste wat 'n bedrogspul of strikroofpoging is om sensitiewe gebruikersinligting soos bankbesonderhede en wagwoorde te bekom. Tans bestaan meer as 55.00% van alle e-posnetwerkverkeer uit ongevraagde gemorspos wat gebruikers se aanlyn posbusse oorlaai. Tradisionele gemorsposfilterstelselbenaderings is tans onsuksesvol in die oplos van die gemorsposprobleem. Dit is deels te wyte aan spammers wat op 'n gereelde basis nuwe inhoud vir gemorsposboodskappe genereer en dit moeilik maak vir gemorsposfilters om gemorspos volgens 'n vaste patroon te klassifiseer.

Die hoofdoel van hierdie studie is om die geskiktheid van die gebruik van 'n Veralgemeende additiewe neurale netwerk (VANN) om gemorspos te filtreer met 'n spesifieke outomatiese konstruksie-algoritme te bepaal. Die VANN is 'n relatiewe nuwe gekontroleerde masjienleertegniek wat in staat is om komplekse patrone in data te herken. Dit is ook in staat om aan te pas by veranderinge met die verloop van tyd. Die gebruik van VANN-modelle word voorgestel vir klassifikasieprobleme waar dit belangrik sou wees om die verhouding tussen insetveranderlikes en die verwagte teikenwaarde te verstaan. In hierdie studie word die definisie van gemorspos, die gevolge van onbeheerde gemorspos en huidige gemorsposfiltertegnieke ondersoek. Die huidige stand van die gemorsposprobleem word opgesom, gevolg deur 'n bespreking oor kunsmatige neurale netwerke wat patroonherkenningvermoëns het. 'n Literatuuroorsig met betrekking tot die VANN word gegee en bevat 'n bespreking van beide die interaktiewe en outomatiese konstruksiemetodes vir die VANN. Laasgenoemde sal gebruik word as 'n gemorsposfilter om die gemorsposprobleem te probeer verminder. 'n Aantal gemorsposeksperimente is op vyf publieke gemorsposkorpuse (Enron, GenSpam, PU1, SpamAssassin en TREC2005) uitgevoer - elk met verskillende voorbereidingsmetodes en evalueringmaatstawwe. Die Sakvorming- en Bevordering-ensembletegnieke wat moontlik kan verbeter op die VANN-resultate word ook in ag geneem. Die VANN en ensembles word dan vergelyk met ander gemorsposfiltertegnieke wat toegepas is op die vyf korpuse voordat dit met mekaar vergelyk word. Resultate dui aan dat die VANN 'n geskikte gemorsposfilter is wat gemorspos kan verminder. Dit vergelyk goed met ander gemorsposfiltertegnieke wat in die literatuur voorkom. Beide ensemblemetodes is boonop in die meeste gevalle in staat om te verbeter op die resultate van die VANN.

Sleutelwoorde: AutoGANN, bevordering, ensemble, e-pos, gemorspos, klassifikasie, kunsmatige neurale netwerke, MLP, multilaagperseptrone, sakvorming, VANN, Veralgemeende additiewe neurale netwerke.

TABLE OF CONTENTS

Acknowledgements	iv
Abstract	v
Opsomming	vi
Table of contents	viii
List of figures	xi
List of tables	xii
List of algorithms	xiv
Abbreviations	xv
1 Introduction	1
1.1 Problem statement	2
1.2 Research objectives	3
1.3 Method of investigation	4
1.4 Dissertation outline	4
1.5 Conclusions	5
2 Spam E-mail	7
2.1 Background	9
2.2 Consequences of unmanaged spam e-mails	12
2.2.1 Time impact	13
2.2.2 Financial impact	14
2.2.3 Security impact	15
2.3 Spam e-mail counteract measures	17
2.3.1 Nonfilter solutions	19
2.3.1.1 CAN-SPAM Act of 2003	20
2.3.1.2 ePrivacy Directive	21
2.3.1.3 ISP policies	21
2.3.2 Filter solutions	21
2.3.2.1 Content-based filters	23
2.3.2.2 List-based filters	23
2.3.2.3 Other filters	24
2.4 Current state of the spam problem	29
2.5 Conclusions	30

3	Artificial Neural Networks	32
3.1	History	33
3.2	The Neuron model	35
3.2.1	The single-input neuron	35
3.2.2	The multiple-input neuron	36
3.2.3	The Perceptron	37
3.2.4	A layer of neurons	40
3.3	The Multilayer Perceptron	42
3.3.1	The Backpropagation algorithm	43
3.3.1.1	The Chain rule	44
3.3.1.2	Backpropagating the sensitivities	46
3.3.1.3	Summary	49
3.4	Conclusions	49
4	Generalized Additive Neural Networks	50
4.1	Generalized additive models and smoothing	51
4.2	The GANN architecture	53
4.2.1	Interactive construction methodology	57
4.2.2	Automated construction methodology	58
4.3	Ensemble techniques	59
4.3.1	Bagging	62
4.3.2	Boosting	63
4.4	Conclusions	64
5	Experimental Design and Results	65
5.1	Experimental design	66
5.1.1	GANN experiments	66
5.1.2	Ensemble experiments	66
5.1.3	Preprocessing of corpora	67
5.1.3.1	Stop-word removal	68
5.1.3.2	Lemmatisation and stemming	68
5.1.4	Representation of data	69
5.1.5	Evaluation measures	70
5.2	The Enron data set	73
5.2.1	GANN and ensemble results	74
5.3	The GenSpam data set	77
5.3.1	GANN and ensemble results	79
5.4	The PU1 data set	81
5.4.1	GANN and ensemble results	83
5.5	The SpamAssassin data set	85
5.5.1	GANN and ensemble results	86
5.6	The TREC2005 data set	87
5.6.1	GANN and ensemble results	87
5.7	Conclusions	88
6	Discussion	89
6.1	Comparison to other techniques	89
6.1.1	Enron corpus	89

6.1.2	GenSpam corpus	91
6.1.3	PU1 corpus	92
6.1.4	SpamAssassin corpus	94
6.1.5	TREC2005 corpus	94
6.2	Model accuracy	95
6.3	Model comprehensibility	98
6.4	Model construction and ease of use	98
6.5	Conclusions	99
7	Conclusions	100
7.1	Research objectives	100
7.2	Research results	103
7.3	Research limitations	103
7.4	Recommendations for future work	104
7.5	Conclusions	104
	Reference list	105

LIST OF FIGURES

2.1	First commercial spam e-mail: Green card lottery	10
2.2	Spam e-mail externalities	13
2.3	E-mail framework	17
2.4	E-mail framework components	18
2.5	CAPTCHA	26
2.6	reCAPTCHA	26
2.7	KittenAuth CAPTCHA	27
2.8	Mathematics CAPTCHA	27
2.9	Social CAPTCHA	27
2.10	PlayThru CAPTCHA	28
2.11	No CAPTCHA reCAPTCHA	28
3.1	Biological neuron	33
3.2	Single-input neuron	35
3.3	Multi-input neuron	36
3.4	Hard-limit activation function	37
3.5	Hard-limit activation function symbol	37
3.6	The Perceptron	38
3.7	Perceptron decision boundary	38
3.8	Linearly inseparable problem	40
3.9	A single-layer of neurons	40
3.10	The Multilayer Perceptron (MLP) neural network architecture	42
4.1	GANN architecture	55
4.2	Enhanced GANN architecture	56
4.3	Fundamental reasons for enhanced ensemble performance over a single classifier	61
4.4	Example of Bagging voting	63
5.1	Bagging and Boosting ensembles in SAS [®] Enterprise Miner [™]	67
5.2	Stop-word removal example	68
5.3	Lemmatisation and stemming examples	69
5.4	GenSpam message representation	79
5.5	PU1 message before encryption	82
5.6	PU1 message after encryption	82

LIST OF TABLES

2.1	Yearly decrease of spam in global e-mail traffic.	30
4.1	GANN subarchitecture identifiers.	56
5.1	E-mail classification classes.	70
5.2	Confusion matrix.	71
5.3	Enron corpora description.	73
5.4	GANN and ensemble results for the Enron 1 subset.	74
5.5	GANN and ensemble results for the Enron 2 subset.	74
5.6	GANN and ensemble results for the Enron 3 subset.	75
5.7	GANN and ensemble results for the Enron 4 subset.	76
5.8	GANN and ensemble results for the Enron 5 subset.	76
5.9	GANN and ensemble results for the Enron 6 subset.	77
5.10	GenSpam corpora description.	78
5.11	GenSpam encryption types.	78
5.12	GANN and ensemble results for the GenSpam Training subset.	79
5.13	GANN and ensemble results for the GenSpam Adaption subset.	80
5.14	GANN and ensemble results for the GenSpam Combination subset.	81
5.15	PU1 subset description.	82
5.16	GANN and ensemble results for the PU1 Bare subset.	83
5.17	GANN and ensemble results for the PU1 Stop subset.	83
5.18	GANN and ensemble results for the PU1 Lemm subset.	84
5.19	GANN and ensemble results for the PU1 Lemm Stop subset.	85
5.20	SpamAssassin corpus description.	85
5.21	GANN and ensemble results for the SA corpus.	86
5.22	TREC2005 corpus description.	87
5.23	GANN and ensemble results for the TREC2005 corpus.	87
6.1	GANN and ensemble techniques compared to the five NB techniques applied to the Enron 1 subset.	90
6.2	GANN and ensemble techniques compared to the five NB techniques applied to the Enron 2 subset.	90
6.3	GANN and ensemble techniques compared to the five NB techniques applied to the Enron 3 subset.	90
6.4	GANN and ensemble techniques compared to the five NB techniques applied to the Enron 4 subset.	90
6.5	GANN and ensemble techniques compared to the five NB techniques applied to the Enron 5 subset.	90
6.6	GANN and ensemble techniques compared to the five NB techniques applied to the Enron 6 subset.	91
6.7	GANN and ensemble techniques compared to the five techniques applied to the GenSpam Training subset.	92
6.8	GANN and ensemble techniques compared to the five techniques applied to the GenSpam Adaption subset.	92

6.9	GANN and ensemble techniques compared to the five techniques applied to the GenSpam Combination subset.	92
6.10	GANN and ensemble techniques compared to the NB technique applied to the PU1 Bare subset.	93
6.11	GANN and ensemble techniques compared to the NB technique applied to the PU1 Stop subset.	93
6.12	GANN and ensemble techniques compared to the NB technique applied to the PU1 Lemm subset.	93
6.13	GANN and ensemble results compared to the NB technique applied to the PU1 Lemm Stop subset.	93
6.14	GANN and ensemble techniques compared to the DBN and SVM techniques applied to the SpamAssassin data set.	94
6.15	GANN and ensemble techniques compared to the four yorSPAM techniques applied to the TREC2005 data set.	95
6.16	GANN and ensemble accuracy results for the fifteen experiments on the five data sets.	96
6.17	Best accuracy method for each of the fifteen experiments on the five data sets.	96
6.18	Best overall method for accuracy.	97

LIST OF ALGORITHMS

4.1	Interactive construction algorithm of the GANN.	58
4.2	Automated construction algorithm of the GANN.	60
4.3	The Bagging algorithm.	62

ABBREVIATIONS

ACC	accuracy
AI	artificial intelligence
ANN	artificial neural network
ARPANET	Advanced Research Projects Agency Network
BC	blind copy
BLR	Bayesian logistic regression
CAN-SPAM Act of 2003	Controlling the Assault of Nonsolicited Pornography and Marketing Act of 2003
CAPTCHA	completely automated public Turing test to tell computers and humans apart
CC	carbon copy
DBN	deep belief network
ePrivacy Directive	European union privacy and electronic communications directive
E-mail	electronic mail
FB	flexible Bayes
GAM	generalized additive model
GANN	generalized additive neural network
HM	ham misclassification
HP	ham precision
HR	ham recall
HTML	hypertext mark-up language
IETF	internet engineering task force
ILM	interpolated language model
IP	internet protocol
ISP	internet service provider
kNN	k -nearest neighbour
LEMM	lemmatisation
MI	mutual information
MLP	multilayer perceptron

MN Bool	multinomial Boolean
MN TF	multinomial term frequency
MNB	multinomial naïve Bayes
MSE	mean squared error
MV Bern	multivariate Bernoulli
MV Gauss	multivariate Gaussian
MV	multivariate
NB	naïve Bayesian
RFC	request for comments
SA	SpamAssassin
SARS	South African Revenue Service
SAS®	statistical analysis system
SM	spam misclassification
SMTP	simple mail transport protocol
SP	spam precision
SR	spam recall
SVM	support vector machine
TCP/IP	transmission control protocol/internet protocol
TCR	total cost ratio
UBE	unsolicited bulk e-mail
UCE	unsolicited commercial e-mail
URL	uniform resource locator
WACC	weighted accuracy

CHAPTER 1

INTRODUCTION

The internet is expanding at a rapid pace as new devices are connected to this global network on a daily basis. Although the amount of data being generated is staggering, present-day computers help humans to understand and manage it better. The internet provides a variety of information and communication services of which electronic mail (e-mail) is one of the most popular communication services. Other services include voice over internet protocol like Skype, instant messaging like WhatsApp, social media websites like Facebook, shopping portals like Takealot, online gaming like Steam and cloud storage like Google Drive. Most online services require an e-mail address for user authentication, password recovery, newsletter sign-up, linking of multiple services and overall communication. Unfortunately, users can not be certain how secure these services are in keeping their e-mail addresses private or with which third parties these services are sharing data. As a result, the users' inboxes are likely to get cluttered with unwanted messages from unknown sources which might lead to frustration when navigating the inbox or can even compromise the user's computer system if the messages are malicious. This study focuses on e-mails and how a machine learning technique can be used to assist users in managing their inboxes against unsolicited e-mail messages.

The spam e-mail problem has been unresolved for more than 20 years since the first spam e-mail message was sent in 1994 (Singel, 2010; Cranor & LaMacchia, 1998). Many solutions have been proposed to address the spam problem which include a variety of spam filtering techniques: content-based filters, list-based filters and other filters like collaborative filters, support vector machine filters, naïve Bayesian filters and artificial neural network (ANN) techniques. Since spammers (people who send spam messages) regularly adapt their spamming patterns to outsmart filters, a more dynamic approach is required to address the problem. One way of achieving this is through the use of ANNs as it is able to recognise complex patterns and discover useful relationships in data. A supervised machine learning technique will be used to address the spam e-mail classification problem. Supervised machine learning

is the ability of a computer system to learn from example data and adapt to change when exposed to new data without being explicitly programmed. A Generalized additive neural network (GANN) is a predictive model that discovers patterns with supervised learning by mapping the input values to an expected target value (Kotsiantis, Zaharakis & Pintelas, 2006). For a spam filter to be effective, it is necessary to adapt over time. Fortunately, a GANN has this capability and can thus be regarded as a possible candidate for spam filtering. In this study, a GANN is considered to classify spam e-mails with an automated construction algorithm (Du Toit, 2006).

The problem statement for this study is presented in Section 1.1. A list of research objectives is presented in Section 1.2. The method of investigation is discussed in Section 1.3 followed by the dissertation outline in Section 1.4. Finally, a chapter summary is given in Section 1.5.

1.1 Problem statement

Currently, e-mail is an important, cost friendly and convenient medium for daily communications. According to Alhadlaq (2016), e-mail is one of the most preferred communication methods with a noteworthy technological influence on communication in general. Unfortunately, more than 55.00% of all e-mail network traffic comprise unsolicited messages known as spam (Vergelis, Shcherbakova, Demidova & Gudkova, 2016). The fact that spam contributes to so much of the overall percentage of e-mail network traffic, makes it responsible for causing unnecessary network load. Other difficulties caused by spam include financial problems, security issues and time constraints for users. The delivery of important legitimate e-mails could get delayed as network congestion occurs due to spam messages. Organisations are impacted in a financial manner since employees spend working time to manage their inboxes. Spam e-mails might also pose a security risk as it could contain attachments infected with viruses or spyware that can harm the recipients' system (Lee, Kim, Kim & Park, 2010). It also impacts each recipient in a nonproductive manner by claiming our precious time (Caliendo, Clement, Papiés & Scheel-Kopeinig, 2008). Spam can be perceived as intrusive and continues to be a constant annoyance.

Most internet service providers attempt to reduce the costs incurred from spam by implementing spam filter systems. A spam filter is a software system that checks incoming messages for spam before delivering it to the designated recipient. These filters require maintenance on a regular basis to perform optimally (Goodman, Cormack & Heckerman, 2007). Researchers need to stay on the forefront of this problem to counter-attack new tactics used by spammers. They periodically find new ways to bypass existing spam filters and are able to distribute spam e-mail messages on a large scale because

the distribution of spam e-mails is inexpensive and easy to execute. To date, spam is still an ongoing problem with no solution.

The spam e-mail classification problem involves identifying the correct class (spam or nonspam) to which an incoming e-mail message belongs. Previous research on the GANN related to spam e-mail detection provided only preliminary investigations performed on either small or single spam data sets in evaluating the GANN's classification accuracy. In the next section the research objectives are outlined.

1.2 Research objectives

The primary aim of this study is to determine the feasibility of employing a GANN to filter spam e-mail messages with an automated construction algorithm for the GANNs. This study will contribute to existing literature regarding the use of supervised machine learning techniques for spam e-mail classification. The implementation of an automated construction algorithm for the GANN is tested empirically as a possible solution to mitigate spam. Two ensemble techniques (Bagging and Boosting) that might improve on the GANN's results are also examined. The secondary objectives, which will contribute to achieving the primary research objective, are as follows:

1. Define spam and describe literature relating to the problem of spam e-mails and the classification thereof.
2. Investigate the impact spam e-mails could have on individuals and companies when left unmanaged.
3. Discuss artificial neural networks in general and the Multilayer Perceptron (MLP) neural network architecture which forms the basis of GANNs.
4. Describe the GANN model and the associated automated construction algorithm.
5. Give an overview of the Bagging and Boosting ensemble techniques that are applied to the GANN which might improve on the results obtained by the GANN.
6. Discuss the experimental design used to compare the GANN to other spam filtering techniques found in the literature based on a number of different metrics.
7. Apply the ensemble techniques to the results obtained by the GANN to possibly improve the accuracy of the GANN model.

8. Compare the GANN model and the ensemble techniques in terms of accuracy, model interpretability and ease of construction.

The steps followed to achieve each secondary objective, which contributes to accomplishing the primary research aim, are described next.

1.3 Method of investigation

A comprehensive literature study on the e-mail application, spam e-mails and the classification thereof will be performed to provide background for this study. The need for a reliable spam filter is determined by investigating the spam e-mail trend and lack of a solution over the past few years. Next, the architecture of artificial neural networks, the Multilayer Perceptron (MLP) and the basic GANN structure will be considered before the automated construction algorithm of the GANN is discussed. The MLP architecture forms the basis of a GANN. The automated construction algorithm was implemented by Du Toit (2006) in the AutoGANN system. The construction of GANNs will be investigated prior to its application on five publicly available spam corpora. Experiments will be conducted by building GANN models with the AutoGANN system which is integrated in the SAS[®] Enterprise Miner[™] environment, to search for good models that classify spam e-mails. In addition, the two ensemble techniques, Bagging (Breiman, 1996) and Boosting (Freund & Schapire, 1996), will also be applied to the GANN model to try to improve on the GANN's results. The five spam corpora used in the experiments are Enron (Metsis, Androutsopoulos & Paliouras, 2006), GenSpam (Medlock, 2006), PU1 (Androutsopoulos, Koutsias, Chandrinos & Spyropoulos, 2000), SpamAssassin (SA) (Tzortzis & Likas, 2007) and TREC2005 (Cormack & Lynam, 2005). Different preprocessing techniques and evaluation measures will be applied to each data set. The experimental design entails a detailed description of each experiment performed and is discussed to ensure objective comparisons. This allows researchers to replicate the experiments and obtain similar results, thus encouraging future research. The implemented approach is scientific of nature and focuses on the use of experimental methods to derive empirical evidence from careful observation. Results obtained from the experiments are analysed in a quantitative manner. In the next section the dissertation outline is discussed.

1.4 Dissertation outline

The structure of the dissertation is outlined in this section and a brief overview of each chapter is given. In Chapter 2 a general overview of spam e-mails is provided. The incentives for spamming and

three consequences of unmanaged spam are discussed. The current state of the spam e-mail problem is investigated to motivate the use of a relatively new supervised machine learning technique (the GANN) to address the issue. Current anti-spamming techniques are also discussed.

In Chapter 3 a brief history on artificial neural networks is given. The neuron model and a layer of neurons are described before the structure of the MLP is discussed. This type of neural network is currently the most widely used. The Backpropagation algorithm, a learning technique for MLPs, is discussed since this learning technique overcame the theoretical pitfalls of basic ANNs and criticism on the Perceptron model's training capabilities. With the Backpropagation algorithm it is possible to train ANNs consisting of multiple layers of Perceptrons which could then solve nonlinearly separable problems. Context is provided by the MLP for the description of the GANN architecture.

In Chapter 4 related literature on the GANN is considered. A discussion on Generalized additive models (GAMs) and smoothing is presented since a GANN is the neural network implementation of a GAM. The GANN architecture is discussed followed by a discussion on the interactive and automated construction methodologies for the GANN. The Bagging and Boosting ensemble methods applied to the GANN to possibly improve the classification accuracy obtained with the AutoGANN system are also discussed. Next, the GANN and ensemble experiments on five publicly available corpora are considered.

The experimental design, preprocessing steps, evaluation measures for testing the performance of the GANN and experimental results obtained by each experiment are discussed in Chapter 5.

A comparison and detailed discussion on the results per measurement obtained by the GANN and the Bagging and Boosting ensemble techniques, compared to other spam filtering techniques found in the literature, are given in Chapter 6. Model comprehensibility and the ease of model construction for the GANN and ensemble techniques will also be discussed.

In Chapter 7 a final conclusion on the GANN's capabilities to mitigate spam e-mails is given. The research limitations for this study and possible opportunities for further research are also considered.

1.5 Conclusions

This chapter gave an overview of the complete study. E-mail as an important and convenient means of communication, as well as its use for most online services to function properly, was briefly discussed. The problem statement for the spam e-mail classification problem was also provided. To address the spam e-mail classification problem a supervised machine learning technique, the automated construction

algorithm for the GANN, is considered in an attempt to mitigate the issue. Multiple research objectives were stated that will contribute to accomplishing the primary research aim which is determining the feasibility of a GANN to accurately classify spam e-mails. A series of experiments have been conducted to reach a conclusion. A literature study on spam e-mails is discussed next in Chapter 2.

CHAPTER 2

SPAM E-MAIL

The technological world is rapidly evolving and improving on current technologies showing no sign of slowing down. Information industries are expanding and becoming more information-intensive, requiring a means for quick and easy communication. One of the most widely used communication tools from the technological world is electronic mail more commonly known as e-mail (Alhadlaq, 2016; Aceto & Pescapè, 2012; Gomez & Moens, 2012). An e-mail message is the virtual representation of a written message with the exception of the option to add digital attachments consisting of various file types and formats. The attachments enable the end users to share information easily and can be represented as some of the following examples: documents, pictures, compressed archives and audio or video files. E-mail is therefore a digital message transfer that occurs between a sender and one or more recipients.

The e-mail application was developed by Ray Tomlinson in 1971 (Tomlinson, n.d.). In the early stages of development, the application provided primitive services to both researchers and military institutions who had access to the Advanced Research Projects Agency Network (ARPANET). Established in 1969, ARPANET was the first packet switching network to utilise the transmission control protocol/internet protocol (TCP/IP) and revolutionised communication for basic data sharing. ARPANET became the precursor of what we nowadays refer to as the internet (Hafner & Lyon, 1998). At the time, the potential of the e-mail application was unknown as the internet was still in its early evolving stages. As the internet expanded with thousands of computer clusters being interconnected with each other on a global scale, it transformed data generation and information sharing techniques. It was not until 12 April 1994, when immigration lawyers Canter and Siegel from Phoenix decided to use the e-mail application as a promotional tool to enhance business processes (Everett-Church, 1999), that e-mail changed into a prevalent communication medium currently used worldwide by billions of people on a regular basis. Present e-mail systems are sophisticated and provide the end user with numerous advantages over other communication methods. These advantages can be compared to services which

provide the end users with a convenient, affordable and immediate way of communicating with other individuals or groups situated anywhere in the world.

For e-mail to continue as a functional application, internet standards have been defined. E-mail messages with or without attachments, can be sent via the internet using the simple mail transport protocol (SMTP) by adhering to internet standards adopted from the internet engineering task force (IETF). Information regarding the IETF SMTP specifications, communications protocols, procedures and events are described in the request for comments (RFC) publication 5321 (Klensin, 2008). The feasibility of e-mails depends on these standards and ensure e-mail messages can be sent globally to other devices capable of interpreting such digital messages. Unfortunately, there are e-mail users who abuse the characteristics of e-mail systems creating an unpleasant experience for various users that in actuality jeopardises the very utility of e-mails as a communication medium. These people are known as spammers who aim to reach the masses with spam e-mail messages. Spam e-mail is commonly defined as unsolicited bulk e-mail (UBE) messages sent to multiple recipients where the sender and receivers have no known relationship (Cranor & LaMacchia, 1998). It is therefore irrelevant, inappropriate and intrusive junk e-mail messages that provide recipients with unwanted content, claiming some of their valuable time and resources. Examples may include any e-mail messages with content related to a competition, chain mail where the recipient is asked to forward copies of the e-mail message to multiple people to avoid misfortune, a newsletter and a mailing list. Other examples of spam include phishing scams where spammers impersonate different institutions for the purpose of scamming the user into giving up sensitive information that is used for identity theft, pornography or prostitution. Commercial spam e-mails usually, but not always, focus their content on money-making techniques like adverts of various products or services. These messages are referred to as unsolicited commercial e-mail (UCE). An e-mail message with the opposite properties of junk mail is referred to as ham, legitimate or nonspam. Spammers are adaptable to change, fooling anti-spamming e-mail systems with their pattern changing techniques. By continually changing the spam content of e-mail messages, they avoid getting easily detected by anti-spamming systems. Current content-based anti-spamming systems are faced with a challenge where keeping up with new emerging patterns is getting more difficult.

In this chapter, background literature about the e-mail application is presented in Section 2.1. This will give a better understanding of the origin of spam e-mails and why it is still an ongoing problem. The need to optimise anti-spamming systems is also emphasised to ensure that communication channels are not overwhelmed by spam. In Section 2.2 possible consequences of unmanaged spam e-mails for both companies and individuals as a direct result of UBE and UCE are discussed. Current solutions to reduce spam e-mails are discussed in Section 2.3. This includes a more dynamic anti-spamming approach

which uses artificial neural networks. Section 2.4 summarises the current state of the spam problem and progress made by research communities and practitioners in an attempt to reduce the amount of global spam e-mail network traffic. A conclusion to the chapter will be presented in Section 2.5.

2.1 Background

The term spam originated from a 1970s British skit titled *Monty Python's Flying Circus* (Ivey, 1998). The comedy sketch depicted a cafe where almost every item listed on the menu included SPAM (a canned precooked meat product). After the waitress received her customers' orders, a group of Vikings disrupt all other conversations by loudly singing "Spam, spam, spam, spam, lovely spam, wonderful spam." until told to "shut up" by the waitress. This continued several times, creating an unsettled environment for other guests making it difficult for them to converse and enjoy their meal in peace. Currently, SPAM[®] refers to the meat product produced by Hormel Foods Corporation (USA). According to the Oxford Dictionary of British & World English, the word spam is defined as "Irrelevant or unsolicited messages sent over the internet, typically to large numbers of users, for the purposes of advertising, phishing, spreading malware, etc."

On 12 April 1994 Canter and Siegel are believed to have sent the first commercial spam e-mail message (Figure 2.1) by advertising legal services on Usenet (Singel, 2010; Cranor & LaMacchia, 1998). Free green cards were offered in the form of a lottery competition targeted at immigrants outside the United States of America. The company made national headlines for introducing a new way of doing global business through mass advertising. Not long after the first commercial spam e-mail message was sent, the partners wrote a book titled *How to make a fortune on the information superhighway: everyone's guerrilla guide to marketing on the internet and other on-line services* (Canter & Siegel, 1994), sharing business successes and encouraging other businesses to make use of e-mail as a marketing tool (O'Connor, 2006). This resulted in frustration for many consumers who received these irrelevant messages frequently.

The Canter and Siegel instance started the commercial spam evolution allowing e-mail to evolve from a primitive service into a prevalent communication and advertising tool. According to The Radicati Group (2015), more than 205.6 billion e-mails are sent daily around the world. A large portion of these e-mails are business-related making e-mail the predominant form of communication in the business sector. Ducker & Payne (2010) indicated that information and communication technologies are capable of providing enterprises with a competitive advantage allowing businesses to move forward, improve on current processes and compete against adversaries (Pavic, Koh, Simpson & Padmore, 2007; Fink &

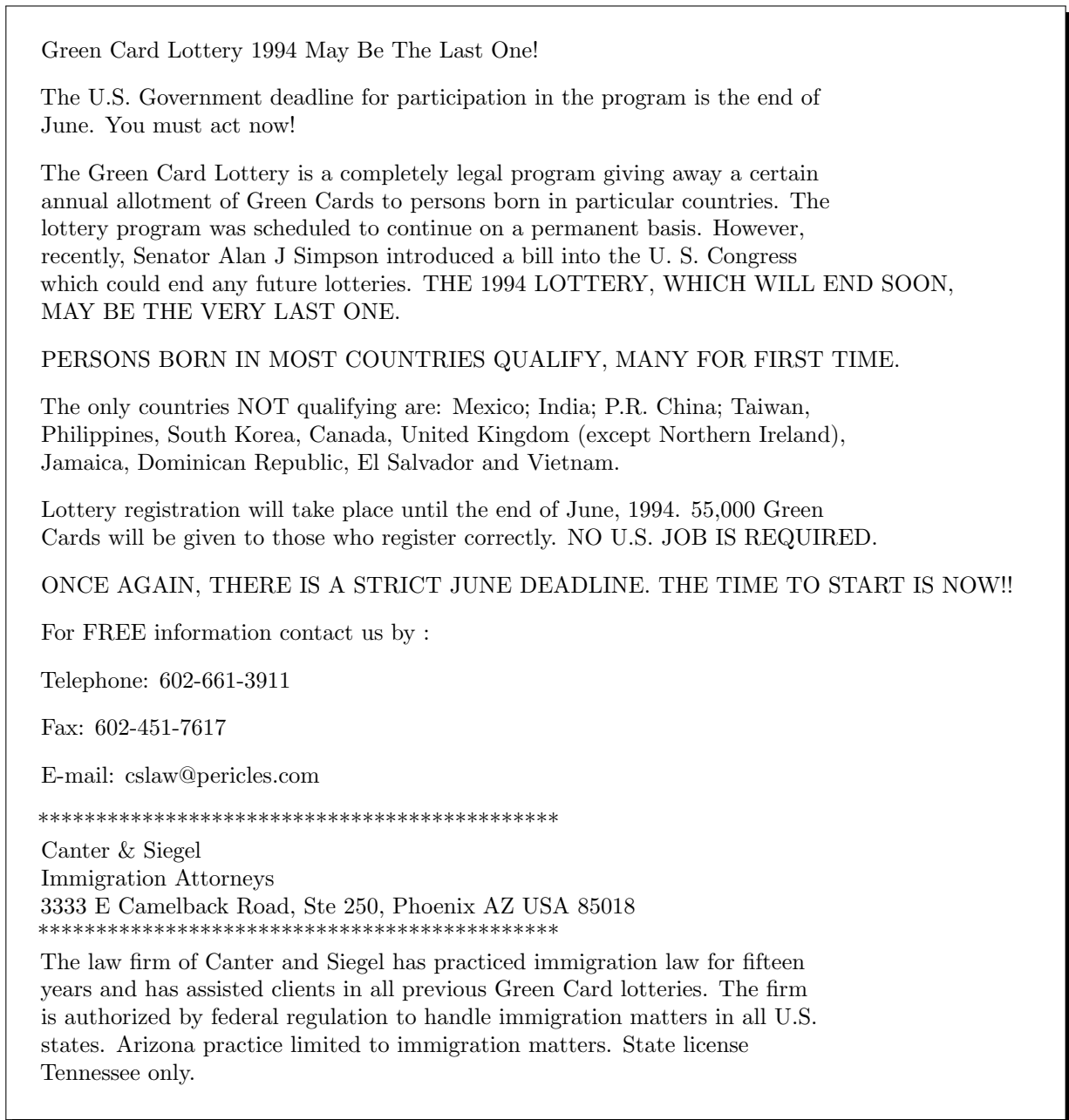


FIGURE 2.1: First commercial spam e-mail: Green card lottery.

Disterer, 2006). Communication with the customers, suppliers, and stakeholders should be effortless and hassle-free making e-mail the perfect communication method. Companies and small businesses mainly rely on their consumers for financial support, brand establishment and the overall success of making it in the competitive corporate world. E-mail is one of the most popular information-sharing and communication technology tools available at an extremely low cost (Gomez & Moens, 2012). If adopted and used wisely, the way products and services are marketed could be done more effectively with less effort saving both time and money (Apulu & Latham, 2011). The competitive advantage could entail better customer relations, increased profits from higher quality products and services offered

as well as promote awareness of products and services advertised globally (Ongori & Migiro, 2010). As storage media, sophisticated computer hardware and high speed internet services become more affordable, it is almost no surprise that spam e-mails are thriving. Spammers are able to use these technology advancements in sending millions of spam e-mail messages worldwide within a few minutes. The internet provides them with the tools for content creation, distribution and publication which ensures that the spam-based advertising businesses will continue to generate revenue. The digital age of information generation and sharing has numerous advantages in taking the world forward. With this came unforeseen problems like those associated with e-mail where the recipient's inbox is flooded with unwanted junk mail. This is due to the abuse of e-mail services by spammers who saw a money-making opportunity. A few years ago no one anticipated the harm spam could cause. Presently, spam is still an ongoing problem as the research community has not found a solution to the problem yet. Some attempts have proven effective in reducing spam, but only for a certain time period until spammers adapt to new anti-spamming techniques. When no action is taken against spam, there might be unforeseen consequences for end users. Spammers do have motives behind e-mail spamming. According to Hayati & Potdar (2008) the five incentives in order of importance for spamming are:

1. Revenue generation: The lucrative nature of the spamming business brings in about \$200 million revenue annually for spammers worldwide (Rao & Reiley, 2012). With little effort from the spammers' side and the overall low cost associated with sending bulk advertising messages, selling a few products will result in profit. Another means for generating profit is spamming on behalf of other businesses (Zhang, 2005).
2. Higher search engine ranking: Spammers send out mass e-mails containing ads and uniform resource locator (URL) links that represents web addresses pointing to another website that could represent a business website like an online store where products are for sale. With increasing visitors being directed to these websites, search engine rankings increase with the extra traffic letting the page show up more frequently in search engine results. This improves the chances of more sales and income for the spammers via advertising.
3. Promoting products and services: Technology advancement makes it easy to distribute ads on a global scale, expanding a business's products and service range to a much bigger audience (Ongori & Migiro, 2010). Spammers abuse this service by monetising synthetic content.

In contrast to the first three spamming motivations which are passive and do not intrude on a user's security or privacy, the last two motivations are actively aimed at malicious tasks and obtaining user-specific information.

4. Stealing information: A user's online security is directly affected by means of hidden malicious code on their system capable of executing actions like gathering keystrokes. These malicious applications are embedded and transferred via spam e-mails. Once installed, malware loaded onto the user's computer could show annoying advertisement popups and when clicked on by the user it opens a web browser and redirects them to a specific website managed by the spammer. It also attempts to obtain the user's e-mail address information to send out more e-mail spam on behalf of the user making the spam messages appear to be from a secure known sender. The malicious code gets distributed giving the spammer a back door entry point to multiple user computers.
5. Phishing: Spammers try to obtain personal and sensitive information like banking details, passwords, etc. to get access to certain accounts. With the information they could steal money and do online transactions by posing as the legitimate entity. Information could also be sold to other spammers. A phishing attack is achieved by forging the identity of a trusted source and posing as them. Most phishing scams are e-mails claiming to be from the bank, a social media site, etc. asking the user to verify their personal information by logging in onto the hoax site created by the spammer which is a URL link attached in the spam message.

Even though phishing is at the bottom of the list, it is gaining popularity with spammers and becoming one of the preferred methods to generate revenue by scamming end users into giving up personal information. The next section explores consequences of unmanaged spam e-mails followed by counteract measures to reduce spam. By exploring various anti-spam methods, the current state of spam control is determined.

2.2 Consequences of unmanaged spam e-mails

The internet is a global system of computer clusters put together by billions of devices that are interconnected with each other (Vermesan & Friess, 2013). Sometimes referred to as the Internet of Things, these smart devices allow people to share and manage information globally in an effective manner at affordable prices and blazing speeds. The e-mail application was built upon this continuous evolving architecture and became one of the preferred methods when facilitating information and communication tasks. From the billions of e-mails sent daily, more than 55.00% of the e-mail network traffic represent spam e-mails (Vergelis *et al.*, 2016). Companies are impacted by the unresolved spam problem to a greater extent than individual users, because they make use of e-mail on a much greater scale (The Radicati Group, 2016). Siponen & Stucke (2006) stated that spam can lead to direct financial losses and cause problems in different areas. This results from misused network traffic, and

storage space and computational power being wasted on delivering, storing and analysing spam e-mails. Those who receive spam are burdened by negative externalities and financial losses associated with the distribution of spam messages (Subramaniam, Jalab & Taqa, 2010; Goodman *et al.*, 2007; Melville, Stevens, Plice & Pavlov, 2006; Keizer, 2005). A depiction of these external aspects which affects e-mail users are presented in Figure 2.2. Each of these areas (time, finance and security) is discussed next with emphasis on how the issues are interconnected.

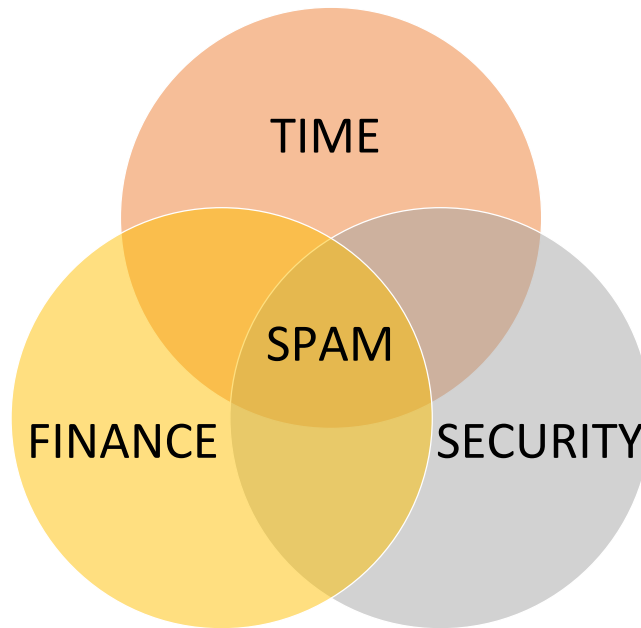


FIGURE 2.2: Three possible areas by which individuals and companies are impacted by as a result of unmanaged spam e-mails.

2.2.1 Time impact

Sending single or multiple digital messages to a friend, relative or any other entity is made effortless by clicking on a button. Users can decide when to reply to the e-mail messages, for example at a later time when it is best suited and accommodated by one's busy schedule. There is no doubt that e-mail saves us a lot of time when it comes to communicating arrangements or sharing information. However, as easy as it is to send and receive these messages, receiving too much e-mails have exactly the opposite effect by evoking frustration (Siponen & Stucke, 2006). Several hours could be wasted to filter through all the chain mail, promotions and social media notifications to get to important e-mail messages.

The most common negative effect from mismanaged spam is productive time loss (Caliendo *et al.*, 2008). Companies are greatly impacted by this and spend millions of dollars each year on damages incurred by losses in employee productivity. Working time is spent to filter through spam in order to advance

business goals. As a side effect, important tasks are put on hold and takes longer to complete. This may introduce scope creep where project deadlines are not met because of increasing work backlog. Support centres that rely greatly on e-mail communication for record keeping and user interaction will probably be affected the most as finding the relevant user request could turn into a highly time-consuming task. The possibility of accidentally deleting a user's requests while cleaning up spam messages is also a concern. Recovering from this requires time and might delay overall support and productivity in the workplace.

Spam have numerous negative effects. New prevention measures being implemented in response to the spam problem also takes up time. Filter configuring, training and necessary updates could contribute to even more time losses. All these scenarios have one common factor which is productive time loss.

2.2.2 Financial impact

E-mail is very affordable, if not the most affordable communication option when taking certain factors like time, distance and audience into consideration. Anyone can simply sign-up for multiple free e-mail accounts from various providers such as Gmail, Yahoo and Hotmail. Transferring a message around the world is free of charge, because postage service fees do not apply to e-mails as is the case with traditional mail. The only cost incurred is the required internet connection where some internet service providers (ISPs) may charge for bandwidth usage. When removing the human factor from the delivery process, there is very little delay in message transfer times. Recipients are notified immediately when new incoming messages enter their e-mail inboxes. The internet enables smart devices to communicate with each other and pushes your notifications to multiple devices (Gubbi, Buyya, Marusic & Palaniswami, 2013). These services are some of the main reasons why e-mail is such a widely used communication tool, because e-mails can provide users with instant notifications on most devices available today. However, most e-mail users do not even realise the amount of money service providers spend on infrastructure and that e-mail services require domains, storage space and other sophisticated hardware to operate and ensure an acceptable online experience.

In order to address the negative time impact spam e-mail has on productivity, financial support is needed. Sustaining a stable, fast and reliable internet connection is part of a business's core process. The company and ISP are challenged with spam issues that create additional traffic overhead consuming network resources. The sophisticated mail server hardware and software that filters out unwanted messages are costly and requires regular maintenance from highly paid specialists. Caliendo *et al.* (2008) pointed out that the cost incurred also depends on a filter's installation procedure, training time

and monitoring of the results regarding misclassifications. Expenses are bound to network data traffic transfers, quality of service, infrastructure cost and time needed for training as well as educating staff. As a direct result, damage to enterprises caused by spam can accumulate to about \$20 billion annually worldwide (Rao & Reiley, 2012). Neglecting any prevention measure against spam will result in the amount being much higher.

According to Cisco (2014), maliciously intended spam remains constant. With social engineering attacks, spammers present themselves as a trustworthy source whilst phishing for banking details and personal information. When individuals or companies fall victim to these attacks, it could result great financial losses that are irreversible. Spammers are always finding ways to exploit current prevention techniques, hiding their identities by using zombie computers to do their bidding. Zombie computers are used for malicious activities under remote direction without the knowledge of the owner, because the computers were compromised with malware. It is difficult for authorities to identify the spammers responsible for a fraudulent attack. These attacks could therefore be costly.

2.2.3 Security impact

E-mail is used for different purposes, depending on user requirements. The best known example for business use is marketing (Apulu & Latham, 2011). Private use is taking the form of communication, user identification and verification to obtain web services. Most websites require users to sign-up with a valid e-mail address to obtain access to certain services like social media sites and cloud storage. In exchange for the services provided by the website, website owners obtain valuable customer information and are able to stay in touch and provide their clients with relevant information through newsletters. Unfortunately, some website owners abuse this communication channel by bombarding clients with product advertisements. Spam constantly evolve and adapt to new filters. It is even targeted at country-specific activities like the South African Revenue Service's eFiling system. SARS represents the revenue service responsible for collecting tax in South Africa. Citizens are able to complete their tax returns online by using the eFiling system. Spammers take advantage of this annual event by posing as SARS and sending out phishing spam asking registered SARS eFiling users to verify their banking details before SARS can refund them with a generous amount (SARS, 2015). The traditional advertising role of spam is shifting towards malicious activities that include phishing scams and other tasks punishable by law. Even with anti-spam laws in place, spammers are still finding ways to bypass the system, go undetected and show little interest in complying with, for example, the CAN-SPAM Act of 2003 (Yu, 2011; Grimes, 2007) and the European Union Privacy and Electronic Communications Directive (ePrivacy Directive). Embedded code could be injected within e-mail attachments to infect a

victim's computer when opening the attachment. Certain URLs could also redirect users to unknown malicious websites responsible for downloading more malware automatically. Some e-mail messages contain sensitive personal information and private conversations that users want to keep secure. When a user's information is exposed on the world wide web it could become a discouraging task to identify all the network servers and paths the data was routed along due to the network architecture of the internet being so complex and dynamic in nature. This is mainly because the internet keeps on expanding with new network clusters, creating different paths for data to be transmitted by. E-mail users should therefore think twice about what they share with friends, be it via e-mail or a social media portal.

According to Ben-Itzhak (2008), the world wide web can be a dangerous place where cybercrime takes on different forms. It, for example, hosted one of the biggest online black markets, namely the Silk Road, where illegal drugs, products and services were for sale, but was shut down recently (Christin, 2013). Preventing these computer crimes are costly and resource-intensive because cybercrime is done discretely. The Silk Road was only accessible via The Onion Router which is an anonymous network that routes all internet traffic through a volunteering network consisting of multiple relays making it extremely difficult to identify its users and their activities (Jansen, Bauer, Hopper & Dingleline, 2012). The same applies to e-mail spamming. Spammers use different techniques to hide their identity from law enforcement by masking their IP addresses using proxies or virtual private networks. The proxy acts as an intermediary for internet requests while a virtual private network encrypts traffic send over public networks like the internet via virtual tunnelling where a dedicated connection path is established between the host and destination server allowing for increased privacy. Companies like Google must allocate resources (time, money and workforce capacity) to the development of anti-spamming software. The anonymity that can be achieved with these services enable spammers to send spam messages more freely knowing they will not get caught that easily.

Figure 2.2 clearly shows how the three areas (time, finance and security) are closely interconnected. If spam is left unattended, it will escalate and cause problems in the neighbouring areas. In an attempt to reduce spam and prevent unforeseen issues associated with it, researchers, organisations like SPAMHAUS (SPAMHAUS, 2015b) and governments took action against spammers by implementing counteract measures. This was necessary to take back more control of the world wide web and the inbox. The following section explains how spam is being dealt with.

2.3 Spam e-mail counteract measures

A number of spam e-mail counteract measures is available to minimise the amount of junk mail users receive in their inbox. These counteract measures function in different ways and are categorised into either filter or nonfilter solutions. By understanding the structure of an e-mail message the implementation of these prevention techniques can be done more effectively to counteract spam e-mails. An e-mail message is more complex than defining it as a block of text conveying a message. It has a predefined structure that consists of two parts namely the header and body (Mir & Bandy, 2010). Each part contributes to the overall e-mail message design enabling senders to easily construct new messages. Likewise, recipients benefit from this predefined structure because important information relative to each message, like who the sender is and what the e-mail entails, is fairly easy to distinguish. The framework of an e-mail message is depicted in Figure 2.3.

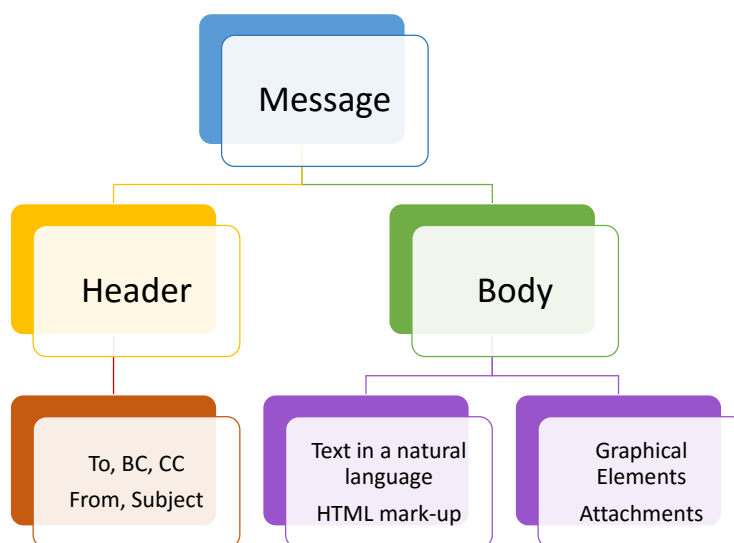


FIGURE 2.3: Framework of an e-mail message (adapted from Blanzieri & Bryl, 2008).

The header contains detailed and specific information about the e-mail message presented by multiple fields. The most common fields are *To*, *From*, *Subject*, *CC* and *BC* although many other properties also exist. Fields like *To* and *From* are self-explanatory while the *CC* field refers to a carbon copy send to additional recipients who should take notice of the e-mail message's contents. The *BC* field also contains recipients, but they are hidden from the other recipients, thus *BC* refers to a blind copy. The *Subject* field contains a short description of the e-mail contents which is sometimes referred to as the body of the message. The message body is the actual content which the sender wants the recipients to read and could include HTML or graphical elements other than text written in a natural language. Other general characteristics of the message, like noncontent features such as the number of

attachments included or overall message size, are also used by counteract measures (Hershkop, 2006). In Figure 2.4 an example e-mail message (middle block) is represented with emphasis on the different components of the e-mail framework (branching from the middle block) that can be used by different counteract measures to help determine if the message is spam or nonspam.



FIGURE 2.4: An example of what can be measured for analysis to determine the type of message (adapted from Blanzieri & Bryl, 2008).

Figure 2.4 shows that a diverse number of message attributes can be used for data analysis when conducting feature extraction. The process of feature extraction depends on the message attributes needed for a specific counteract measure. The most popular message attributes used are the unstructured sets of body tokens (bottom right) and header tokens (top right). Selected fields of the header like IP addresses (top left) are more suitable for prevention techniques which implement an access control list. Nearly all anti-spamming techniques rely on the contents of an e-mail message to determine its class.

When it comes to data mining (analysis of data for meaningful significance), Fawcett (2003) identified some problem areas where spam prevention techniques might struggle to perform optimally. The lack of up-to-date real-world spam corpora causes the problem of insufficient training data which is necessary for the algorithms to make accurate classification choices. Concept drift, where the underlying data distribution changes over time, affects the performance of the classification function as current learning concepts needs to be adjusted on a regular basis in order to stay current with new content generated by reactive spammers. These spammers continuously try to outwit current prevention techniques by adapting their methods using different tactics such as content obscuring to disguise their spam messages as legitimate and safe (Guzella & Caminhas, 2009). The problem related to assigning error cost to messages are also troublesome. All e-mail classification algorithms are prone to find the best trade-off between ham and spam misclassifications. It is more acceptable to compromise on spam classification accuracy because wrongly classifying nonspam messages as spam could result in the loss of valuable information (Androutsopoulos, Koutsias, Chandrinou & Spyropoulos, 2000; Androutsopoulos, Koutsias, Chandrinou, Paliouras & Spyropoulos, 2000). When spam is incorrectly labelled as nonspam it could indicate a less effective solution regarding spam classification. Androutsopoulos, Paliouras & Michelakis (2004) came forth with a suggested solution for determining a reasonable trade-off between these two misclassification error costs which are elaborated on in Section 5.1.5. According to Androutsopoulos *et al.* (2004) it is sensible to imply that the relative cost of these errors are based on user-defined parameters, because users are diverse and have varying demands. What is defined as spam by one person, could be nonspam for another person. SPAMHAUS (2015a) therefore defines spam as not being “about content, it is about consent” although many filters still rely on the contents/attributes of a message (header, subject, body, attachments, graphics) for detection and classification purposes as depicted by Figure 2.4. The next sections explore different counteract measures which include nonfilter approaches and different filter options. In addition, a description of how each one could be implemented is given.

2.3.1 Nonfilter solutions

Nonfilter solutions use methods that implement legislation punishable by law (Moustakas, Ranganathan & Duquenoy, 2005) or methods that result in recurring costs for the sender. Rules and regulations set by ISPs regarding the use of their services eliminate most irrelevant and unnecessary e-mails. Examples include the CAN-SPAM Act of 2003, the ePrivacy Directive (Blanzieri & Bryl, 2008), and ISP policies.

2.3.1.1 CAN-SPAM Act of 2003

The CAN-SPAM Act of 2003 (Controlling the Assault of Nonsolicited Pornography and Marketing Act of 2003) was enacted by the American federal government and signed by President Bush on 16 December 2003 (Lee, 2005). The act took effect since beginning January 2004 and undertakes to effectively manage the spam problem by enforcing hefty penalties and restrictions on the sending of unsolicited commercial e-mails (Clarke, Flaherty & Zugelder, 2005). A set of guidelines are stipulated within the act that companies and salespeople must follow to ensure their commercial e-mails are abiding the law:

- Deceptive headings: Header information like the subject, sender and date fields may not contain misleading or false text. The recipients must not be tricked into opening messages that entail content other than described in the heading field. It should be clear whether the messages are of an advertisement nature or not. The penalty for this violation is about US\$100-\$250 per message.
- Opt-out feature: All commercial e-mail messages must provide the receiver with an unsubscribe mechanism from future messages. This option should be clearly visible and operational for at least 30 days from the time that the messages was first sent. The feature should work correctly and prevent the recipients from receiving any future messages they've opt-out from. This offence varies between US\$25-\$250 per message that does not adhere to or shows the opt-out mechanism.
- Transmission of commercial e-mails after objection: It is considered unlawful if the opt-out request was not honoured within 10 days. Any messages sent after the 10-day period are subject to penalties between US\$25-\$250 per e-mail address.
- Provide a physical mailing address: All commercial e-mails must include the sender's (marketing company or salesperson) valid physical postal address. Omitting the address is also considered unlawful with penalties between US\$25-\$250 per message.

By enforcing these guidelines, e-mail users could benefit from less time spent managing e-mail, gain improved network speeds, be less vulnerable to computer security threats and avoid unnecessary frustration. Spammers are at a disadvantage risking jail time or hefty penalties and expected to see a reduction in their annual revenue generation. Some criticise the act for promoting spam because the guidelines enable spammers to legitimise spam and send spam legally. The act is only applicable to the USA giving spammers the opportunity to send spam from abroad.

2.3.1.2 ePrivacy Directive

Directive 2002/58/EC is a European Union Directive implemented on 31 October 2003 (Lugaresi, 2004). It addresses several regulations concerning the distribution of data and how information must be managed. The privacy of data, confidentiality of digital information, rules that must be followed when cookies (stored web browser information like user credentials, page preferences, online search history, etc.) are used and spamming regulations are all topics of interest. According to the directive the recipient must give foregoing permission before any form of unsolicited commercial communication transpire. The use of e-mail as a marketing tool is otherwise prohibited unless the recipient gives consent via an opt-in (subscribe) agreement.

2.3.1.3 ISP policies

There are many ISPs, each with their own policies. These policies serve as an agreement between the user and service provider regarding the usage of resources and services provided by the ISP. It is accompanied by a fair usage policy and user acceptance policy whereby the user agrees and acknowledges that if he/she abuse any resources or services granted to them which affects the performance of the network or experience of other users in a negative way (like spamming), the ISP has the right to terminate the service of the user in question without notice. This is an effective way of managing spam and other illegal activities on the network that are in contrast with the ISP's policies.

2.3.2 Filter solutions

One of a filter's main functions is to save end users' e-mail-reading time by limiting the number of junk e-mail messages they receive. A spam filter can be represented by the following classifier function:

$$f(e, \theta) = \begin{cases} c_{spam}, & \text{if the e-mail message } e \text{ is considered spam.} \\ c_{ham}, & \text{if the e-mail message } e \text{ is considered not spam.} \end{cases} \quad (2.1)$$

where e is an e-mail message requiring classification, θ presents a vector of various parameters and c_{spam} , c_{ham} are e-mail message labels (Blanzieri & Bryl, 2008). Spam e-mail classification can be regarded as a special case of the text categorisation problem (Meng, Lin & Yu, 2011) where messages are automatically assigned to respective categories. Problems associated with text classification techniques

include the lack of example data and the cost related in categorising unknown data (Kiritchenko & Matwin, 2001).

Spam filter configurations can be implemented at three points during the delivering process. The first being at routers (Agrawal, Kumar & Molle, 2005) where list-based filters are befitting. E-mail spam blocked at this level will not be delivered to the recipient, thus sparing network resources. The next point is at the designated mail server of the recipient's ISP. Here a number of anti-spamming techniques, such as content-based filters in compliance with legislation rules, particular ISP service usage policies, and collaborative filters could be applied. The endpoint is the destination mailbox that belongs to the recipient. Implementing filters at the user side requires more sophisticated solutions like artificial neural networks capable of analysing a user's e-mail preferences and adapting to continuous changes of the spammers by altering its methods. It does not defend against resource abuse as all messages will be delivered nonetheless by labelling each message as either spam or legitimate. Currently, spam filters have proven to be the most effective measure against spam messages (Clark, 2008). According to a study conducted by Siponen & Stucke (2006) about the practicality of various anti-spamming tools and techniques within companies, spam filters are the best method to protect against spam. Goodman *et al.* (2007) stated that machine learning components are present in most spam filtering systems. Spam filtering is therefore expected to maintain an important role as a practical application of machine learning.

Despite the fact that existing filters achieve high accuracy and are improving, the spam problem persists. Spammers are reactive, making each message unique by personalising it with the recipient's information or including random blocks of text which increases the difficulty of filters to identify messages according to a predefined pattern. Presenting an e-mail in graphic format makes it very difficult for content-based filters to analyse, unless more sophisticated filtering techniques like optical character recognition is used to help with the detection process. Another evading technique is to formulate the e-mail message using HTML where various properties for the contents can be set. With continuous attempts by spammers to outsmart spam filters, it is possible for these unwanted messages to reach the recipient's inboxes despite efforts made by the spam filter. When spam e-mails pass through the spam filter and are misclassified as legitimate messages, it could affect the recipients (individuals as well as companies) in several ways as portrayed by Figure 2.2. Depending on the filtering technique, spam filters can be categorised into one of the following groups either for detecting or preventing spam: content-based filters, list-based filters and other filters. Each of these groups is discussed next.

2.3.2.1 Content-based filters

Content-based filters evaluate the authenticity of messages by using words or phrases found in the different messages. Examples include word-based, heuristic (rule-based) and Bayesian filters. With this technique the focus is on spam message detection by evaluating the contents of the message.

- **Word-based filter:** This is one of the most basic content-based filters where various words within the message contribute to classifying the message type. This type of filter can be quite effective, but is prone to false positives where legitimate messages are considered spam. Spammers easily bypass this filter by spelling/misspelling words in a unique manner. Frequent updates to the list of blocked terms are required for this filter to work optimally.
- **Heuristic filter:** This filter works in the same way as the word-based filter, but it also considers phrases when trying to determine the type of message. A ranking system is used where suspicious or unfamiliar words, not regularly found in standard e-mails are assigned a higher rank than typical words. By adding up all the ranking points an overall score is obtained. If the score is higher than a certain threshold as set by the user, the messages is regarded as spam. The opposite is also true and a score below the threshold will be considered nonspam. Heuristic filters are easy to configure but may also contribute to false positives. Spammers are able to bypass this filter as well by determining what word combinations to avoid in their messages using trial and error.
- **Bayesian filter:** Using a mathematical probability in determining whether a message is spam or nonspam, the Bayesian filter is the most advanced content-based filter when compared to word-based filters and heuristic filters. This filter improves over time but needs to be manually trained to effectively identify spam. Two lists are generated to keep words or phrases found in both spam and nonspam messages separate. If more words correspond with the words in the spam list when analysing a message, the message is labelled as spam. If more words are from the nonspam list, the message is labelled as nonspam. This technique works well for a single user to suit his/her content interests, but requires dedicated training sessions before reaching optimal performance.

2.3.2.2 List-based filters

List-based filters use different types of lists in an attempt to stop spam. E-mail senders are grouped as trusted users or spammers which determine if their messages are blocked or allowed past the filter.

Examples of this type of filter include blacklist, whitelist, greylist and real-time black hole list filters.

- **Blacklist filter:** A list containing multiple spam e-mail and IP addresses of the spammers and computers sending spam messages. A blacklist needs to be maintained on a regular basis. When an e-mail is received, the e-mail or IP address is searched on the list. If the list contains an entry, the message is rejected from reaching the recipient. Spammers bypass this filter by using different e-mail or IP addresses not yet added to the blacklist.
- **Whitelist filter:** The whitelist filter allows users to specify from which e-mail or IP addresses of senders they are permitted to receive e-mails from. It performs in the opposite manner as a blacklist which specifies a list of blocked senders. Maintenance is crucial as all messages from senders not on the list will be blocked.
- **Greylist filter:** Relying on the assumption that spammers will only send a batch of spam messages once, a greylist filter adds e-mail or IP addresses to the list of trusted senders if an unknown message was sent a second time. Otherwise messages will be blocked from reaching the recipients' inbox. This technique is not suited for time-sensitive messages as a delay in delivery time will occur.
- **Real-time black hole list filter:** This filter is nearly the same as a blacklist filter with the exception of less maintenance. The list is maintained by third parties saving setup time but offering users less control over what the list contains.

2.3.2.3 Other filters

Other filters include the use of statistical and mathematical models to help distinguish between spam and nonspam messages. Popular examples include collaborative filters, support vector machines, naïve Bayesian classifiers, challenge and response systems, artificial neural networks and ensemble methods.

- **Collaborative filter:** The performance of this filter depends on its community base and is ISP specific. Users are required to flag e-mails they receive into spam or nonspam groups. By doing this they are contributing to a central database that keeps record of e-mail messages sent on the network. The database is used by millions of people worldwide and the data integrity is related to the judgement of its users. This technique is effective in preventing the distribution of mass spam e-mails to other network users if a number of community users come to terms that a message is indeed considered spam. The strength of this filter (its community base of active users) is also its

weakness as a group of spammers could join the community and affect the integrity of the data by flagging spam e-mail messages as legitimate messages.

- Support vector machine: As a learning process influenced by statistical learning theory, support vector machines (SVMs) aim to generalise well to test data (Youn & McLeod, 2007). SVMs are used by different applications including image classification and handwriting recognition. It learns by examples and applies supervised learning to infer class labels from training data making it a nonprobabilistic binary linear classifier. Data points are assigned to one of the two class labels (spam or ham) when considering a SVM as a spam filter.
- Naïve Bayesian classifier: The naïve Bayesian classifier is a probabilistic classifier used in text categorisation problems to determine to which category a document belongs (Sebastiani, 2002). If used for e-mail classification these documents represent e-mail messages. It applies the Bayes' theorem on training data to determine a document's class by considering the frequencies of the training dataset which could be determined by means of a bag-of-words representation (Mitchell, 1997).
- Challenge and response system: This system is capable of blocking most spam e-mails generated and sent via automated systems. Spammers use automated mailing programs to reach a large audience. The completely automated public Turing test to tell computers and humans apart (CAPTCHA) system is an online service available free of charge with the aim of protecting various websites and online services for example e-mail, social media websites, blogs, forums and cloud storage services from getting spammed. The majority of online services use and require e-mail verification (some even implement two-step verification which asks for a generated code or SMS code, known as a one-time pin, to ensure the authenticity of the user) after sign-up. It is important to challenge the service request the account holder initiated to keep services functioning optimal and prevent unauthorised access from an unknown entity. The purpose of the CAPTCHA system regarding spam e-mails is to pose a challenging problem for automated spamming techniques, like bots, that are easily solvable by humans but difficult for computers (Gossweiler, Kamvar & Baluja, 2009).

Examples include optical character recognition (Figures 2.5 and 2.6) following an image-of-fuzzy-text approach. Users are required to type in the characters represented by the distorted image. An improved version of CAPTCHA, called reCAPTCHA was presented by Google to help digitise books while solving the challenge. KittenAuth (Figure 2.7) works on the concept of identifying animals (mostly cats) by selecting them from a group of images (Warner, 2006). Mathematics CAPTCHA (Figure 2.8) requires basic human reasoning to solve a math problem.

Social CAPTCHA (Figure 2.9) is commonly found on social media websites like Facebook. Users must be able to identify their friends from a series of pictures. More examples include Audio CAPTCHA that plays instructions for solving the problem and PlayThru CAPTCHA (Figure 2.10) which is an alternative to the original CAPTCHA concept where users play a game instead of deciphering text and solving mathematical problems. The use of these hard artificial intelligence (AI) problems has contributed to significant advances in the field of AI, as was hoped and believed by Von Ahn, Blum, Hopper & Langford (2003). Due to new developments in the field of AI and improvements in computational power, some of these CAPTCHA techniques have been solved by computer programs. Addressing this issue is an improved version of reCAPTCHA that Shet (2014) refers to as “No CAPTCHA reCAPTCHA” (Figure 2.11). This new reCAPTCHA technique requires users to simply check a box stating “I’m not a robot”. It is driven by an advanced risk analysis system basing its decision on a user’s interaction with the CAPTCHA. If for some reason the risk analysis system can not accurately make a decision, it will use current CAPTCHAs to assist in its reasoning process.



FIGURE 2.5: CAPTCHA (Thoma, 2011).



FIGURE 2.6: reCAPTCHA (Thoma, 2011).

When a challenge and response system is implemented as a spam filter, the sender will receive a response from the server and should proof he/she is not a computer but indeed a human. The test must be completed successfully, and sometimes within a certain time frame, in order for a message to successfully reach its destination, otherwise the message delivery process is



FIGURE 2.7: KittenAuth CAPTCHA (Seopher, 2007).

Qualifying question

Just to prove you are a human, please answer the following math challenge.

Q: Find the least real zero of the polynomial:
 $p(x) = x^2 + 6x + 8$.

A:

mandatory

Note: If you do not know the answer to this question, reload the page and you'll (probably) get another, easier, question.

FIGURE 2.8: Mathematics CAPTCHA (Ruder Bošković Institute, 2008).

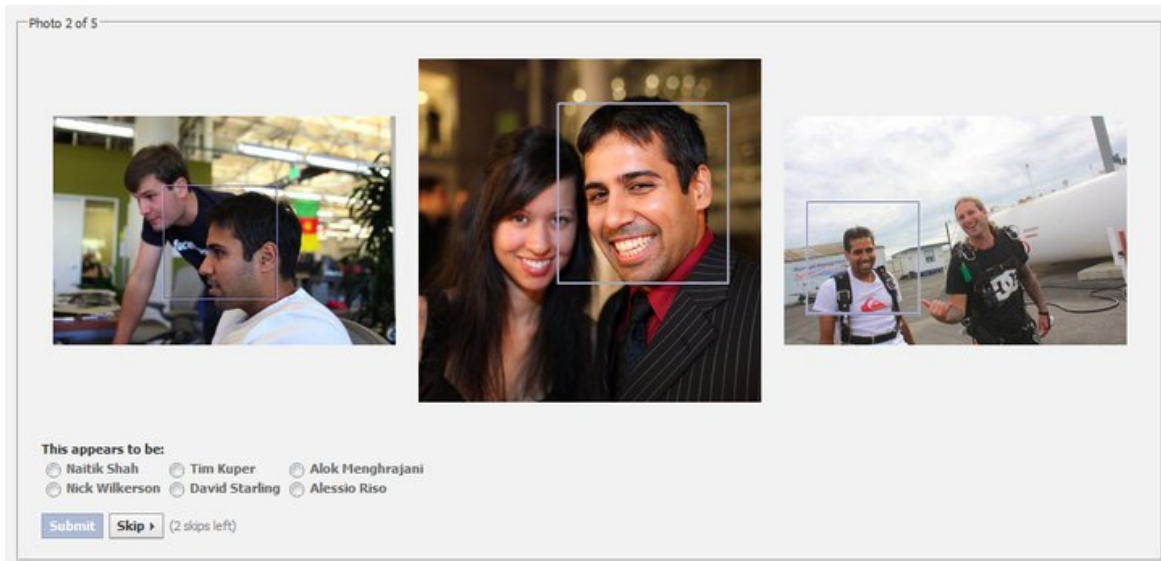


FIGURE 2.9: Social CAPTCHA (Thoma, 2011).

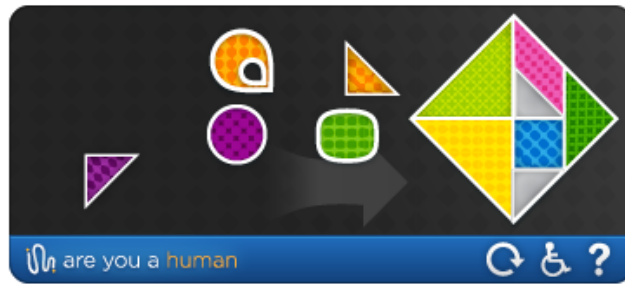


FIGURE 2.10: PlayThru CAPTCHA (Reynen, 2012).

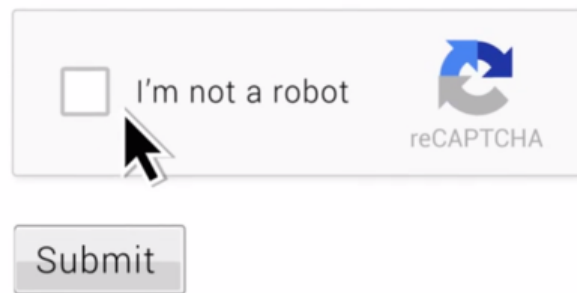


FIGURE 2.11: No CAPTCHA reCAPTCHA (Shet, 2014).

terminated. To overcome frustration experienced by trusted senders, these systems are sometimes accompanied with a whitelist filter to remember human senders. As a result, a sender only needs to pass the test once. The downside of a challenge and response system is the possibility that both sender and recipient may implement such a system resulting in a deadlock where e-mails are never delivered, because the anti-spam systems are fighting against each other i.e. challenging each other for human verification (Alkahtani, Gardner-Stephen & Goodwin, 2011). Legitimate e-mail messages sent from automated systems, like newsletters signed-up for by the user, will be rejected. It may also occur that the sender never completes the challenge due to the degree of difficulty (Figure 2.8) and time allocated for completing the challenge. On the positive side these systems are proactive in limiting the number of auto-generated spam messages one will receive. The CAPTCHA system inherits characteristics of the Turing test based on hard artificial intelligence problems (Von Ahn *et al.*, 2003). For practical purposes these problems should be easily solved by humans in a very short time and present a time consuming and difficult task for computers or bots to figure out. These problems are represented in various formats and demonstrate the comprehensiveness of spamming techniques, thus it should be accessible to most users including those who are either visually or hearing impaired.

- Artificial neural network: Artificial neural networks, which is the focus of this study, and more specifically Generalized additive neural networks (GANNs) (Du Toit, 2006) which is a special

type of neural network, is discussed in Chapter 4. This technique is based on the functioning of the human brain. Being good at pattern recognition and fitting nonlinear functions (Zhang, Patuwo & Hu, 1998) this technique is worth considering as a spam e-mail classifier, because it should be able to identify properties that distinguish nonspam messages from spam messages. In this study this special type of artificial neural network is used to filter spam e-mails. Since none of the aforementioned anti-spam methods are able to eradicate spam on its own the combination of different anti-spamming techniques may provide improved detection capabilities. This is achieved with ensemble methods.

- Ensemble method: The filter solutions that were mentioned can be used in combination with each other. When multiple classifiers are used alongside another it is referred to as an ensemble (Kuncheva, 2014). By applying more than one classifier to the spam e-mail problem the expected outcomes may be an increase in spam detection accuracy, but may also result in a less favourable outcome than those of the single optimised classifiers. When using too many of the various individual classifiers alongside one another, for example combining a Bayesian filter with a whitelist filter and challenge and response system, the filter could become too aggressive and lose its effectiveness by either blocking too much or too few messages. It is therefore important to have a balanced filter that is not too strict or too flexible, because many spam prevention systems if implemented incorrectly, could result in the loss of important information and annoy e-mail users. Filters all have the same function, but achieve the objective in different ways. The Bootstrap aggregating (Bagging) (Breiman, 1996) and Boosting (Freund & Schapire, 1996) ensemble algorithms are used in this study to determine the usefulness of ensemble methods on the results obtained by the GANN classifier and are discussed in Chapter 4.

In the next section the current state of the spam problem, considering the last six years, is investigated. The progress of finding a solution for the spam problem which uses the counteract measures previously discussed is illustrated.

2.4 Current state of the spam problem

According to Kaspersky Lab statistics (Vergelis *et al.*, 2016; Vergelis, Shcherbakova & Demidova, 2015; Gudkova, 2014; Gudkova, 2013) as shown in Table 2.1, the fraction of overall worldwide spam e-mail that is part of all e-mail network traffic is slowly decreasing. The last column in Table 2.1 represents the decrease of spam in global e-mail traffic compared to the previous year. The table illustrates that

the overall increased level of anti-spam protection applied throughout the last five years is indeed paying off.

Year	Spam in global e-mail traffic	Decrease of spam in global e-mail traffic
2010	82.20%	3.00%
2011	80.30%	1.90%
2012	72.10%	8.20%
2013	69.60%	2.50%
2014	66.76%	2.80%
2015	55.28%	11.48%

TABLE 2.1: Yearly decrease of spam in global e-mail traffic.

The reduction in spam e-mail shown in Table 2.1 indicates progress towards a solution. Over the past six years, spam is down from 85.20% to 55.28%. Unfortunately, more than 55.00% of all e-mail network traffic is still spam. Overall, progress in the right direction is made with worldwide spam for each of the last six years being less than the previous year. This is an indication that people are becoming more aware and willing to eradicate the issue at hand. Even though the spam e-mail problem will not disappear overnight, researchers, communities and governments are trying various methods to stop the billions of spam e-mails received each year. The profitable spamming business will continue and so will prevention techniques keep on improving until the spam problem hopefully has been solved. In the next section this chapter is concluded.

2.5 Conclusions

In this chapter spam was defined in the context of e-mail messages as being unsolicited bulk e-mails. Background information regarding the origin of spam was given to help understand the context of the spam problem. The motivation behind spamming was examined and it was concluded that revenue generation, promoting of products or services and stealing of sensitive information by means of phishing techniques are the success drivers behind spamming (Hayati & Potdar, 2008). Three externalities (time, finance and security) influenced by spamming were discussed with emphasis on how they are interconnected. Next, different anti-spamming techniques were investigated. Filtering and nonfiltering solutions were mentioned and explained. It was found that filtering options currently provide the most successful solutions. From the different solutions available it is evident that currently no single

filter is capable of eradicating spam e-mails. Combinations of current methods provide a good overall filter with high detection rates, but currently no solution exist. To better address the ever-changing spamming techniques used by spammers, a dynamic automated spam filter is required. In this study a filter that uses an artificial neural network capable of both identifying changes in spammers' patterns and adapting to change is considered. In Chapter 4 an automated construction algorithm for a GANN that complies with these requirements is examined. The solution to be presented attempts to mitigate the unresolved spam problem e-mail users are still facing. The next chapter elaborates on artificial neural networks to provide background knowledge on how the GANN is constructed.

CHAPTER 3

ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANNs), also referred to as neural networks, are data-driven, rather flexible self-adaptive models (Zhang *et al.*, 1998), and generally constitutes a nonlinear function (Zhang, 2010). The flexibility of a neural network to perform data analysis tasks, learn from experience, and generalise, make it a popular solution in various fields of study. ANNs are well-known for its prediction, pattern recognition and classification capabilities (Zhang *et al.*, 1998). Implemented successfully in areas like robotics, automotive, banking, oil and gas (Hagan, Demuth, Beale & De Jesús, 2014), neural networks are capable of addressing problems for a variety of disciplines. This emphasises the adaptive characteristics of these networks as they are not bound to a specific problem domain. In this study, Generalized additive neural networks (GANNs) are applied to the spam problem to determine if it can accurately classify spam e-mails. What constitutes spam differs for each e-mail user and depends in part on their subject of interest. A GANN, which is a supervised machine learning technique, may be flexible enough to model these patterns. This flexibility is desirable as spammers regularly challenge spam filters by changing their spamming behaviour.

In this chapter a brief history on ANNs and how it mimics a biological human brain on a high level is presented in Section 3.1. In Section 3.2 a simple ANN is considered. The mathematical model for both the single-input and multiple-input neuron is described followed by a discussion on a layer of neurons. In Section 3.3 the more complex Multilayer Perceptron (MLP) neural network, the most common neural network which forms the basis for a GANN, is presented. A learning technique for MLPs, known as the Backpropagation algorithm, is also discussed since this technique overcame some of the theoretical pitfalls of basic ANNs. A conclusion of this chapter is given in Section 3.4.

3.1 History

In the early 1940s, research about certain brain theories were conducted by Warren McCulloch, a neurophysiologist, and his collaborator, Walter Pitts who was a logician (McCulloch & Pitts, 1943). They conducted a study on neural activity by examining the operation of biological neurons. From their research a model of artificial neurons was proposed that still stands as a theory on how the biological brain performs cognitive functions. The proposed model is based on the human brain where linked neurons form a network. They suggested that a neuron could be in one of two states triggered by an activation switch. The activation switch was either on or off in response to stimulation of the adjacent neurons. These neurons were based on the composition of the human brain which comprises a neural network that has about 10^{11} interconnected neurons each consisting of a cell body, axon, dendrites and synapses (the connection between multiple dendrites) (Hagan *et al.*, 2014; Russell & Norvig, 2010). An illustrated example of a biological neuron is given in Figure 3.1. Electrochemical reactions are responsible for signal transfers between the connected neurons and control brain activity in charge of cognition processes such as thought and knowledge advancements (Russell & Norvig, 2010).

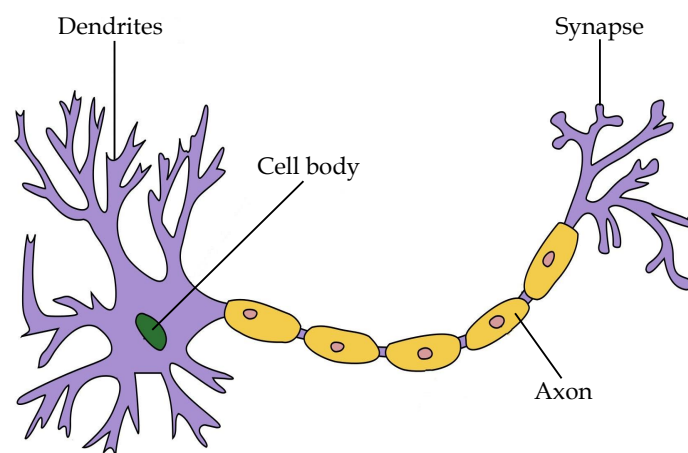


FIGURE 3.1: Biological neuron (adapted from Jarosz, 2009).

In the composition of the biological neuron shown in Figure 3.1, each part has a specific function to perform. The dendrites transfer signals to the cell body which sends out a new signal to neighbouring neurons via the axon once it reaches a certain threshold. The synapse (connection between axons and dendrites joining the neurons) is responsible for the electrochemical reaction causing a potential difference at the cell body (Negnevitsky, 2005). As stated by Hagan *et al.* (2014), the arrangement of neurons and the strength of each neuron connection driven by complex chemical reactions, establish the function of the neural network.

In 1947 McCulloch and Pitts proposed a way to use neural networks for recognising visual inputs (Pitts & McCulloch, 1988). With their research they proved that with a fixed network of artificial neurons, a Turing machine (Turing, 1936) could be simulated. This machine is an abstract device which can be used to emulate the logic of any computer algorithm. It hypothetically represents a computer that uses a set of rules to manipulate symbols. This realisation encouraged the application of the ANN models for learning generalisation rules, as well as using the models for sensory perception in pattern recognition and classification tasks. Thus, the manner in which the human brain processes information and recognises patterns have contributed to the development of ANNs.

Research conducted on ANNs between the 1940s and 1960s were mostly based on mathematical models. It was not until Rosenblatt (1958) introduced the Perceptron model that practical neural network implementations commenced. The Perceptron model was based on the neuron proposed by McCulloch & Pitts (1943). It is an artificial neuron that would classify given inputs into one of two distinct classes. The approach used by the Perceptron is based on pattern recognition to decide the class type. This is achieved with the Perceptron learning rule which was also introduced by Rosenblatt (1962). This learning rule is still being used by modern neural networks as a machine learning technique for predictive analysis (Kufandirimbwa & Gotor, 2012).

During the 1960s, advancements in ANNs were primarily focused on solving various classification problems. This resulted in the construction of numerous neural network simulations in computers. It addressed a vast collection of pattern recognition problems which ranged from speech and handwritten text classification tasks to recognising visual shapes. The training process involved adjusting the weights (the strength of connections between two nodes) according to a learning algorithm. The process was repeated until the neural network produced the correct output pattern based on the given inputs. When training was completed, the ANN would be capable of classifying other examples it was presented with. However, the accuracy of the classifier was dependent in part on the total training time and the diversity of samples used.

Research on ANNs progressed very slowly during the 1970s due to criticism on the training capabilities of the Perceptron model. Minsky & Papert (1969), with Minsky a student of McCulloch, stated that a single-layer Perceptron neural network could only solve linearly separable problems. According to Hagan *et al.* (2014), this statement, the lack of institutional research funding and the need for more powerful computers during that time resulted in less research being conducted on ANNs.

In the 1980s new computer advancements were introduced along with the backward propagation of errors technique (Rumelhart & McClelland, 1986). The Backpropagation algorithm made it possible to train ANNs consisting of multiple layers of Perceptrons which could then solve nonlinearly separable

problems. These ANNs are referred to as Multilayer Perceptrons (MLPs). The distribution of published results by Rumelhart & McClelland (1986), gradually promoted interest among researchers to further investigate ANNs. Before elaborating on MLPs and its training process, the neuron model which forms the structure of an ANN is considered in the next section. Note that Sections 3.2 and 3.3 were obtained from Hagan *et al.* (2014).

3.2 The Neuron model

The first artificial neuron (Hagan *et al.*, 2014) under discussion is the single-input neuron. A more advanced extension of this artificial neuron, the multiple-input neuron, is then considered. After that, Rosenblatt's Perceptron model which represents a basic ANN, is considered followed by a discussion on neuron layers.

3.2.1 The single-input neuron

The single-input neuron, depicted by Figure 3.2, has only one scalar input p . The transmission strength (weighted input) is determined by the product wp , where w denotes the weight. The activation (transfer) function produces the final output a . This function receives the net input n as argument, which is the sum of the weighted input wp and the scalar bias b . The latter has a constant input value of 1.0. The scalar parameters w and b are adjusted while training is performed to allow the network to adapt to more accurate output.

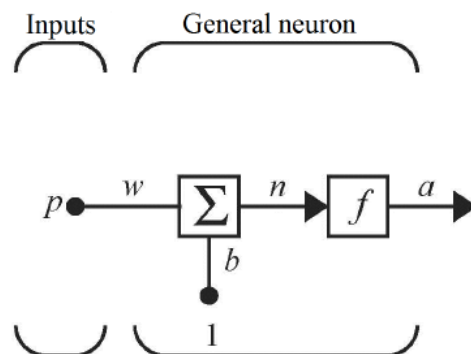


FIGURE 3.2: Single-input neuron (Hagan *et al.*, 2014).

As an example of calculating the final output a with input $p = 2$, consider the parameter values $w = 4$ and $b = 1.5$. The output a is then calculated as follows:

$$a = f(n) = f(wp + b) = f(4(2) + 1.5) = f(9.5), \quad (3.1)$$

where f denotes the activation function. The brain is without doubt more complex than an ANN, but they do share some similarities. Two analogies of networks of artificial neurons and biological neurons are as follows: both networks have constituent parts that are highly interconnected and the network function is subject to the connections between neurons (Hagan *et al.*, 2014). In most real-world problems multiple-input neurons are used instead of just one. This type of neuron is discussed next.

3.2.2 The multiple-input neuron

The multiple-input neuron is an extension of the single-input neuron. An example of the multiple-input neuron is depicted in Figure 3.3.

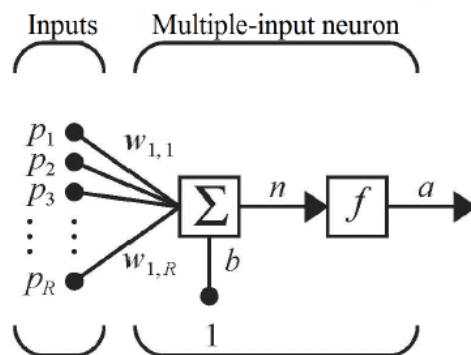


FIGURE 3.3: Multiple-input neuron (Hagan *et al.*, 2014).

For each of the individual inputs there exists a corresponding weight. Let p_1, p_2, \dots, p_R represent the different inputs and $w_{1,1}, w_{1,2}, \dots, w_{1,R}$ the corresponding weights from the weight matrix \mathbf{W} discussed in Section 3.2.4. As in the case of the single-input neuron, the activation function f receives the net input n as parameter and produces the final output a . The net input n is defined as:

$$n = \mathbf{W}\mathbf{p} + b = [(w_{1,1})(p_1) + (w_{1,2})(p_2) + \dots + (w_{1,R})(p_R)] + b. \quad (3.2)$$

The final output of the multiple-input neuron can be written as

$$a = f(n) = f(\mathbf{W}\mathbf{p} + b). \quad (3.3)$$

If, for example, there were three inputs p_1, p_2 and p_3 with associated weights $w_{1,1}, w_{1,2}$ and $w_{1,3}$ such that $p_1 = 2, w_{1,1} = 4; p_2 = 5, w_{1,2} = 3; p_3 = 1, w_{1,3} = 3$ and $b = -2.5$, then the final output can be calculated as follows:

$$a = f[(4(2) + 3(5) + 3(1)) - 2.5] = f(23.5). \quad (3.4)$$

In the next section, the architecture of a Perceptron with a hard-limit activation function is taken as an example of a multiple-input neuron.

3.2.3 The Perceptron

A Perceptron constitutes a simple ANN with an architecture that includes a single-layer (Section 3.2.4) using the hard-limit transfer function (Hagan *et al.*, 2014). This function is defined as

$$a = \text{hardlim}(n) \begin{cases} 1 & \text{if } n \geq 0 \\ 0 & \text{otherwise,} \end{cases} \quad (3.5)$$

where n is the net input. The function can be denoted graphically as in Figure 3.4.

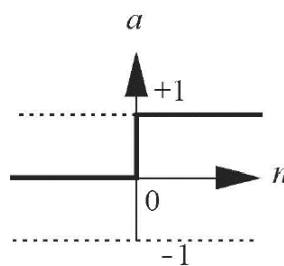
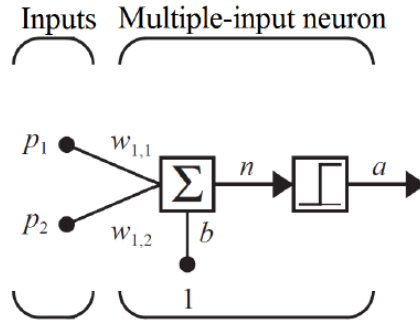


FIGURE 3.4: Hard-limit activation function (Hagan *et al.*, 2014).

The symbol used to represent the hard-limit activation function graphically is shown in Figure 3.5. In order to illustrate a classification example for two categories, consider Rosenblatt's (1958) Perceptron shown in Figure 3.6.



FIGURE 3.5: Hard-limit activation function symbol (Hagan *et al.*, 2014).

FIGURE 3.6: The Perceptron (Hagan *et al.*, 2014).

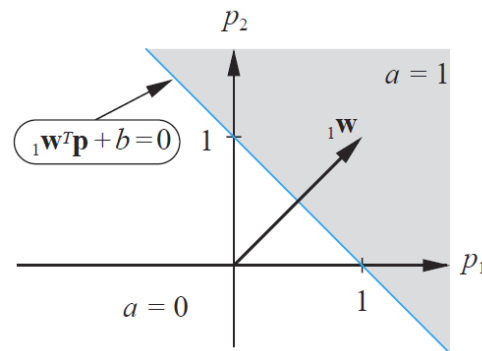
The Perceptron from Figure 3.6 has two inputs, namely p_1 and p_2 . The weight matrix \mathbf{W} , which consists of a single row vector, contains the respective weights $w_{1,1}$ and $w_{1,2}$ for each of the inputs. The subscript of ${}_1\mathbf{w}^T$ refers to the row of \mathbf{W} . The output for the two-input single-output Perceptron is determined by:

$$\begin{aligned} a &= \text{hardlim}(n) = \text{hardlim}(\mathbf{W}\mathbf{p} + b) = \text{hardlim}({}_1\mathbf{w}^T\mathbf{p} + b) \\ &= \text{hardlim}[(w_{1,1})(p_1) + (w_{1,2})(p_2)] + b. \end{aligned} \quad (3.6)$$

Using the input vectors, the decision boundary is placed where the net input n is zero:

$$n = {}_1\mathbf{w}^T\mathbf{p} + b = [(w_{1,1})(p_1) + (w_{1,2})(p_2)] + b = 0. \quad (3.7)$$

Let $w_{1,1} = 1$, $w_{1,2} = 1$ and $b = -1$. The decision boundary for the two-input single-output Perceptron will be able to solve a linearly separable problem where each side of the dividing line represents an output class a with $a \in \{0, 1\}$ as illustrated in Figure 3.7.

FIGURE 3.7: Decision boundary for a two-input single-output Perceptron (Hagan *et al.*, 2014).

Using the above-mentioned values, the decision boundary can be calculated as follows:

$$n = {}_1\mathbf{w}^T\mathbf{p} + b = [(w_{1,1})(p_1) + (w_{1,2})(p_2)] + b = p_1 + p_2 - 1 = 0. \quad (3.8)$$

The boundary line for the input space in Figure 3.7 is derived from (3.8) by calculating the intercepts on the axes. These intercepts are determined by setting one of the inputs equal to zero, while determining the other. In the example, the intercepts on the axes for p_1 and p_2 are both 1 and calculated as follows:

$$\text{If } p_2 = 0, \quad p_1 = -\frac{b}{w_{1,1}} = -\frac{-1}{1} = 1. \quad (3.9)$$

$$\text{If } p_1 = 0, \quad p_2 = -\frac{b}{w_{1,2}} = -\frac{-1}{1} = 1. \quad (3.10)$$

Next, input values are tested for determining the side of the boundary that corresponds to an output of 1. For example, if the input $\mathbf{p} = \begin{bmatrix} 1.5 & 4 \end{bmatrix}^T$, the output of the network is:

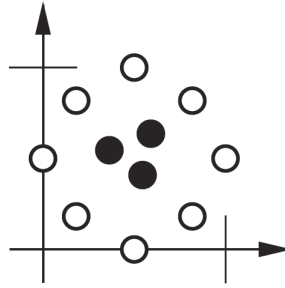
$$a = \text{hardlim}({}_1\mathbf{w}^T \mathbf{p} + b) = \text{hardlim}\left(\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1.5 \\ 4 \end{bmatrix} - 1\right) = \text{hardlim}(4.5) = 1. \quad (3.11)$$

As a result, the shaded region above the boundary line of Figure 3.7 represents all network outputs equal to 1 for input vectors of that region. The area below the boundary line represents all network outputs of 0 for input vectors found in the unshaded area. Let input $\mathbf{p} = \begin{bmatrix} 0.5 & -1 \end{bmatrix}^T$, then the output will be:

$$a = \text{hardlim}({}_1\mathbf{w}^T \mathbf{p} + b) = \text{hardlim}\left(\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 0.5 \\ -1 \end{bmatrix} - 1\right) = \text{hardlim}(-1.5) = 0. \quad (3.12)$$

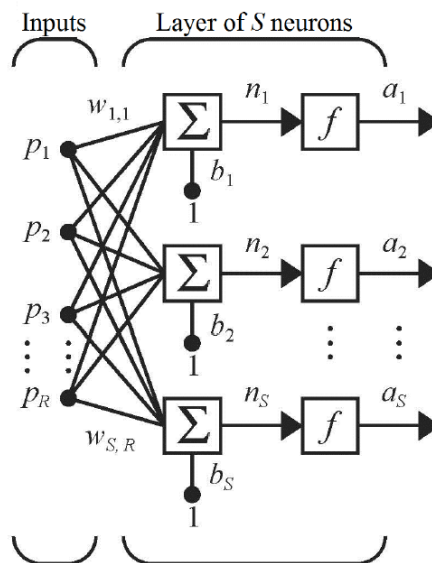
The weight vector ${}_1\mathbf{w}$ is always orthogonal to the decision boundary pointing in the direction where the output is 1. As stated by Minsky & Papert (1969), the single-layer Perceptron is capable of only solving linearly separable problems where the input space can be divided into two distinct classes. A linearly inseparable problem is shown in Figure 3.8. The black circles (\bullet) indicate an output class of 1, while a clear circle (\circ) indicates an output class of 0. Determining the decision boundary which separates the input space for these classes is not feasible with a Perceptron.

To address more complex problems like the linearly inseparable problem shown in Figure 3.8 and since using only a single neuron could produce insufficient results, the concept of a layer of neurons is discussed next. The Multilayer Perceptron capable of solving arbitrary classification problems is considered thereafter.

FIGURE 3.8: Linearly inseparable problem (Hagan *et al.*, 2014).

3.2.4 A layer of neurons

A neural network could either contain a single-layer of neurons (Figure 3.9) or form a multilayer network when multiple neurons are joined together.

FIGURE 3.9: A single-layer of neurons (Hagan *et al.*, 2014).

In the single-layer of neurons depicted by Figure 3.9, there are S neurons and R inputs. Note that the inputs do not represent a separate layer based on the view of Hagan *et al.* (2014). It is not uncommon for the number of inputs to differ from the number of neurons, i.e., $R \neq S$. All the input elements of vector \mathbf{p} are connected to every neuron through the weight matrix \mathbf{W} . Let p_1, p_2, \dots, p_R represent the different inputs; $w_{1,1}, w_{1,2}, \dots, w_{1,R}$ the corresponding weights from the weight matrix \mathbf{W} ; b_1, b_2, \dots, b_S the biases of the related neurons, n_1, n_2, \dots, n_S the net inputs and a_1, a_2, \dots, a_S the outputs of the activation functions. Hence, the layers are formed by the following: the weight matrix \mathbf{W} , the bias vector \mathbf{b} , summations of the weighted inputs and the corresponding biases which provides the net input (defined in 3.2) to the different activation functions f , and an output vector \mathbf{a} . The different neurons

could each have a unique activation function generating part of the final output from the same input values. According to Hagan *et al.* (2014), this is achievable with a single (composite) layer of neurons where two networks from Figure 3.9 are combined in parallel. Each layer has its own weight matrix \mathbf{W} . The weight matrix of a network is the entry point through which the input elements of vector \mathbf{p} enters the network and can be defined as:

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}, \quad (3.13)$$

where $w_{i,j}$ denotes the weight to target neuron i from source neuron j . Thus, $w_{1,3}$ represents a connection to neuron 1 from the input neuron 3. To obtain all vector elements from \mathbf{W} for a specific neuron i , represented by a single horizontal row in the weight matrix, the following is defined:

$${}_i\mathbf{w} = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,R} \end{bmatrix}. \quad (3.14)$$

Consequently, the weight matrix can also be represented as:

$$\mathbf{W} = \begin{bmatrix} {}_1\mathbf{w}^T \\ {}_2\mathbf{w}^T \\ \vdots \\ {}_S\mathbf{w}^T \end{bmatrix}, \quad (3.15)$$

where ${}_i\mathbf{w}^T$ is the transpose of ${}_i\mathbf{w}$. The i^{th} element of the output vector can be expressed as:

$$a_i = f(n_i) = f([{}_i\mathbf{w}^T(\mathbf{p})] + b_i), \quad (3.16)$$

where n_i is the net input of the input vector \mathbf{p} , weight vector ${}_i\mathbf{w}^T$, and the bias b_i . In the next section the Multilayer Perceptron neural network, which is more robust than a single-layer ANN is discussed. The Multilayer Perceptron is also more powerful than a single-layer neural network and forms part of the GANN architecture discussed in Chapter 4.

3.3 The Multilayer Perceptron

The Multilayer Perceptron (MLP) is the most common artificial neural network, capable of pattern recognition and used for supervised prediction (Potts, 1999; Zhang *et al.*, 1998). It can identify trends or anomalies within a data set making it one of the most popular predictive data analysis techniques. An MLP is the preferred model for many problems concerning optimisation, classification and simplification of complex models (Berry & Linoff, 1997). The topology of an MLP consists of two or more distinct layers as shown in Figure 3.10 (Hagan *et al.*, 2014). As with the single-layer network (Figure 3.9) the inputs are not considered a layer on its own.

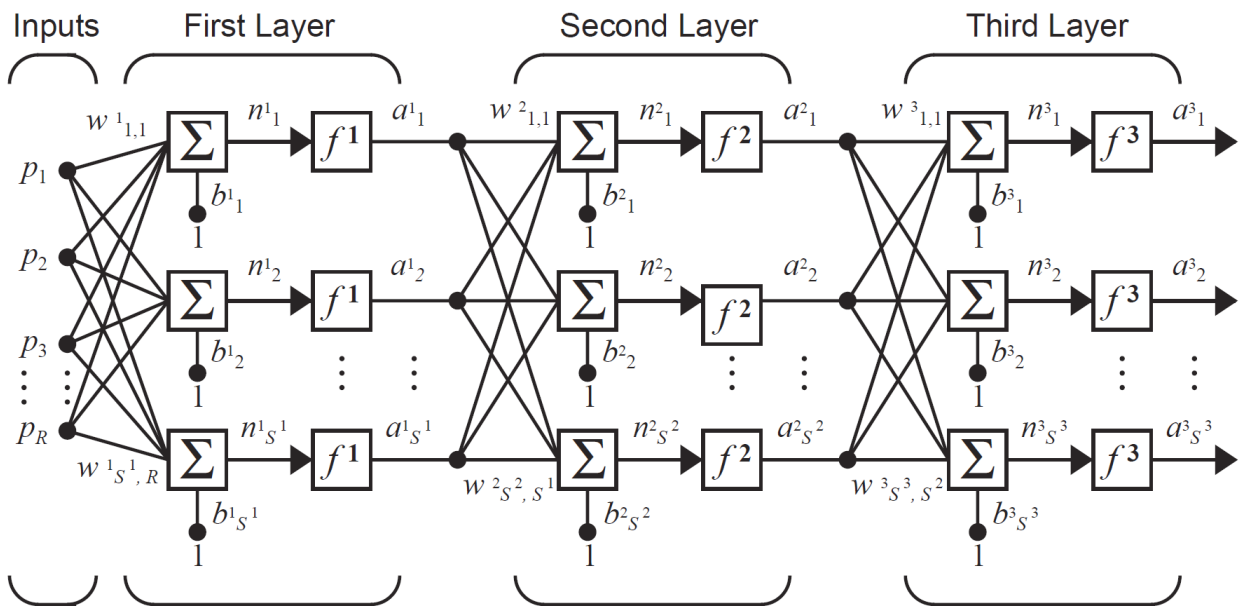


FIGURE 3.10: The Multilayer Perceptron neural network architecture (Hagan *et al.*, 2014).

The MLP neural network of Figure 3.10, has R inputs, namely p_1, p_2, \dots, p_R . For each layer a superscript is appended to the variables which presents the number of the corresponding layer. The first layer has S^1 neurons, while the number of neurons may vary for each layer. Every layer also has a weight matrix, where, e.g., \mathbf{W}^2 is the weight matrix for the second layer. In Figure 3.10, the two hidden layers are represented as First layer and Second layer respectively. The third layer produces the network output and is called the output layer. The last layer producing the final network output will always represent the output layer. The layers are highly interconnected in a feed-forward manner: the input data is passed through the network layer-by-layer (left-to-right) to the output layer. A network with M layers will give the following output:

$$\mathbf{a} = \mathbf{a}^M. \quad (3.17)$$

Weights are initially assigned random values by the neural network and adjusted to minimise misclassification of the training data set during supervised training. While training commences, the external inputs are passed to the first hidden layer:

$$\mathbf{a}^0 = \mathbf{p}. \quad (3.18)$$

An activation function denoted as \mathbf{f}^1 defines the output \mathbf{a}^1 . The type of activation function may differ for each neuron. If for instance the hard-limit activation function (3.5) is used, the MLP is useful for binary classification schemes, thus suitable to classify an input pattern into one of two groups like spam or nonspam. The final output for the above mentioned MLP neural network is \mathbf{a}^3 and can be defined as:

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{W}^3 \mathbf{f}^2(\mathbf{W}^2 \mathbf{f}^1(\mathbf{W}^1 \mathbf{p} + \mathbf{b}^1) + \mathbf{b}^2) + \mathbf{b}^3) = \mathbf{f}^3(\mathbf{W}^3 \mathbf{a}^2 + \mathbf{b}^3). \quad (3.19)$$

The manner in which the data signal is progressed through the network can be defined as:

$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1} \mathbf{a}^m + \mathbf{b}^{m+1}) \text{ for } m = 0, 1, \dots, M - 1, \quad (3.20)$$

where M is the total number of layers. Neural networks consisting of multiple layers and neurons could learn complex classification models as long as enough training examples exist and sufficient training time is given. With the Backpropagation algorithm, the weights are adjusted during learning to map inputs to the desired output by minimising the mean squared error (MSE). This algorithm is discussed next.

3.3.1 The Backpropagation algorithm

The Backpropagation algorithm is a learning technique used for Multilayer Perceptron neural network training (Hagan *et al.*, 2014). Given a data set which contains both input patterns and the known target output values, the weights are optimised to map the inputs to the outputs. Let D represent a data set containing $\{\mathbf{p}_q, \mathbf{t}_q\}$ entries, where \mathbf{p}_q denotes the inputs and \mathbf{t}_q is the target value for the corresponding \mathbf{p}_q input:

$$D = \{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \dots, \{\mathbf{p}_Q, \mathbf{t}_Q\}. \quad (3.21)$$

By adjusting the weights, the MSE could be minimised for more accurate classification results. The

MSE is used with Backpropagation training to determine the difference between network output data and the target output data. A network is considered to have converged if the MSE is satisfactorily small. This enables the neural network to generalise well and not overfit the model to the training data. If the data was overfitted, then the classification accuracy of new data will be inadequate, since the neural network memorised the data (Hagan *et al.*, 2014; Russell & Norvig, 2010). The MSE for a network which has only one output value is determined by:

$$F(\mathbf{x}) = E[e^2] = E[(t - a)^2], \quad (3.22)$$

where \mathbf{x} represents a vector of network weights and biases, and e is the difference between the desired target output t and the network output a . For networks with multiple output values (3.22) can further be generalised as follows:

$$F(\mathbf{x}) = E[\mathbf{e}^T \mathbf{e}] = E[(\mathbf{t} - \mathbf{a})^T (\mathbf{t} - \mathbf{a})]. \quad (3.23)$$

Hence the MSE can be determined by:

$$\hat{F}(\mathbf{x}) = (\mathbf{t}(k) - \mathbf{a}(k))^T (\mathbf{t}(k) - \mathbf{a}(k)) = \mathbf{e}^T(k) \mathbf{e}(k), \quad (3.24)$$

where k denotes the iteration number. In (3.24) a squared error replaces the expectation of the squared error. The steepest descent algorithm used to approximate the MSE is:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha \frac{\partial \hat{F}}{\partial w_{i,j}^m}, \quad (3.25)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial \hat{F}}{\partial b_i^m}, \quad (3.26)$$

where α denotes the learning rate of the network. Next, the partial derivatives must be calculated with the Chain rule of calculus since the error is an indirect function of the hidden layer weights.

3.3.1.1 The Chain rule

In order to demonstrate the Chain rule, consider an explicit function f of the single variable n . Calculating the derivative of f with respect to a third variable w , the Chain rule is defined as:

$$\frac{df(n(w))}{dw} = \frac{df(n)}{dn} \times \frac{dn(w)}{dw}. \quad (3.27)$$

As an example let $f(n) = e^n$ and $n = 2w$, so that $f(n(w)) = e^{2w}$ then:

$$\frac{df(n(w))}{dw} = \frac{df(n)}{dn} \times \frac{dn(w)}{dw} = (e^n)(2). \quad (3.28)$$

The Chain rule (3.27) will be used to determine the derivatives in (3.25) and (3.26):

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial w_{i,j}^m}, \quad (3.29)$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = \frac{\partial \hat{F}}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial b_i^m}. \quad (3.30)$$

Since the net input to layer m is an explicit function of the weights and bias in that layer, the second term for (3.29) and (3.30) can respectively be computed as follows:

$$n_i^m = \sum_{j=1}^{S^{m-1}} w_{i,j}^m a_j^{m-1} + b_i^m. \quad (3.31)$$

Thus:

$$\frac{\partial n_i^m}{\partial w_{i,j}^m} = a_j^{m-1}, \quad \frac{\partial n_i^m}{\partial b_i^m} = 1. \quad (3.32)$$

Next, if the sensitivity of \hat{F} to changes made in the i^{th} element of the net input at layer m is defined as:

$$s_i^m \equiv \frac{\partial \hat{F}}{\partial n_i^m}, \quad (3.33)$$

then (3.29) and (3.30) can be simplified to

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = s_i^m a_j^{m-1} \quad \text{and} \quad (3.34)$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = s_i^m. \quad (3.35)$$

This allows the steepest descent algorithm, (3.25) and (3.26), to be written as:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha s_i^m a_j^{m-1} \quad \text{and} \quad (3.36)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha s_i^m. \quad (3.37)$$

If (3.36) and (3.37) were to be expressed in matrix form, it becomes:

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m (\mathbf{a}^{m-1})^T \quad \text{and} \quad (3.38)$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha \mathbf{s}^m, \quad (3.39)$$

where

$$\mathbf{s}^m \equiv \frac{\partial \hat{F}}{\partial \mathbf{n}^m} = \begin{bmatrix} \frac{\partial \hat{F}}{\partial n_1^m} \\ \frac{\partial \hat{F}}{\partial n_2^m} \\ \vdots \\ \frac{\partial \hat{F}}{\partial n_{s^m}^m} \end{bmatrix}. \quad (3.40)$$

The next step in the Backpropagation algorithm involves the computation of the sensitivities \mathbf{s}^m .

3.3.1.2 Backpropagating the sensitivities

Backpropagating the sensitivities describes the recurrence relationship of sensitivities and the manner in which they are calculated. The sensitivity at layer m is determined from the sensitivity at layer $m+1$. The Jacobian matrix (3.41) is used to derive the recurrence relationship for the sensitivities:

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} \equiv \begin{bmatrix} \frac{\partial n_1^{m+1}}{\partial n_1^m} & \frac{\partial n_1^{m+1}}{\partial n_2^m} & \cdots & \frac{\partial n_1^{m+1}}{\partial n_{S^m}^m} \\ \frac{\partial n_2^{m+1}}{\partial n_1^m} & \frac{\partial n_2^{m+1}}{\partial n_2^m} & \cdots & \frac{\partial n_2^{m+1}}{\partial n_{S^m}^m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_1^m} & \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_2^m} & \cdots & \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_{S^m}^m} \end{bmatrix}. \quad (3.41)$$

Next, an expression for this matrix must be found. Consider the i, j element of the matrix:

$$\begin{aligned} \frac{\partial n_i^{m+1}}{\partial n_j^m} &= \frac{\partial \left(\sum_{l=1}^{S^m} w_{i,l}^{m+1} a_l^m + b_i^{m+1} \right)}{\partial n_j^m} = w_{i,j}^{m+1} \frac{\partial a_j^m}{\partial n_j^m} \\ &= w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} = w_{i,j}^{m+1} f^m(n_j^m), \end{aligned} \quad (3.42)$$

where

$$f^m(n_j^m) = \frac{\partial f^m(n_j^m)}{\partial n_j^m}. \quad (3.43)$$

Thus, the Jacobian matrix can be written as:

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} = \mathbf{W}^{m+1} \dot{\mathbf{F}}^m(\mathbf{n}^m), \quad (3.44)$$

where

$$\dot{\mathbf{F}}^m(\mathbf{n}^m) = \begin{bmatrix} f^m(n_1^m) & 0 & \cdots & 0 \\ 0 & f^m(n_2^m) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & f^m(n_{S^m}^m) \end{bmatrix}. \quad (3.45)$$

Now, using the Chain rule (Section 3.3.1.1) in matrix form, the recurrence relation for the sensitivity can be presented as:

$$\begin{aligned} \mathbf{s}^m &= \frac{\partial \hat{F}}{\partial \mathbf{n}^m} = \left(\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} \right)^T \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}} = \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}} \\ &= \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}. \end{aligned} \quad (3.46)$$

The latter expression depicts where the Backpropagation algorithm obtained its name from: by propagating the sensitivities backward through the network starting at the last layer until the first layer is reached as follows:

$$\mathbf{s}^M \rightarrow \mathbf{s}^{M-1} \rightarrow \dots \rightarrow \mathbf{s}^2 \rightarrow \mathbf{s}^1. \quad (3.47)$$

The final step for completing the Backpropagation process requires the starting point \mathbf{s}^M to be known for the recurrence relation of (3.46). This is acquired at the final layer:

$$s_i^M = \frac{\partial \hat{F}}{\partial n_i^M} = \frac{\partial (\mathbf{t} - \mathbf{a})^T (\mathbf{t} - \mathbf{a})}{\partial n_i^M} = \frac{\partial \sum_{j=1}^{S^M} (t_j - a_j)^2}{\partial n_i^M} = -2(t_i - a_i) \frac{\partial a_i}{\partial n_i^M}. \quad (3.48)$$

As

$$\frac{\partial a_i}{\partial n_i^M} = \frac{\partial a_i^M}{\partial n_i^M} = \frac{\partial f^M(n_i^M)}{\partial n_i^M} = \dot{f}^M(n_i^M), \quad (3.49)$$

s_i^M can now be expressed as:

$$s_i^M = -2(t_i - a_i) \dot{f}^M(n_i^M). \quad (3.50)$$

In matrix form, it is written as:

$$\mathbf{s}^M = -2\dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{a}). \quad (3.51)$$

This concludes the Backpropagation algorithm, which provides a very efficient implementation of the Chain rule (Hagan *et al.*, 2014). The next section summarises the algorithm.

3.3.1.3 Summary

The three steps involved when implementing the Backpropagation algorithm are the following:

1. Propagate the input in a feed-forward manner through the network:

$$\mathbf{a}^0 = \mathbf{p}, \quad (3.52)$$

$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\mathbf{a}^m + \mathbf{b}^{m+1}) \text{ for } m = 0, 1, \dots, M-1, \quad (3.53)$$

$$\mathbf{a} = \mathbf{a}^M. \quad (3.54)$$

2. Propagate the sensitivities backward through the network:

$$\mathbf{s}^M = -2\dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{a}), \quad (3.55)$$

$$\mathbf{s}^m = \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}, \text{ for } m = M-1, \dots, 2, 1. \quad (3.56)$$

3. Update the weights and biases utilising the approximate steepest descent rule:

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m (\mathbf{a}^{m-1})^T, \quad (3.57)$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha \mathbf{s}^m. \quad (3.58)$$

In the next section this chapter is concluded.

3.4 Conclusions

A brief history on ANNs were presented in Section 3.1. This gave an overview of how the artificial neural network is based on the functioning of a biological human brain. In Section 3.2 the neuron model, Perceptron model and a layer of neurons were discussed. The architecture of the most widely used neural network (MLP) and its training process, which make use of the Backpropagation algorithm, has been presented in Section 3.3. Since MLPs are regarded as being universal approximators capable of modelling any continuous function (Ripley, 1996), it can be utilised as the univariate functions of Generalized additive models (GAMs). These models form the basis of GANNs which are discussed next.

CHAPTER 4

GENERALIZED ADDITIVE NEURAL NETWORKS

Computers are used in many real-world applications to solve complex pattern recognition problems. The task of classifying spam e-mails can be automated by computers due to its fast processing abilities and fewer computational errors compared to humans. In order to address the complexity of the spam e-mail classification problem, a computer capable of executing parallel processes is desired over a conventional sequential computer. Fortunately, artificial neural networks are well suited for this type of problem because of its flexible nonlinear modelling and powerful pattern recognition abilities (Du Toit, 2006).

In this chapter a neural network known as the Generalized additive neural network (GANN) is discussed and is the focus of this study. In the next chapter this type of neural network is examined to detect spam e-mail. The most common neural network used is known as the Multilayer Perceptron (MLP). It is used by the GANN as univariate (basis) functions, since it is a universal approximator (Ripley, 1996). Related studies performed on the GANN is not limited to the following and include studies done in recent years by Goosen (2011), Campher (2008), Du Toit (2006) and Potts (1999). Sarle (1994) introduced GANNs when the relationship between neural network and statistical models were discussed. He demonstrated that a nonlinear Generalized additive model (GAM) could be implemented as a neural network. An interactive construction algorithm was then proposed by Potts (1999), based on Sarle's proposal, for building GANNs. Du Toit (2006)¹ introduced an automated construction algorithm that constructs GANNs in an attempt to improve on the interactive construction algorithm proposed by Potts (1999). This technique uses model selection criteria supporting the model building process which enables the algorithm to objectivity select the best GANN model. Campher (2008) did

¹The literature presented in this chapter is primarily based on Potts (1999) and Du Toit (2006)

a study where the GANN was compared with decision trees and alternating conditional expectations as the two latter techniques were known to perform well in the field of predictive data mining. The aim of the study was to better understand the GANN's contribution to the data mining field. The outcome of Campher's study provided promising results and found that the GANN compared well to the other techniques. Suggestions for future work included the implementation of Du Toit's AutoGANN system on classification problems to compare the performance with related methods. Du Toit (2006) named the implementation of the automated construction algorithm AutoGANN. A study similar to Campher's was done by Goosen (2011), where the performance of GANN models was compared to MLP models. The outcome provided insight in choosing between the two techniques depending on a specific problem. Future work also suggested the application of AutoGANN to classification problems. Other research done on the GANN automated construction algorithm includes predictive data mining (De Waal & Du Toit, 2011), customer churn prediction (De Waal & Du Toit, 2008), credit scoring (De Waal, Du Toit & De La Rey, 2005) and mortality prediction (Bras-Geraldes, Papoila, Xufre & Diamantino, 2013). Previous research on the GANN related to spam e-mail detection (Labuschagne & Du Toit, 2012; Du Toit & Kruger, 2012; Du Toit & De Waal, 2010; Goosen & Du Toit, 2009), provides only preliminary investigations performed on either small or single spam data sets in evaluating a GANN's classification accuracy. In these studies, it was concluded that a more comprehensive approach was needed to determine the feasibility of classifying spam with a GANN. This approach is the focus of this study.

In this study, a GANN is considered as a potential modelling technique for the spam e-mail classification problem. Since a GANN is the neural network implementation of a GAM, a discussion on GAMs and the Backfitting algorithm is presented. GAMs and smoothing, which estimates additive models with the Backfitting algorithm, are briefly discussed in Section 4.1. Next, the GANN is discussed in Section 4.2 which focuses on the GANN architecture that produces partial residual plots capable of providing meaningful insight into the complexity of the univariate (basis) functions. Both the interactive and automated construction of GANNs are considered. In Section 4.3 the Bagging and Boosting ensemble techniques utilised to possibly improve on the GANN's spam e-mail classification performance, are discussed. A conclusion to this chapter is presented in Section 4.4.

4.1 Generalized additive models and smoothing

A linear regression model has a simple structure and can be interpreted by the user. It is applied to supervised prediction problems where the relation between a dependant variable Y and the independent

variable X is inspected and fitted by a model. Smoothing summarises the trend of a response measurement as a function of one or more predictor measurements. A smoother creates an approximating function used by multiple regression algorithms, like Generalized linear models (McCullagh & Nelder, 1989) which is the generalisation of linear regression models. These Generalized linear models are defined as:

$$g_0^{-1}(E(Y)) = \alpha_0 + \alpha_1 X_1 + \dots + \alpha_p X_p + \epsilon, \quad (4.1)$$

where $E(\epsilon) = 0$ and $var(\epsilon) = \sigma^2$. The link function g_0^{-1} , which is the inverse of the activation function g_0 , restricts the response values to a certain range. The logit link function,

$$g_0^{-1}(E(Y)) = \ln\left(\frac{E(Y)}{1 - E(Y)}\right), \quad (4.2)$$

can be used for probabilities where a response is expected between 0 and 1. If the expected response is to be bound between -1 and 1, the hyperbolic tangent link function,

$$g_0^{-1}(E(Y)) = 1 - \frac{2}{1 + \ln(2E(Y))}, \quad (4.3)$$

is appropriate. Smoothers are nonparametric regression models and therefore do not have a fixed form for the dependence of Y as a function of one or more predictor measurements X_1, \dots, X_p . Smoothing attempts to reduce data noise (focuses less on irrelevant data) and captures important patterns found in the data to estimate additive models utilising the Backfitting algorithm. An additive model (Hastie & Tibshirani, 1990) consists of individual unspecified univariate functions (one for every input). The linear function of these inputs are replaced by an unspecified smooth function which provides a good predictive model, because the effect of each input can be analysed separately. A smoother is useful to explore the relationship between variables in a nonlinear model, such as neural networks, where it can be difficult to infer the relationship. Since MLPs are in theory universal approximators that can model any continuous function (Ripley, 1996), it can be applied to the GAM as the univariate functions. With this implementation, Backfitting is unnecessary since all training methods applicable to MLPs can also be applied to GANN models for determining the model parameters. Therefore, GANN models suffer from the same common optimisation and model complexity issues as MLPs. A detailed discussion on GAMs and smoothing is beyond the scope of this study and can be found in Du Toit (2006). The GANN architecture is subsequently discussed.

4.2 The GANN architecture

When neural networks are applied to prediction problems there are three practical difficulties encountered (Potts, 1999):

1. **Model selection:** Neural networks constitute a rather flexible nonlinear modelling function, which makes it difficult to find the best model. The large number of network configurations which includes the optimal number of hidden layers, the number of neurons per layer, the connections between layers and the activation functions to use, contributes to the complexity of the model. According to Potts (1999) a trial and error approach is the most reliable way to determine the best network configurations for a neural network, but is subjective to human judgement.
2. **Inscrutability:** There exist criticism towards the use of MLPs, because it is considered a black box when it comes to the reasoning, interpretation and understanding of underlying relationships in the data (Potts, 1999). Explaining the reason behind certain input to output mappings is complex, because of the flexible and nonlinear nature of the model.
3. **Troublesome training:** Parameter search on a relative large dataset could result in a resource intensive task. Optimisation of many different weights and biases requires a lot of computational power. Methods for parameter optimisation include the Backpropagation algorithm (Section 3.3.1). Considering a large search space, local minima solutions can be found and are troublesome since various starting points could converge to suboptimal solutions. Potts (1999) suggests multiple runs from different starting points as a possible solution.

Some of the GANN's attributes are inherited from MLPs like the univariate functions that are universal approximators. Fortunately, since a GANN has constraints on its architecture (Potts, 1999) it is able to address these practical difficulties (Potts, 2000). The architecture of a GANN (Potts, 1999) is built upon the neural network implementation of a GAM (Wood, 2006; Hastie & Tibshirani, 1990; Hastie & Tibshirani, 1986). A GAM can be expressed as follows:

$$g_0^{-1}[E(y)] = \beta_0 + f_1(x_1) + f_2(x_2) + \cdots + f_k(x_k), \quad (4.4)$$

where g_0^{-1} is the inverse of the output activation function, $E(y)$ the expected target value, β_0 the bias, f_j the individual univariate functions and x_j the inputs. With a GANN, the univariate functions of the GAM are MLPs since they are universal approximators (Potts, 1999). An MLP has the form:

$$g_0^{-1}[E(y)] = w_0 + w_1 \tanh\left(w_{01} + \sum_{j=1}^k w_{j1}x_j\right) + \cdots + w_h \tanh\left(w_{0h} + \sum_{j=1}^k w_{jh}x_j\right), \quad (4.5)$$

where g_0^{-1} is the inverse activation function, $E(y)$ the expected target value, \tanh the hyperbolic tangent activation function, w_0 , w_h , w_{0h} and w_{jh} the weights and x_j the inputs. In (4.5) the link-transformed expected target value is expressed as a linear combination of nonlinear functions of linear combinations of all the inputs (Du Toit, 2006). The GANN architecture has a separate MLP with a single hidden layer of h units for each input variable. Consequently, the GANN's univariate functions can be stated as follows:

$$f_j(x_j) = w_{1j} \tanh(w_{01j} + w_{11j}x_j) + \cdots + w_{hj} \tanh(w_{0hj} + w_{1hj}x_j). \quad (4.6)$$

In Figure 4.1 an example of the GANN architecture with two inputs, x_1 and x_2 , is shown. The inputs have MLPs with three- and two neurons in the hidden layers respectively. The consolidation layer contains the neurons which correspond to the univariate functions. The weights between the consolidation layer and the output layer is fixed at 1.0. The first univariate function of the network in Figure 4.1 can be defined as:

$$f_1(x_1) = w_{11} \tanh(w_{011} + w_{111}x_1) + w_{21} \tanh(w_{021} + w_{121}x_1) + w_{31} \tanh(w_{031} + w_{131}x_1), \quad (4.7)$$

and the second univariate function can be expressed as:

$$f_2(x_2) = w_{12} \tanh(w_{012} + w_{112}x_2) + w_{22} \tanh(w_{022} + w_{122}x_2). \quad (4.8)$$

Potts (2000) stated that the GANN architecture has a less complex structure as opposed to the more widely used MLP neural network. It is possible to enhance this architecture and include a skip layer which allows for a direct connection between the input and output layers. When enhancing the GANN architecture in this way, the Generalized linear model, $w_{0j}x_j$, can be regarded a special case. The enhanced univariate function is formulated as:

$$f_j(x_j) = w_{0j}x_j + w_{1j} \tanh(w_{01j} + w_{11j}x_j) + \cdots + w_{hj} \tanh(w_{0hj} + w_{1hj}x_j), \quad (4.9)$$

where $w_{0j}x_j$ represents the skip layer.

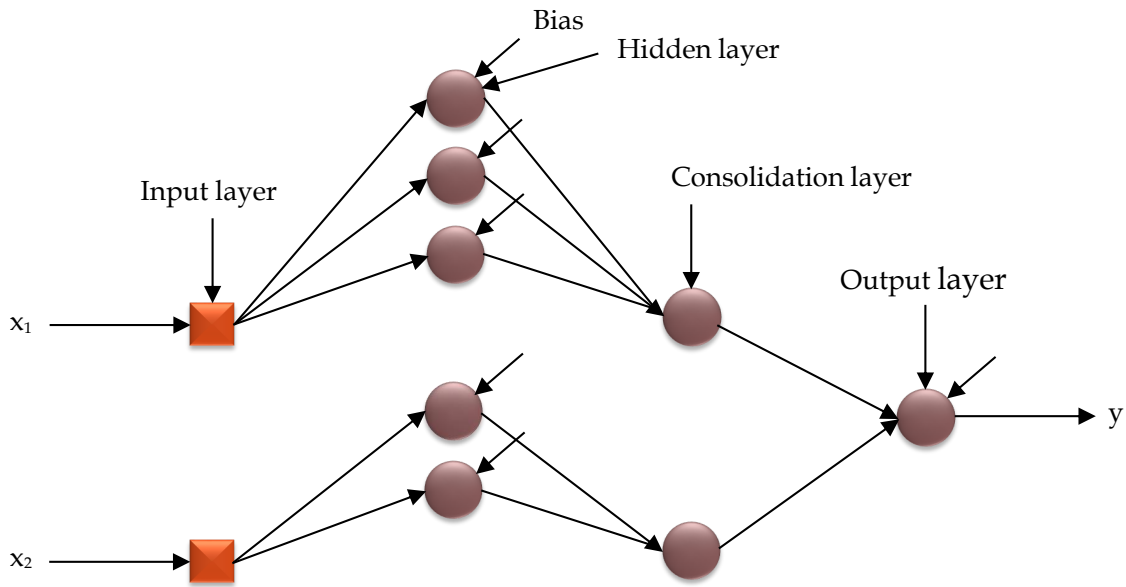


FIGURE 4.1: GANN architecture (adapted from Du Toit, 2006).

Figure 4.2 shows an example of an enhanced GANN architecture which includes a skip layer for inputs x_1 and x_3 . All the inputs have an MLP where the number of neurons in the hidden layer differs. The first input has three neurons in the hidden layer with a skip layer. The second input has two neurons in the hidden layer, but no associated skip layer. The last input has only one neuron in the hidden layer as well as a skip layer.

The first univariate function can be defined as:

$$f_1(x_1) = w_{01}x_1 + w_{11} \tanh(w_{011} + w_{111}x_1) + w_{21} \tanh(w_{021} + w_{121}x_1) + w_{31} \tanh(w_{031} + w_{131}x_1), \quad (4.10)$$

and the second univariate function can be defined as:

$$f_2(x_2) = w_{12} \tanh(w_{012} + w_{112}x_2) + w_{22} \tanh(w_{022} + w_{122}x_2). \quad (4.11)$$

The last univariate function can be defined as:

$$f_3(x_3) = w_{03}x_3 + w_{13} \tanh(w_{013} + w_{113}x_3). \quad (4.12)$$

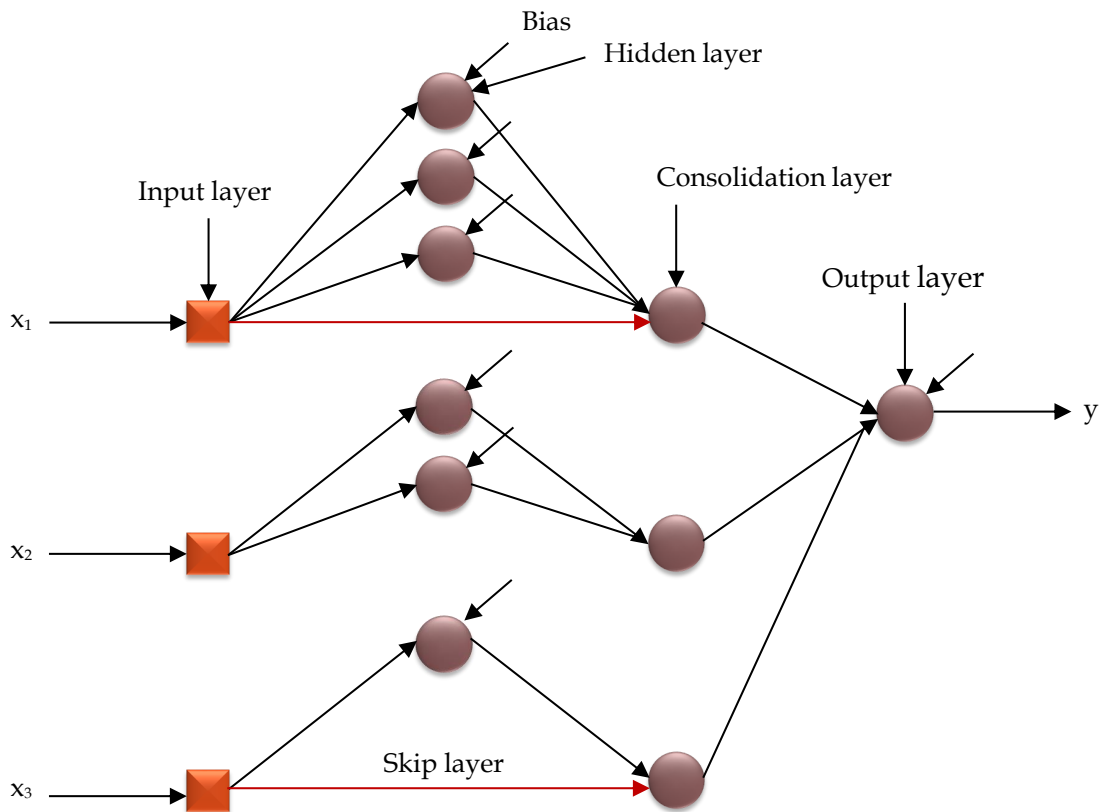


FIGURE 4.2: Enhanced GANN architecture (adapted from Du Toit, 2006).

Du Toit (2006), proposed another way of representing the GANN architectures by using subarchitecture identifiers. Table 4.1 depicts the GANN sub-architecture identifiers which starts from 0 and ends with 9. These identifiers are used in the description of the automated construction algorithm in Section 4.2.2.

GANN subarchitecture identifier	GANN subarchitecture description
0	Input removed from the model
1	MLP with a skip layer and 0 hidden nodes
2	MLP with a skip layer and 1 hidden node
3	MLP with a skip layer and 2 hidden nodes
4	MLP with a skip layer and 3 hidden nodes
5	MLP with a skip layer and 4 hidden nodes
6	MLP with no skip layer and 1 hidden node
7	MLP with no skip layer and 2 hidden nodes
8	MLP with no skip layer and 3 hidden nodes
9	MLP with no skip layer and 4 hidden nodes

TABLE 4.1: GANN subarchitecture identifiers.

Identifier 0 indicates that the input was removed from the model. Identifiers 1 to 5 include a skip layer while 6 to 9 do not have a skip layer. The number of hidden nodes vary. The GANN architecture of Figure 4.1 and Figure 4.2 can be represented using this representation as [8,7] and [4,7,2] respectively.

The interactive construction methodology of the GANN is discussed in the next section. This methodology was first used to build GANN models and will help in visualising the complexity for each of the univariate functions.

4.2.1 Interactive construction methodology

Potts (1999) proposed Algorithm 4.1 with six steps for constructing a GANN interactively. The interactive construction process starts with a GANN architecture consisting of a single MLP which has one hidden neuron and a skip layer for every input rather than a linear model. The linear fit (skip layer) is merely used for initialisation. The constrained form of the GANN is used to simplify model selection and optimisation.

For over half a century a distinct number of diagnostic plots have been adopted to examine the nonlinear relationships between the input and target variables in multiple regression models (Potts, 1999). Two collective approaches to investigate the assumption of linearity include formal tests and informal graphical methods (Cai & Tsai, 1999). An informal graphical method was introduced by Ezekiel (1924), which is still regularly used and was termed the *partial residual plot* by Larsen & McCleary (1972). The interactive model selection process for the GANN uses this visual diagnostic method that are plots of the fitted univariate functions, $\hat{f}_j(x_j)$, overlaid on the partial residuals, pr_j , versus the corresponding j th input (Potts, 1999). Partial residuals are defined as follows:

$$pr_j = g_0^{-1}(y) - \alpha - \sum_{l \neq j} \hat{f}_l(x_l) = (g_0^{-1}(y) - g_0^{-1}(\hat{y})) + \hat{f}_j(x_j). \quad (4.13)$$

When g_0^{-1} is nonlinear, a first order estimate is commonly used:

$$pr_j = \frac{\partial g_0^{-1}(y)}{\partial y} (y - \hat{y}) + \hat{f}_j(x_j). \quad (4.14)$$

These partial residual plots help to reduce the black box effect of MLPs to some degree (Du Toit, 2006). Utilising graphical methods, the relationship between inputs and the target can be explored to interpret the significance of each input on the fitted model (Potts, 1999). The j th partial residual is the

discrepancy among the actual values and the fitted model portion that does not involve x_j . According to Berk & Booth (1995) partial residuals based on the GAM fit is more reliable than those that are based on a linear fit when visualising the underlying curve of the partial residual plots. They also showed that it is common practice to begin with four parameters when GAM estimation is performed.

1. Construct a GANN with a skip layer for every input (assuming inputs are already standardised) and include one neuron in the hidden layer. From

$$f_j(x_j) = w_{0j}x_j + w_{1j} \tanh(w_{01j} + w_{11j}x_j) + \dots + w_{hj} \tanh(w_{0hj} + w_{1hj}x_j) \quad (4.15)$$

the univariate functions are initialised to

$$f_j(x_j) = w_{0j}x_j + w_{1j} \tanh(w_{01j} + w_{11j}x_j), \quad (4.16)$$

for this specific step.

2. Determine initial estimates of β_0 and w_{0j} by fitting a Generalized linear model.
3. Initialise the remaining three parameters in each hidden layer as random values from a normal distribution with mean zero and variance equal to 0.1.
4. Fit the full GANN model.
5. For each of the fitted univariate functions, inspect the functions overlaid on their partial residuals.
6. Prune (remove) neurons from the hidden layers with evidently linear effects and grow (add) neurons to the hidden layers where the nonlinear trend appears to be underfitted. The final estimates from previous fits can be used as initial values if this step is repeated.

ALGORITHM 4.1: Interactive construction algorithm of the GANN.

In the next section the automated construction methodology of the GANN is considered. This methodology is based on the creation of a search space of possible GANN models in combination with an effective search procedure to find “good” models by utilising some model selection criteria.

4.2.2 Automated construction methodology

The automated construction of GANNs (Du Toit, 2006) is an improvement in comparison to the interactive construction algorithm proposed by Potts (1999). For one, no human interaction is required to build the GANN models. The modeller is presented with more time to focus on the results rather than subjectively interpreting the partial residual plots. With the automated approach, partial residual plots are mostly used to gain insight into the models constructed rather than building the models. In this study partial residual plots were not considered since the focus was on the results obtained by the GANN. The automated construction of GANNs also alleviates the problems associated when working

with a large numbers of inputs, which is the complexity of the problem and the time constraint. A model selection criterion is used to automatically search for models in an objective manner through a formal measure of fit process. During this process the predictive accuracy of the models are evaluated to determine the “good” models. Since neural networks are sometimes prone to overfitting of the data, because of a large number of parameters that needs to be estimated, it can result in the training (or in-sample) data to be fitted well but poorly for test (or out-of-sample) data. Overfitting could occur for various reasons, for example when having a small data set the modelling technique could quickly memorise the data and might lead to inconclusive solutions. Another reason might be the redundancy of specific target values, where the target values are not varying enough to properly make a distinct conclusion. To alleviate the effect of overfitting, two model selection approaches are commonly adopted in the neural network literature (Qi & Zhang, 2001). These are a cross-validation approach (Stone, 1974) and an in-sample model selection criterion method. A combination of the two approaches were applied to this study and is discussed in Section 5.1.1. With cross-validation, the data is divided into three sets, namely training, validation and test. Models are tested on the test data set if such a set is present. With the automated construction algorithm, feature selection (Guyon & Elisseeff, 2003; Blum & Langey, 1997) is automatically performed. The automated construction algorithm (Du Toit, 2006) uses a best-first search strategy (Rich & Knight, 1991) and consists of seven steps as presented by Algorithm 4.2. Note that in Steps 5 and 6 of Algorithm 4.2, $sub(x_i)$ denotes the subarchitecture of input x_i as described in Table 4.1. A more detailed description of the automated construction algorithm can be found in Du Toit (2006).

In the next section the ensemble techniques applied to the GANN are considered. These techniques are used to possibly improve the classification accuracy obtained with the AutoGANN system.

4.3 Ensemble techniques

Thus far, supervised learning has been considered to predict the class type of an e-mail message based on a single hypothesis function h . In this section ensemble methods which use a group of hypotheses functions that determine a final decision are discussed. Ensemble algorithms usually obtain better predictive performance than a single technique by combining classifiers or by changing the underlying distribution of training data (Aggarwal & Zhai, 2012). Although combining classifiers may improve classification performance, ensembles are time-consuming and models become hard to interpret. Bootstrap aggregating (Bagging) (Breiman, 1996) and Boosting (Freund & Schapire, 1996) are both powerful ensemble algorithms used in machine learning to improve the prediction accuracy of

1. Construct a GANN with a skip layer for every input. Initialise the univariate functions to

$$f_j(x_{ji}) = w_{0j}x_{ji}, \quad (4.17)$$

which gives a single parameter for each input.

2. Determine initial estimates of the constant term and w_{0j} by fitting a Generalized linear model.
3. Fit the full GANN model and evaluate it by using the model selection criterion. Indicate that the model is available for expansion by setting the *expanded* flag to false (i.e. indicate the model's successors have not yet been generated). Next, denote this model as the root of the tree.
4. Perform a search on the tree for the best GANN model m where the *expanded* flag is set to false using the model selection criterion. If such a model is found, indicate that the model is expanded by setting the *expanded* flag to true. If no such model is found when the *expanded* flag is still false, the tree is searched for the best model. Finally, report the model and terminate the program.
5. For each input x_i from the model m identified in Step 4: If $1 \leq \text{sub}(x_i) \leq 9$, create a GANN model (node) n with the subarchitecture of x_i set to $\text{sub}(x_i) - 1$ and leave the remaining subarchitectures of m unmodified. Determine whether node n was previously created in the tree. If not, use the model selection criterion to evaluate n . Add n , to the parent node m , as a child and set its *expanded* flag to false.
6. For each input x_i from the model m identified in Step 4: If $0 \leq \text{sub}(x_i) \leq 8$, create a GANN model (node) n with the subarchitecture of x_i set to $\text{sub}(x_i) + 1$ and leave the remaining subarchitectures of m unmodified. Determine whether node n was previously created in the tree. If not, use the model selection criterion to evaluate n . Add n , to the parent node m , as a child and set its *expanded* flag to false.
7. Return to Step 4.

ALGORITHM 4.2: Automated construction algorithm of the GANN.

classifier learning systems. These two ensemble techniques are described and utilised to determine the performance gain over ordinary GANNs in the context of spam e-mail classification accuracy. The only research that could be found on the effect of Bagging and Boosting ensemble methods applied to the results obtained by a GANN was that of Labuschagne & Du Toit (2014). In their research they applied the ensemble methods to improve spam e-mail classification results. The results obtained are presented and discussed in Chapter 5.

According to Dietterich (2000), single classifiers may perform weaker, in terms of classification accuracy, than an ensemble, because of three underlying problems not addressed. These problems include the following:

1. The statistical problem: This problem arises when insufficient training data in correlation with the size of the hypothesis space \mathbb{H} exists. The situation is depicted in Figure 4.3(a). Finding

the true hypothesis is difficult since the lack of training data could cause the learning algorithm to find several hypotheses h_i which fits the data equally well. By averaging the outputs of the individual hypotheses, a more accurate approximation of the true function f can be achieved.

2. The computational problem: When a parameter search algorithm needs to find the global optimal solution in a large hypothesis space \mathbb{H} , it is prone to construct local optimal solutions. Even with sufficient training data it could still prove computationally difficult for the learning algorithm to find the best hypothesis since many different starting points exist. Numerous hypotheses h_i , each representing a distinct local optimal solution, can be fitted to the training data. By combining the outputs of the different hypotheses, a better generalisation of the global optimal solution f can be provided compared to a single local optimal solution, as shown in Figure 4.3(b).
3. The representational problem: This problem occurs when the true function f does not reside in the hypothesis search space \mathbb{H} as depicted by Figure 4.3(c). The learning algorithm will continue to search through partial training samples until a hypothesis h_i fits the training data. Constructing an ensemble of the different hypotheses taken from the hypothesis space, therefore increasing the overall search space, may provide a better approximation of the true function f .

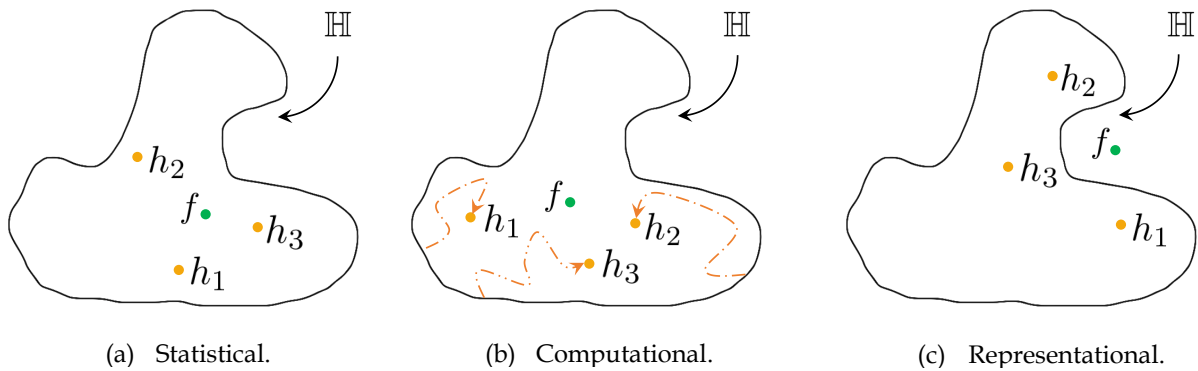


FIGURE 4.3: Fundamental reasons for enhanced ensemble performance over a single classifier (adapted from Dietterich, 2000).

Ensembles have the advantage to overcome (or even exclude) these three shortcomings existing learning algorithms are prone to (Dietterich, 2000). For this fundamental reason, it is possible to construct rather good ensembles. Bagging and Boosting are data-centred methods, as they aggregate multiple hypotheses from the same learning algorithm over different distributions of the training data to train a classifier. It creates distinctive models rather than combining different types of classifiers. The results reported are a combination or average of these models' output. Individual hypotheses are likely to have a larger error than a combination of multiple hypotheses, thus ensembles normally generate a

classifier with a smaller error on the training data. Next, the Bagging and Boosting ensemble methods are discussed.

4.3.1 Bagging

Bagging (Breiman, 1996) is a meta-algorithm that was initially designed for classification. It is a special case of model averaging and the first effective method of ensemble learning (Opitz & Maclin, 1999). Although it is mostly applied to decision tree models, it can be utilised to any model for regression or classification, i.e. neural networks.

In Bagging sampling with replacement is used to randomly construct different versions of a training set from the original training set S . The newly constructed training sets (called bootstrap samples) are each unique because some versions may include overlapping sample data from the original training set, while some sample data from the original training set might not even be present in any of the constructed training set versions. Each training set is used to train a different model. The results of these models are aggregated through averaging (in the case of regression) or voting (in the case of classification) to produce the final output C^* (Sewell, 2011). Algorithm 4.3 (Bauer & Kohavi, 1999) presents the Bagging algorithm followed by Figure 4.4 which illustrates how voting would determine the class type from three different bootstrap samples.

Input: training set S , model I , integer T (number of bootstrap samples)

1. for $i = 1$ to T {
2. $S' =$ bootstrap sample from S (independent and identically distributed sample with replacement).
3. $C_i = I(S')$
4. }
5. $C^*(x) = \arg \max_{y \in Y} \sum_{i: C_i(x)=y} 1$ (the most often prediction label y)

Output: classifier C^* .

ALGORITHM 4.3: The Bagging algorithm.

In Figure 4.4, the generated bootstrap samples, S_1 , S_2 and S_3 , are aggregated to build a final classifier C^* . The output of this classifier is determined by the class predicted most often.

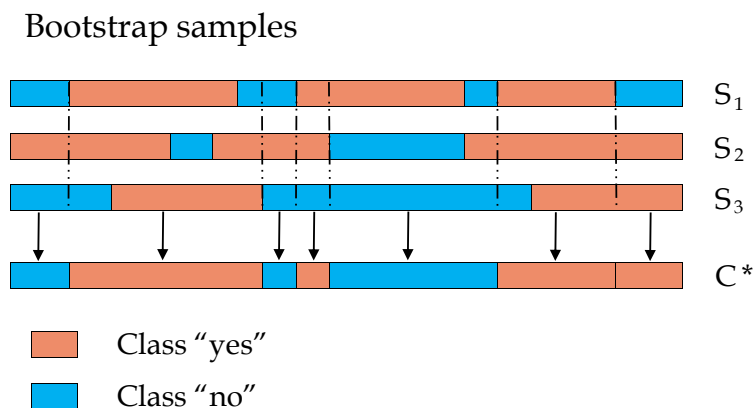


FIGURE 4.4: Example of Bagging voting.

Each dotted line indicates a random portion of the bootstrap samples which are used to perform a vote. The majority class of each random portion contributes to the final classifier C^* as indicated by the arrows. The final output for the newly build classifier will be yes, as demonstrated by Figure 4.4 since the predominant class is yes. Boosting is discussed in the next section.

4.3.2 Boosting

Boosting (Freund & Schapire, 1996), is a meta-algorithm and a widely used ensemble method (Sewell, 2011). Similar steps as Bagging are performed but it differs by building the ensemble incrementally rather than constructing the training models independently (Aggarwal & Zhai, 2012). The construction of sequential training models helps in adjusting the samples to accommodate previously computed inaccuracies by applying heavier weights on misclassified observations.

In Boosting a weight w_i is maintained for every instance h_i in the training set S . The training set for successive classifiers depend upon the performance of the previous classifier(s). A final classifier C^* is constructed using a weighted voting schema. Higher weights have a greater influence on the next hypothesis to be learned. With each iteration, the weights of misclassified instances are increased while those of correctly classified instances are decreased (Opitz & Maclin, 1999). The resampling of data ensures that each consecutive classifier is provided with the most informative training data. The objective is to minimise the expected error over different input distributions (Bauer & Kohavi, 1999).

According to Patel & Patel (2015), there are no combination rule or single ensemble generation algorithm that is universally better than others. They also concluded that the effectiveness on real-world data depends on the diversity of the classifier and characteristics of the data. In the next section the conclusions to the chapter are presented.

4.4 Conclusions

This chapter mentioned some of the earlier work performed on the GANN. A brief discussion on Generalized additive models and smoothing were presented in Section 4.1. Since the neural network implementation of a GAM is a GANN, a discussion on GAMs and smoothing were necessary. The GANN architecture was discussed in Section 4.2. A GANN has the ability to recognise patterns giving it the advantage to adapt to new spamming techniques. It can also perform variable selection to ensure only the most important inputs are selected as part of the modelling process. When the interactive construction algorithm is used for building GANN models, the interpretation of partial residual plots relies on human judgement which is subjective. This may lead to suboptimal models being constructed. The automated construction algorithm overcame the reliance on human judgement by using an objective model selection criterion to build the GANN models. Section 4.3 described the Bagging and Boosting ensemble techniques. The two ensemble methods are implemented for spam e-mail classification in an attempt to improve on the results obtained by the AutoGANN system.

In the next chapter the application of the automated construction algorithm of the GANN on five spam corpora is considered. With a model selection criterion, objectivity is ensured when models are constructed. In addition, cross-validation is performed to reduce the risk of overfitting the training data which ensures accurate results. From the results obtained, partial residual plots may be generated providing graphical results that aid the researcher in obtaining insight into the constructed models. This ability of a GANN also helps overcome the black box effect commonly associated with neural networks, to some degree. The experimental design and results obtained are discussed next.

CHAPTER 5

EXPERIMENTAL DESIGN AND RESULTS

The automated construction algorithm for the GANN was applied to five publicly available spam corpora to determine how accurately it could classify spam e-mails. Due to numerous privacy and legal restrictions applied to most private e-mails containing sensitive information not intended for public use, publicly available spam corpora were used. The use of publicly available spam corpora is well-documented in various spam filtering literature reviews and surveys (Guzella & Caminhas, 2009; Blanzieri & Bryl, 2008; Méndez, Fdez-Riverola, Glez-Peña, Díaz & Corchado, 2007; Cormack, 2007). The privacy concerns of e-mail users who feel uncomfortable publicising their private e-mails are the main cause for scarce benchmarking corpora of late (Cormack, 2007; Méndez, Fdez-Riverola, Díaz, Iglesias & Corchado, 2006). Continuing with this tendency will restrict the creation of more recent corpora (Méndez *et al.*, 2007). Unfortunately, most of these public spam corpora are outdated and do not conform to current spamming trends or real-world e-mail data. This introduces the problem of generalising the performance estimates observed from filters since it is difficult to predict how well a filter will perform once implemented in a realistic online setting. A plausible reason for the use of these less up-to-date corpora is prior research presenting a means for comparative filter benchmarking. Diverse methods are able to use similar circumstances for evaluation purposes by following the same preprocessing procedures.

The five different corpora used in this study for experimental purposes are the Enron (Metsis *et al.*, 2006), GenSpam (Medlock, 2006), PU1 (Androutsopoulos, Koutsias, Chandrinou, Paliouras & Spyropoulos, 2000), SpamAssassin (SA) (Tzortzis & Likas, 2007) and the Text REtrieval Conference 2005 (TREC2005) (Cormack & Lynam, 2005) data sets. Different classifiers found in the literature were applied to these corpora and the results obtained are compared to the performance of the automated construction algorithm for GANNs in Chapter 6.

This chapter describes the experiments conducted using the implementation of the automated construction algorithm, called the AutoGANN system (Du Toit, 2006). The different experiments performed on each of the five mentioned data sets are subsequently described. In Section 5.1 an introduction to the GANN and ensemble experiments are provided. In addition, the preprocessing techniques as well as the evaluation measures applied to the different data sets are considered. The Enron data set and its experimental results are discussed in Section 5.2. In Section 5.3 the GenSpam data set is considered along with the results of the experiments performed on this data. The PU1 data set is presented in Section 5.4 and the experimental results are discussed. In Section 5.5 the SpamAssassin (SA) data set and the results of the experiments performed are presented and discussed. The last data set, TREC2005, and the experiments conducted on this data are discussed in Section 5.6. Finally, the chapter is concluded in Section 5.7

5.1 Experimental design

All experiments were performed with SAS[®] Enterprise Miner[™] 5.3 software on a HP Compaq Elite 8300 CMT system running Windows XP SP3 (32-bit) with 4GB DDR2 memory and an Intel[®] Core[™] i5-3470 CPU at 3.20GHz. A custom-built application was used for preprocessing of the corpora. An in depth discussion of the GANN experiments follows.

5.1.1 GANN experiments

Cross-validation and an in-sample model selection approach were implemented where the complete data set was used for both training and validation. Each data set was divided into k -folds where in-sample model section was applied to each fold. The AutoGANN system would search for a good model during the 12 hours it was executed, which is considered enough time to determine a good model (Du Toit, 2006). The initial architecture of all GANN experiments were determined with an intelligent start method. With this method a subset of the available subarchitectures was used to set the search space to $\{0, 1, 2, 3, 4, 5\}$ in order to reduce the search space size (Du Toit, 2006). The evaluation measures used for model performance in this study are considered in Section 5.1.5. A description of the ensemble experiments is discussed next.

5.1.2 Ensemble experiments

In order to determine the effect of ensembles on the GANN, two well-known ensemble methods, Bagging

and Boosting (Section 4.3) were considered. Both these ensembles are supported by SAS[®] Enterprise Miner[™] (Maldonado, Dean, Czika & Haller, 2014) by using the Start Groups and End Groups nodes. In the Start Groups node, the preferred model property was set to either Bagging or Boosting. The index count property determines the number of iterations that will be executed. For the experiments an index count equal to the number of cross-validation folds was selected. The Ensemble node combines the posterior probabilities of different model nodes while the Model Comparison node was used to compare the two ensemble methods.

Figure 5.1 illustrates how this network of experiments would appear in the SAS[®] Enterprise Miner[™] software environment.

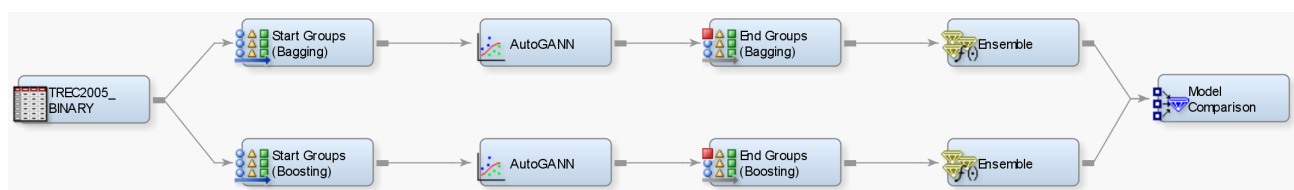


FIGURE 5.1: Bagging and Boosting ensembles in SAS[®] Enterprise Miner[™].

The left node represents the data set used. From Figure 5.1 it is the TREC2005 data set in a specific format. Columns in this file represent words and the rows individual e-mails (Section 5.1.4). A one (1) indicates that a word is present in the e-mail message while zero (0) indicates it is not. The last column in this file indicates the message class type (spam or nonspam). There is a Start Group node, an AutoGANN node, an End Group node and Ensemble node for each ensemble method. The Start Group node receives the data set. The Ensemble node calculates an average from the different iterations and the Model Comparison node gives the output. Next, different preprocessing techniques are discussed.

5.1.3 Preprocessing of corpora

Most of the data sets mentioned are in raw format consisting of many e-mail messages. In order to use these data sets in any of the experiments, preprocessing is performed. The raw data is transformed into an understandable format for use by the models. Commonly used procedures include character encoding, HTML removal, cut-off values, stop-word removal, lemmatisation and stemming. Character encoding entails the encryption of content where e.g. the word “apple” is represented in a numerical form as 73364. HTML removal is the removal of all mark-up language code found in the data. A cut-off value could indicate that only words longer than 5 characters are allowed while shorter words are discarded. Stop-word removal, lemmatisation and stemming are explained in more detail next.

5.1.3.1 Stop-word removal

Stop-word removal is the process of discarding words commonly found in a text corpus. These words are generally neutral and contribute little meaning for determining the class of an e-mail. Examples of common stop-words include words such as a, and, the, is, or, etc. which is found in both spam and nonspam messages. There are several approaches to stop-word removal, where the easiest way of implementation is to use a language-specific stop-word dictionary. A different approach involves constructing a list of stop-words specific to the text corpus under consideration by determining the term frequency of each unique word found in the corpus. Once the stop-list is populated with high-frequency words, n of the most frequent words can be removed from the text corpus. An example of stop-word removal is presented in Figure 5.2. This preprocessing step in effect reduces computational overhead and ensures only informative input terms are used by the classifier to determine the class label.

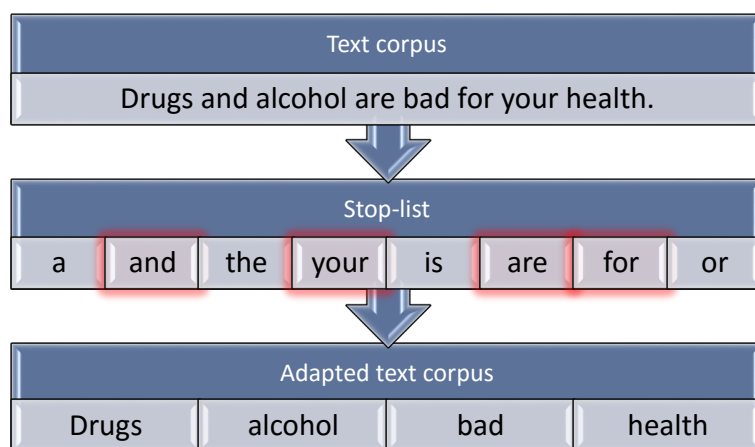


FIGURE 5.2: Stop-word removal example.

From Figure 5.2, the text corpus is reduced by half of its original size. The removal of the stop-words, highlighted in red, resulted in four words remaining. These words (Drugs, alcohol, bad, health) will contribute the most information for determining the message type.

5.1.3.2 Lemmatisation and stemming

The lemma (base form) of a word is determined through a process called lemmatisation (lemm) where the different inflected forms of a word within the same context is grouped together. This preprocessing technique should not be confused with a similar technique known as stemming. Stemming does not take into consideration the meaning and context of a word like lemm does. Lemm is a much more computationally intensive technique to perform than stemming. The benefit is a reduction in the

number of input terms and model complexity since only the lemma is used. According to Toman, Tesar & Jezek (2006), the performance impact of these techniques on text classification tasks are barely noticeable. In Figure 5.3, examples of the two techniques are shown.

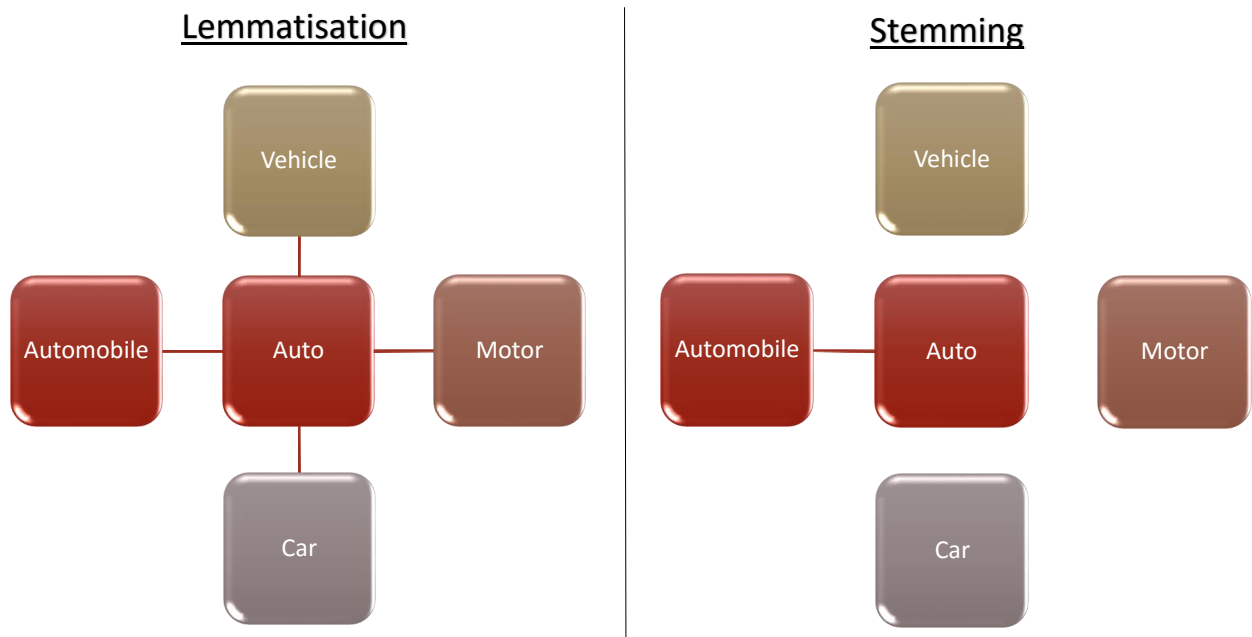


FIGURE 5.3: Lemmatisation and stemming examples.

From Figure 5.3, the lemmatisation technique depicts a relationship between the different inflected forms of the word Auto which represents the lemma. Considering the stemming technique, only Auto and Automobile will match. The other three words will be seen as separate terms with no relationship between them. This will result in four words (Auto, Motor, Car and Vehicle) when stemmed compared to the one word (Auto) with lemmatisation.

5.1.4 Representation of data

One part of the preprocessing steps is the transformation of data into a usable format for conducting experiments. Some representation techniques include tokenising, bag-of-words and feature selection. With tokenising, individual elements (words) are obtained from a large text corpus (e-mails) and divided into separate words. These words are represented by a binary bag-of-words which removes duplicates and is used to determine word occurrences. The split usually occurs on the white spaces between each word. If, for example, the phrase, “Drugs and alcohol are bad for your health”, was tokenised after stop-word removal was performed, as shown in Figure 5.2, four tokens would remain. The four tokens would be represented by the words drugs, alcohol, bad and health respectively.

Once the e-mail messages were converted into a bag-of-words representation (Mitchell, 1997), feature selection by mutual information (MI) (Sahami, Dumais, Heckerman & Horvitz, 1998) was performed to identify the top 100 word attributes (Du Toit, 2006) contributing the most information for use by the classifiers. When applying feature selection each word has the same probability of being chosen. $MI(X,C)$ is defined as:

$$\sum P(X = x, C = c) \log \left(\frac{P(X = x, C = c)}{P(X = x)P(C = c)} \right), \quad (5.1)$$

where X is the candidate word, C the category, $x \in \{0, 1\}$ where 0 indicates the absence of a word and 1 the presence of a word, $c \in \{Spam, Legitimate\}$, $P(X, C)$ is the joint probability distribution function of X and C , and $P(X)$ and $P(C)$ are the marginal probability distribution functions of X and C respectively (Piedra-Fernandez, Cantón-Garbín & Wang, 2010). A data set is then constructed with the top 100 words for use in the GANN. The outcome of these experiments are evaluated by certain measures discussed in the next section.

5.1.5 Evaluation measures

The accuracy, effectiveness and safety of the automated construction algorithm of a GANN, as a spam filter, are measured through the use of evaluation measures. To evaluate the classification performance of a spam filter based on text classification tasks, measurements such as accuracy and error rate are commonly used. In this section, evaluation measures that report on the performance and effectiveness of a spam classifier are briefly discussed. The solution proposed by Androutsopoulos *et al.* (2004) regarding the error cost calculation between the trade-off of misclassified nonspam and spam messages, is also explained.

Let N_L and N_S denote the entire data set of legitimate and spam messages respectively, that the filter must classify, and $n_{X \rightarrow C}$ the number of messages associated with category X which is classified by the filter as belonging to category C . In Table 5.1 the four possible ways to classify the e-mail messages are shown.

Class	Classification	Notation
False positives	Legit \rightarrow Spam	$n_{L \rightarrow S}$
False negatives	Spam \rightarrow Legit	$n_{S \rightarrow L}$
True positives	Spam \rightarrow Spam	$n_{S \rightarrow S}$
True negatives	Legit \rightarrow Legit	$n_{L \rightarrow L}$

TABLE 5.1: E-mail classification classes.

With the help of a confusion matrix it is possible to evaluate the different prediction classes. The confusion matrix (Table 5.2) consists of rows that represents instances of a predicted class as well as columns representing the instances of an actual class. Below Table 5.2 are the definitions of the evaluation metrics used in this study.

Predicted \ Actual	Spam	Legit
Spam	True positive $n_{S \rightarrow S}$	False negative $n_{S \rightarrow L}$
Legit	False positive $n_{L \rightarrow S}$	True negative $n_{L \rightarrow L}$

TABLE 5.2: Confusion matrix.

Spam recall (SR) measures the percentage of spam messages that the filter manages to block and is an indication of the effectiveness of the filter. SR is defined as:

$$\text{SR} = \frac{n_{S \rightarrow S}}{n_{S \rightarrow L} + n_{S \rightarrow S}}. \quad (5.2)$$

Ham recall (HR) measures the percentage of legitimate messages that successfully pass through the filter and is an indication of the filter's effectiveness. HR is defined as:

$$\text{HR} = \frac{n_{L \rightarrow L}}{n_{L \rightarrow S} + n_{L \rightarrow L}}. \quad (5.3)$$

Spam precision (SP) measures the degree to which the blocked messages are indeed spam, indicating the filter's safety. SP is defined as:

$$\text{SP} = \frac{n_{S \rightarrow S}}{n_{L \rightarrow S} + n_{S \rightarrow S}}. \quad (5.4)$$

Ham precision (HP) measures the degree to which the messages that passed through the filter are indeed ham, indicating the filter's safety. HP is defined as:

$$\text{HP} = \frac{n_{L \rightarrow L}}{n_{S \rightarrow L} + n_{L \rightarrow L}}. \quad (5.5)$$

Spam misclassification (SM) measures the percentage of spam messages that the filter failed to block. SM is defined as:

$$SM = 1 - SR. \quad (5.6)$$

Ham misclassification (HM) measures the percentage of ham messages that the filter block. HM is defined as:

$$HM = 1 - HR. \quad (5.7)$$

Weighted accuracy (WACC) (Androutsopoulos, Koutsias, Chandrinou & Spyropoulos, 2000) makes the accuracy sensitive to the cost of misclassification errors. For this case, each legitimate message is treated as λ messages. Every false positive counts as λ errors and as λ successes when classified correctly. If λ is set to 1, all legitimate and spam messages are weighed the same. More realistic results are obtained when using the WACC cost-sensitive measure, because λ assigns a higher cost to false positives (Clark, 2008). Misclassifying legitimate messages as spam can be more severe than letting a spam message pass through the filter. In this study λ is set to 1 based on the scenario where spam messages are only flagged and remains in the recipient's inbox. This is therefore equivalent to measuring the accuracy (ACC). WACC is defined as:

$$WACC = \frac{\lambda \cdot n_{L \rightarrow L} + n_{S \rightarrow S}}{\lambda \cdot (n_{L \rightarrow L} + n_{L \rightarrow S}) + n_{S \rightarrow L} + n_{S \rightarrow S}}. \quad (5.8)$$

Finally, total cost ratio (TCR) (Androutsopoulos, Koutsias, Chandrinou & Spyropoulos, 2000) is an indicator of spam filter performance. The filter is compared to the baseline where no filter is present and all spam and ham are allowed. Higher TCR values are desired as it measures the average performance of the filter. If TCR values are smaller or equal to 1 (baseline), it is better not to use a filter. TCR is defined as:

$$TCR = \frac{n_{S \rightarrow L} + n_{S \rightarrow S}}{\lambda \cdot n_{L \rightarrow S} + n_{S \rightarrow L}}. \quad (5.9)$$

In the sections to follow the experiments conducted on the five different data sets are discussed. Each experiment uses different preprocessing steps and evaluation measures as described in the corresponding literature. The results obtained with the GANN and ensemble experiments are presented. The first experiment discussed is performed on the Enron data set.

5.2 The Enron data set

The Enron corpus (Metsis *et al.*, 2006) is one of the few mass e-mail collections available to the research community which replicates real-world e-mail data. The Enron corpus consists of six subsets representing the messages from six different users respectively. Each subset contains legitimate and spam messages accumulated over a fixed time period. These time periods are presented in a month/year format. The six subsets are labelled as Enron 1 to Enron 6 and summarised in Table 5.3. The spam ratio column indicates what percentage of the subset consists of spam messages. Since Enron is publicly available, none of the data sets were encrypted to protect the privacy of the users whose e-mails were included in the corpus. Each message is represented in the form of a text file with a prefix number in its name corresponding to the order in which the messages were received. There are two versions available. A raw version containing the original messages (including duplicates) and the preprocessed version which the authors provided (Metsis *et al.*, 2006). In this study the latter version, which consists of only the body and subject sections, excluding the header part as indicated by the headers column, was used. With the preprocessing all duplicate messages (including those sent by the corresponding subset owner) were removed. Spam messages written in non-Latin character sets and HTML tags found in all messages were also removed.

Subset	# Messages			Metadata			Accumulation period		
	Legit	Spam	Total	Spam ratio	Encrypted	Headers	Creation year	Legit	Spam
Enron 1	3672	1500	5172	29.00%	No	No	2006	12/99 - 01/02	12/03 - 09/05
Enron 2	4361	1496	5857	26.00%	No	No	2006	12/99 - 05/01	05/01 - 07/05
Enron 3	4012	1500	5512	27.00%	No	No	2006	02/01 - 02/02	08/04 - 07/05
Enron 4	1500	4500	6000	75.00%	No	No	2006	04/01 - 02/02	12/03 - 09/05
Enron 5	1500	3675	5175	71.00%	No	No	2006	01/00 - 05/01	05/01 - 07/05
Enron 6	1500	4500	6000	75.00%	No	No	2006	06/00 - 03/02	08/04 - 07/05
Enron	16545	17171	33716	51.00%	No	No	2006	12/99 - 03/02	12/03 - 07/05

TABLE 5.3: Enron corpora description.

The Enron corpus was used by Metsis *et al.* (2006) for analysing five variations of the naïve Bayes filter. Two of the models, flexible Bayes (FB) and multivariate Gauss (MV Gauss), use normalised frequency representation. The multinomial model (MN TF) was considered for terms using the frequency-based representation. The remaining two models, multivariate Bernoulli (MV Bern) and multinomial Binary (MN Bool) were used for binary attributes. A detailed discussion on the different variations of naïve Bayes is beyond the scope of this study and can be found in Metsis *et al.* (2006). The results obtained by the GANN and ensembles are discussed next.

5.2.1 GANN and ensemble results

Tables 5.4 to 5.9 summarises the results obtained by the GANN with a 6-fold cross-validation and the ensemble techniques for the six Enron subsets respectively.

Measure	GANN	Bagging	Boosting
Accuracy	88.23%	93.96%	94.59%
Spam recall	87.27%	94.18%	94.51%
Spam precision	75.80%	86.28%	87.79%
Spam misclassification	12.73%	5.82%	5.49%
Ham recall	88.62%	93.88%	94.63%
Ham precision	94.46%	97.53%	97.68%
Ham misclassification	11.38%	6.12%	5.37%
Total cost ratio	2.46	4.81	5.37

TABLE 5.4: GANN and ensemble results for the Enron 1 subset.

The GANN filter obtained an accuracy of 88.23% on the Enron 1 subset. The number of blocked spam messages is 87.27% with a precision of 75.80% as indicated by SR and SP respectively. For this subset, 12.73% of spam messages were misclassified. Considering ham classification, 88.62% of ham messages passed through the GANN filter (indicated by HR) with a HP of 94.46%. Ham misclassification was 11.38%. A TCR value of 2.46 was obtained indicating that it is better to use the GANN filter than no filter at all. Both ensemble methods improved on the GANNs results. The Boosting technique performed the best with the highest TCR value of 5.37. The Enron 2 subset is discussed next.

Measure	GANN	Bagging	Boosting
Accuracy	86.48%	96.54%	96.54%
Spam recall	97.46%	95.82%	95.15%
Spam precision	65.91%	91.10%	91.63%
Spam misclassification	2.54%	4.18%	4.85%
Ham recall	82.71%	96.79%	97.02%
Ham precision	98.96%	98.54%	98.32%
Ham misclassification	17.29%	3.21%	2.98%
Total cost ratio	1.89	7.38	7.38

TABLE 5.5: GANN and ensemble results for the Enron 2 subset.

For the Enron 2 subset an accuracy result of 86.48% was obtained by the GANN. A high SP value of 97.46% was obtained which is an indication of the filter's effectiveness. According to the low SP value of 65.91% which indicates the degree to which the blocked messages are indeed spam, the GANN filter's safety was unsatisfactory. The GANN filter also misclassified 2.54% of spam messages. Regarding ham classification, the GANN performed better with a HR and HP of 82.71% and 98.96% respectively. The HM value of 17.29% is not good since important nonspam messages might have been blocked by the GANN filter. Regardless, the GANN still obtained a TCR value of 1.89 which is higher than 1.00 and indicates that the GANN is a feasible solution. The Bagging and Boosting ensembles greatly improved on the GANNs results and performed equally well with TCR values of 7.38 each. Next, the Enron 3 subset is considered.

Measure	GANN	Bagging	Boosting
Accuracy	88.55%	95.74%	96.06%
Spam recall	89.07%	93.18%	94.18%
Spam precision	74.10%	91.35%	91.59%
Spam misclassification	10.93%	6.82%	5.82%
Ham recall	88.36%	96.70%	96.76%
Ham precision	95.58%	97.43%	97.80%
Ham misclassification	11.64%	3.30%	3.24%
Total cost ratio	2.38	6.39	6.91

TABLE 5.6: GANN and ensemble results for the Enron 3 subset.

The GANN filter performed relatively the same on the Enron 3 subset as it did on the Enron 1 subset. On this subset an accuracy of 88.55% was obtained, with the filter being 89.07% effective as indicated by SR. The GANN filter's safety (SP) was lower at 74.10%. The number of misclassified spam messages is 10.93%. An HR and HP value of 88.36% and 95.58% were obtained respectively. The GANN filter blocked 11.64% of nonspam messages and has a TCR value of 2.38. The GANN is therefore a feasible spam filter compared to not using a filter. Both the ensemble methods improved on the GANNs results. The Boosting ensemble technique performed the best as indicated by the highest TCR value of 6.91. The Enron 4 subset (Table 5.7) will be discussed next.

The GANN obtained an accuracy of 72.27% on the Enron 4 subset. The SR and SP values were 72.38% and 88.55% respectively. The number of misclassified spam messages which the GANN filter failed to block was 27.62%. Considering ham classification, HR and HP values of 71.93% and 46.47% were obtained respectively. The HM was 28.07% where nonspam messages were classified as spam messages.

Measure	GANN	Bagging	Boosting
Accuracy	72.27%	97.54%	97.79%
Spam recall	72.38%	99.00%	99.28%
Spam precision	88.55%	97.75%	97.81%
Spam misclassification	27.62%	1.00%	0.72%
Ham recall	71.93%	93.18%	93.34%
Ham precision	46.47%	96.89%	97.74%
Ham misclassification	28.07%	6.82%	6.66%
Total cost ratio	2.70	30.51	33.96

TABLE 5.7: GANN and ensemble results for the Enron 4 subset.

Fortunately, the GANN is still a feasible solution for classifying spam as indicated by TCR (2.70) which is greater than 1.00. Both the ensemble techniques improved on the GANNs results. The Boosting ensemble performing the best with the highest TCR value of 33.69. The results obtained on the Enron 5 subset are considered next.

Measure	GANN	Bagging	Boosting
Accuracy	88.99%	96.72%	97.20%
Spam recall	91.70%	98.57%	98.91%
Spam precision	92.71%	96.86%	97.19%
Spam misclassification	8.30%	1.43%	1.09%
Ham recall	82.33%	92.18%	93.01%
Ham precision	80.19%	96.35%	97.22%
Ham misclassification	17.67%	7.82%	6.99%
Total cost ratio	6.45	21.62	25.34

TABLE 5.8: GANN and ensemble results for the Enron 5 subset.

The GANN showed promising results on the Enron 5 subset with an accuracy of 88.99%. Both the SR and SP results are good. The GANN filter was effective in blocking 91.70% of spam messages as indicated by SR. The GANN filter is considered safe since the degree to which the blocked spam messages were indeed spam is 92.71% as indicated by SP. The number of misclassified spam messages is 8.30%. The HR and HP values were 82.33% and 80.19% respectively. An HM value of 17.67% was obtained indicating the percentage of nonspam messages incorrectly classified as spam messages. The TCR is above 1.00 with a value of 6.45 indicating that the GANN filter is a feasible solution

for classifying spam. The Bagging and Boosting ensembles greatly improved on the GANNs results with TCR values of 21.62 and 25.34 respectively. The Boosting ensemble performed the best since it obtained the highest TCR value. The Enron 6 subset results are presented next.

Measure	GANN	Bagging	Boosting
Accuracy	83.25%	95.96%	95.96%
Spam recall	96.33%	98.83%	99.06%
Spam precision	83.77%	95.90%	95.71%
Spam misclassification	3.67%	1.17%	0.94%
Ham recall	44.00%	87.35%	86.69%
Ham precision	80.00%	96.15%	96.84%
Ham misclassification	56.00%	12.65%	13.31%
Total cost ratio	4.48	18.56	18.56

TABLE 5.9: GANN and ensemble results for the Enron 6 subset.

The GANN filter obtained an accuracy of 83.25% on the Enron 6 subset. The percentage of blocked spam messages (SR) is 96.33% which is good with a precision of 83.77%. For this subset, 3.67% of spam messages were misclassified. Considering ham classification, 44.00% of ham messages passed through the GANN filter (indicated by HR) with a HP of 80.00%. Ham misclassification was 56.00%. A TCR value of 4.48 was obtained indicating that it is better to use the GANN filter than no filter at all. Both the ensemble methods improved on the GANNs results. The Bagging and Boosting techniques performed equally well with TCR values of 18.56 each.

Of the six Enron subsets the GANN's best performance was on the Enron 5 subset with a TCR value of 6.45. The ensemble techniques improved on the GANN's results for all six Enron subsets. The ensembles performed equally well on the Enron 2 and Enron 6 subsets with TCR values of 7.38 and 18.56 respectively. Of the two ensemble techniques, the Boosting technique performed the best. The GANN is still a recommended filter with TCR values of more than 1.00. In the next section the experiments conducted on the GenSpam corpus are considered.

5.3 The GenSpam data set

The GenSpam corpus (Medlock, 2006) was created to provide a more recent collection of e-mails since other corpora were getting outdated. The study conducted by Medlock (2006), used the GenSpam

corpus to classify messages using an interpolated language model. The corpus consists of three subsets named GenSpam Training, GenSpam Adaption and GenSpam Test. Three experiments were conducted by using the GenSpam Training subset, GenSpam Adaption subset and a combination of both (GenSpam Combination) as training data. The classifier was evaluated on the GenSpam Test subset. Detailed information on the three subsets is presented in Table 5.10.

Subset	# Messages			Metadata			
	Legit	Spam	Total	Spam ratio	Encrypted	Headers	Creation year
GenSpam Training	8158	30088	38246	79.00%	Partially	No	2005
GenSpam Adaption	300	300	600	50.00%	Partially	No	2005
GenSpam Test	754	797	1551	51.00%	Partially	No	2005
GenSpam	9212	31185	40397	77.00%	Partially	No	2005

TABLE 5.10: GenSpam corpora description.

The average spam ratio for the complete GenSpam data set is 77.00%, indicating that 23.00% of the messages are legitimate messages. Preprocessing included partially encrypted entries which refer to the following five token types (Table 5.11) that were encrypted or replaced for privacy reasons in order to protect the users whose ham messages are included in the subsets. None of the subsets used headers in determining the message class.

Token type	Encryption value
Proper names	&NAME
Individual characters	&CHAR
Numbers	&NUM
E-mail addresses	&E-MAIL
Internet URLs	&URL

TABLE 5.11: GenSpam encryption types.

Figure 5.4 illustrates how a GenSpam message is represented with encryption values. The results obtained by the automated construction algorithm of the GANN and the two ensemble techniques on the GenSpam subsets are discussed next.

```

<MESSAGE>
<FROM> net </FROM>
<TO> ac.uk </TO>
<SUBJECT>
<TEXT_NORMAL> ^ Re : Hello everybody </TEXT_NORMAL>
</SUBJECT>
<DATE> Tue, 15 Apr 2003 18:40:56 +0100 </DATE>
<CONTENT-TYPE> text/plain; charset="iso-8859-1" </CONTENT-TYPE>
<MESSAGE_BODY>
<TEXT_NORMAL>
^ Dear &NAME ,
^ I am glad to hear you 're safely back in &NAME .
^ All the best
^ &NAME
^ - On &NUM December &NUM : &NUM &NAME ( &EMAIL ) wrote :
...
</TEXT_NORMAL>
</MESSAGE_BODY>
</MESSAGE>

```

FIGURE 5.4: GenSpam message representation (adapted from Medlock, 2006).

5.3.1 GANN and ensemble results

Table 5.12 shows the GANN and ensemble results obtained with the GenSpam Training subset.

Measure	GANN	Bagging	Boosting
Accuracy	82.85%	83.17%	82.59%
Spam recall	93.60%	93.85%	93.60%
Spam precision	77.63%	77.92%	77.31%
Spam misclassification	6.40%	6.15%	6.40%
Ham recall	71.49%	71.88%	70.95%
Ham precision	91.36%	97.71%	91.30%
Ham misclassification	28.51%	28.12%	29.05%
Total cost ratio	3.00	3.05	2.95

TABLE 5.12: GANN and ensemble results for the GenSpam Training subset.

The GANN obtained an accuracy of 82.85% on the GenSpam Training subset. The SR value of 93.60% indicates the percentage of spam messages the GANN filter managed to block, but according to the SP value only 77.63% of blocked spam messages were indeed spam. Overall, 6.40% of all spam messages were misclassified. The HR value is rather low at 71.49% but with 91.36% precision as shown by the HP value. A low HR value indicates that the GANN has a high false negative rate which is why HM is 28.51%. Higher TCR values indicates a better filter. As depicted by the TCR values, it is better to implement the GANN as a spam filter rather than not having a filter, since the TCR values are

all greater than 1.00. The Bagging ensemble performed slightly better than the GANN with a TCR value of 3.05. Unfortunately, the Boosting ensemble was unable to improve on the GANN's results. Nonetheless, an ensemble was able to improve the GANN's results on the GenSpam Training subset. The second experiment used the GenSpam Adaption subset. The GANN and ensemble results for this experiment are presented in Table 5.13.

Measure	GANN	Bagging	Boosting
Accuracy	86.72%	81.37%	83.49%
Spam recall	88.46%	80.43%	87.20%
Spam precision	86.08%	82.82%	81.86%
Spam misclassification	11.54%	19.57%	12.80%
Ham recall	84.88%	82.36%	79.58%
Ham precision	87.43%	79.92%	85.47%
Ham misclassification	15.12%	17.64%	20.42%
Total cost ratio	3.87	2.76	3.11

TABLE 5.13: GANN and ensemble results for the GenSpam Adaption subset.

An accuracy result of 86.72% was obtained with the GANN on the GenSpam Adaption subset. The SR and SP values are very similar with 88.46% and 86.08% respectively. A SM value of 11.54% was obtained indicating the percentage of spam messages the filter failed to block. The HR value shows that 84.88% of legitimate messages pass through the filter with a precision of 87.43% as indicated by the HP value. The false positives (when ham messages are misclassified as spam messages) are indicated by the HM value of 15.12%. Using the GANN filter is better than no filter with a TCR value of 3.87. Both ensemble methods performed well with TCR values greater than 1.00, but was unable to improve on the GANN's results. Of the three techniques the GANN performed the best on the GenSpam Adaption subset.

The third and last experiment combines the GenSpam Training and GenSpam Adaption subsets into one subset (GenSpam Combination subset) used for training. The results obtained by the GANN for the GenSpam Combination subset is shown in Table 5.14.

The GANN obtained an accuracy of 84.01% on the GenSpam Combination subset. A SR value of 93.73% was obtained, indicating the percentage of spam messages blocked by the GANN filter. According to the SP value 79.05% of the blocked messages were in fact spam. The SM value of 6.27% is good, indicating low false negatives (when spam messages are misclassified as ham messages). A HR value of

Measure	GANN	Bagging	Boosting
Accuracy	84.01%	83.17%	83.24%
Spam recall	93.73%	93.22%	93.60%
Spam precision	79.05%	78.21%	78.12%
Spam misclassification	6.27%	6.78%	6.40%
Ham recall	73.74%	72.55%	72.28%
Ham precision	91.75%	91.01%	91.44%
Ham misclassification	26.26%	27.45%	27.72%
Total cost ratio	3.21	3.05	3.07

TABLE 5.14: GANN and ensemble results for the GenSpam Combination subset.

73.74% was obtained with a precision of 91.75% as indicated by HP. The HM value is high with 26.26% of legitimate messages blocked by the GANN filter. This indicates a high false positive rate where ham messages are misclassified as spam messages. Overall the GANN performed well with a TCR value of 3.21 and is thus better than not using a spam filter. Both ensemble methods achieved similar results with TCR values greater than 1.00. Unfortunately, the ensembles were unable to improve on the GANN's results. The GANN was the best classifier of the three techniques on the GenSpam Combination subset.

Overall the GANN filter performed better than the Bagging and Boosting ensemble techniques on the GenSpam Adaption and GenSpam Combination subsets when considering TCR values. Its TCR value was only 0.05 lower on the GenSpam Training subset compared to the Bagging technique with a TCR value of 3.05. On all three GenSpam subsets the HM values were much higher than the SM values which indicates that the GANN performed well as a spam filter. The PU1 corpus and the experiments performed are considered in the next section.

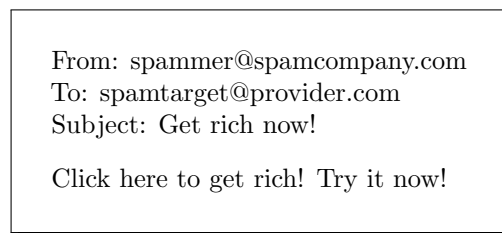
5.4 The PU1 data set

The PU1 corpus (Androutsopoulos, Koutsias, Chandrinou & Spyropoulos, 2000) was analysed using the naïve Bayesian (NB) filter and has four subsets, each consisting of 1099 messages, with different preprocessing steps performed. The preprocessing techniques include no preprocessing (Bare), stop-word removal (Stop), lemmatization (Lemm) or both techniques (Lemm and Stop). Legitimate (nonspam) and spam messages were collected over periods of 22 and 36 months respectively. The PU1 corpus has a spam ratio of 44.00% and each subset is summarised in Table 5.15.

Subset	# Messages			Metadata		
	Legit	Spam	Total	Spam ratio	Encrypted	Preprocessing
PU1 Bare	618	481	1099	44.00%	Yes	None
PU1 Stop	618	481	1099	44.00%	Yes	Stop
PU1 Lemm	618	481	1099	44.00%	Yes	Lemm
PU1 Lemm Stop	618	481	1099	44.00%	Yes	Lemm and Stop

TABLE 5.15: PU1 subset description.

Each of the four subsets were encrypted to protect the privacy of the individuals it belongs to. To achieve anonymity, the content (words, punctuation symbols, numbers, etc.) of these messages were obfuscated by substituting the content with an arbitrarily chosen integer. The four different subsets were also subjected to the preprocessing steps of HTML and attachment removal, duplicate messages and header removal except for the subject field. Thus, only the subject field and message body of an e-mail message were used. An e-mail message before and after encryption are shown in Figures 5.5 and 5.6 respectively.



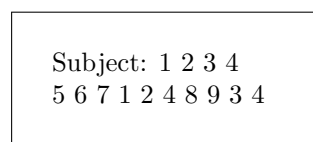
```

From: spammer@spamcompany.com
To: spamtarg@provider.com
Subject: Get rich now!

Click here to get rich! Try it now!

```

FIGURE 5.5: PU1 message before encryption (adapted from Androutsopoulos et al., 2000).



```

Subject: 1 2 3 4
5 6 7 1 2 4 8 9 3 4

```

FIGURE 5.6: PU1 message after encryption (adapted from Androutsopoulos et al., 2000).

All experiments used 10-fold cross-validation. Only the top 100 words calculated with mutual information (MI) were used by the AutoGANN system as suggested by Du Toit (2006). The GANN and ensemble results are discussed next.

5.4.1 GANN and ensemble results

The GANN and ensemble results obtained on the PU1 Bare subset is presented in Table 5.16.

Measure	GANN	Bagging	Boosting
Accuracy	95.50%	97.28%	96.60%
Spam recall	94.98%	98.45%	96.89%
Spam precision	94.81%	95.48%	95.41%
Spam misclassification	5.02%	1.55%	3.11%
Ham recall	95.92%	96.37%	96.37%
Ham precision	96.04%	98.76%	97.55%
Ham misclassification	4.08%	3.63%	3.63%
Total cost ratio	9.79	16.08	12.87

TABLE 5.16: GANN and ensemble results for the PU1 Bare subset.

The GANN achieved an accuracy value of 95.50%. Both the SR and SP values were above 90.00% at 94.98% and 94.81% respectively. The SM value of 5.02% is good, indicating very few spam messages passed through the GANN filter. Regarding ham classification, the GANN filter also achieved good results with a HR value of 95.92% and a HP value of 96.04%. False negatives were low with a HM value of 4.08%. With a TCR value of 9.79 the GANN is recommended over not using a filter. Both ensemble techniques were able to improve on the GANN's results. The Bagging ensemble performed the best with the highest accuracy and TCR values of 97.28% and 16.08 respectively. The GANN and ensemble results on the PU1 Stop subset are presented in Table 5.17 and discussed next.

Measure	GANN	Bagging	Boosting
Accuracy	95.35%	95.68%	95.91%
Spam recall	94.83%	95.34%	93.78%
Spam precision	94.50%	94.85%	96.79%
Spam misclassification	5.17%	4.66%	6.22%
Ham recall	95.74%	95.95%	97.57%
Ham precision	96.00%	96.34%	95.26%
Ham misclassification	4.26%	4.05%	2.43%
Total cost ratio	9.36	10.16	10.72

TABLE 5.17: GANN and ensemble results for the PU1 Stop subset.

An accuracy result of 95.35% was obtained with the GANN filter. According to the SR value, 94.83% of spam messages were blocked at a precision of 94.50% as indicated by the SP value. Only 5.17% of spam messages were misclassified. For ham classification similar results were obtained. The HR value is 95.74% with a HP value of 96.00%. The percentage of misclassified ham messages is 4.26%. The GANN filter achieved a TCR value of 9.36 indicating it would perform better compared to not using a filter. The Bagging and Boosting ensembles were able to improve on the GANN results with TCR values of 10.16 and 10.72 respectively. The Boosting technique performed the best on the PU1 Stop subset of the three techniques. Next, the GANN and ensemble results on the PU1 Lemm subset are discussed. In Table 5.18 the results obtained by the GANN and ensembles techniques on the PU1 Lemm subset are presented.

Measure	GANN	Bagging	Boosting
Accuracy	93.98%	95.45%	95.00%
Spam recall	92.05%	92.71%	95.31%
Spam precision	93.92%	96.74%	93.37%
Spam misclassification	7.95%	7.29%	4.69%
Ham recall	95.46%	97.58%	94.76%
Ham precision	94.03%	94.53%	96.31%
Ham misclassification	4.54%	2.42%	5.24%
Total cost ratio	7.19	7.29	8.73

TABLE 5.18: GANN and ensemble results for the PU1 Lemm subset.

The GANN filter obtained an accuracy value of 93.98%. A SR value of 92.05% indicates the percentage of spam messages blocked by the filter while SP suggest that 93.92% of the spam messages were indeed spam. Nearly 8.00% of spam messages were misclassified. For HR and HP values the GANN showed promising results of 95.46% and 94.03 respectively. Only 4.54% of ham messages were misclassified as spam messages. A TCR value of 7.19 is better than the baseline of 1.00. Using a GANN filter is thus better than not using a filter. Both ensemble methods improved on the GANN's results. Boosting performed the best with a TCR value of 8.73. The last PU1 experiment was conducted on the PU1 Lemm Stop subset. In Table 5.19 the results obtained by the GANN and the ensemble techniques for the PU1 Lemm Stop subset are presented.

An accuracy value of 95.08% was obtained with the GANN filter. Both SR and SP values were similar at 94.47% and 94.31% respectively. The SM value was 5.53%. The GANN obtained good results for ham classification with HR and HP values of 95.55% and 95.68% respectively. Only a few ham mess-

Measure	GANN	Bagging	Boosting
Accuracy	95.08%	95.91%	95.45%
Spam recall	94.47%	94.79%	92.71%
Spam precision	94.31%	95.79%	96.74%
Spam misclassification	5.53%	5.21%	7.29%
Ham recall	95.55%	96.77%	97.58%
Ham precision	95.68%	96.00%	94.53%
Ham misclassification	4.45%	3.23%	2.42%
Total cost ratio	8.91	10.67	9.60

TABLE 5.19: GANN and ensemble results for the PU1 Lemm Stop subset.

ages were misclassified as indicated by HM with a value of 4.45%. The TCR value obtained is 8.91, indicating it is better to use a GANN filter compared to not using a filter. The Bagging and Boosting ensemble techniques were able to improve on the GANN's results. Bagging achieved the highest accuracy and TCR values of 95.91% and 10.67 respectively, thus performing the best on the PU1 Lemm Stop subset of the three techniques.

Overall the GANN performed well as a spam filter. The SM values were higher than the HM values on all four PU1 subsets. This indicates that the GANN did not block too many nonspam messages, which is good. Although the ensembles performed better than the GANN, the GANN is still a recommended filter with a TCR value higher than 8.00. The next corpus to be considered is the SpamAssassin (SA) data set.

5.5 The SpamAssassin data set

In Table 5.20 a summary of the SpamAssassin (SA) corpus is given.

Corpus	# Messages			Metadata		
	Legit	Spam	Total	Spam ratio	Encrypted	Creation year
SpamAssassin (SA)	4150	1897	6047	31.00%	No	2003

TABLE 5.20: SpamAssassin corpus description.

The SA corpus contains 4150 legitimate messages and 1897 spam messages. In total the SA corpus consists of 6047 messages with a 31.00% spam ratio. A spam-filtering approach with Deep Belief

Networks (DBN) (a type of feedforward neural network with many hidden layers), which used the SA corpus, was proposed by Tzortzis & Likas (2007). Preprocessing of the corpus included stop-word removal, HTML tag removal and random splitting of the SA corpus into ten partitions for use with 10-fold cross-validation. Only the top 100 words calculated with mutual information (MI) were used. In the next section the GANN and ensemble results obtained are discussed.

5.5.1 GANN and ensemble results

Table 5.21 shows the GANN and ensemble results obtained with the SA corpus.

Measure	GANN	Bagging	Boosting
Accuracy	94.24%	93.76%	94.17%
Spam recall	87.85%	87.24%	87.63%
Spam precision	93.39%	92.47%	93.41%
Spam misclassification	12.15%	12.76%	12.37%
Ham recall	97.16%	96.75%	97.17%
Ham precision	94.60%	94.30%	94.49%
Ham misclassification	2.84%	3.25%	2.83%
Total cost ratio	5.45	5.03	5.39

TABLE 5.21: GANN and ensemble results for the SA corpus.

An accuracy result of 94.24% was obtained with the GANN filter. According to the SR value, 87.85% of spam messages were blocked by the filter at a precision of 93.39% as indicated by SP. The percentage of misclassified spam messages is 12.15%. Considering ham classification, the GANN obtained a good HR value of 97.16% with a HP value of 94.60%. False negatives are low with an HM value of 2.84%. A TCR value greater than 1.00 was achieved (5.45) indicating that the use of a GANN filter is better than not using a filter. The Bagging and Boosting ensemble techniques performed well with TCR values greater than 5.00. Unfortunately, the ensemble techniques were unsuccessful in improving on the GANN's results. The GANN performed the best of the three techniques when considering TCR values. The GANN also performed well with a lower HM value compared to the SM value.

The last experiment conducted with the automated construction algorithm for the GANN was on the TREC2005 corpus which is discussed next.

5.6 The TREC2005 data set

The TREC2005 corpus (Cormack & Lynam, 2005), also referred to as trec05p-1/full, provides a large collection of e-mails. The data set was part of the Text REtrieval Conference (TREC) 2005 Spam Track which provided one of the most realistic data sets for laboratory evaluation. The purpose of Spam Track is to measure an e-mail filter’s effectiveness by considering the ham and spam misclassification rates and model its intended usage as close as possible. Table 5.22 describes the TREC2005 corpus.

Corpus	# Messages			Metadata		
	Legit	Spam	Total	Spam ratio	Encrypted	Creation year
trec05p-1/full (TREC2005)	39399	52790	92189	57.00%	No	2005

TABLE 5.22: TREC2005 corpus description.

The TREC2005 corpus consists of 92189 messages of which 39399 messages are ham (legitimate) and 52790 messages are spam. The data set has a spam ratio of 57.00%. None of the messages are encrypted and is in a raw format like it was when originally provided. Preprocessing involved attachment removal, tokenisation of the messages, stop-word removal, and feature selection by mutual information (MI) to determine the top 100 words contributing the most information gain for use by the classifier. The GANN and ensemble results obtained with the TREC2005 corpus are discussed next.

5.6.1 GANN and ensemble results

Table 5.23 shows the GANN results obtained with the TREC2005 corpus.

Measure	GANN	Bagging	Boosting
Accuracy	64.37%	64.43%	64.43%
Spam recall	99.44%	99.44%	99.46%
Spam precision	64.71%	61.76%	61.76%
Spam misclassification	0.56%	0.56%	0.54%
Ham recall	17.43%	17.51%	17.50%
Ham precision	95.90%	95.90%	96.03%
Ham misclassification	82.57%	82.49%	82.50%
Total cost ratio	1.61	1.61	1.61

TABLE 5.23: GANN and ensemble results for the TREC2005 corpus.

The GANN filter obtained an accuracy of 64.37%. According to the SR value, 99.44% of spam messages were blocked. However, 64.71% of the blocked messages were indeed spam as indicated by the SP value. The SM value is very good with only 0.56% of spam messages misclassified as ham messages. Considering ham classification, the GANN filter obtained a HR value of 17.43% and a HP value of 95.90%. The low HR value resulted in many false negatives as depicted by the HM value of 82.57%. The GANN was still able to obtain a TCR value greater than 1.00 which indicates the GANN filter is better than no filter. Overall, both ensembles performed the same as the GANN with TCR values of 1.61. The HM value for the GANN was much higher than the SM value which indicates that the GANN performed well as a spam filter. However, the high HM value indicates a high false positive rate which results in a large number of ham messages mistakenly classified as spam. As a consequence, the TCR value is very low. In the next section this chapter is concluded.

5.7 Conclusions

In this chapter five experiments using the automated construction algorithm of the GANN were discussed. Each experiment used a different public spam corpus (Enron, GenSpam, PU1, SpamAssassin or TREC2005) which adhered to different preprocessing techniques. The GANN and ensemble experiments for each data set were outlined in Section 5.1 and a general discussion on the preprocessing techniques and data representation was given. In Sections 5.2 to 5.6 the GANN and ensemble results for the Enron, GenSpam, PU1, SA and TREC2005 experiments were discussed respectively. For all the experiments, the GANN filter is recommended compared to not using a filter, since the TCR values were always above 1.00. In some cases, the SM values were lower than the HM values and vice versa, indicating that the corpus used could have an impact on the GANN models which affect these values. In Chapter 6 a discussion on the GANN's overall performance is given. The GANN is compared to techniques found in the literature that were applied to each corpus.

CHAPTER 6

DISCUSSION

In Chapter 5 the experimental design and results obtained, together with the GANN and two ensemble techniques, applied to five publicly available spam corpora were each presented and discussed. This chapter focuses on the overall performance of the GANN. A comparison between the GANN, the Bagging and Boosting ensemble techniques and the spam filters found in the literature (mentioned in Chapter 5) are discussed in Section 6.1. The GANN and ensemble techniques are then compared in terms of accuracy in Section 6.2. The comprehensibility of the results obtained by these models is considered in Section 6.3. Next, model construction and ease of use with the SAS[®] Enterprise Miner[™] system for the three techniques are discussed in Section 6.4. A conclusion to this chapter is presented in Section 6.5.

6.1 Comparison to other techniques

In this section the GANN and ensemble techniques are compared to different spam filtering methods found in the literature. In Section 6.1.1 the techniques applied to the Enron corpus are discussed followed by the techniques used on the GenSpam corpus (Section 6.1.2), PU1 corpus (Section 6.1.3), SpamAssassin corpus (Section 6.1.4), and the TREC2005 corpus (Section 6.1.5).

6.1.1 Enron corpus

Metsis *et al.* (2006) applied five variations of the naïve Bayes technique (flexible Bayes (FB); multivariate Gauss (MV Gauss); multinomial term frequency model (MN TF); multivariate Bernoulli (MV Bern) and a multinomial binary model (MN Bool)) to the Enron corpus which consists of six subsets (Enron 1, Enron 2, . . . , Enron 6). The results compared to the GANN and the ensemble techniques for each

subset are presented by Tables 6.1 (Enron 1), 6.2 (Enron 2), 6.3 (Enron 3), 6.4 (Enron 4), 6.5 (Enron 5) and 6.6 (Enron 6) respectively. The best result per measurement (ham recall and spam recall) is indicated in bold.

Measure	FB	MV Gauss	MN TF	MV Bern	MN Bool	GANN	Bagging	Boosting
Ham recall	97.64%	94.83%	94.00%	93.19%	95.25%	88.62%	93.88%	94.63%
Spam recall	90.50%	93.08%	95.66%	97.08%	96.00%	87.27%	94.18%	94.51%

TABLE 6.1: GANN and ensemble techniques compared to the five NB techniques applied to the Enron 1 subset.

Measure	FB	MV Gauss	MN TF	MV Bern	MN Bool	GANN	Bagging	Boosting
Ham recall	98.83%	96.97%	96.78%	97.22%	97.83%	82.71%	96.79%	97.02%
Spam recall	93.63%	95.80%	96.81%	91.05%	96.68%	97.46%	95.82%	95.15%

TABLE 6.2: GANN and ensemble techniques compared to the five NB techniques applied to the Enron 2 subset.

Measure	FB	MV Gauss	MN TF	MV Bern	MN Bool	GANN	Bagging	Boosting
Ham recall	95.36%	88.81%	98.83%	75.41%	99.88%	88.36%	96.70%	96.76%
Spam recall	96.94%	97.55%	95.04%	97.42%	96.94%	89.07%	93.18%	94.18%

TABLE 6.3: GANN and ensemble techniques compared to the five NB techniques applied to the Enron 3 subset.

Measure	FB	MV Gauss	MN TF	MV Bern	MN Bool	GANN	Bagging	Boosting
Ham recall	96.61%	99.39%	98.30%	95.86%	99.05%	71.93%	93.18%	93.34%
Spam recall	95.78%	80.14%	97.79%	97.70%	97.79%	72.38%	99.00%	99.28%

TABLE 6.4: GANN and ensemble techniques compared to the five NB techniques applied to the Enron 4 subset.

Measure	FB	MV Gauss	MN TF	MV Bern	MN Bool	GANN	Bagging	Boosting
Ham recall	90.76%	97.28%	95.65%	90.08%	95.65%	82.33%	92.18%	93.01%
Spam recall	99.56%	95.42%	99.42%	97.95%	99.69%	91.70%	98.57%	98.91%

TABLE 6.5: GANN and ensemble techniques compared to the five NB techniques applied to the Enron 5 subset.

Measure	FB	MV Gauss	MN TF	MV Bern	MN Bool	GANN	Bagging	Boosting
Ham recall	89.97%	95.87%	95.12%	82.52%	96.88%	44.00%	87.35%	86.69%
Spam recall	99.55%	91.95%	98.08%	97.92%	98.10%	96.33%	98.83%	99.06%

TABLE 6.6: GANN and ensemble techniques compared to the five NB techniques applied to the Enron 6 subset.

For the ham recall measurement, the best GANN performance of 88.62% was achieved on the Enron 1 subset while the worst GANN performance of 44.00% was achieved on the Enron 6 subset. The GANN was only able to improve on the MV Bern technique of the Enron 3 subset. The Bagging and Boosting ensembles were able to improve on eight and ten NB techniques respectively of the thirty NB spam filters in all of the six Enron subsets. Both ensembles were able to improve on the GANN's results in terms of ham recall.

Considering the spam recall measurement, the best (97.46%) and worst (72.38%) GANN performances were obtained on the Enron 2 and Enron 4 subsets respectively. The GANN achieved greater results on the Enron 2 subset than all NB techniques and performed better than the MV Gauss technique on the Enron 6 subset. The ensembles were able to improve on the GANN's results except for the Enron 2 subset. The ensembles outperformed all the NB techniques on the Enron 4 subset. The Bagging and Boosting ensembles performed better on sixteen and fifteen NB techniques respectively of the thirty NB spam filters in all of the six Enron subsets. Next, the techniques applied to the GenSpam corpus are compared to the GANN and ensemble methods.

6.1.2 GenSpam corpus

The GenSpam corpus consists of three subsets (Training, Adaption and Combination). For each of these sets the following five classification models were applied: multinomial naïve Bayes (MNB), support vector machine (SVM), Bayesian logistic regression (BLR) and a unigram and bigram interpolated language model (ILM) (Medlock, 2006). The results obtained by the GANN and the ensemble techniques, for each data set, are presented by Tables 6.7, 6.8 and 6.9 respectively. The performance measures applied were accuracy, ham recall and spam recall. Values in bold represent the best result per measurement.

The GANN and both ensemble techniques performed worse for accuracy and ham recall than the five techniques applied to all three GenSpam subsets. Improvements were seen for the GANN and the Bagging and Boosting methods regarding spam recall on the GenSpam Training and Combination sets when compared to the SVM and BLR techniques. The GANN and ensemble methods were unable to

Measure	MNB	SVM	BLR	ILM Unigram	ILM Bigram	GANN	Bagging	Boosting
Accuracy	94.52%	94.33%	94.07%	95.87%	96.84%	82.85%	83.17%	82.59%
Ham recall	93.22%	98.37%	98.62%	96.74%	96.36%	71.49%	71.88%	70.95%
Spam recall	95.89%	90.05%	89.26%	94.96%	97.35%	93.60%	93.85%	93.60%

TABLE 6.7: GANN and ensemble techniques compared to the five techniques applied to the GenSpam Training subset.

Measure	MNB	SVM	BLR	ILM Unigram	ILM Bigram	GANN	Bagging	Boosting
Accuracy	95.04%	97.87%	96.91%	95.68%	96.65%	86.72%	79.58%	83.49%
Ham recall	93.35%	97.24%	97.37%	93.73%	96.49%	84.88%	82.36%	79.58%
Spam recall	96.82%	98.54%	96.42%	97.75%	96.82%	88.46%	80.43%	87.20%

TABLE 6.8: GANN and ensemble techniques compared to the five techniques applied to the GenSpam Adaption subset.

Measure	MNB	SVM	BLR	ILM Unigram	ILM Bigram	GANN	Bagging	Boosting
Accuracy	94.58%	96.07%	95.74%	97.87%	97.94%	84.01%	83.17%	83.24%
Ham recall	92.97%	98.87%	98.87%	96.74%	97.37%	73.74%	72.55%	72.28%
Spam recall	96.29%	93.10%	92.44%	99.07%	98.54%	93.73%	93.22%	93.60%

TABLE 6.9: GANN and ensemble techniques compared to the five techniques applied to the GenSpam Combination subset.

improve on any of the other techniques (MNB, ILM unigram and ILM bigram) when considering accuracy, ham recall and spam recall. The GANN was thus outperformed by these three techniques applied to the GenSpam corpus when classifying spam e-mails. Next, the GANN and ensembles are compared to the techniques applied to the PU1 corpus.

6.1.3 PU1 corpus

The PU1 data set was divided into four subsets (PU1 Bare, PU1 Stop, PU1 Lemm and PU1 Lemm Stop), each with different preprocessing techniques applied to them. The filtering technique applied to each subset was the naïve Bayesian (NB) filter (Androutsopoulos, Koutsias, Chandrinos & Spyropoulos, 2000). The GANN and ensemble techniques were compared to the NB filter in terms of accuracy, spam precision, spam recall and the total cost ratio measurement with $\lambda = 1$. When $\lambda = 1$, legitimate and spam messages are weighed the same i.e. misclassifying legitimate messages as spam is considered the same as letting a spam message pass through the filter. The results are shown in Tables 6.10 (PU1 Bare), 6.11

(PU1 Stop), 6.12 (PU1 Lemm) and 6.13 (PU1 Lemm Stop). The best result per measurement is indicated in bold.

Measure	NB	GANN	Bagging	Boosting
Accuracy	91.08%	95.50%	97.28%	96.60%
Spam precision	95.11%	94.81%	95.48%	95.41%
Spam recall	83.98%	94.98%	98.45%	96.89%
Total cost ratio	4.90	9.79	16.08	12.87

TABLE 6.10: GANN and ensemble techniques compared to the NB technique applied to the PU1 Bare subset.

Measure	NB	GANN	Bagging	Boosting
Accuracy	91.17%	95.35%	95.68%	95.91%
Spam precision	96.76%	94.50%	94.85%	96.79%
Spam recall	84.19%	94.83%	95.34%	93.78%
Total cost ratio	4.95	9.36	10.16	10.72

TABLE 6.11: GANN and ensemble techniques compared to the NB technique applied to the PU1 Stop subset.

Measure	NB	GANN	Bagging	Boosting
Accuracy	89.80%	93.98%	95.45%	95.00%
Spam precision	98.25%	93.92%	96.74%	93.37%
Spam recall	78.14%	92.05%	92.71%	95.31%
Total cost ratio	4.29	7.19	9.60	8.73

TABLE 6.12: GANN and ensemble techniques compared to the NB technique applied to the PU1 Lemm subset.

Measure	NB	GANN	Bagging	Boosting
Accuracy	90.34%	95.08%	95.91%	95.45%
Spam precision	97.96%	94.31%	95.79%	96.74%
Spam recall	79.60%	94.47%	94.79%	92.71%
Total cost ratio	4.53	8.91	10.67	9.60

TABLE 6.13: GANN and ensemble results compared to the NB technique applied to the PU1 Lemm Stop subset.

The GANN was able to outperform the NB filter in terms of accuracy, spam recall and total cost ratio (TCR) on all four subsets. The Bagging ensemble improved on the GANN's results for all four subsets on all measurements. The Boosting ensemble improved on the GANN results regarding the accuracy and TCR measurements on all four subsets. Of the four techniques the Bagging ensemble performed the best with the highest TCR value on three subsets except the PU1 Stop subset where the Boosting ensemble obtained the highest TCR value. Implementing any of the four techniques is better than not having a filter, since the TCR values are greater than 1.00. The SpamAssassin data set and two filtering techniques are considered next for comparison to the GANN and the ensembles.

6.1.4 SpamAssassin corpus

The Deep Belief Network (DBN) and support vector machine (SVM) spam filtering approaches were applied to the SpamAssassin (SA) data set (Tzortzis & Likas, 2007). The performance measures applied were accuracy, ham recall, ham precision, spam recall and spam precision. The GANN and the ensemble models' results compared to the DBN filter and the SVM filter are presented in Table 6.14. Bold values show the best result obtained per measurement.

Measure	DBN	SVM	GANN	Bagging	Boosting
Accuracy	97.50%	97.32%	94.24%	93.76%	94.17%
Ham recall	98.39%	98.24%	97.16%	96.75%	97.17%
Ham precision	98.02%	97.89%	94.60%	94.30%	94.49%
Spam recall	95.51%	95.24%	87.85%	87.24%	87.63%
Spam precision	96.40%	96.14%	93.39%	92.47%	93.41%

TABLE 6.14: GANN and ensemble techniques compared to the DBN and SVM techniques applied to the SpamAssassin data set.

The GANN and both the Bagging and Boosting ensemble methods were unable to improve on any of the five performance measures for either the DBN nor the SVM spam-filtering techniques. The DBN filter outperformed the other filters. The Boosting ensemble was the only ensemble method to improve on the GANN's results in terms of ham recall and spam precision. The techniques applied to the TREC2005 corpus are considered next.

6.1.5 TREC2005 corpus

Four k -nearest neighbour (kNN) filters were applied to the TREC2005 data set, namely yorSPAM1,

yorSPAM2, yorSPAM3 and yorSPAM4 (Cormack & Lynam, 2005). The results compared to the GANN and the ensemble methods are presented in Table 6.15. In this table the GANN misclassification results are shown. Note that lower values are considered better and the best (lowest) value is indicated in bold per measurement.

Measure	yorSPAM1	yorSPAM2	yorSPAM3	yorSPAM4	GANN	Bagging	Boosting
Ham misclassification	2.44%	0.92%	1.29%	2.99%	82.57%	82.49%	82.50%
Spam misclassification	2.43%	1.74%	1.20%	1.36%	0.56%	0.56%	0.54%

TABLE 6.15: GANN and ensemble techniques compared to the four yorSPAM techniques applied to the TREC2005 data set.

The four different kNN variations outperformed the GANN and ensemble methods when considering ham misclassification. However, the GANN and ensemble techniques outperformed the kNN methods when spam misclassification was considered. The ensemble techniques were able to slightly improve on the GANN's results except for the Bagging ensemble result obtained with the spam misclassification measurement that was the same as the GANN's result of 0.56%. For this experiment the GANN was better in classifying spam than any of the four kNN techniques. A comparison between the GANN and the Bagging and Boosting ensemble techniques, based only on accuracy, is discussed next.

6.2 Model accuracy

The accuracy measurement gives an indication of how well each of the models performed when classifying spam messages. In Table 6.16 the accuracy results obtained by each experiment is summarised. The best results when the GANN and ensemble techniques are compared, are indicated in bold. Table 6.17 shows the method which obtained the best accuracy for the corresponding experiment. This table is derived from the results of Table 6.16. In Table 6.18 the best overall method is presented based on the number of accuracy wins.

From Table 6.16 the best accuracy result for the GANN was obtained by the PU1 Bare experiment with a value of 95.50%. The lowest accuracy value of 64.37% was obtained by the TREC2005 experiment. The Bagging technique performed best on the Enron 4 experiment, with an accuracy value of 97.54%. The Boosting ensemble technique achieved the best accuracy result of the three methods with an accuracy value of 97.79% also on the Enron 4 experiment. Both the ensemble methods performed equally poor on the TREC2005 experiment with accuracy values of 64.43%.

Experiment	GANN	Bagging	Boosting
Enron 1	88.23%	93.96%	94.59%
Enron 2	86.48%	96.54%	96.54%
Enron 3	88.55%	95.74%	96.06%
Enron 4	72.27%	97.54%	97.79%
Enron 5	88.99%	96.72%	97.20%
Enron 6	83.25%	95.96%	95.96%
GenSpam Training	82.85%	83.17%	82.59%
GenSpam Adaption	86.72%	79.58%	83.49%
GenSpam Combination	84.01%	83.17%	83.24%
PU1 Bare	95.50%	97.28%	96.60%
PU1 Stop	95.35%	95.68%	95.91%
PU1 Lemm	93.98%	95.45%	95.00%
PU1 Lemm Stop	95.08%	95.91%	95.45%
SpamAssassin	94.24%	93.76%	94.17%
TREC2005	64.37%	64.43%	64.43%

TABLE 6.16: GANN and ensemble accuracy results for the fifteen experiments on the five data sets.

Experiment	Best method
Enron 1	Boosting
Enron 2	Bagging & Boosting equal
Enron 3	Boosting
Enron 4	Boosting
Enron 5	Boosting
Enron 6	Bagging & Boosting equal
GenSpam Training	Bagging
GenSpam Adaption	GANN
GenSpam Combination	GANN
PU1 Bare	Bagging
PU1 Stop	Boosting
PU1 Lemm	Bagging
PU1 Lemm Stop	Bagging
SpamAssassin	GANN
TREC2005	Bagging & Boosting equal

TABLE 6.17: Best accuracy method for each of the fifteen experiments on the five data sets.

The best method column in Table 6.17 was derived from the results obtained in Table 6.16 depending on the technique or techniques that performed the best regarding accuracy. For the TREC2005 experiment the Bagging and Boosting ensembles performed equally well with an accuracy of 64.43% and will therefore not contribute to the total wins in Table 6.18. The same is applicable for the Enron 2 and Enron 6 experiments where Bagging and Boosting accuracy values were the same (96.54% and 95.96% respectively).

Method	Accuracy wins
GANN	3
Bagging	4
Boosting	5

TABLE 6.18: Best overall method for accuracy.

All the other experiments had one method that outperformed the rest. According to Table 6.18 the Boosting ensemble performed the best, achieving the highest accuracy results on five of the experiments (Enron 1, Enron 3, Enron 4, Enron 5 and PU1 Stop). The Bagging ensemble method achieved four accuracy wins (GenSpam Training, PU1 Bare, PU1 Lemm and PU1 Lemm Stop experiments). The GANN achieved the best accuracy results on three experiments (GenSpam Adaption, GenSpam Combination and SpamAssassin).

Overall the ensemble methods performed better than the GANN with a combined nine accuracy wins compared to the GANN's three accuracy wins. The ensemble methods were mostly able to improve on the results obtained by the single GANN model which was the expected result as suggested by Aggarwal & Zhai (2012) and Dietterich (2000). Dietterich (2000) referred to three underlying problems which a single classifier does not address and might cause worse classification accuracy results than an ensemble method. The overall low accuracy values for the TREC2005 experiment might be due to the extremely large data set which according to Dietterich (2000) is subjective to the computational and representational problems. The global optimal solution is in a large hypothesis space \mathbb{H} and the true function f is possibly not present in the hypothesis search space \mathbb{H} . Since the other data sets were smaller, higher accuracy results might be achieved (found to be above 72.00% in the experiments) since the only problem to consider was the statistical problem where insufficient training data in correlation with the size of the hypothesis space \mathbb{H} could cause lower accuracy results (Dietterich, 2000). In the next section the comprehensibility of the GANN and ensemble models are considered.

6.3 Model comprehensibility

Depending on the problem that needs solving, interpretation and understanding of the results produced by the model are quite important. The GANN is able to produce partial residual plots overcoming, to some degree, the black box effect posed by MLPs and more complex models like ensembles. The partial residual plots provide meaningful insight into the complexity of the model. The relationship between inputs and the target can be explored to interpret the significance of each input on the fitted model (Potts, 1999) (Sections 4.2.1 and 4.2.2). Since ensembles have a more complex architecture than the GANN, it is difficult to interpret and understand the relationships between the input attributes and the target value. Examples of problems, where it is important to understand the relationship between attributes and a target value, include credit scoring or customer churn prediction. The construction and ease of use by the GANN and ensemble models are considered next.

6.4 Model construction and ease of use

In this study all models were constructed with the SAS[®] Enterprise Miner[™] system. The ensemble models were assembled with the built-in Start Groups and End Groups Nodes (Maldonado *et al.*, 2014) while the GANN models were built with the AutoGANN system represented by a model node. For the GANN and both ensemble techniques only a few parameters need to be set before the search for a good model is initiated. No user input is required while the search is taking place. The SAS[®] Enterprise Miner[™] system provides a graphical user interface for both the AutoGANN system and ensemble methods, making it very user-friendly to construct these models.

With the AutoGANN system (Du Toit, 2006) a time must be allocated beforehand which determines how long the search process will take. With no guidelines of how long it will take to find good models, Du Toit (2006) suggested 12 hours, which was followed in this study. The AutoGANN system can identify good models in a relatively short time (Du Toit, 2006) and proved to have high predictive accuracy. This system is relatively easy to use and is capable of addressing different problems like predictive data mining and credit scoring. It integrates numerous heuristic features capable of decreasing the time it takes to find the best GANN model. The GANN model can then be compared to other models found in the literature using various fit statistics (Du Toit, 2006). The SAS[®] software also includes a number of ensemble methods like Bagging and Boosting. A SAS[®] software license is required to use the AutoGANN system which is relatively expensive. However, there are exceptions where free licenses can be obtained when attending an academic institution. A conclusion to this chapter is presented in the next section.

6.5 Conclusions

In this chapter the GANN and the Bagging and Boosting ensemble methods were compared to different spam filtering techniques applied to each of the five public spam corpora (Chapter 5) in Section 6.1. The GANN and ensemble techniques were compared with each other with regard to accuracy in Section 6.2. The comprehensibility of the GANN and ensemble results were considered in Section 6.3. The construction and ease of use for the GANN and ensemble models using the SAS[®] Enterprise Miner[™] system were discussed in Section 6.4. With the SAS[®] software, implementing these models is relatively easy via the user-friendly graphical interface. Time is saved as most of the processes are automated and require no user interaction. In the final chapter, the conclusion to this study is presented.

CHAPTER 7

CONCLUSIONS

The purpose of this study was to investigate the use of a GANN as feasible spam filter to accurately classify spam and reducing it. Spam e-mail is still an ongoing problem with no solution to solve the issue. The automated construction algorithm for the GANN was implemented by the AutoGANN system and investigated to perform this classification task. The GANN was discussed and evaluated with regard to its spam e-mail classification capabilities. It was compared to a number of spam filtering techniques found in the literature and two ensembles methods, namely Bagging and Boosting.

In this chapter the research objectives are reviewed in Section 7.1. The research results and research limitations are presented in Section 7.2 and Section 7.3 respectively. Future work is discussed in Section 7.4 before concluding this chapter in Section 7.5.

7.1 Research objectives

The primary research aim of this study was to investigate the GANN's capabilities to accurately classify spam and therefore reducing it. A list of research objectives was defined in Section 1.2 which contributes to accomplishing the primary research aim. These objectives and how they were addressed in this study are summarised next:

1. Define spam and describe literature relating to the problem of spam e-mails and the classification thereof:

The definition of spam e-mails and a history of the e-mail application was presented in Chapter 2. Spam e-mails were defined as unsolicited bulk e-mail messages sent to multiple recipients where the sender and receivers have no known relationship. In Section 2.1 the origin of the word "spam"

was described and the first spam message sent (Figure 2.1) was shown. The five incentives for spamming (revenue generation, higher search engine rankings, promoting products and services, stealing information and phishing) were also discussed. Chapter 2 provided context for this study by emphasising the spam e-mail classification problem (Section 2.4) which is still ongoing. This chapter also guided the experimentation for this study by suggesting a relatively new spam e-mail classification approach in which the automated construction algorithm for the GANN implemented by the AutoGANN system is considered as a possible solution to mitigate spam e-mails.

2. Investigate the impact spam e-mails could have on individuals and companies when left unmanaged:

Three areas of possible consequences (time, finance and security) of unmanaged spam, which impacts both individuals and companies, were identified in Section 2.2. Figure 2.2 illustrated how these areas are interconnected and showed that spam is responsible for problems in all three these areas. Available counteract measures (filter and nonfilter solutions) that might prevent these consequences and also help mitigate spam e-mails were discussed in Section 2.3. Neural networks were considered as one of the more dynamic counteract measures to address the spam problem since it is good at pattern recognition and fitting of nonlinear functions.

3. Discuss artificial neural networks in general and the Multilayer Perceptron (MLP) neural network architecture which forms the basis of GANNs:

A brief history on artificial neural networks (ANNs) and how it mimics a biological human brain on a high level was given in Section 3.1. The neuron model was described for both the single-input and multiple-input neuron in Sections 3.2.1 and 3.2.2 respectively. A discussion on Perceptrons (Section 3.2.3), which constitutes a simple ANN and a layer of neurons (Section 3.2.4), was needed to provide context for the discussion and construction of the most common artificial neural network used, known as the Multilayer Perceptron (MLP) (Section 3.3). Within this section the Backpropagation algorithm (Section 3.3.1), a learning technique for MLPs, was explained since it overcame the theoretical pitfalls of basic ANNs and criticism on the training capabilities of the Perceptron model. Not only did Chapter 3 provide context about artificial neural networks, it also showed on what basis the GANN is constructed.

4. Describe the GANN model and the associated automated construction algorithm:

Related studies that used the GANN were considered in Chapter 4 to show that it is a powerful modelling technique. A discussion on Generalized additive models (GAMs) and smoothing were presented (Section 4.1) since a GANN is the neural network implementation of a GAM. This

provided the needed background information to discuss the GANN's architecture (Section 4.2). This section included examples of the construction of the GANN's architecture. The interactive and automated construction methodologies for the GANN were both considered in Sections 4.2.1 and 4.2.2 respectively. The latter is an improvement on the former by automating most of the processes needed to build a GANN model.

5. Give an overview of the Bagging and Boosting ensemble techniques that are applied to the GANN which might improve on the results obtained by the GANN:

The GANN was discussed in Section 4.2 and ensemble techniques were considered in Section 4.3. More specifically, the Bagging ensemble was discussed in Section 4.3.1 and the Boosting ensemble was explained in Section 4.3.2. It was noted that a set of classifiers might theoretically perform better than an individual classifier as explained in Section 4.3.

6. Discuss the experimental design used to compare the GANN to other spam filtering techniques found in the literature based on a number of different metrics:

Chapter 5 gave an overview of the methodology followed in the empirical investigation presented in this study where the GANN and ensemble experiments were discussed in Sections 5.1.1 and 5.1.2 respectively. The various preprocessing steps (Section 5.1.3), and representation techniques (Section 5.1.4) like tokenising, bag-of-words and feature selection by mutual information, applied to the corpora, were considered. The different evaluation measures that were used to evaluate each experiment were outlined in Section 5.1.5. All experiments conducted were presented in detail in Sections 5.2 (Enron), 5.3 (GenSpam), 5.4 (PU1), 5.5 (SpamAssassin) and 5.6 (TREC2005) respectively. Results obtained by the GANN and ensemble techniques were compared to techniques found in the literature in Sections 6.1.1 (Enron), 6.1.2 (GenSpam), 6.1.3 (PU1), 6.1.4 (SpamAssassin) and 6.1.5 (TREC2005).

7. Apply the ensemble techniques to the results obtained by the GANN to possibly improve the accuracy of the GANN model:

For each experiment the GANN and ensemble results were presented in Chapter 5 and Chapter 6. A detailed discussion of the accuracy results was provided in Chapter 6.

8. Compare the GANN model and the ensemble techniques in terms of accuracy, model interpretability and ease of construction:

In Chapter 6 the GANN was compared to the Bagging and Boosting ensembles in terms of how well it performed regarding accuracy (Section 6.2). The model comprehensibility (Section 6.3) and model construction and ease of use (Section 6.4) for the GANN and ensemble methods were also discussed.

Considering the above objectives and what has been observed, the final research results can be derived and is discussed next.

7.2 Research results

As a supervised machine learning technique the GANN has accurate pattern recognition capabilities and the ability to adapt to change when exposed to new training data. A GANN's models are also relatively easy to construct and comprehend. The use of GANN models is suggested for classification problems where it might be important to understand the relationship between input attributes and the expected target value. In this study these relationships were not considered important, and consequently not explored any further.

The primary research aim for this study was to determine the feasibility of employing a GANN to filter spam e-mail messages with an automated construction algorithm for the GANNs. Empirical evidence was obtained using experiments. The GANN was tested on five publicly available spam corpora (Enron, GenSpam, PU1, SpamAssassin (SA) and TREC2005) as a possible solution to mitigate spam. In all cases it was found that the GANN is a feasible spam filter based on total cost ratio values greater than 1.00. For these five corpora the GANN mostly compared well to the techniques found in the literature that were applied to the respective data sets. From the results obtained it was observed that the GANN is able to very accurately classify spam e-mails on certain spam corpora and therefore mitigate the issue. The primary research aim of this study was therefore achieved. The secondary objectives were also achieved and by comparing the GANN to the ensembles methods, it was shown that both ensemble methods performed better than the single GANN classifier (Table 6.18) in most cases. The Boosting ensemble performed the best when considering classification accuracy to filter spam e-mails. Overall, the GANN provided comparable results to other techniques found in the literature. In the next section the research limitations are discussed.

7.3 Research limitations

Spam is still an ongoing problem as the research community has not found a solution yet. There currently is no solution to solve the spam problem, however, over the past five years, researchers and practitioners have managed to decrease the amount of spam in e-mail traffic by nearly 30.00%. Some solutions have proven to mitigate the issue. Unfortunately, outdated spam data sets are limiting research findings. It is understandable that users are concerned about their privacy and thus do not

contribute to the release of more recent spam corpora. Recommendations for future work are discussed next.

7.4 Recommendations for future work

The following may be opportunities for further research:

1. Investigate the feasibility of the proposed method by comparing it to present-day spam-filtering solutions (commercial or otherwise).
2. Collect or construct a more recent spam corpus that can be used to train the GANN, thus providing more recent results.
3. Consider other ensemble approaches and focus on classification accuracy. Only three different approaches (GANN, Bagging and Boosting) were tested in this study, therefore further research is needed to generalise. Determine the ensemble combinations working best for the GANN which achieves the overall highest accuracy results.
4. Perform an e-mail classification accuracy survey. Consider the most popular spam-filtering techniques proposed in the literature and compare it to the GANN. This might provide results suitable for generalisation.

This chapter is concluded in the next section.

7.5 Conclusions

The initial objectives for this study and how it was addressed were summarised in Section 7.1. In Section 7.2 the final research findings were presented. Limitations to this study were discussed in Section 7.3 followed by possible future research opportunities outlined in Section 7.4. This chapter concludes the study on classifying spam with Generalized additive neural networks.

REFERENCE LIST

- Aceto, G. & Pescapè, A. (2012). On the recent use of email through traffic and network analysis: the impact of OSNs, new trends, and other communication platforms. *SIGMETRICS Performance Evaluation Review*, 39(4):61–70.
- Aggarwal, C. C. & Zhai, C. X. (2012). *Mining text data*. New York: Springer.
- Agrawal, B., Kumar, N. & Molle, M. (2005). Controlling spam emails at the routers. *in* ‘Proceedings of the IEEE International Conference on Communications (ICC)’. Vol. 3. pp. 1588–1592.
- Alhadlaq, I. (2016). How technology influences communication. *International journal of scientific and engineering research*, 7(1):960–963.
- Alkahtani, H., Gardner-Stephen, P. & Goodwin, R. (2011). A taxonomy of email spam filters. *in* ‘Proceedings of the 12th International Arab Conference on Information Technology (ACIT)’. pp. 351–356.
- Androutsopoulos, I., Koutsias, J., Chandrinou, K. V. & Spyropoulos, C. D. (2000). An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. *in* ‘Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval’. pp. 160–167.
- Androutsopoulos, I., Koutsias, J., Chandrinou, K. V., Paliouras, G. & Spyropoulos, C. D. (2000). An evaluation of naive bayesian anti-spam filtering. *in* ‘Proceedings of the workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning’. pp. 9–17.
- Androutsopoulos, I., Paliouras, G. & Michelakis, E. (2004). Learning to filter unsolicited commercial e-mail. Technical Report 2004/2. National Centre for Scientific Research (NCSR) “Demokritos”.
- Apulu, I. & Latham, A. (2011). Drivers for information and communication technology adoption: a case study of Nigerian small and medium sized enterprises. *International journal of business and management*, 6(5):51–60.
- Bauer, E. & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: bagging, boosting, and variants. *Machine learning*, 36(1/2):105–139.

- Ben-Itzhak, Y. (2008). 'Infosecurity 2008 - new defence strategy in battle against e-crime'. Date of access: 8 Nov 2015.
<http://www.computerweekly.com/opinion/Infosecurity-2008-New-defence-strategy-in-battle-against-e-crime>
- Berk, K. N. & Booth, D. E. (1995). Seeing a curve in multiple regression. *Technometrics*, 37(4):385–398.
- Berry, M. J. A. & Linoff, G. (1997). *Data mining techniques: for marketing, sales, and customer support*. New York: John Wiley & Sons.
- Blanzieri, E. & Bryl, A. (2008). A survey of learning-based techniques of email spam filtering. *Artificial intelligence review*, 29(1):63–92.
- Blum, A. L. & Langey, P. (1997). Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271.
- Bras-Geraldes, C., Papoila, A., Xufre, P. & Diamantino, F. (2013). Generalized additive neural networks for mortality prediction using automated and genetic algorithms. in 'Proceedings of the 2nd International IEEE Conference on Serious Games and Applications for Health (SeGAH)'. pp. 1–8.
- Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.
- Cai, Z. & Tsai, C. (1999). Diagnostics for nonlinearity in generalized linear models. *Computational statistics and data analysis*, 29(4):445–469.
- Caliendo, M., Clement, M., Papiés, D. & Scheel-Kopeinig, S. (2008). The cost impact of spam filters: measuring the effect of information system technologies in organizations. *IZA discussion paper*, 3755:1–35.
- Campher, S. E. S. (2008). Comparing generalised additive neural networks with decision trees and alternating conditional expectations. Master's thesis. School for Computer, Statistical and Mathematical Sciences. North-West University, South Africa.
- Canter, L. A. & Siegel, M. S. (1994). *How to make a fortune on the information superhighway: everyone's guerrilla guide to marketing on the internet and other on-line services*. New York: HarperCollins.
- Christin, N. (2013). Traveling the silk road: A measurement analysis of a large anonymous online marketplace. in 'Proceedings of the 22nd International Conference on World Wide Web'. pp. 213–224.

- Cisco (2014). ‘Cisco 2014 annual security report’. Date of access: 11 Feb 2014.
http://www.cisco.com/web/offer/gist_ty2_asset/Cisco_2014_ASR.pdf
- Clark, K. P. (2008). A survey of content-based spam classifiers. *CiteSeerX*, pp. 1–19.
- Clarke, I., Flaherty, T. B. & Zugelder, M. T. (2005). The CAN-SPAM Act: new rules for sending commercial e-mail messages and implications for the sales force. *Industrial marketing management*, 34(4):399–405.
- Cormack, G. V. (2007). Email spam filtering: a systematic review. *Foundations and Trends in Information Retrieval*, 1(4):335–455.
- Cormack, G. V. & Lynam, T. R. (2005). TREC 2005 spam track overview. *in* ‘Proceedings of the 14th Text REtrieval Conference (TREC-2005)’. pp. 1–17.
- Cranor, L. F. & LaMacchia, B. A. (1998). Spam. *Communications of the ACM*, 41(8):74–83.
- De Waal, D. A. & Du Toit, J. V. (2008). Gaining insight into customer churn prediction using generalized additive neural networks. *in* ‘Proceedings of the Southern Africa Telecommunication Networks and Applications Conference (SATNAC)’. p. 5.
- De Waal, D. A. & Du Toit, J. V. (2011). Automation of generalized additive neural networks for predictive data mining. *Applied artificial intelligence: an international journal*, 25(5):380–425.
- De Waal, D. A., Du Toit, J. V. & De La Rey, T. (2005). An investigation into the use of generalized additive neural networks in credit scoring. *in* ‘Proceedings of Credit Scoring & Credit Control IX’. p. 10.
- Dietterich, T. G. (2000). Ensemble methods in machine learning. *in* ‘Proceedings of the International Conference on Multiple Classifier Systems - Lecture Notes in Computer Science’. Vol. 1857. pp. 1–15.
- Du Toit, J. V. (2006). Automated construction of generalized additive neural networks for predictive data mining. PhD thesis. School for Computer, Statistical and Mathematical Sciences. North-West University, South Africa.
- Du Toit, J. V. & De Waal, D. A. (2010). Spam detection using generalized additive neural networks. *in* ‘Proceedings of the Southern Africa Telecommunication Networks and Applications Conference (SATNAC)’. p. 6.

- Du Toit, J. V. & Kruger, H. A. (2012). Filtering spam e-mail with generalized additive neural networks. *in* 'Proceedings of the 11th Annual IEEE Conference on Information Security for South Africa (ISSA)'. pp. 1–8.
- Ducker, M. & Payne, J. (2010). Information communication technology as a catalyst to enterprise competitiveness: Research report. *United states agency international development (USAID) from the american people*, pp. 1–37.
- Everett-Church, R. (1999). The spam that started it all. Technical report. Wired Magazine. Date of access: 20 May 2014.
<http://www.wired.com/1999/04/the-spam-that-started-it-all/>
- Ezekiel, M. (1924). A method for handling curvilinear correlation for any number of variables. *Journal of the american statistical association*, 19(148):431–453.
- Fawcett, T. (2003). In vivo spam filtering: a challenge problem for KDD. *ACM SIGKDD explorations newsletter*, 5(2):140–148.
- Fink, D. & Disterer, G. (2006). International case studies: to what extent is ICT infused into the operations of SMEs. *Journal of enterprise information management*, 19(6):608–624.
- Freund, Y. & Schapire, R. E. (1996). Experiments with a new boosting algorithm. *in* 'Proceedings of the 13th International Conference'. Machine Learning. pp. 148–156.
- Gomez, J. C. & Moens, M.-F. (2012). PCA document reconstruction for email classification. *Computational statistics and data analysis*, 56(3):741–751.
- Goodman, J., Cormack, G. V. & Heckerman, D. (2007). Spam and the ongoing battle for the inbox. *Communications of the ACM*, 50(2):25–33.
- Goosen, J. C. (2011). Comparing generalized additive neural networks with multilayer perceptrons. Master's thesis. School for Computer, Statistical and Mathematical Sciences. North-West University, South Africa.
- Goosen, J. C. & Du Toit, J. V. (2009). Spam detection with generalized additive neural networks. *in* 'Proceedings of the Southern Africa Telecommunication Networks and Applications (SATNAC) Conference'. p. 2.
- Gossweiler, R., Kamvar, M. & Baluja, S. (2009). What's up CAPTCHA: a CAPTCHA based on image orientation. *in* 'Proceedings of the 18th International Conference on World Wide Web'. pp. 841–850.

- Grimes, G. A. (2007). Compliance with the CAN-SPAM Act of 2003. *Communications of the ACM*, 50(2):56–62.
- Gubbi, J., Buyya, R., Marusic, S. & Palaniswami, M. (2013). Internet of Things (IoT): a vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660.
- Gudkova, D. (2013). Kaspersky security bulletin: spam evolution 2012. Technical report. Kaspersky Lab. Date of access: 15 May 2014.
http://www.securelist.com/en/analysis/204792276/Kaspersky_Security_Bulletin_Spam_Evolution_2012
- Gudkova, D. (2014). Kaspersky security bulletin: spam evolution 2013. Technical report. Kaspersky Lab. Date of access: 15 May 2014.
http://media.kaspersky.com/pdf/LK_KSB_2013_spam_EN.pdf
- Guyon, I. & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3:1157–1182.
- Guzella, T. S. & Caminhas, W. M. (2009). A review of machine learning approaches to spam filtering. *Expert systems with applications*, 36:10206–10222.
- Hafner, K. & Lyon, M. (1998). *Where wizards stay up late: the origins of the internet*. New York: Simon and Schuster.
- Hagan, M. T., Demuth, H. B., Beale, M. H. & De Jesús, O. (2014). *Neural network design*. 2nd edn. USA: Martin Hagan.
- Hastie, T. J. & Tibshirani, R. J. (1986). Generalized additive models. *Statistical science*, 1(3):297–318.
- Hastie, T. J. & Tibshirani, R. J. (1990). *Generalized additive models*. Vol. 43 of *Monographs on statistics and applied probability*. London: Chapman and Hall.
- Hayati, P. & Potdar, V. (2008). Evaluation of spam detection and prevention frameworks for email and image spam: a state of art. *in* ‘Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services (iiWAS)’. pp. 520–527.
- Hershkop, S. (2006). Behavior-based email analysis with application to spam detection. PhD thesis. Graduate School of Arts and Sciences. Columbia University, Columbia.
- Ivey, K. C. (1998). Spam: the plague of junk e-mail. *Computer applications in power*, 11(2):15–16.

- Jansen, R., Bauer, K. S., Hopper, N. & Dingledine, R. (2012). Methodically modeling the tor network. *in* 'Proceedings of the 5th USENIX Workshop on Cyber Security Experimentation and Test (CSET)'. p. 9.
- Jarosz, Q. (2009). 'Neuron hand-tuned'. Date of access: 22 Sep 2015.
https://commons.wikimedia.org/wiki/File:Neuron_Hand-tuned.svg
- Keizer, G. (2005). 'Spam could cost businesses worldwide \$50 billion'. InformationWeek. Date of access: 8 Nov 2015.
[http://www.informationweek.com/spam-could-cost-businesses-worldwide-\\$50-billion/d/d-id/1030669](http://www.informationweek.com/spam-could-cost-businesses-worldwide-$50-billion/d/d-id/1030669)
- Kiritchenko, S. & Matwin, S. (2001). Email classification with co-training. *in* 'Proceedings of the 2001 Conference of the Center for Advanced Studies on Collaborative Research (CASCON)'. p. 10.
- Klensin, J. C. (2008). 'Simple mail transfer protocol'. Internet RFC 5321. Date of access: 9 Jul 2013.
<https://tools.ietf.org/pdf/rfc5321.pdf>
- Kotsiantis, S. B., Zaharakis, I. D. & Pintelas, P. E. (2006). Machine learning: a review of classification and combining techniques. *Artificial intelligence review*, 26(3):159–190.
- Kufandirimbwa, O. & Gotora, R. (2012). Spam detection using artificial neural networks (perceptron learning rule). *Online journal of physical environmental science research*, 1(2):22–29.
- Kuncheva, L. I. (2014). *Combining pattern classifiers: methods and algorithms*. 2nd edn. New Jersey: John Wiley & Sons.
- Labuschagne, P. & Du Toit, J. V. (2012). Spam classification using a generalized additive neural network approach. *in* 'Proceedings of the Southern Africa Telecommunication Networks and Applications Conference (SATNAC)'. p. 2.
- Labuschagne, P. & Du Toit, J. V. (2014). Spam email classification with generalized additive neural networks using ensemble methods. *in* 'Proceedings of the Southern Africa Telecommunication Networks and Applications Conference (SATNAC)'. p. 6.
- Larsen, W. A. & McCleary, S. J. (1972). The use of partial residual plots in regression analysis. *Technometrics*, 14(3):781–790.
- Lee, S. M., Kim, D. S., Kim, J. H. & Park, J. S. (2010). Spam detection using feature selection and parameters optimization. *in* 'Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)'. pp. 883–888.

- Lee, Y. (2005). The CAN-SPAM Act: a silver bullet solution. *Communications of the ACM*, 48(6):131–133.
- Lugaresi, N. (2004). European union vs. spam: a legal response. *in* ‘Proceedings of the First Conference on E-mail and Anti-Spam (CEAS)’. p. 8.
- Maldonado, M., Dean, J., Czika, W. & Haller, S. (2014). Leveraging ensemble models in SAS[®] enterprise miner[™]. *in* ‘Proceedings of the SAS Global Forum 2014 Conference’. p. 16.
- McCullagh, P. & Nelder, J. A. (1989). *Generalized linear models*. Vol. 37 of *Monographs on statistics and applied probability*. 2nd edn. London: Chapman and Hall.
- McCulloch, W. S. & Pitts, W. H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical biophysics*, 5:115–133.
- Medlock, B. (2006). An adaptive, semi-structured language model approach to spam filtering on a new corpus. *in* ‘Proceedings of the Third Conference on E-mail and Anti-Spam (CEAS)’. p. 8.
- Melville, N., Stevens, A., Plice, R. K. & Pavlov, O. V. (2006). Unsolicited commercial e-mail: empirical analysis of a digital commons. *International journal of electronic commerce*, 10(4):143–170.
- Méndez, J. R., Fdez-Riverola, F., Díaz, F., Iglesias, E. L. & Corchado, J. M. (2006). A comparative performance study of feature selection methods for the anti-spam filtering domain. *in* ‘Proceedings of the 6th Industrial Conference on Data Mining (ICDM)’. pp. 106–120.
- Méndez, J. R., Fdez-Riverola, F., Glez-Peña, D., Díaz, F. & Corchado, J. M. (2007). Relaxing feature selection in spam filtering by using case-based reasoning systems. *in* ‘Proceedings of the 13th Portuguese Conference on Artificial Intelligence (EPIA)’. pp. 53–62.
- Meng, J., Lin, H. & Yu, Y. (2011). A two-stage feature selection method for text categorization. *Computers and mathematics with applications*, 62(7):2793–2800.
- Metsis, V., Androutsopoulos, I. & Paliouras, G. (2006). Spam filtering with naive Bayes: Which naive Bayes. *in* ‘Proceedings of the Third Conference on E-mail and Anti-Spam (CEAS)’. p. 9.
- Minsky, M. & Papert, S. (1969). *Perceptrons*. Cambridge: MIT Press.
- Mir, F. A. & Banday, M. T. (2010). Control of spam: a comparative approach with special reference to India. *Information and communications technology law*, 19(1):27–59.
- Mitchell, T. M. (1997). *Machine learning*. New York: McGraw-Hill.

- Moustakas, E., Ranganathan, C. & Duquenoy, P. (2005). Combating spam through legislation: a comparative analysis of US and European approaches. *in* 'Proceedings of the Second International Conference on E-mail and Anti-Spam (CEAS)'. p. 8.
- Negnevitsky, M. (2005). *Artificial intelligence: a guide to intelligent systems*. 2nd edn. United Kingdom: Pearson.
- O'Connor, P. (2006). An analysis of email marketing practices of international hotel chains: compliance with legislative requirements. *Information and communication technologies in tourism*, pp. 487–496.
- Ongori, H. & Migiro, S. O. (2010). Information and communication technology adoption in SMEs: literature review. *Journal of chinese entrepreneurship*, 2(1):93–104.
- Opitz, D. & Maclin, R. (1999). Popular ensemble methods: an empirical study. *Journal of artificial intelligence research*, 11:169–198.
- Oxford Dictionary of British & World English (2015). 'Spam'. Date of access: 7 Nov 2015.
<http://www.oxforddictionaries.com/definition/english/spam>
- Patel, T. & Patel, N. (2015). Study of ensemble learning algorithm with stacking framework. *International journal for scientific research and development*, 3(3):2800–2802.
- Pavic, S., Koh, S., Simpson, M. & Padmore, J. (2007). Could e-business create a competitive advantage in UK SMEs. *Benchmarking: an international journal*, 14(3):320–351.
- Piedra-Fernandez, J. A., Cantón-Garbín, M. & Wang, J. Z. (2010). Feature selection in AVHRR ocean satellite images by means of filter methods. *IEEE transactions on geoscience and remote sensing*, 48(12):4193–4203.
- Pitts, W. & McCulloch, W. S. (1988). How we know universals: The perception of auditory and visual forms. *in* 'Neurocomputing: foundations of research'. Cambridge: MIT Press. pp. 29–41.
- Potts, W. J. E. (1999). Generalized additive neural networks. *in* 'Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining'. pp. 194–200.
- Potts, W. J. E. (2000). *Neural network modeling course notes*. New York: SAS Institute.
- Qi, M. & Zhang, G. P. (2001). An investigation of model selection criteria for neural network time series forecasting. *European journal of operational research*, 132:666–680.
- Rao, J. M. & Reiley, D. H. (2012). The economics of spam. *The journal of economic perspectives*, 26(3):87–110.

- Reynen, S. (2012). 'Project review Wednesday: are you a human PlayThru'. Date of access: 14 Nov 2015.
<http://atendesigngroup.com/blog/project-review-wednesday-are-you-human-playthru>
- Rich, E. & Knight, K. (1991). *Artificial intelligence*. 2nd edn. New York: McGraw-Hill.
- Ripley, B. D. (1996). *Pattern recognition and neural networks*. Cambridge: Cambridge University Press.
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–408.
- Rosenblatt, F. (1962). *Principles of neurodynamics*. New York: Spartan Books.
- Ruder Bošković Institute (2008). 'Quantum random bit generator service: sign up'. Date of access: 15 Nov 2015.
<http://random.irb.hr/signup.php>
- Rumelhart, D. E. & McClelland, J. L. (1986). *Parallel Distributed Processing: explorations in the Microstructure of Cognition*. Vol. 1 of *Foundations*. Cambridge: MIT Press.
- Russell, S. & Norvig, P. (2010). *Artificial intelligence: a modern approach*. 3rd edn. New Jersey: Pearson.
- Sahami, M., Dumais, S., Heckerman, D. & Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. *in* 'Proceedings of the AAAI Workshop on Learning for Text Categorization'. p. 8.
- Sarle, W. S. (1994). Neural networks and statistical models. *in* 'Proceedings of the 19th Annual SAS[®] Users Group International Conference'. pp. 1538–1550.
- SARS (2015). 'Scams and phishing attacks'. Date of access: 11 Mar 2015.
<http://www.sars.gov.za/TargTaxCrime/Pages/Scams-and-Phishing.aspx?k=SARSScamSource:email>
- Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys*, 34(1):1–47.
- Seopher (2007). 'CAPTCHA 101'. Date of access: 4 Nov 2015.
http://www.seopher.com/articles/captcha_101
- Sewell, M. (2011). Ensemble learning. *Research note*, 11(2):1–12.

- Shet, V. (2014). 'Are you a robot? introducing no CAPTCHA reCAPTCHA'. Date of access: 8 Nov 2015.
<https://googleonlinesecurity.blogspot.co.za/2014/12/are-you-robot-introducing-no-captcha.html>
- Singel, R. (2010). April 12, 1994: immigration lawyers invent commercial spam. *Wired Magazine*, p. 1. Date of access: 20 Apr 2014.
<http://www.wired.com/2010/04/0412canter-siegel-usenet-spam/>
- Siponen, M. & Stucke, C. (2006). Effective anti-spam strategies in companies: an international study. *in* 'Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS)'. pp. 127c–136c.
- SPAMHAUS (2015a). 'The definition of spam'. Date of access: 23 Mar 2015.
<https://www.spamhaus.org/consumer/definition/>
- SPAMHAUS (2015b). 'The SPAMHAUS project'. Date of access: 17 Mar 2015.
<https://www.spamhaus.org/>
- Stone, M. (1974). Cross-validators choice and assessment of statistical predictions. *Journal of the royal statistical society*, 36(2):111–147.
- Subramaniam, T., Jalab, H. A. & Taqa, A. Y. (2010). Overview of textual anti-spam filtering techniques. *International journal of the physical sciences*, 5(12):1869–1882.
- The Radicati Group (2015). Email market, 2015-2019. Executive summary. The Radicati Group, A Technology Market Research Firm. Date of access: 12 Oct 2015.
http://www.radicati.com/wp/wp-content/uploads/2015/02/Email_Market_2015-2019,_Executive_Summary.pdf
- The Radicati Group (2016). Email statistics report, 2016-2020. Executive summary. The Radicati Group, A Technology Market Research Firm. Date of access: 24 May 2016.
<http://www.radicati.com/wp/wp-content/uploads/2016/03/Email-Statistics-Report-2016-2020-Executive-Summary.pdf>
- Thoma, M. (2011). 'CAPTCHA'. Date of access: 4 Nov 2015.
<http://martin-thoma.com/captcha/>
- Toman, M., Tesar, R. & Jezek, K. (2006). Influence of word normalization on text classification. *in* 'Proceedings of the 1st International Conference on Multidisciplinary Information Sciences & Technologies (InSciT)'. Vol. 4. pp. 354–358.

- Tomlinson, R. (n.d.). The first network email: a history from Ray Tomlinson, the inventor of email. *Raytheon BBN technologies*, p. 2. Date of access: 15 May 2014.
http://www.raytheon.com/newsroom/rtnwcm/groups/public/documents/content/rtn12_tomlinson_email.pdf
- Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *in* 'Proceedings of the London Mathematical Society'. Vol. 42. pp. 230–265.
- Tzortzis, G. & Likas, A. (2007). Deep belief networks for spam filtering. *in* 'Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)'. Vol. 2. pp. 306–309.
- Vergelis, M., Shcherbakova, T. & Demidova, N. (2015). Kaspersky security bulletin: spam in 2014. Technical report. Kaspersky Lab. Date of access: 10 Jun 2015.
<http://securelist.com/analysis/kaspersky-security-bulletin/69225/kaspersky-security-bulletin-spam-in-2014/>
- Vergelis, M., Shcherbakova, T., Demidova, N. & Gudkova, D. (2016). Kaspersky security bulletin: spam and phishing in 2015. Technical report. Kaspersky Lab. Date of access: 31 Aug 2016.
https://cdn.securelist.com/files/2016/02/KSB_SpamPhishing_2015.pdf
- Vermesan, O. & Friess, P. (2013). *Internet of things: converging technologies for smart environments and integrated ecosystems*. Denmark: River Publishers.
- Von Ahn, L., Blum, M., Hopper, N. J. & Langford, J. (2003). CAPTCHA: using hard AI problems for security. *in* 'Proceedings of the 22nd International Conference on Theory and Applications of Cryptographic Techniques (EUROCRYPT)'. pp. 294–311.
- Warner, O. (2006). 'The cutest human-test: KittenAuth'. Date of access: 4 Nov 2015.
http://thepcspy.com/read/the_cutest_humantest_kittenauth/
- Wood, S. N. (2006). *Generalized additive models: an introduction with R*. Texts in statistical science. London: Chapman & Hall.
- Youn, S. & McLeod, D. (2007). A comparative study for email classification. *in* 'Proceedings of International Joint Conferences on Computer Information System Sciences and Engineering (CISSE)'. pp. 462–467.
- Yu, S. (2011). Email spam and the CAN-SPAM Act: a qualitative analysis. *International journal of cyber criminology*, 5(1):715–735.

-
- Zhang, G. P. (2010). Neural networks for data mining. *in* 'Data mining and knowledge discovery handbook'. 2nd edn. New York: Springer. pp. 419–444.
- Zhang, G., Patuwo, B. E. & Hu, M. Y. (1998). Forecasting with artificial neural networks: the state of the art. *International journal of forecasting*, 14:35–62.
- Zhang, L. (2005). The CAN-SPAM Act: an insufficient response to the growing spam problem. *Berkeley technology law journal*, 20(1):301–332.