UNIVERSITY OF PRETORIA

# Spamming Mobile Botnet Detection Using Computational Intelligence

Submitted in fulfillment of the requirements for the degree

**Magister Scientiae (Computer Science)**

in the

**Faculty of Engineering, Built-Environment and Information Technology**

**Ickin Vural**

## Abstract

This dissertation explores a new challenge to digital systems posed by the adaptation of mobile devices and proposes a countermeasure to secure systems against threats to this new digital ecosystem.

The study provides the reader with background on the topics of spam, Botnets and machine learning before tackling the issue of mobile spam.

The study presents the reader with a three tier model that uses machine learning techniques to combat spamming mobile Botnets. The three tier model is then developed into a prototype and demonstrated to the reader using test scenarios.

Finally, this dissertation critically discusses the advantages of having using the three tier model to combat spamming Botnets.

## Keywords

**Spam, Malware, Bot, Botnet, Mobile, Computational intelligence, Artificial immune system, Bayesian Spam Filtering and Neural Networks**

## Table of Contents

## List of Figures

## List of Tables

## List of Charts

## List of Equations

## List of Algorithms

# Part I: Introduction

# 1 Introduction

## 1.1 Introduction

Digital security can, in many ways, be likened to an arms race; the belligerents who are security experts defending systems from attack, and criminals and adventurers that target digital systems.

These two sides are locked in a battle in which the development of security counter measures is matched by new, more sophisticated attacks. The field of digital security is one that finds its ecosystem in a constant state of evolution. Threats of yesteryear are replaced with new and unforeseen threats and, every now and then, an old threat reappears to wreak havoc. Every successful attack is copied by others and evolved to combat counter measures put in place by defenders of digital systems.

Mobile devices and their applications are meant to improve our lives. Unfortunately, these technologies are often the target of malicious intent. Digital viruses and worms are probably the most well known examples of malicious intent; another example of malicious software is Botware (1), which is software that installs its self on a device with the purpose of gaining control over it (2).

The remainder of the chapter is structured as follows; a motivation for this research in Section 1.2, a problem statement in Section 1.3, the objectives of this research in Section 1.4, some commonly used terminology in Section 1.5 and finally we describe the layout of this dissertation in Section 1.6.

## 1.2 Motivation

The field of computer security, which for many years focused on paradigms such as network security, information security and workstation security, is facing a paradigm shift with the ever-increasing popularity of mobile devices such as smart phones and tablet devices. As many of the current threats to mobile devices are similar to those that threaten desktop machines connected to the internet, many of the same solutions can be adapted to deal with mobile devices.

Nevertheless, mobile devices present their own unique challenges such as a fragmented operating system market (with operating systems such as Apple OS, Android, Windows Mobile, RIM, etc. (2)). The mobile space also hosts a proliferation of manufacturers building devices on different standards. These factors as well as the more limited processing and data storage capabilities of mobile devices (4) means that security solutions have to be programmed with these limitations in mind.

Several years ago, the word "spam" was used colloquially to represent unwanted junk email sent from desktop machines. The migration of computing from desktop devices to smart phones and tablets (5) has led to the appearance of threats, such as spam, that initially only affected computers. Spam is also no longer just limited to email. Various types of spam are encountered today, for example, SMS spam and instant messaging spam (SPIM) (3).

Various jurisdictions have implemented legislation to control spam. One particular example is the introduction of the Electronic Communications and Transactions Act, 2002 (4). Unsolicited mail now has a legal definition and the sending of spam is prohibited. Spammers, if identified, are liable for a fine and/or prosecution. Thus, spammers will attempt to cover their trail to prevent identification.

There are a number of identity concealment techniques used by spammers. The most common is the use of Botnets (5) as their use allows the spammer to send spam from devices that cannot be linked to them. The owner of the device usually has no idea that their machine has been compromised until their Internet connection is shut down by an ISP. As most ISPs block bulk mail if they suspect it might be spam (9), the spammers who control these Botnets typically send low volumes of mail from numerous infected computers. It is done in this fashion so as not to arouse suspicions when bulk mail originates from a single computer. The following section defines the threat posed in more detail.

## 1.3 Problem Statement

The threat that this dissertation addresses is the migration of spamming Botnets onto mobile devices. Botnets are now capable of infecting mobile devices and using them to send mobile spam. Therefore, a significant cause of spam email (Botnets) could be a significant generator of spam SMSs in the very near future.

Network forensics is the capture, recording and analysis of network events in order to discover the source of problem incidents (6). Using a spamming Botnet detector implementation installed on a network server as a means of network forensics to detect Botnets has two primary drawbacks: firstly, the analysis, which leads to the detection of Botnets, only happens after the spam message has already been sent. This is not ideal as people do not like receiving spam and every spam SMS sent costs the device owner money.

The second drawback of such a Botnet detector is that the capture of packet information, including user data, raises privacy issues and in terms of the Electronic Communications Privacy Act (ECPA), ISPs are forbidden from eavesdropping or disclosing intercepted contents without user permission for limited operations monitoring or under court order (6).

A tool for detecting spamming Botnets on mobile devices is best installed on the mobile device in order to intercept spam SMSs before they are sent. Installing the Botnet detecting software on a mobile device poses another set of problems. Firstly, the processing power and data storage capabilities of mobile devices are limited. This could result in a degradation of performance while the software attempts to process the message data and determine if it is a valid message.

Secondly, the update of this software will require users to continuously install software updates on their devices. An implementation installed on a network server will have access to large amounts of processing and data storage capabilities. Also, this software can be updated at a centralised location. The problems listed here will be addressed by means of the objectives detailed in the next section.

## 1.4 Objectives

The aim of this research is to introduce a novel way of combating a not much researched threat to mobile devices, namely SMS spam. The author implemented a three tier model to detect spamming Botnets. The three tiers that were used by the author are, Artificial Immune Systems, Bayesian Filters and an Artificial Neural Network.

The Artificial Immune System is installed on a mobile device, The Bayesian Filter and the Artificial Neural Network is installed on a network server. The implementations must be capable of detecting spam SMSs sent by malware installed on a mobile device. The applications must be capable of classifying SMS messages as spam and non-spam. The applications should only stop those messages classified as being spam and allow those that are not classified as spam to be sent through.

This research started off by asking what spam is and it's prevalence in the mobile ecosystem. Questions such as the economic impact of spam were discussed. The following was studied: how spam was sent and traced; anti-spam techniques; spammer identification techniques; and an in depth study of Botnets. Once mobile spam was identified as a problem that the Author would like to study, the author needed to identify how mobile spam was sent. The author then had to understand how mobile Botnets functioned.

Initially, the Author devised a network forensics implementation that monitored network data and used fuzzy logic techniques (11) to monitor SMS sending behaviour to identify Botnets. Later, this was adapted to monitor anomalies in SMS sending behaviour at a device level. This required selecting a computational intelligence technique to identify infected devices sending spam. The Author eventually implemented three prototypes, which will be discussed further in this paper. These prototypes were designed to ensure they could identify spam SMSs being sent from a mobile device, correctly identifying messages as being spam and non-spam. The next section will provide the reader with an overview of some of the terminology used in this dissertation.

## 1.5 Terminology

In order to avoid misinterpreting this dissertation it is essential that the terminology used is correctly interpreted. The terminology will be defined and explained in greater detail throughout the dissertation. However, to ensure that the a few critical points are well defined, the researcher will now provide a brief description of what is meant by the terms **spam**, **malware**, **bot**, **Botnet**, **mobile**, **computational intelligence**, **artificial immune system**, **Bayesian spam filtering** and **neural networks**

### 1.5.1 Spam

Spam means the use of any electronic communications medium to send unsolicited messages in bulk form (7). Spam mail is a spam message sent via email. It is estimated that spam constitutes over 60 percent of all email traffic on the Internet (8). Spam costs consumers and ISPs lots of money in bandwidth charges and, despite the growing number of technological means for combating spam, the spammers somehow manage to stay one step ahead and the deluge shows little sign of abating.

### 1.5.2 Malware

Malware, short for malicious software, is programming (code, scripts, active content, and other software) designed to disrupt or deny operation; gather information that leads to loss of privacy or exploitation; gain unauthorized access to system resources; and other abusive behaviour. Examples include various forms of adware, diallers, hijack ware, slag code (logic bombs), spyware, Trojan horses, viruses, web bugs and worms (9). Between the period July 2011 and October 2011, malware in the Android market increased by 472 % (10).

### 1.5.3 Bot

Internet bots, also known as web robots or bots, are software applications that run automated tasks over the internet. In this dissertation, a Bot is a device that has been taken over by Bot software (malware) to maliciously use it as a Spambot (11).

### 1.5.4 Botnet

A Botnet is a network consisting of devices taken over by Bot software that are connected to the Internet. When a device becomes compromised it becomes part of a Botnet controlled by a Botnet controller (12).

### 1.5.5 Mobile

Mobile devices are devices such as smartphones (13) and computer tablets that can connect to a network wirelessly.

### 1.5.6 Computational Intelligence

Computational intelligence (14) methodologies inspired by nature which are used to solve complex real world problems which traditional (first principles, probabilistic, black-box, etc.) methodologies and approaches are ineffective or infeasible.

### 1.5.7 Artificial Immune System

An artificial immune system (15) is a program that simulates the natural immune system. It is a type of computational intelligence technique that uses learning and memory to solve a problem.

### 1.5.8

### 1.5.9 Bayesian spam filtering

Bayesian spam filtering is a statistical technique that makes use of probabilities to classify email as spam or non-spam (16). This classification is done by correlating the message contents with spam and non-spam messages and then using Bayesian inference (17) to calculate the probability of it being a spam message or not.

### 1.5.10 Artificial Neural Networks

An artificial neural network (ANN) (18) is a flexible mathematical structure which is capable of identifying complex nonlinear relationships between input and output data sets (19). ANN models have been found to be useful and efficient, particularly in problems for which the characteristics of the processes are difficult to describe when using physical equations. The next chapter will detail the structure of this dissertation for the reader.

## 1.6 Layout of Dissertation

This dissertation is structured as follows: Part I is the introductory section and includes this first chapter in which the motivation for the research, the problem statement and goals and objectives of the study are provided. Part II consists of two background chapters which introduce the reader to the technologies and concepts used in this dissertation. Chapter 2 provides the reader with background information on the topics of spam and Botnets so that the reader can better understand the challenges of combating spamming Botnets. Chapter 3 focuses on artificial immune systems, Bayesian filters and artificial neural networks so as to provide the reader with background on the technologies used in the three tier model.

Part III consists of two chapters, namely Chapter 4 and 5. Chapter 4 details the model requirements that need to be addressed by the three tier model and Chapter 5 proposes three models, one for each tier of the three tier model implemented by the author. Part IV contains Chapters 6 and 7 in which the prototype is proposed and implemented. Chapter 6 introduces the reader to the equipment on which the prototypes are implemented and details how the three prototypes are set up. Chapter 7 details the demonstration scenarios that were used to test the three tier model.

Part V contains the concluding chapter, Chapter 8, which contains a brief summary of the extent to which the research problems have been solved and addressed in this dissertation. The chapter is concluded by proposing areas of further research. Chapter 1 concludes with a graphical representation (Figure 1.1) of the layout of the dissertation.

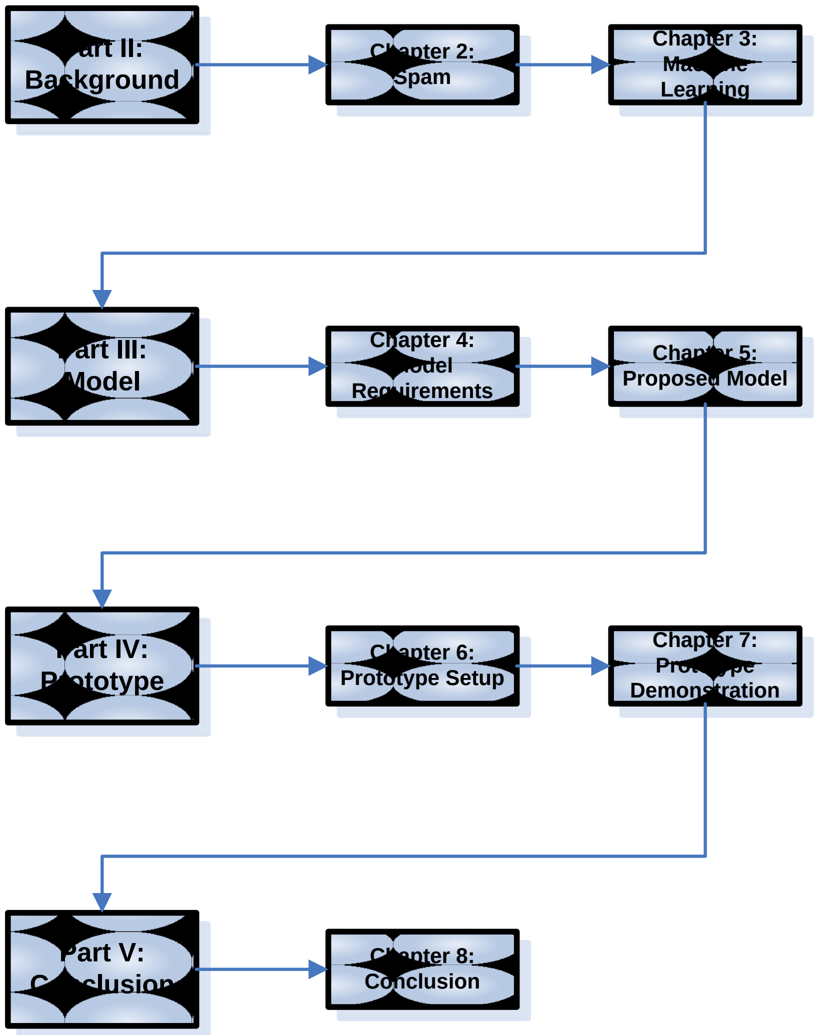

Part II: Background
Chapter 2: Spam
Chapter 3: Machine Learning
Part III: Model
Chapter 4: Model Requirements
Chapter 5: Proposed Model
Part IV: Prototype
Chapter 6: Prototype Setup
Chapter 7: Prototype Demonstration
Part V: Conclusion
Chapter 8: Conclusion

Figure 1.1: Dissertation layout.

# Part II: Background

# 2 Spam

## 2.1 Introduction

This dissertation explores the migration of spamming Botnets to mobile devices therefore it is best to start this dissertation of with background information pertaining to spam and Botnets. Chapter 2 is divided into two major sections. Section 2.2 provides background information on spam, including what it is and why it is a significant problem. Section 2.3, details information about spammer strategies in particular the use of Botnets by spammers to send spam. This chapter is then concluded with Section 2.4.

## 2.2 Spam

Section 2.2.1 provides a definition of spam and, as will be discussed, the definition is not as clear cut as one might assume. This is followed by Section 2.2.2 which discusses the different types of spam including mobile spam which is the subject of this dissertation. Section 2.2.3 discusses how big of a problem spam really is and its cost to society and business. Section 2.2.4 discusses the economics of spam, i.e., the motive for sending spam.

### 2.2.1 The Definition of Spam

Unsolicited bulk communication, also known as spam, is the practise of sending unwanted e-mail messages, often with commercial content, in large quantities to an indiscriminate set of recipients (20). The sending of unsolicited bulk communications with the intention to advertise products and generate sales is economically viable because senders have no operating costs beyond the management of their mailing lists. As the cost of setting up a spamming operation is low, spammers are numerous. Therefore, the volume of unsolicited bulk communications has increased dramatically over the past few years (21).

Spam is generally used to advertise a service or a product. An example of spam is an unsolicited email message from an unknown or forged address advertising Viagra. The exact definition of spam tends to vary but most definitions limit spam to messages sent to many recipients without their consent. There are three categories of unsolicited mail:

- **Unsolicited commercial email** (UCE) consists of advertisements which have been sent out without the request or the consent of the recipients. Examples of UCE include advertisements for medication such as Viagra, pornographic websites, etc.

- **Unsolicited bulk email** (UBE) consists of messages which have been sent to many parties without their request or their consent. Examples of UBE may include scams. A common spam scam involves fake emails which appear to come from a legitimate agency such as a bank requesting verification of credit card number (and other personal details) but instead allow thieves to steal identities.

- **Unsolicited automated email** (UAE) is a related definition used by Graham as another attempt to define what people really see as spam (22). He points out that if a neighbour sent a "For sale" advertisement for something he wanted, he'd be delighted, despite it being both unsolicited and commercial. The use of the word "automated" instead of "bulk" illustrates that a low-volume message may still be spam.

The above mentioned categories, as well as combinations thereof, may be used as a definition for spam. Although these cases seem pretty straightforward, it can be a challenge to define each in sufficiently precise terms for the legal pursuit of offenders.

With the introduction of the "Electronic Communications and Transactions Act, 2002" unsolicited emails now have a legal definition and the sending of spam is illegal (4). Spammers, if identified, are liable for a fine and prosecution.

An interesting thing to note is that for some business dealings, consent actually is implicit. When a user buys something from an online retailer, he doesn't mind when they send a receipt, but would not like to receive a newsletter every day.

Depending on the jurisdiction, the legal definitions of spam varies. Some definitions may exclude mail sent by companies with an existing relationship with the recipient. For example, if I bought a car from a dealership they may be allowed to send me SMSs advertising specials on services as my number was obtained legally. This leads to quite a number of questions as to what is valid communication:

- Was consent to receive electronic mail given?
- What implicit consent is given in a business transaction?
- Can an individual be said to have "solicited" electronic mail from a company if they signed up for an unrelated service from the same company and gave the company permission to send messages related to that service?
- Was the contact details entered in a form that is "opt-out" or "opt-in"?

These are obviously questions for the legal experts so we shall move on to the next section, which introduces the reader to the different types of spam.

### 2.2.2 Types of Spam

Figure 2.1 shows the different types of spam that are commonly encountered today. Email spam is the most common form of spam and the one that most people are most familiar with.

Comment spam is of the kind that inflicts the comments section of newspaper websites, where adverts are inserted in the comments section. Messaging spam, also known as spam over instant messaging (SPIM), is the kind of spam that one would receive over an instant messenger application such as Google Talk, Lync or Whatsapp (23).

Mobile spam, which is discussed in more detail later in this paper, is spam received on one's mobile device in the form of SMSs. Voice over internet protocol (VOIP) spam is the kind of spam that one receives through automated voice messages over a VOIP phone (24).

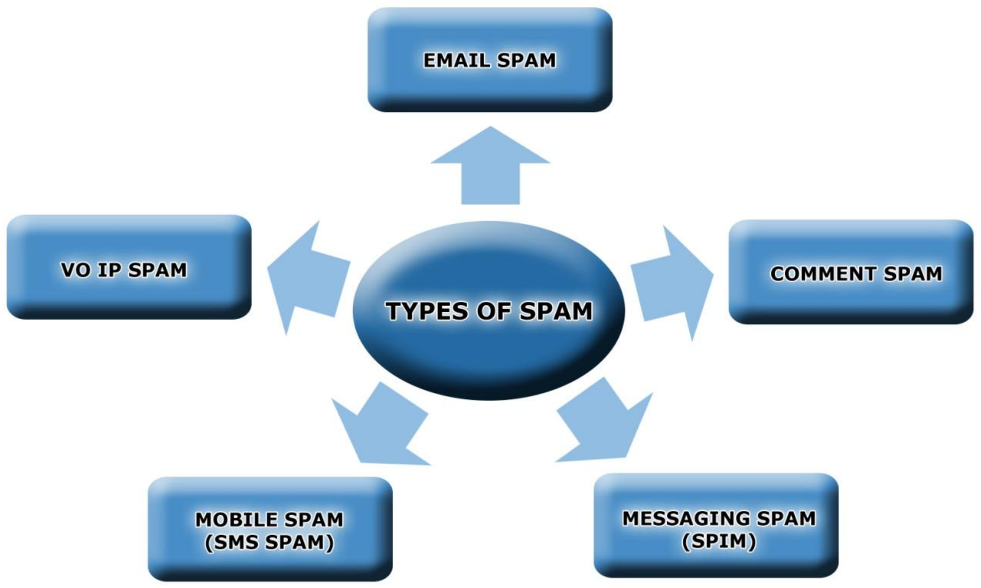


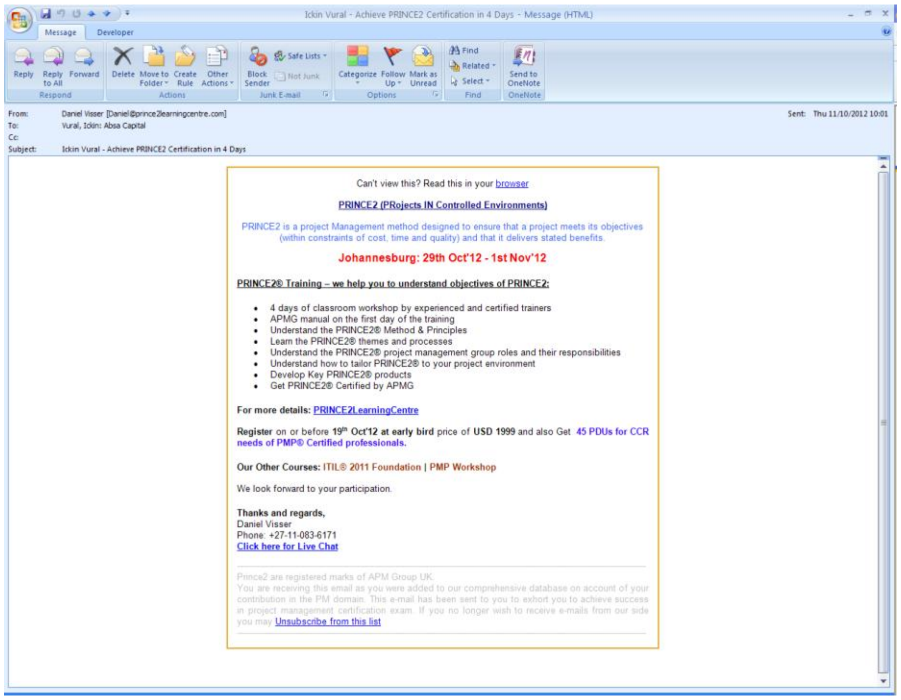
**Figure 2.1 Types of Spam**

The following sections discuss the different types of spam in greater detail in the subsections that follow.

#### *2.2.2.1 Email Spam*

Email is perhaps the most common form of spam. There is an audience for email advertising, regardless of the product that is being sold. Spammers are trying to reach these people. Spammers do not know which people comprise this group. To reach them, they send spam to as many people as possible. This is done because they don't know who will respond to the message and who will not.

For a company, spamming can be very lucrative if done right. For example, take a company that is selling fake Viagra for R 50 per pack. If the company sends out 10 million emails and the response rate is just 0.1% it will make half a million rand. The following figures, Figure 2.2 is a good example of unsolicited commercial email.



From: Daniel Visser [Daniel@prince2learningcentre.com]
Sent: Thu 11/10/2012 10:01
To: Vural, Ickin: Absa Capital
Cc:
Subject: Ickin Vural - Achieve PRINCE2 Certification in 4 Days

Can't view this? Read this in your browser

**PRINCE2 (PRojects IN Controlled Environments)**

PRINCE2 is a project Management method designed to ensure that a project meets its objectives (within constraints of cost, time and quality) and that it delivers stated benefits.

**Johannesburg: 29th Oct'12 - 1st Nov'12**

**PRINCE2® Training – we help you to understand objectives of PRINCE2:**

- 4 days of classroom workshop by experienced and certified trainers
- APMG manual on the first day of the training
- Understand the PRINCE2® Method & Principles
- Learn the PRINCE2® themes and processes
- Understand the PRINCE2® project management group roles and their responsibilities
- Understand how to tailor PRINCE2® to your project environment
- Develop Key PRINCE2® products
- Get PRINCE2® Certified by APMG

**For more details:** PRINCE2LearningCentre

**Register** on or before **19th Oct'12 at early bird** price of **USD 1999** and also Get **45 PDUs for CCR needs of PMP® Certified professionals.**

**Our Other Courses:** ITIL® 2011 Foundation | PMP Workshop

We look forward to your participation.

**Thanks and regards,**
Daniel Visser
Phone: +27-11-083-6171
Click here for Live Chat

Prince2 are registered marks of APM Group UK.
You are receiving this email as you were added to our comprehensive database on account of your contribution in the PM domain. This e-mail has been sent to you to exhort you to achieve success in project management certification exam. If you no longer wish to receive e-mails from our side you may Unsubscribe from this list

**Figure 2.2: Example of Unsolicited commercial mail contents.**

### *2.2.2.2 Mobile Spam*

Spam is not limited to email but also exists in text messaging services (SMS). In the case of SMS, spam can cost even more than it would when received through mail. For example, a user has subscribed to receive a notification via SMS when he/she receives email at her mail account. She will have to pay for every SMS received regardless if it announces a spam or a valid email.

Until recently, mobile networks have been relatively isolated from the Internet. Mobile networks are now well integrated with the Internet and, with the proliferation of smart phones running on operating systems such as Android and Apple iOS (2), threats to the Internet have started to migrate to mobile networks.

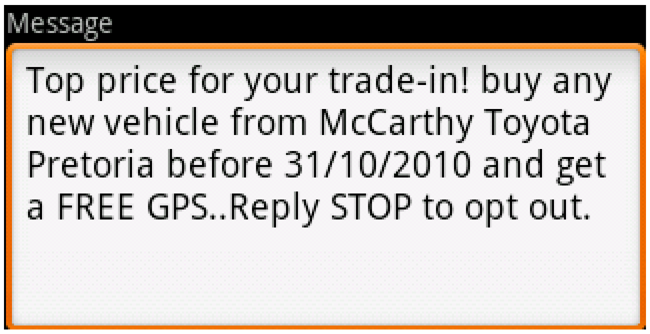
Applications that people access on a desktop computer such as email, audio media and video are now available on mobile computing devices (25). Greater adaptation of these devices will encourage more users to access personal and financial data on their phones. This means that the threats to the Internet, such as spyware, phishing and spam, are migrating to mobile devices.

Desktop operating systems are getting harder to exploit but devices such as cellular telephones (also known as cell phones or mobile phones) have code bases that are largely unexplored with updates to new versions that contain security flaws (26) occurring frequently.

The increased processing power of mobile phones and the growing number of features and applications included with them, make mobile phones an ideal candidate for exploitation by malware. Malware writers are attacking commercial programs for mobile devices.

A malicious program such as J2ME/RedBrowser (27), which is a Trojan horse program, pretends to access WAP web pages via SMS messages. In reality, instead of retrieving WAP pages, it sends SMS messages to premium rate numbers, thus costing the user more than intended.

Figure 2.4 is an example of an unsolicited bulk SMS message received by the author. This message was sent by the dealership where the author purchased his car.



Top price for your trade-in! buy any new vehicle from McCarthy Toyota Pretoria before 31/10/2010 and get a FREE GPS..Reply STOP to opt out.

**Figure 2.2: SMS message received by author**

#### *2.2.2.3 VOIP Spam*

Voice over Internet Protocol (VoIP) is a key enabling technology for transmitting data for communication over a network. VoIP spam (28) is unwanted, automatically-dialled, pre-recorded phone calls using VoIP. It is similar to E-mail spam and is sometimes referred to by the acronym SPIT (29).

The problem of spam in VoIP networks has to be solved in real time as compared to email systems (VoIP). Many of the techniques devised for email spam detection rely upon content analysis and, in the case of VoIP, it is too late to analyse the media after picking up the receiver. The spam calls need to be stopped before the telephone rings this is most likely done by the use of black lists which block numbers known to be used for telephone spam.

#### *2.2.2.4 Messaging Spam*

Messaging spam, also known as SPIM (spam being rapidly outpaced by SPIM), is a type of spam targeting users of instant messaging (IM) services. Instant messaging systems, such as Yahoo! Messenger, AIM and Windows Live Messenger, are targets for spammers.

Many IM systems offer a directory of users, including demographic information such as age and sex. Advertisers can gather this information, sign on to the system and send unsolicited messages, which could include commercial scam-ware, viruses and links to paid links for the purpose of click fraud.

#### *2.2.2.5 Comment Spam*

This is a type of spam that occurs when spammers post adverts on the comment section of websites (30). It is also commonly referred to as news group spam. This type of spam is particularly attractive to spammers as it allows them to target a specific demographic with their adverts. For example, spam adverts placed on The Economist magazine would reach a different audience as opposed to adverts placed on Sports Illustrated. The next section describes the cost of spam for both mobile device users as well as network service providers.

### 2.2.3 The costs of Spam

Spam is one of the most significant threats to the internet and accounts for around 60% of all email traffic (8). Spam costs consumers and ISPs lots of money in bandwidth charges.

The costs of spam include lost productivity and fraud. These costs are borne by the general public, institutions that store and retrieve mail for their employees and by internet service providers. Institutions and internet service providers have been forced to add extra capacity to cope with the high volumes of unsolicited bulk communications (31).

As discussed in Section 2.2.4, mobile spam may be even more costly with recipients paying premium rates to respond to SMS messages received. An example of this is when recipients reply "Opt-Out" to these messages. Despite the growing number of technological means for combating spam, the spammers somehow manage to stay one step ahead and the deluge, which shows little sign of abating.

Spam is a problem that has attracted the attention of many organisations, and governments. Despite the efforts of companies and governments to reduce spam, the amount of spam sent and received daily continues to rise. The section that follows discusses the economic rationale behind spam i.e what is the profit motive for spammers.

### 2.2.4 The Economics of Spam

Spam makes money for those who send it, as its cost versus benefit ratio is so low. If even a small percentage of spam recipients respond to the advertised product, spammers will make money.

Spammers generally do not pay much for sending spam. They accomplish this by exploiting open mail servers that do their task for them. The spammer need only send one email message to an incorrectly configured mail server to reach thousands of email addresses, with the bulk of the transfer being handled by the misconfigured mail server. Recipients, in turn, need to pay access costs or telephone costs in order to receive content they didn't ask for. ISPs have to bear the bulk of the cost for bandwidth overuse by spammers. This cost is often passed on to the consumer through increased internet access fees or a degraded level of service.

With the introduction of the "Electronic Communications and Transactions Act, 2002", unsolicited emails now have a legal definition and the sending of spam is illegal (4). Spammers, if identified, are liable for a fine and prosecution. Spammers attempt to cover their trail to prevent identification. This will be discussed in greater detail in the next chapter. Now that we have introduced the reader to the consequences of spam and sending spam, the next chapter will discuss the way in which spammers send spam.

## 2.3 Spammer strategies

To fully understand the threat posed by spamming it is necessary to provide some background on spammer strategies, particularly the use of Botnets. Section 2.3.1 gives an overview of spammer strategies. The process of obtaining email addresses and mobile numbers is discussed in Section 2.3.2 and Section 2.3.3 discusses how spammers conceal their identities so as to evade detection and possible prosecution. Section 2.3.4 expands on this by further exploring mobile Botnet which is the core topic of this dissertation.

### 2.3.1 The process of obtaining email addresses and mobile numbers

In order to reach many people, spammers must first gather email addresses or mobile numbers to use. Obtaining an electronic address (this could be an email address or mobile number) is the most important requirement for sending spam. There are a number of ways to gather this information. A few examples are listed as follows:

1. Email addresses and mobile numbers can be harvested from public locations (37).
2. Email addresses or mobile numbers can be guessed.
3. Social engineering which is process by which a hacker deceives others into disclosing valuable data that will benefit the hacker in some way (32).
4. Buying lists from those who have harvested email addresses and mobile numbers (39).
5. Hacking into IT systems and stealing this information (40).

The following section describes the most commonly used method of sending spam while simultaneously concealing the spammer's identity.

### 2.3.2 How Spammers conceal their identities

Spammers conceal their identities for a number of reasons. If they are based in a jurisdiction which has strict anti-spamming laws, they do not want to be traced for fear of prosecution. If they are based in a jurisdiction which has weak anti-spamming laws then the primary motive is not to be traced and blacklisted as many ISP's will block any mail from blacklisted sites.

The techniques studied in this section are spoofing, bot-networks, open proxies and untraceable internet connections.

#### *2.3.2.1 Spoofing*

Spoofing is the process whereby a spammer would insert fictitious headers into the email address to hide the network address of their computer. The spammer will usually insert fake "From" and "Reply-To" headers into the email. These headers would point to a non-existent network address or, more commonly, an innocent third parties' network address (33).

SMS spoofing occurs when a sender manipulates address information. Often it is done in order to impersonate a user that has roamed onto a foreign network and is submitting messages to the home network. Frequently, these messages are addressed to destinations outside the home network, with the home SMSC essentially being “hijacked” to send messages into other networks. The impact of this activity is threefold:

- The home network can incur termination charges caused by the delivery of these messages to interconnect partners. This is a quantifiable revenue leakage.
- These messages can be of concern to interconnect partners. Their customers may complain about being spammed or the content of the messages may be politically sensitive. Interconnect partners may threaten to cut off the home network unless a remedy is implemented. Home subscribers will be unable to send messages into these networks.
- While fraudsters normally used spoofed-identities to send messages, there is a risk that these identities may match those of real home subscribers. The risk, therefore, emerges that genuine subscribers may be billed for roaming messages they did not send. If this situation occurs, the integrity of the home operator’s billing process may be compromised, with potentially huge impact on the brand. This is a major churn risk.

Not all spoofing is malicious, there are also legitimate use cases for SMS spoofing, the legitimate use cases for SMS spoofing include:

- A sender transmits an SMS message from an online computer network for lower, more competitive pricing and for the ease of data entry from a full size console. They must spoof their own number in order to properly identify themselves.
- A sender does not have a mobile phone and they need to send an SMS from a number that they have provided the receiver in advance as a means to activate an account.
- A sender adopts the default network gateway identifier provided by an online service, in order to send an anonymous SMS rather than specifying a number of their own choosing.
- A third party sends a message to a virtual number, which then forwards (resends) the message to one or more recipients in such a way that the true originator address (rather than the virtual number) appears as the sender ID and the recipient(s) can

reply, call, sort, save, or otherwise process the message in the expected way.

An SMS spoofing attack is often first detected by an increase in the number of SMS errors encountered during a bill-run. These errors are caused by the spoofed subscriber identities. Operators can respond by blocking different source addresses in their Gateway-MSCs, but fraudsters can change addresses easily to by-pass these measures. If fraudsters move to using source addresses at a major interconnect partner, it may become unfeasible to block these addresses due to the potential impact on normal interconnect services

#### *2.3.2.2 Botnets*

A Bot-Network consists of a set of machines that have been taken over by a spammer using Bot software sent over the Internet. This Bot software hides itself on its host machine and periodically checks for instructions from its human Bot-Network administrator.

Today, Botnets are often controlled using Internet Relay Chat (34). The owner of the computer usually has no idea that their machine has been compromised until its Internet connection is shut down by an ISP.

As most ISPs block bulk mail if they suspect it is spam, the spammers who control these Bot-Networks typically send low volumes of mail at any one time so as not to arouse suspicions. Thus, the spam mail can be traced to an innocent individuals network address and not the spammers network address.

While the number of Botnets appears to be increasing, the number of bots in each Botnet is actually dropping. In the past, Botnets with over 80 000 machines were common (34). Currently, Botnets with a few hundred to a few thousands infected machines are common. One reason for this is that smaller Botnets are more difficult to detect and may be easier to sell or rent. When Procter & Gamble ran a security check of its 80,000 PCs, it found that 3,000 were infected with Botnet software (35). The section that follows describes Botnets that have spread to mobile devices.

#### *2.3.2.3 Open Proxies*

An open proxy is a machine that allows computers to connect through it to other computers on the Internet. Open proxies exist because they enable unhindered Internet usage in countries that restrict access to certain sites for political or social reasons.

An Internet user in a country that restricts Internet access can access blocked sites by using an open proxy in a country that does not restrict Internet access. Spammers use open proxies to hide their network addresses. The recipient of a spammers' email will not see the spammers' network address on the email but the open proxy's network address. It is estimated that 60% of all spam is sent using an open proxy (33).

#### *2.3.2.4 Open mail relays*

Emails sent over the Internet pass through a number of gateways on their way from the sender to the receiver. These gateways are called mail relays. Each time an email passes through a mail relay it has a Received header inserted, this will have the network address of the computer that connected to the mail relay.

An open mail relay is a misconfigured mail relay that accepts mail from any computer on the Internet and forwards it to any other computer on the Internet as opposed to a normal mail relay that accepts mail from a limited number of computers on the Internet and forwards it to a limited number of computers (36). This helps the spammer conceal his identity as it appears that the mail is from the open relay and not from the spammer. However, as the spammers network address is still found in the emails headers the spammer would insert fake headers into the email.

Open mail relays are usually used together with open proxies to conceal the network address of the spammer.

##### *2.3.2.5 Untraceable Internet connections*

Spammers can also conceal their identities (45) by accessing the Internet from Internet cafes, university computer labs and by using stolen 3G cards. Thus, there is no way of tracing spammers who access the Internet using these methods. Even if the network address of the computer used is identified this cannot be connected to the identity of the spammer.

### 2.3.3 Botnets on mobile networks

As this dissertation investigates Botnets on mobile networks the Author will expand upon this method of sending spam. Sending mobile spam via a Botnet serves two purposes. The first purpose is to conceal the identity of the spammer, the second is to pass the cost of sending spam SMSs to the mobile device owner.

The connection between the Internet and mobile devices acts as a gateway for malware to move from the Internet to mobile networks. As more and more financial transactions takes place over mobile devices this puts valuable information at risk. The challenge for businesses and banks in the near future will be to produce secure mobile applications while ensuring ease of use at the same time (37).

An implementation that would enable users to identify Botnets on their mobile devices or spam being sent from mobile devices would slow the emergence of SMS spam and enhance the security of devices on a network. Now that the reader has been made aware of the various spammer strategies the following section concludes the chapter.

## 2.4 Conclusion

This chapter provided the reader with background information on spam and Botnets. Section 2.2 described spam, including what it is and why it is a significant problem. The second part of Chapter 2, Section 2.3, detailed information regarding spammer strategies, particularly the use of Botnets by spammers to send spam. The dissertation continues with Chapter 3, which provides the reader with some background on various machine learning techniques that are used by the author to combat spamming Botnets.

# 3 .Machine Learning

## 3.1 Introduction

This dissertation explores ways to combat spamming Botnets on mobile devices. The author feels that some background information on machine learning techniques, which will be used by the author to combat spamming Botnets, is required. This chapter will focus on the three machine learning techniques implemented by the author to detect spamming mobile Botnets.

Chapter 3 is divided into three major sections. Section 3.2 describes some background information on artificial immune systems, Section 3.3 details information about Bayesian filters. Section 3.4 provides the reader with insight into artificial neural networks. The chapter is then concluded with Section 3.5.

## 3.2 Artificial Immune Systems

This Section gives an overview of artificial immune systems (AIS). The author implemented one of the prototypes using an AIS, therefore, before continuing with this dissertation it is important that the reader has a good understanding of artificial immune systems.

The rest of this section is structured as follows: Section 3.2.1 provides an introduction to artificial immune systems. Section 3.2.2 describes how the natural immune system can be modelled by artificial immune systems to enable systems to learn to classify data adaptively.

### 3.2.1 The Artificial Immune System

This subsection explains the components of the "spam immune system" and how they work together to learn and unlearn message signatures in order to classify them as spam or non-spam. Artificial immune systems can be used to model many systems, but for the purpose of this dissertation, we will look at how an artificial immune system is modelled to classify spam.

The AIS is modelled on the natural immune system (NIS). The author will not go into too much detail on the NIS. Further background information on the natural immune system can be found in the Appendix A.

The goal of a NIS is to differentiate between self cells and potentially dangerous non-self cells. Self cells are cells naturally belonging to the body and non-self cells are foreign cells, such as viruses and bacteria, which might threaten the body.

In an anti-spam system one needs to differentiate between self (legitimate) messages and non-self (spam) messages. Theories of how the NIS works, can serve as a starting point for creating computer systems as many of the techniques used in artificial intelligence are inspired by the principles and processes found in nature.

It is the classification of self and non-self cells that makes the NIS an appealing model for spam detection, which also requires a classification between the legitimate (self) messages and spam (non-self) messages.

The aim of the artificial immune system implementation that the author developed (as will be detailed later in the  study) is to alert the user or network provider if the mobile device is being used to send SMS spam. The first subsection describes how the SMS signature is modelled as an antigen. This is followed by a subsection explaining how the artificial immune System distinguishes between Spam and non-spam SMS messages. Table 1 has been included to explain some of the terms used in this study.

**Table 3-1 Glossary of terms**

| Term | Explanation |
|---|---|
| Self | Refers to a valid message (i.e. not spam). |
| Non self | Refers to an invalid message (i.e. spam). |
| Antigen | Is a message signature that is not recognised as being self and is thus possibly spam. |
| Pathogen | Is a message positively identified as being spam. |
| Signature | The digital representation of an SMS message. |
| Signature library | Storage module in which signatures are stored. |

#### *3.2.1.1 Signature: SMS Message and Signature library*

In the context of this domain, an antigen refers to the target or solution that is potentially bad, e.g., whether the message we need to check is spam (38). Antigens have to be encoded in some way to be represented in digital form. This could be in binary or real number format. For the implementation of the artificial immune system in this study, the SMS message was converted to a digital representation (signature) for the system to process.

This was done by reading in the message and identifying characteristics such as the number of capital letters used, the length of the message, use of punctuation and the presence of URLs. It is by storing and analysing these messages that a profile of the user's SMS sending behaviour can be built. The selection of the characteristics to be stored is critical in building a valid profile of the user.

The AIS prototype  also has a signature storage module in which randomly-generated signatures are stored. This is called the signature library. The library is initially populated with randomly generated signatures as a starting point. This practice is common with many artificial intelligence techniques. The library changes with each successive generation by a process of selection until we have a set of signatures that represent an optimum solution. This process is explained in Figure 3.1.

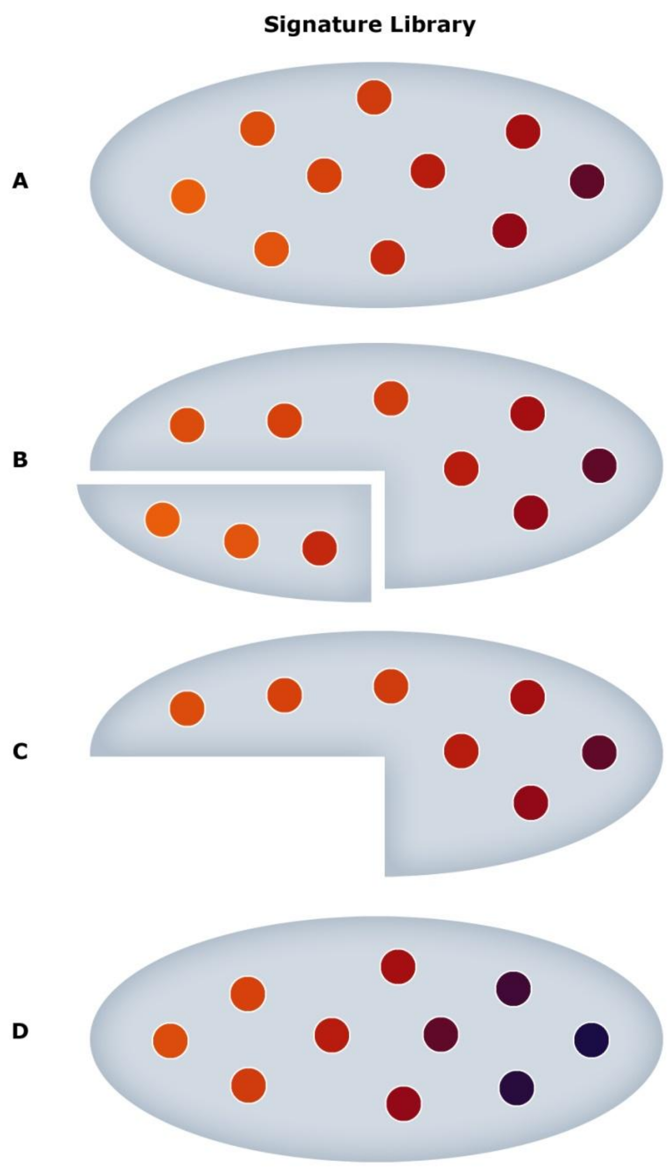


Figure 3.1 : Signature Library

As shown in Figure 3.1 A, the signature library is first populated by randomly generated signatures, each represented by a red dot. This list is then pruned to remove all signatures that match self (non-spam) signatures, as shown in Figure 3.1 B, with randomly generated new signatures being generated in order to replace the signatures that were removed (see Figure 3.1 D).

The newly-generated signatures, as well as those retained from the previous iteration, will form the next generation of the signature library. This process is repeated until the signature library contains a set of signatures that match non self (spam) signatures, indicating that the signature library has now reached an optimum. The following section describes the self and non-self messages.

#### *3.2.1.2 Self and non Self*

A message that is identified as non-self, i.e., sent by someone else (48), should be classified as spam by our AIS system. The user of our anti-spam solution should have the opportunity to provide input to our system, indicating whether a message constitutes spam or not. Thus, the system learns what spam is by initially basing its decisions on the user's input.

The system should then be able to identify a message as spam when a message is encountered that is not usually sent by the user. The system should be adaptive so as to cater for changes in users' sending behaviour and learn to relearn what constitutes spam in the case that a user's sending behaviour might change due to, for example, social influences.

The AIS implementation should then still be able to identify an SMS sent by the user as a valid SMS, even though the user's sending behaviour might be changing. It is essential that the user correctly identifies to the system that a legitimate message is a non-spam message, as opposed to identifying a legitimate message as a spam message.

For example, a user may tolerate messages incorrectly identified as spam (which the system will then learn to identify as non-spam), but the user will be less tolerant of spam messages being incorrectly marked as non-spam. The section that follows describes how the affinity measure is calculated as well as how the antibodies are digitally represented in our AIS.

#### 3.2.1.3 Affinity measure and Digital Antibodies

The affinity measure or similarity measure is used to determine how similar a message signature is to a signature library of legitimate (self) message signatures. Such libraries of message signatures are also known as digital antibodies (49). As explained in Section 3.2.1.4, these antibodies are randomly generated by the system and matched to antigens using a Euclidian distance formula.

The Euclidian distance formula is, in this case, acting as our fitness function by determining how closely our antibodies bind to their target. These digital antibodies are thus used to determine whether or not an SMS message is a self (non-spam) or non-self (spam) message. The Euclidian distance formula, in mathematical terms, is the ordinary distance between two points that one would measure with a ruler (39).

The Euclidian distance was selected for the AIS prototype implemented in this study, but other methods, such as the Pearson Correlation Coefficient (40), can be used instead. The aim of the latter, which is similar to using the Euclidian Distance, is to match the digital representation of the message signature with the digital signatures in our signature library.

#### 3.2.1.4 Detection binding

Detection of foreign objects in the NIS of the body is done through binding (52). This means that antigens are the target which antibodies attack (bind to). The immune system's antibodies may bind to many different antigens, although some will bind more closely than others. The binding process does not always happen due to an exact match but can also bind due to a partial match known as approximate binding.

The antigens to which a given antibody will bind, must, therefore, be similar in shape but do not need to be exactly compatible. A given detector (antibody) can bind to many cells and a given cell might have multiple detectors which can bind to it. The strength of the binding depends on how closely the two shapes can match. In a nutshell, the antibodies will attack the antigens (in the case of spam, the message is "attacked") and determine if it is a pathogen (non-self) (53). In the case of spam, the spam message that needs to be destroyed constitutes the pathogen. In the case that it is not a pathogen, i.e., the message is not spam, the message is left alone.

There are approximately $10^{16}$ different foreign proteins which the NIS must recognize, yet the repertoire of the NIS contains a much smaller number of actual receptor (in our case, signatures in the signature library) types, closer to $10^{8}$ (41). This is accomplished by using approximate binding. Thus, the immune system can use a smaller number of antibodies to detect a large number of potential pathogens, as long as pathogens have similar shapes.

Figure 3.2 shows an example of binding and non-binding. In the first example the antibody on the left binds to the antigen on the right (i.e., in the case of spam, identifies it as spam). In the second example the antibody on the left does not bind to the antigen (thus, in the case of spam, it is not identified as spam). Just because an antibody does not bind to an antigen does not mean the antigen is not a pathogen (actual spam), but that the immune system has not learnt to recognise it as such yet.

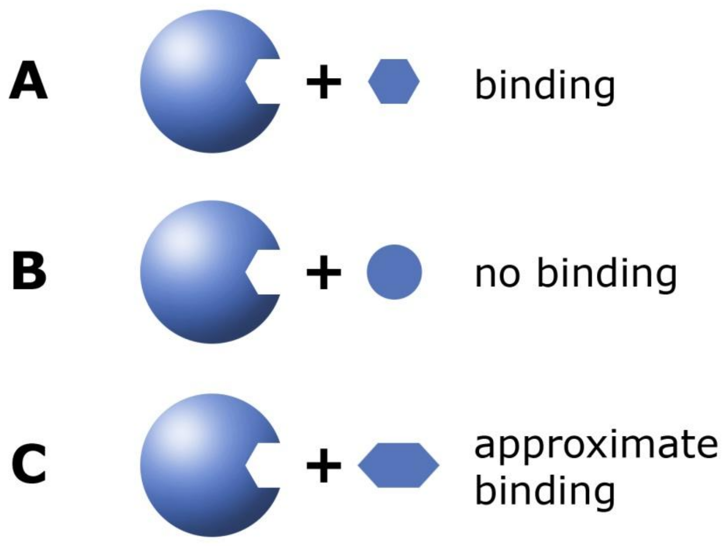


Figure 3.2: Detection binding

The AIS should, as is the case with the NIS, use approximate binding to identify spam messages as this would reduce the number of antigens that need to be stored in the signature library. Section 3.3 will now introduce Bayesian filtering, which is used by the author as part of the anti-spam solution.

## 3.3 Bayesian filtering

This section gives the reader an overview of Bayesian filtering which is one of the techniques used by the author to combat spamming Botnets. As this study explores the detection of spamming Botnets, the use of Bayesian filtering in relation to spam will be discussed.

Firstly, Bayesian filtering is introduced in Section 3.3.1. This is then followed by an explanation of Bayesian filtering applied to mobile spam SMSs in Section 3.3.2.

### 3.3.1 Bayesian filtering

Bayesian spam filtering is a statistical technique that makes use of probabilities to classify email as being spam or non-spam (42). This classification is done by correlating the message contents with spam and non-spam messages and then using Bayesian inference (43) to calculate the probability of it being a spam message or not.

The process works as follows. Some words will have a particular probability of occurring in spam messages and legitimate messages. The spam filter will not know the probabilities of these words occurring in advance, but is trained to learn the probabilities by use of both black and white lists. Black lists contain message samples that are definitely spam and white lists contain message samples that are definitely legitimate.

These black and white lists are built up manually by the user. This is accomplished by adding available spam messages to the black list and available legitimate messages to the white list. The following figure, Figure 3.3, shows how the spam and ham lists are compiled.

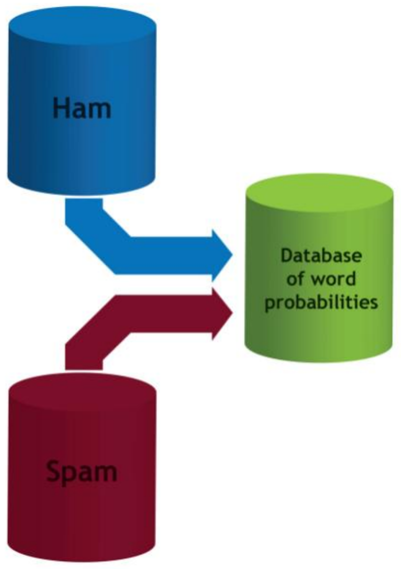


Figure 3.3: Compiling ham and spam lists for a Bayesian filter

The size of the dataset used to train the filter is important as filter performance climbs with increased data to train upon (22). The size of the dataset used to train the filter in this experiment was limited by the number of legitimate and spam SMSs the author was able to collect. The data selection process is expanded upon later in the study. The section that follows discusses the application of Bayesian filtering to mobile spam.

### 3.3.2 Bayesian filtering applied to mobile spam

Having a good term representation is one of the most important parts for getting a good classifier (44). The spam filter should be designed with the SMS message characteristics in mind, namely only 160 characters are allowed on a standard SMS message (44).

The problem with this limitation is that fewer words mean less information can be transferred. Also, due to this constraint, people tend to use acronyms when writing SMSs (44). SMS messages lack structured fields and their text is rife with abbreviations and idioms (45). Moreover, the abbreviations used by SMS users are not standard for a language, but they depend on the user communities.

Such language variability provides more terms to the word collection. For example, the sender of an SMS could replace the word “tomorrow” with the abbreviation “tmw”. This would usually be understandable to the SMS recipient, but there would now be two permutations of the word “tomorrow” to analyse, hence, expanding the collection of words for which their spam probability needs to be calculated. Figure 3.4 shows how the Bayesian classifier can analyse a message and then output a probability that indicates weather the message is spam or ham (it may sometimes be unable to make a clear determination).

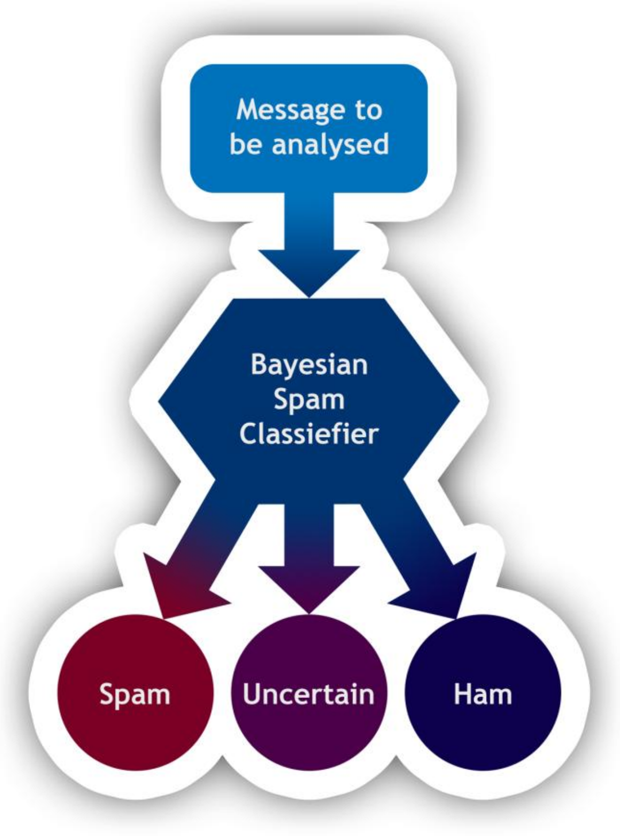


**Figure 3.4: A model for a Bayesian spam classifier.**

Now that the reader has been made introduced to Bayesian the next section will introduce the reader to artificial neural networks which is one of the machine learning techniques implemented by the author to test spam detection on mobile devices.

## 3.4 Artificial neural networks

Artificial neural networks (ANN) are computational methodologies that perform multifactorial analyses. Inspired by networks of biological neurons, ANN models contain layers of simple computing nodes that operate as nonlinear summing devices (46).

These nodes are richly interconnected by weighted connection lines and the weights are adjusted when data is presented to the network during a "training" process. Successful training can result in artificial neural networks that perform tasks such as predicting an output value, classifying an object, approximating a function, recognizing a pattern in multifactorial data (60) and completing a known pattern.

The following subsections provide the reader with greater detail on how artificial neural networks are modelled. Section 3.4.1 explores the concept of learning, Section 3.4.2 then details the pros and cons of using ANN. Section 3.4.3 discusses the types of ANN. Section 3.3.4 explains how neural networks are trained.

### 3.4.1 Learning

Learning is a process in which different events are associated with different outcomes, i.e., substantiating the cause and effect principle. Before introducing the reader to ANN the author felt that the reader required some detail on machine learning. This section will examine the different types of learning mechanisms.

#### *3.4.1.1 Learning through association*

Learning through association is simply learning through the cause and effect principle. One example of learning by association is Boolean algebra (47). Boolean algebra is the logical association between truth and falsehood. The logic of truth and false can be represented by AND, OR and NOT operators. Almost all arithmetic expressions can be represented by Boolean algebra and modern computers use Boolean logic for their operation. For example, consider the statement a person must be at least 18 years old to drive a car, from this we can rationally deduce that someone driving a car is at least 18 years old.

Another example of association is decision tree algorithms (48). Every node on the tree is evaluated and a decision is made as to which child node to proceed to until a solution to the problem is found. This method will fail when the parameters on the node being evaluated do not produce a sharp distinction so as to allow the algorithm to know which child node to evaluate. In cases where it is some correlation between the parameters themselves that means it is better to not just evaluate them but their combinations as well. A neural network is designed to handle such situations.

In general, decision making is an attempt to find the best association between known features and known outcomes, by assigning a certain weight to each association we can select the association that is most likely.

#### *3.4.1.2 Feature identification*

Identification of useful features that will be used to build associations is a pertinent issue in machine learning. A good feature is one that produces proximity to some unique region in the feature space, leading to the formation of a cluster of similar objects in the region.

The greater the separation between the clusters of objects in the feature space the better the parameter. This separation is usually measured as a distance measure. A commonly used distance measure is the Euclidean distance formula. The Euclidian distance formula, in mathematical terms, is the ordinary distance between two points that one would measure with a ruler (49).

#### 3.4.1.3 *Artificial Neural Network Learning*

An artificial neural network models a system based on the information fed into it. If we build a good model it should be possible for the network to predict the correct output from the inputs. Real systems can be very complex and thus difficult to duplicate. Neural networks can, if given enough data, correctly model the system that produced the data. Neural networks can be one of two types: those that use supervised learning and those that use unsupervised learning (50).

Supervised learning networks learn from training data that contains many examples of possible inputs and their corresponding outputs from a real system. Thus, the network attempts to mimic the training data.

For unsupervised learning, the training data consists of a collection of patterns with no distinction between inputs and outputs from this data the network attempts to group the patterns into different clusters. Thus the network makes a self-evaluation of the possible sources of the variants in the data. The following section discusses the pros and cons of using ANN.

### 3.4.2 Pros & Cons of using Artificial Neural Networks

This section details the pros and cons of using ANN to classify data. The advantages and disadvantages are listed below.

Advantages:

- A neural network can perform tasks that a linear program cannot (65).
- When an element of the neural network fails, it can continue without any problem by their parallel nature.
- A neural network learns and does not need to be reprogrammed (66).

Disadvantages:

- The neural network needs training to operate (67).
- Requires high processing time for large neural networks (68).

Another aspect of artificial neural networks is that there are different architectures, which consequently require different types of algorithms. While they appear to be complex systems, neural networks are relatively simple to implement. The following subsection introduces the reader to the two most common neural networks.

### 3.4.3 Types of Artificial Neural Networks

There are two main types of artificial neural networks:

- **Feed-forward neural networks**, where the data from input to output units is strictly feed-forward (69). The data processing can extend over multiple (layers of) units, but no feedback connections are present, that is, connections extending from outputs of units to inputs of units in the same layer or previous layers.
- **Recurrent neural networks** that do contain feedback connections (70). Contrary to feed-forward networks, the dynamical properties of the network are important. In some cases, the activation values of the units undergo a relaxation process such that the neural network will evolve to a stable state in which these activations do not change anymore. In other applications, the change of the activation values of the output neurons is significant, such that the dynamic behaviour constitutes the output of the neural network (51).

The following figure, Figure 3.5, shows the different layers in a neural network.

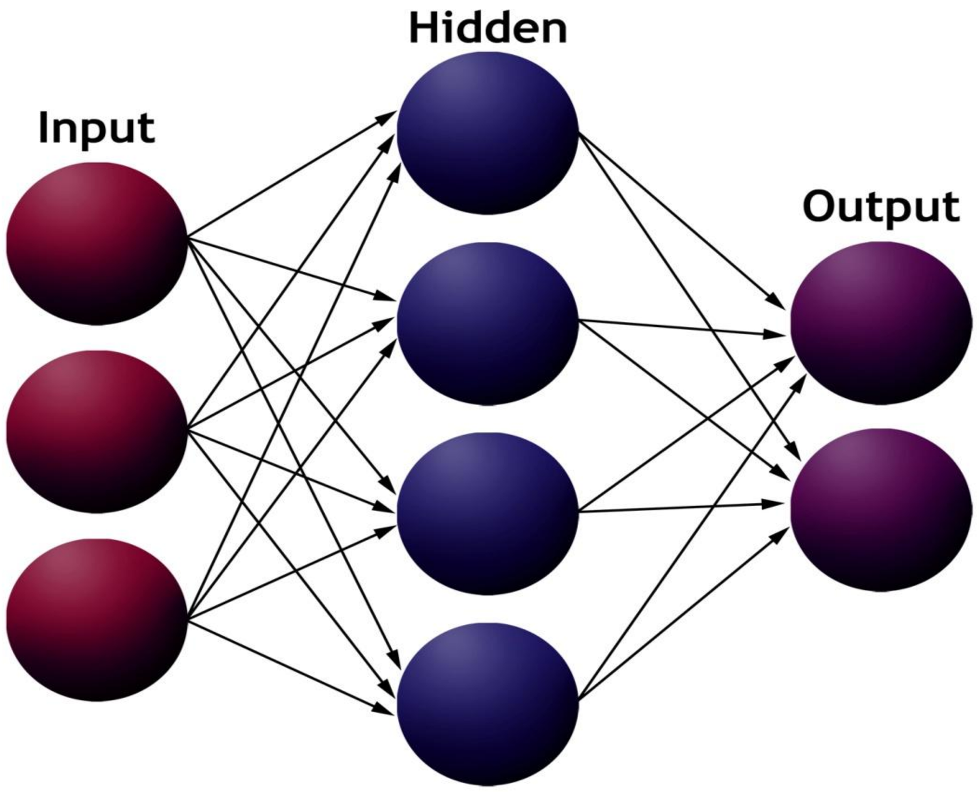


Figure 3.5: Layers in a neural network.

The following subsection details how ANNs are trained.

### 3.4.4 Training Artificial Neural Networks

An artificial neural network has to be configured such that the set of inputs produces the desired set of outputs. One way to do this is to 'train' the neural network by feeding it teaching patterns and letting it change its weights according to some learning rule. We can categorise the learning situations in three distinct sorts. These are:

- **Supervised learning** or Associative learning, in which the network is trained by providing it with input and matching output patterns (72). These input-output pairs can be provided by an external teacher, or by the system which contains the neural network (self-supervised).

- **Unsupervised learning** or Self-organisation, in which an (output) unit is trained to respond to clusters of pattern within the input (73). In this paradigm, the system is supposed to discover statistically salient features of the input population. Unlike the supervised learning paradigm, there is no a priori set of categories into which the patterns are to be classified.Rrather, the system must develop its own representation of the input stimuli.

- **Reinforcement Learning.** This type of learning may be considered as an intermediate form of the above two types of learning (74). Here, the learning machine does some action on the environment and gets a feedback response from the environment. The learning system grades its action good (rewarding) or bad (punishable), based on the environmental response and accordingly adjusts its parameters. Generally, parameter adjustment is continued until an equilibrium state occurs, following which there will be no more changes in its parameters. The self-organising neural learning may be categorised under this type of learning. Now that the reader has been introduced to the training of artificial neural networks the next section concludes the chapter.

## 3.5 Conclusion

Chapter 3 gave the reader detailed information on the three machine learning techniques used by the author in this study. Section 3.2 described the artificial immune systems, Section 3.3 detailed information on Bayesian filters. The final section, Section 3.4, provided the reader with background on artificial neural networks.

Part III will introduce the reader to the three models developed by the authors to combat spam. These models will be based on the three machine learning techniques introduced in this chapter.

# Part III: Model

# 4 Model Requirements

## 4.1 Introduction

Part II of this dissertation provided the required background knowledge needed to proceed with the research. This background information puts one in the position to understand the problem that this study focuses on, namely, the migration of spamming Botnets onto mobile devices. As one can imagine, with the proliferation of mobile devices mobile security has become increasingly important.

In previous research (52) the researcher came to understand that the majority of spam on a network is sent by Botnets. This lead the researcher to investigate the possibility of Botnet detection using artificial intelligence and network forensics techniques (53). This research was later expanded to the study the detection of mobile Botnets (54).

During the course of this research the researcher came up with a model that used anomaly detection to detect Botnets (55). The use of an artificial immune system installed on a mobile device (56) addressed privacy concerns regarding data privacy but resulted in the degradation of service in sending SMSs due to processing power limitations on mobile devices. To address the processing and data storage limitations of mobile devices, the researcher came up with an implementation using neural networks to detect spamming Botnets that could be installed on a network provider's server farm (57). This followed on earlier work that tested the accuracy of Bayesian filters in detecting mobile spam (58).

Due to the above assumptions and reasoning, the researcher decided that it would be more cost-effective, in terms of processing and storage requirements, as well as time-effective, in terms of processing speed, to propose a concept in which security (spamming Botnet detection) is implemented at a network server level.

Due to the above assumptions, with regards to privacy the researcher also decided to propose a model in which security is implemented on a mobile device. Thus, this study examines both the detection of spamming Botnets at a mobile device level and the detection of spamming mobile Botnets at a mobile network operator level.

For the purpose of demonstrating spamming Botnet detection with the use of machine learning techniques, the author implemented three models. These three models are described in greater detail in Chapter 5.

The first model proposed by the author is for an AIS installed on a mobile device. The second proposed model is for a Bayesian filter to be installed on a network provider's server. The third model is for a neural network installed on a network provider's server.

Firstly, before detailing the three models to be implemented, one should focus on the unique requirements of spamming mobile Botnet detection. These will be examined in depth in the section to follow Section 4.2. The chapter is then concluded with Section 4.3.

## 4.2 Requirements for mobile anti Botnet solution

There are many important factors that need to be considered when implementing a spamming Botnet detector. The factors that are addressed in this study are the following:

- Learning with positive and negative data.
- Message detail.
- Remote analysis vs. analysis on a device.
- Privacy.
- Performance.

The factors listed above are the main ones that the author thought it important to detail. The reasoning behind the choice of these factors will become apparent in the subsections to follow, where each factor is addressed individually. It is, however, important to remember that the core of the argument regarding the importance of these factors concerns the manner in which they influence the design decision of how to implement a security application for SMS spam.

### 4.2.1 Learning with negative and positive data

This section describes the data that will be used to train the Botnet detectors to differentiate between spam SMSs and valid SMSs. The following two subsections, Sections 4.2.1.1 and 4.2.1.2, will explain, in greater detail, the use of negative and positive data in machine learning.

As will be described artificial immune systems can be trained with positive data only, while Bayesian filters require both positive and negative data. Artificial neural networks can be trained with positive data only, but as the prototype the author implemented will use positive and negative data the author has grouped it with Bayesian filters under learning with positive and negative data.

#### *4.2.1.1 Learning with positive data – Artificial Immune Systems*

A unique characteristic of AISs is that training only requires positive examples. What is meant by this is that the AIS implementation only needs to be trained on positive data (valid messages), i.e., the training set only needs to consist of valid SMS messages and not negative data (spam messages). Thus, there is no need to train the AIS implementation to learn what a spam message will look like as the AIS will deduce this by figuring out what constitutes valid SMS signatures.

This is ideal when a profile of what the spam SMSs will look like does not exist and, thus, the AIS cannot be provided with spam SMSs to train on. After training, the AIS should have the ability to classify non-self (spam) patterns from the self (non-spam) patterns. This makes the AIS an ideal candidate for classification problems where only one class of pattern is available for training.

When building the signature library, which stores the antigens, the choice was between targeting valid SMSs and spam SMSs, as one cannot possibly build a profile of all the different types of SMS spam. This is due to the fact that, what constitutes Spam is forever changing. Also, a list of all possible Spam profiles would be very large. In addition, there is a question of where SMS spam would be obtained from.

The approach followed by the authors in this implementation was to build a profile of the user's SMS-sending behaviour as this could be done by storing and analysing message signatures. The prototype would be trained on the user's messages, learning to identify messages as self. Thus, everything else becomes non self, i.e., spam.

#### *4.2.1.2 Learning with negative and positive data – Neural networks and Bayesian filters*

The detection of SMS spam is a typical classification problem where patterns, also known as signatures, need to be classified as legitimate or not. Thus, it can be regarded as a binary classification problem in which the data set, consisting of SMS messages, are classified as either spam or non-spam.

Artificial neural networks and Bayesian Filters are ideal for this sort of classification problem as we can train them on legitimate (positive) and non-legitimate (negative or spam) messages. From this data, the neural network and Bayesian filter should be able to deduce whether messages are spam or not. To train the neural network and Bayesian filter to correctly determine wether a message is spam or ham, the neural network and Bayesian filter are trained on both black and white lists. Black lists contain message samples that are definitely spam and white lists contain message samples that are definitely legitimate.

These black and white lists are built up manually by the user. This is accomplished by adding available spam messages to the black list and available legitimate messages to the white list. Figure 4.1 shows how the black and white lists are used to train the neural network and Bayesian filter to identify spam.

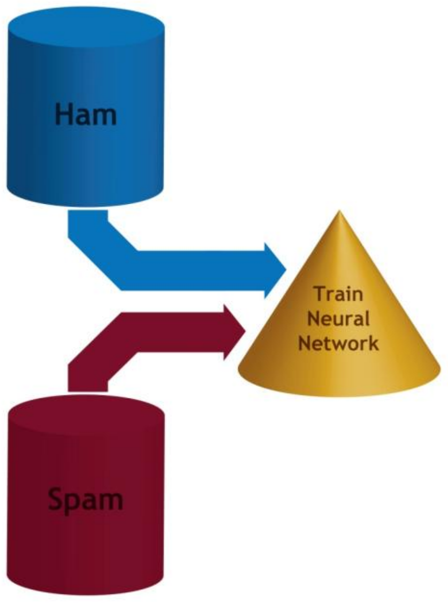


**Figure 4.1: Training sets for artificial neural networks.**

### 4.2.2 Message Detail

Unlike email messages, which can be very long with images and tables attached, SMS messages are limited to about 160 characters. The limitation that results from having the message length limited to 160 characters is that fewer words means less information that can be transferred, thus less information available to be analysed by our application.

The limit placed on message length also means people tend to use acronyms when writing SMSs. SMS messages also lack structured fields and their text is rife with abbreviations and idioms. Moreover, the abbreviations used by SMS users are not standard for a language, but are dependent on the user communities.

Such language variability provides more terms to the word collection. For example, the sender of an SMS could replace the word “tomorrow” with the abbreviation “tmw”. This would usually be understandable to the SMS recipient, but there would now be two permutations of the word “tomorrow” to be analysed by our application.

The size of the SMS message directly impacts the amount of data available to implementation when determining whether or not a message is valid or not. Shorter messages mean that there is less data to assist the implementation in classifying the message. The language variability will directly impact the training data as this means the implementation will need to be trained on a larger data set.

### 4.2.3 Remote analyses vs. Analysis on the device

There are two ways in which one could analyse a mobile device user's SMS data. The first possibility would be to analyse the SMSs sent on the ISP's servers and use this to build a profile of the user's SMS sending behaviour.

The second solution is to implement the detection algorithm on the device, thus, taking care of the privacy implications discussed in Section 4.2.3, as well as allowing for user feedback in the learning process. The pros and cons of both approaches are discussed further in this section.

#### *4.2.3.1 Remote analyses*

By installing the Botnet detector on a network server and  analysing  the SMSs sent on the ISPs servers the application will be able to leverage a great deal of processing and data storage capabilities available at an ISP’s data centre. There are several drawbacks to this approach.

Firstly, there are privacy implications of analysing SMS data on an ISPs server, but even if these concerns were addressed, the classification algorithm would have to determine whether or not a message is valid or invalid without the user's input. Thus, it would not be able to learn based on user feedback.

#### *4.2.3.2 Analysis on the device*

By implementing the detection algorithm on the device, we would take care of any privacy implications resulting from installing the device on a server, as well as allowing for user feedback in the learning process.

The major disadvantage of this approach is that the prototype needs to be installed on a mobile device which has limited processing power and storage space, especially when compared to an ISPs server. Therefore, the prototype had to be programmed to use minimum storage and be optimised to make use of as little memory as possible.

Also, in a live scenario it is possible that the device could be infected by malware before the algorithm has a chance to build a valid user SMS sending profile.

### 4.2.4 Privacy

When building a spamming Botnet detector that intercepts, captures, records, and analyses network events in order to discover the source of security threat, care must be taken not to break any local laws regarding the handling of data. There are two possible systems that can be implemented (59):

- "Catch-it-as-you-can" systems, in which all packets passing through certain traffic points are captured and written to storage with analysis being done subsequently in batch mode. This approach requires large amounts of storage (60).
- "Stop, look and listen" systems, in which each packet is analysed in a rudimentary way in memory and only certain information is saved for future analysis. This approach requires less storage, but may require a faster processor to keep up with incoming traffic (60).

One concern with the "catch-it-as-you-can" approach is one of privacy since all packet information (including user data) is captured. Internet service providers (ISPs) are expressly forbidden by the Electronic Communications Privacy Act (ECPA) (61) in South Africa from eavesdropping or disclosing intercepted contents, except with user permission, for limited operations monitoring, or under a court order.

### 4.2.5 Performance

When designing a model from a spamming Botnet detector it is important to design it so that the user does not notice any degradation of service. First and foremost, false positives (valid SMSs incorrectly classified as spam) need to be kept to an absolute minimum as a user will be more tolerant of spam messages being sent than valid messages not being sent. Another very important consideration is that the time taken to analyse a message and label it as valid (not spam) or invalid (spam) must be kept to a minimum as a user would not like a large delay in sending messages.

## 4.3 Conclusion

There are many important factors that need to be considered when implementing a spamming Botnet detector. The factors that were addressed in this chapter are the following:

- Learning with positive and negative data.
- Message detail.
- Remote analysis vs. analysis on a device.
- Privacy.
- Performance.

These factors were designed almost as a set of rules that can be used as a checklist to determine if the proposed models will indeed work. Such a list could provide anyone who would want to attempt anything along the line of detecting SMS spam with a good starting point. Now that we have a starting point, the next chapter will propose three models to implement a spamming SMS Botnet detector.

# 5 Proposed Model

## 5.1 Introduction

The previous chapter proposed a set of factors that need to be taken into account while proposing a model for a mobile spamming Botnet detector. This chapter details the three tier model and each of its three tiers for implementing a spamming Botnet detector proposed by the author.

The following subsections show how the three tiers adhere to the requirements set out in the previous chapter. Section 5.2 is used to propose a model on how to detect spamming mobile Botnets using an artificial immune system. Section 5.3 proposes a model for detecting Botnets using a Bayesian filter. Section 5.4 is used to propose a model that detects spamming mobile Botnets using an artificial neural network. The chapter is then concluded with Section 5.6.

## 5.2 Model - Artificial Immune System

The first model that the author implemented was a model on how to detect spamming mobile Botnets using an artificial immune system. Section 5.2.1 models this implementation at a very high level. Section 5.2.2 shows how the model adheres to the requirements set out in the previous chapter. Section 5.2.3 summarises the implementation of a model using AIS.

### 5.2.1 Model for a Botnet detector on a Device - Artificial Immune System

Implementing the spam detection algorithm on the mobile device, we take care of any privacy implications resulting from installing the device on a server as well as allowing for user feedback in the learning process. The major disadvantage of this approach is that the prototype needs to be installed on a mobile device which has limited processing power and storage space, especially when compared to an ISPs server. Another point of concern is that it is possible that the device could be infected by malware before the algorithm has a chance to build a valid user SMS sending profile.

Figure 5.1 graphically depicts how the Botnet detector, using an AIS installed on a mobile device, is designed to work. The mobile user enters a text message and sends it to a recipient (Figure 5.1.3). This message is intercepted and certain message characteristics, such as the number of capital letters (the full list of characteristics is defined in Chapter 5), are also extracted for analysis by the Botnet Detector before the message is sent (the AIS should not send out messages identified as spam messages).

The characteristics are sent to the AIS, which will then determine whether the message is valid or not by matching the signature of the message to the signatures in its signature library (this is explained further in Chapter 6). If the AIS can determine that the message is valid, the message is sent onwards (Figure 5.1.1). If the AIS suspects that the message is spam, it prompts the user to confirm whether the message is valid or not (Figure 5.1.2).

If the user confirms the SMS is valid, the message is released and sent onwards to the recipient and the AIS will learn to recognise that type of message as valid. If the user indicates that the message is invalid, the AIS will not learn the new pattern and the message will not be sent.

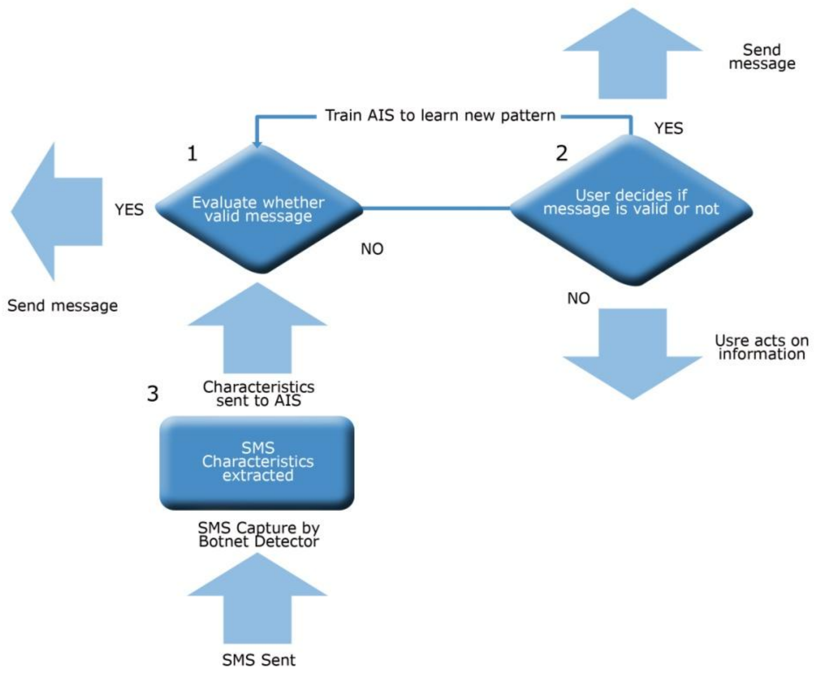

Send message
Train AIS to learn new pattern
YES
1
2
YES
Evaluate whether valid message
User decides if message is valid or not
NO
Send message
NO
Usre acts on information
3
Characteristics sent to AIS
SMS Characteristics extracted
SMS Capture by Botnet Detector
SMS Sent

**Figure 5.1: Model for a Botnet Detector using an Artificial Immune System to detect Spamming Botnets**

It is important to note that this chapter only provides a broad overview of the inner workings of the implementation. A detailed discussion follows in a later chapter.

As the model has been proposed, Section 5.2.2 shows how and why the model adheres to all the factors that were specified in the previous chapter.

### 5.2.2 Adhering to the model requirements when using an AIS

This section shows how the proposed model adheres to the requirements that were set out in the previous chapter. The following subsections detail each one of the points listed in the previous chapter in relation to the AIS.

#### *5.2.2.1 Learning with negative and positive data*

As discussed in Chapter 3, the AIS will be trained only on positive data as the mobile device will build its profile of the user's SMS-sending behaviour.

When building the signature library, which stores the antigens, the choice was between targeting valid SMSs and spam SMSs, as one cannot possibly build a profile of all the different types of SMS spam. This is due to the fact that, what constitutes spam is forever changing. Also, a list of all possible spam profiles would be very large. In addition, there is a question of where SMS spam would be obtained from.

The approach followed by the author in this implementation was to build a profile of the user's SMS-sending behaviour, as this could be done by storing and analysing message signatures. The prototype would be trained on the user's messages, learning to identify messages as self. Therefore, everything else becomes non-self, i.e., spam.

#### *5.2.2.2 Message Detail*

The length of the message was one of the characteristics analysed by the AIS. As was mentioned in the previous chapter, SMS messages have a 160 character limit. However, as this limit is applied to all messages sent from the device, the AIS will build its profile of the users SMSs with the 160 character limit to message size.

#### *5.2.2.3 Remote analysis vs. analysis on a device*

The AIS spam detection algorithm was implemented on the mobile device, thus taking care of any privacy implications resulting from installing the device on a server, as well as allowing for user feedback in the learning process.

The major disadvantage of this approach is that the prototype needs to be installed on a mobile device which has limited processing power and storage space, especially when compared to an ISPs server. Another point of concern is that it is possible that the device could be infected by malware before the algorithm has a chance to build a valid user SMS sending profile.

#### *5.2.2.4 Privacy*

The implementation of an AIS spam detection algorithm on a mobile device takes care of any privacy implications as the mobile user’s private data will not leave the mobile device. This is one of the reasons that the author decided to implement such a model. All the message data is extracted and analysed on the device and not communicated to any third party.

#### *5.2.2.5 Performance*

The implementation of an AIS spam detection algorithm on a mobile device will result in a degradation of service as the limited computing, storage and processing capabilities of the mobile device means that it could take some time for the AIS to correctly identify an SMS as valid (not spam) or invalid (spam). The device could become unresponsive while the AIS calculates wether or not the message is valid. Also, the SMS is not sent until this is computed, so there can be a delay in sending messages.

### 5.2.3 Summary – AIS

The author showed how the proposed model for a spamming Botnet detector, using an artificial immune system, installed on a mobile device adheres to the requirements that were set out in the previous chapter.

The author also listed the factors proposed in the previous chapter to illustrate that the model will indeed work. Each factor was considered as a separate entity and then used to discuss how the model adheres to each factor. The pros and cons of the proposed model were also discussed. The next section will introduce the user to a model that uses Bayesian filtering to detect spamming Botnets.

## 5.3 Model – Bayesian filter

In this section the model for the spamming Botnet detector is introduced and the reasons for it being modelled in this manner are explained, and its advantages and disadvantages are discussed. Section 5.3.1 models this implementation at a very high level. Section 5.3.2 shows how the model adheres to the requirements set out in the previous chapter. Section 5.3.3 summarises the model requirements of an implementation using a Bayesian filter.

### 5.3.1 Model for a Botnet detector on a Network Server – Bayesian Filter

The detection of SMS spam is a typical classification problem where patterns, also known as signatures, need to be classified as legitimate or illegitimate. Therefore, it is a binary classification problem in which the data set, consisting of messages, are classified as either spam or non-spam.

Bayesian filtering is ideal for this sort of classification problem as the spam filter can be trained to identify legitimate and illegitimate (spam) messages. From this data, the filter should be able to deduce whether messages are spam or not.

Figure 5.2 is a visual depiction of how the Botnet detector is designed to work. The mobile user enters a text message and sends it to a recipient. This message is intercepted and analysed by the Bayesian Botnet Detector (BBD), which then determines if the message is valid. If the BBD can determine that the message is valid, the message is sent onwards. Else the network provider will be alerted by the BBD so that the service provider can investigate the malware that is sending the spam messages and remove it from the mobile device.

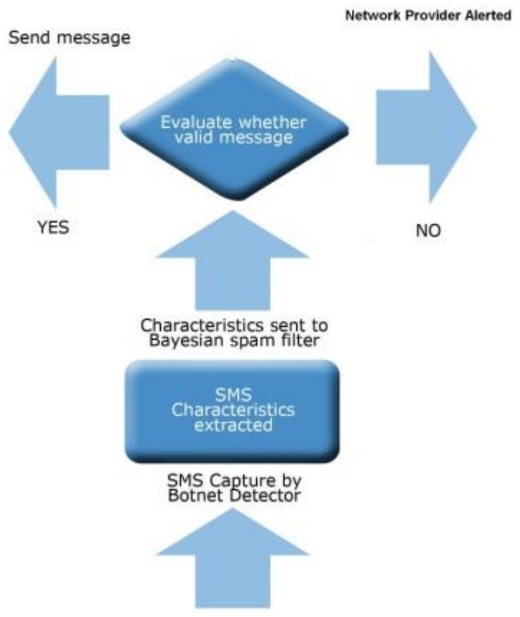

Network Provider Alerted
Send message
Evaluate whether valid message
YES
NO
Characteristics sent to Bayesian spam filter
SMS Characteristics extracted
SMS Capture by Botnet Detector

**Figure 5.1: Model for the Botnet detector**

The premise behind the BBD implementation is that the author believes that a spamming Botnet that has installed itself on a user's mobile device and is sending out spam SMSs without the knowledge of the mobile devices user can be detected by the BBD. If these spam messages are blocked the BBD would have succeeded in preventing the sending of spam, as well as saving the mobile device owner the cost of the spam SMSs being sent.

Additionally, the BBD would alert the mobile device user to the presence of Botnets on their device so that the malware that has installed itself on their device can be removed. Installing the Botnet detector on a network server and analysing the SMSs sent on the ISPs servers has a number of advantages. The application is able to leverage a great deal of processing and data storage capability available at an ISP's data centre.

### 5.3.2 Adhering to the model requirements when using a Bayesian filter

This section shows how the proposed model adheres to the requirements that were set out in the previous chapter. The following subsections detail each one of the points listed in the previous chapter in relation to the Bayesian filter model.

#### *5.3.2.1 Learning with negative and positive data*

The detection of SMS spam is a typical classification problem where patterns need to be classified as legitimate or illegitimate. It is a simple binary classification problem in which the data set (messages) are classified as either spam or non-spam. Bayesian filtering is ideal for this sort of classification problem as we can train the spam filter on legitimate (white list) and illegitimate (black list) messages. From this data the filter should be able to deduce whether messages are spam or not. Thus, this model uses both negative and positive data to train.

The algorithm selected is an implementation of Paul Graham's original Naïve Bayesian Spam filtering algorithm. The statistical approach to detecting spam used by the BBD does not attempt to identify the individual properties of spam, such as the words "click on the link". Rather, we leave it up to the algorithm to determine the probability of a message being spam or not based on the datasets used to train the filter on.

Unlike feature-recognising spam filters, which assign a score to messages, the BBD, which uses Bayesian filtering, assigns a probability to a message as an indication of it being spam or not. As it is measuring probabilities, the BBD considers all the evidence in the SMS message, both good and bad. Words that occur disproportionately rarely in spam, like "later" or "morning", contribute as much to decreasing the probability as possible spam words like "click" and "opt-in" do to increasing it. Therefore, an otherwise innocent SMS message that happens to include the word "Viagra" is not going to be misidentified as spam.

#### *5.3.2.2 Message Detail*

Having a good term representation is one of the most important aspects for getting a good classifier. The spam filter should be designed with the SMS message characteristics in mind, namely only 160 characters, which are allowed on a standard SMS message.

The problem with this limitation is that fewer words mean less information that can be transferred.  As stated in the previous section, SMS messages are limited to 160 characters. Due to this constraint, people tend to use acronyms when writing SMSs. SMS messages lack structured fields and their text is rife with abbreviations and idioms. Moreover, the abbreviations used by SMS users are not standard for a language, but they depend on the user communities. Such language variability provides more terms to the word collection.

For example, the sender of an SMS could replace the word ‘tomorrow’ with the abbreviation “tmw”. This would usually be understandable to the SMS recipient, but there would now be two permutations of the word “tomorrow” to analyse, thereby expanding the collection of words for which their spam probability needs to be calculated. We shall now conclude with a summary of what was discussed.

#### *5.3.2.3 Remote analysis vs. analysis on a device*

Installing the Botnet detector on a network server and analysing the SMSs sent on the ISPs servers has a number of advantages. The application is able to leverage a great deal of processing and data storage capability available at an ISP's data centre.

The major disadvantage of this approach is that the prototype extracts and processes data from a SMS message, which could have privacy implications. Also, the model does not allow for user feedback and is thus not adaptive and must be retrained when the white or black list is updated.

#### *5.3.2.4 Privacy*

Installing the Botnet detector on a network server and analysing the SMSs sent on the ISPs servers means that the prototype extracts and processes data from a SMS message. This could have privacy implications. However, the author believes that as the messages are analysed by a machine and not by a person this should not be too serious of a problem.

#### *5.3.2.5 Performance*

Installing the Botnet detector on a network server and analysing the SMSs sent on the ISPs servers means that the application is able to leverage a great deal of processing and data storage capability available at an ISP's data centre.

### 5.3.3 Summary – Bayesian Filter

The author showed the reader how the proposed model for a spamming Botnet detector, using a Bayesian filter, adheres to the requirements that were set out in the previous chapter. The authors also listed the factors proposed in the previous chapter to explain and show that the model will indeed work.

Each factor was considered as a separate entity and then used to discuss how the model adheres to it. The authors then discussed the pros and cons of the proposed model. The next section will introduce the user to a model that uses artificial neural networks to detect spamming Botnets.

## 5.4 Model – Artificial neural network

The third model that the autho implemented was a model on how to detect spamming mobile Botnets using an artificial neural network. Section 5.4.1 models this implementation at a very high level. Section 5.4.2 shows how the model adheres to the requirements set out in the previous chapter. This is followed by a Section 5.4.3 that summarises the model of an implementation that uses an ANN.

### 5.4.1 Model for a Botnet detector on a Network Server - Using an ANN

Installing the Botnet detector on a network server and analysing the SMSs sent on the ISPs servers, the application is able to leverage a great deal of processing and data storage capability available at an ISPs data centre.

Figure 5.3 visualises how the Botnet detector, installed on a network provider's server using a neural network, is designed to work. The mobile user enters a text message and sends it to a recipient. This message is intercepted and certain message characteristics such as the number of capital letters (the full list of characteristics is defined in a later chapter) are also extracted for analysis by the Botnet Detector before the message is sent (the neural network should not send out messages identified as spam messages).

The characteristics are sent to the neural network, which then determines whether the message is valid or not by inputting the SMS message into the neural network (this is explained further in a later chapter). If the ANN can determine that the message is valid, the message is sent onwards. Else, the network provider will be alerted by the ANN so that the service provider can investigate the malware that is sending these spam messages and remove it from the mobile device

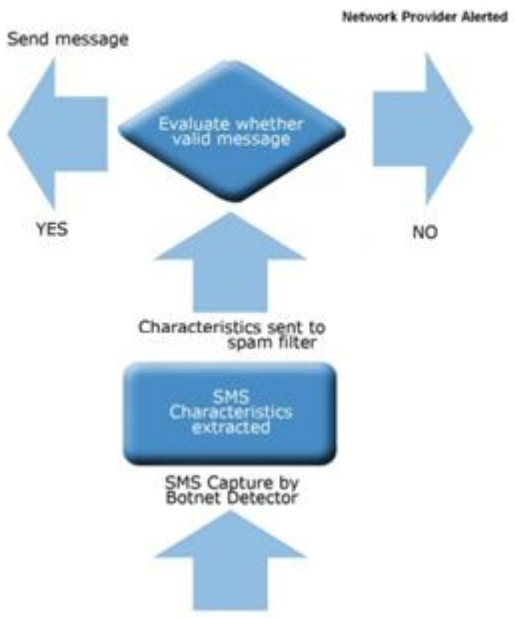

Network Provider Alerted
Send message
Evaluate whether valid message
YES
NO
Characteristics sent to spam filter
SMS Characteristics extracted
SMS Capture by Botnet Detector

*Figure 5.2: Model of* the ANN

The premise behind the ANN implementation is that the authors believe that a spamming Botnet that has installed itself on a user's mobile device, and is sending out spam SMSs without the knowledge of the mobile devices user, can be detected by the ANN. Therefore, if these spam messages are blocked, the ANN would have succeeded in preventing the sending of spam as well as saving the mobile device owner the cost of the spam SMSs being sent. Additionally, the ANN would alert the mobile device user to the presence of Botnets on their device so that the malware, that has installed itself on their device, can be removed. The following section explains how the model requirements are adhered to.

### 5.4.2 Adhering to the model requirements when using a artificial neural network

This section shows how the proposed model adheres to the requirements that were set out in the previous chapter. The following subsections detail each one of the points listed in the previous chapter in relation to the ANN.

#### *5.4.2.1 Learning with negative and positive data*

To train the neural network to correctly determine whether a message is spam or ham, the neural network is trained on both black and white lists, i.e., both negative and positive data. Black lists contain message samples that are definitely spam and white lists contain message samples that are definitely legitimate.  These black and white lists are built up manually by the user. This is accomplished by adding available spam messages to the black list and available legitimate messages to the white list.

The approach followed by the author in this implementation was to build a profile of the user's SMS-sending behaviour as this could be done by storing and analysing message signatures. The prototype would be trained on the user's messages, learning to identify messages as self. Thus, everything else becomes non-self, i.e., spam.

#### *5.4.2.2 Message length*

Unlike email messages, which can be very long and with images and tables attached, SMS messages are limited to about 160 characters. The limitation that results from having the message length limited to 160 characters is that fewer words mean less information that can be transferred, thus less information available to be analysed by our application.

The neural network will need to be trained on a larger word collection to cater for the different permutations of a word, the use of acronyms, abbreviations and idioms. Now we have a model that adheres to all the requirements for a spamming Botnet detector, and the next section will therefore conclude with a summary of what was discussed.

#### *5.4.2.3 Remote analysis vs. analysis on a device*

Installing the Botnet detector on a network server and analysing the SMSs sent on the ISPs servers has a number of advantages. The application is able to leverage a great deal of processing and data storage capability available at an ISPs data centre.

The major disadvantage of this approach is that the prototype extracts and processes data from a SMS message, which could have privacy implications. Also, the model does not allow for user feedback and is thus not adaptive and must be retrained when the white or black list is updated.

#### *5.4.2.4 Privacy*

Installing the Botnet detector on a network server and analysing the SMSs sent on the ISPs servers means that the prototype extracts and processes data from a SMS message. This could have privacy implications. However, the author believes that as the messages are analysed by a machine and not by a person this should not be too serious.

#### *5.4.2.5 Performance*

Installing the Botnet detector on a network server and analysing the SMSs sent on the ISPs servers means that the application is able to leverage a great deal of processing and data storage capability available at an ISPs data centre.

### 5.4.3 Summary – ANN

The author have shown how the proposed model for a spamming Botnet detector using an artificial neural network adheres to the requirements that were set out in the previous chapter.

The authors also listed the factors proposed in the previous chapter to explain and show that the model will indeed work. Each factor was considered as a separate entity and then used to discuss how the model adheres to each factor. The author also discussed the pros and cons of the proposed model. The next section will conclude this chapter.

## 5.5 Conclusion

Chapter 5 proposed three tiers in the three tier model that can be used to detect spamming Botnets. This chapter is also listed the factors proposed in Chapter 4 to explain and show that the model will indeed work. Each factor was considered as a separate entity and then used to discuss how the model adheres to each factor.

Having shown that all the factors can be adhered to by the models developed by the author to detect spamming Botnets, this dissertation will now continue to discuss, in more detail, how a prototype of the model was designed and implemented. Part IV will focus exclusively on the implementation of the prototype and how it performed in demonstration scenarios.

# Part IV: Prototype

# 6 Prototype Setup

## 6.1 Introduction

This chapter is devoted to introducing the reader to the three tier model and its prototype components implemented to detect spamming Botnets. This section details the demonstration scenarios that are used to illustrate the inner workings of each of the three prototypes.

It starts off by providing the reader with information regarding the various devices that were used for various purposes during the demonstrations. The chapter then discusses the environment in which all the tests were performed and shows that the scenarios were sufficient to demonstrate the workings of the protocol.

Section 6.2 explains the demonstration environment, the decision making within this environment and the placement of motes. Section 6.3 introduces the reader to the equipment used to build the three prototypes. Section 6.4 then discusses how the three prototypes were developed and setup. Section 6.5 provides a brief summary of the chapter.

## 6.2 Demonstration environment

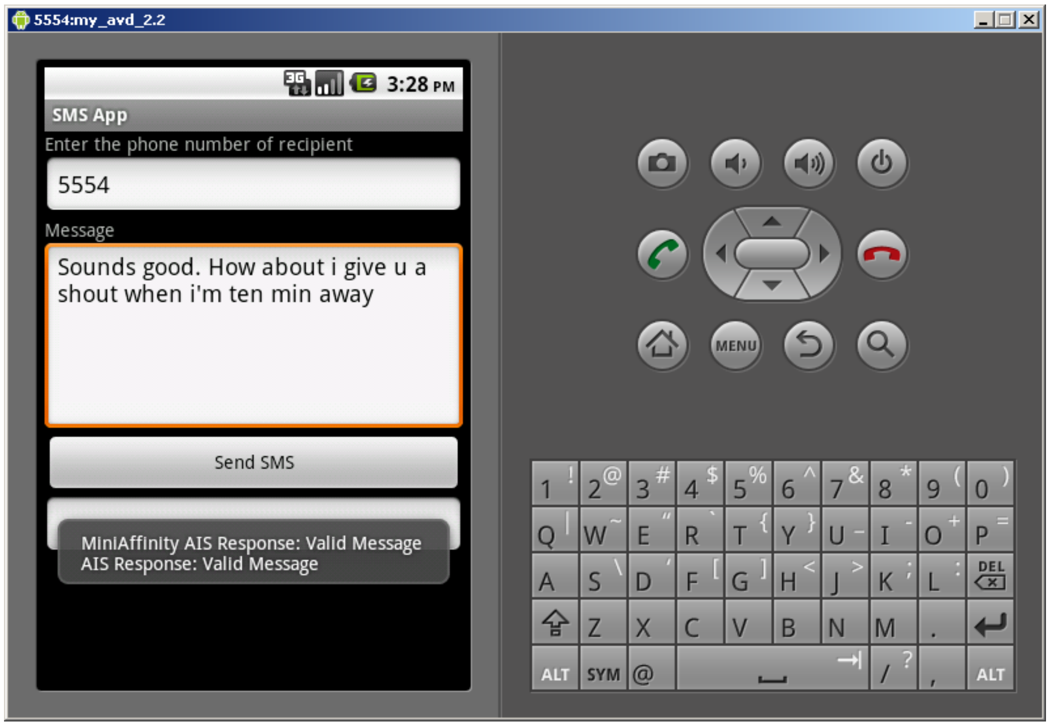
The demonstrations were all completed inside a research computer laboratory. This laboratory allowed the simulation of several different scenarios that are similar to real-life environments. Due to limited funding and the costs involved, the experiments were all carried out in a simulated test environment. The next section will describe, in detail, the software and hardware that was used to build the three prototypes.

## 6.3 Prototype Equipment

This section introduces the reader to the equipment that was used to implement the three prototypes. It starts off with a brief discussion of the actual equipment, followed by a discussion of the programming language and environment. Subsection 6.3.1 is dedicated to introducing the reader to the Android emulator, which was used to test the AIS implementation of the spamming Botnet detector. Once the reader has been familiarised with the Android emulator, the dissertation proceeds to discuss the hardware used to host the prototypes in Section 6.3.2. Sections 6.3.3 and 6.3.4 detail the operating system and programming languages used to build the three prototypes.

### 6.3.1 Android Emulator

The AIS prototype was implemented on an Android mobile smart phone emulator. This section serves to provide the reader with some background on the emulator. The Android emulator is an application that provides a virtual mobile device on which you can run your Android applications. It runs a full Android system stack, down to the kernel level. The version of the Android system that we ran in the emulator in was version 2.2. Figure 10.1 is a screenshot of the Android emulator.



Enter the phone number of recipient
5554
Message
Sounds good. How about i give u a shout when i'm ten min away
MiniAffinity AIS Response: Valid Message
AIS Response: Valid Message

Figure 6.1: Screenshot of the Android emulator

As can be seen by the screenshot in Figure 10.1, the Android emulator supports many features likely to be found in an Android device such as a LCD display, qwerty keyboard and simulated SIM card. The Android operating system is detailed further in Section 6.3.3. The next section will describe the hardware used by the author to host the prototype.

### 6.3.2 Hardware

Three prototypes were implemented by the author. Two of the implementations, the prototype using a Bayesian filter and the prototype using the artificial neural network, were hosted on a simulated network server as the author did not have access to a live mobile network server.

These two prototypes were hosted on the author's desktop machine. Due to cost constraints, the author was unable to procure an actual server. This meant that the performance was not as good as what one would expect on a high spec machine.

Due to the same cost constraints, the author was not able to procure an android device to host the third prototype. Therefore, the AIS was hosted on an Android emulator installed on the authors desktop. This meant that the performance of the AIS, in terms of processing speed, was a lot better than the expected performance on a mobile device, as it was running on a desktop device with greater processing and memory specs.

The three prototypes were implemented on a Microsoft Windows XP machine. This machine which was used to emulate the ISPs server and host the Android emulator had the following hardware specs (see Figure 10.2):

- Intel Xeon® CPU
- 5160 @ 3.00 GHz
- 2.99 GHz, 3.00 GB of RAM

### 6.3.3 Operating system

This subsection will provide the reader with some detail on the operating systems which were used to implement the three prototypes. The Android operating system was designed for use on mobile devices and was used to implement the AIS prototype. The use of the Android emulator described in Section 6.3.1 enabled us to simulate a device running on the Android operating system on the author's Windows machine.

Android is a mobile operating system based on a modified version of the Linux kernel. Android has a large community of developers writing application programs (apps for short) that extend the functionality of the devices. Android is the most popular smart phone operating system in the world (62). Developers write managed code in the Java language, controlling the device via Google-developed Java libraries (63).

The author implemented the prototype with Android Version 2.2 using an eclipse integrated development environment. The author selected the Android operating system primarily because Android is an open source project with lots of developer support and also because of the author's familiarity with the Java programming language.

The ANN and the Bayesian filter were implemented on a machine using Microsoft Windows XP Professional, Version 2002 and Service Pack 3. This operating system is very similar to operating systems used on many servers, therefore it was a good choice in testing the ANN prototype and Bayesian filter.

### 6.3.4 Programming Language

The three prototypes were implemented using two different programming languages. The AIS was programmed using Java with the Android SDK installed. The ANN and Bayesian filter were programmed with the .Net framework using the C# programming language. The next section details the development and setup of the three prototypes.

## 6.4 Prototype Development and Setup

In this section the development and setup of the three prototypes developed to detect spamming Botnets is discussed. The first prototype was designed to be installed on an Android device; the next two prototypes were designed to be implemented on a network server.

The following subsection will discuss the development and setup of an artificial immune system on an Android device. This is followed by a subsection that discusses the development and setup of a Bayesian filter followed by a subsection that discusses the development and setup of a spamming Botnet detecting artificial neural network.

### 6.4.1 Development and Setup of a Mobile Botnet Detector on an Android

This section describes how the prototype was implemented on an Android emulator. The author discuss the message characteristics chosen to extract and train the AIS module with, the algorithm selected to train the AIS, the message signature representation and also how an affinity measure, which is used to match the message signatures to our signature library, is calculated.

#### *6.4.1.1 Message Signature*

The prototype creates a signature (pattern) for each message sent by the mobile device. The signature consists of the following characteristics that are analysed by the AIS to determine the validity of the message:

- The total number of characters in a message including white spaces.
- The total number of characters excluding white spaces.
- The number of capital letters.
- The number of white spaces.
- The number of punctuation characters.
- The number of digits.
- The number of words.

- The presence of URLs.
- The presence of telephone numbers.

The specific characteristics mentioned above were chosen by the author to define the message signature as they allow the implementation to build a profile of the user's sending behaviour. The characteristics chosen are simple to capture yet indicative of sending behaviour.

Use of punctuation, capital letters, as well as message length, may reveal much about a user's SMS sending behaviour. This is because people have different messaging styles with some individuals being more attentive to grammar and thus more likely to use punctuation than others.

The majority of spam emails contain a link to a URL, thus it makes sense to mark the presence of URLs in the SMS message (these links might lead to a website which sells a product the spammer is attempting to advertise). The presence of telephone numbers can also be a useful bit of information to mark as the spammer might include a telephone number to call (quite possibly at a premium rate).

Additional characteristics may be added to the prototype in future in order to increase the accuracy of the implementation. This would enable us to build a better profile of the user's messages. Expanding the list of characteristics would require more processing and storage, but would better define the message signature and, thus, the AISs ability to distinguish between spam and non-spam.

The author felt that the current implementation, though storing and analysing a limited number of message characteristics, should nevertheless be able to detect invalid messages as the marking of URLs and telephone numbers (needed for individual being spammed to be able to purchase the product advertised), combined with the characteristics that define the user (such as message length) should enable us to accurately build a profile.

When deciding what metrics to measure, we made a number of assumptions, namely those individuals that normally send short messages with liberal use of capital letters would not generally send long messages with lots of punctuation and no capital letters. This alone would not be enough to mark the message as spam (he could be typing a message to his mother, where normally he messages his mates) but if this change was accompanied with a message containing URLs and telephone numbers, this could be enough to warrant asking the user for a confirmation before sending the message.

Figure 6.1 shows a sample message that could be sent by the user to a friend.

hey bud wuu2?

**Figure 6.2: Sample message sent by user to a friend**

Figure 6.2 shows a sample message sent by the user to his mother. This message is very different from the previous sample but should, nevertheless, not be identified as spam.

Good morning mommy, trust you are well. Please tell dad I'll be over to fix the blinds later this afternoon, I just want to pop round to Mary's place on my way to the hardware shop. Bye, love you!

**Figure 6.3: Sample message sent by user to their mother**

Figure 6.3 shows a sample message advertising a product and thus unlikely to be sent by the user.

INSURE now and SAVE! Up to 50% OFF car and household insurance!! Don't miss out CALL 0005556677 NOW. This offer ends 01/05/2012… www.carandhouseholdinsurance.com

**Figure 6.4: Sample message that is unlikely to be sent by user, possible spam**

The sample messages provide us an indication of what the AIS should be looking for. It is, however, important to note that we do not tell the botnet detector what to look for, but rather allow the AIS to learn what to look for through trial and error. The next section discusses how the message is digitally represented in the AIS.

#### *6.4.1.2 Signature Library*

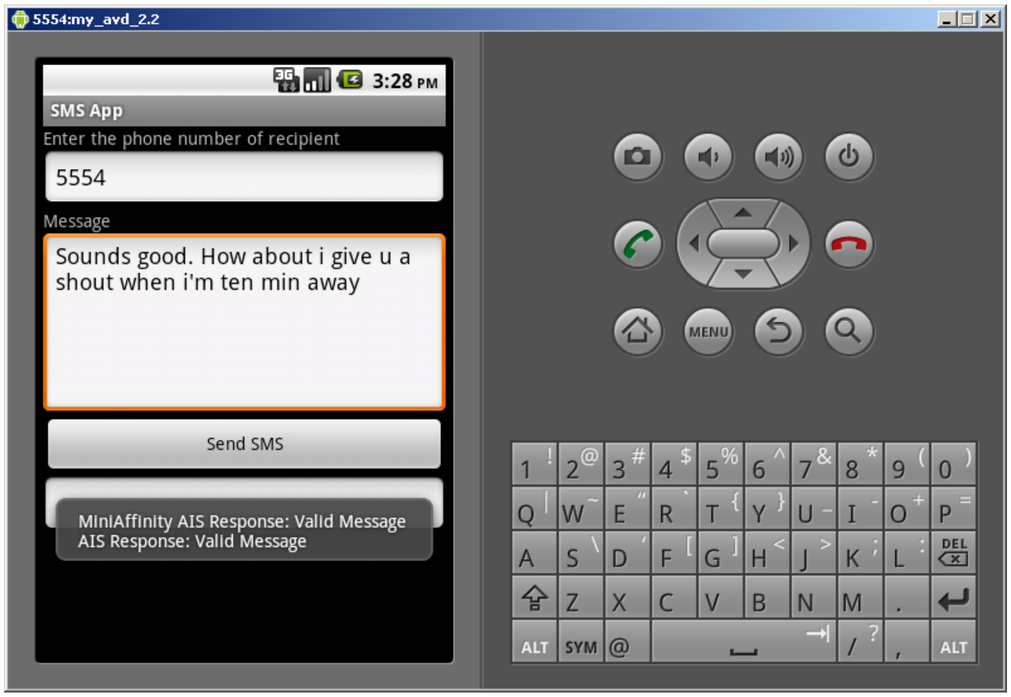
The initial population of the signature library is generated randomly by the AIS, as discussed in Chapter 3. This library is digitally represented as a relational database and stored in a database installed on the device.

During each successive generation (training is continuous) a proportion of the initial population that matches a valid message pattern is selected for deletion. These antigens are replaced by randomly generated new antigens or by mutated clones of existing antigens. The selection of signatures in the signature library for deletion is determined by its affinity measure, this is explained in further detail in the following section.

#### *6.4.1.3 Digital representation of the SMS signature and affinity measure*

The SMS signature (pattern), composed of the characteristics listed above, needs to be represented in a form that the algorithm can process. The attribute values are represented as real numbers and a Euclidean distance function is used as an affinity measure matching the patterns to the signature.

In mathematics, the Euclidean distance is the ordinary distance between two points that one would measure with a ruler. This affinity measure is used to fit the signatures in the signature library to the message signature. The use of a Euclidean function serves the desired purpose for the model as it approximates how closely the patterns fit.


5554:my_avd_2.2

SMS App

Enter the phone number of recipient

5554

Message

Sounds good. How about i give u a shout when i'm ten min away

Send SMS

MiniAffinity AIS Response: Valid Message
AIS Response: Valid Message

Figure 6.5: Sample message

For example, the message in Figure 6.4 can be represented in the message signature library as follows;

*Table 6-1: Antigen-Array 1*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 60 | 48 | 2 | 12 | 1 | 0 | 0 | 0 |

Table 6.1 is the digital representation of a message sent by a user. The message characteristics are stored as real numbers in a database table. This signature can be represented by an integer array. The values in the array are real number representations of the message body.

For example, the first column of this array is a count of the total number of characters including white spaces in the message body. The value of each element in the signature array is then matched to the corresponding element for all the signatures in the signature library if, say, it is compared against the following message signature from our signature library.

*Table 6-2: Antibody – Array 2*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 60 | 48 | 2 | 12 | 5 | 0 | 3 | 5 |

Table 6.2 is the digital representation of an antibody in the signature library. The Euclidean distance between the first element in array 1 and the first element of array 2 is 0. The Euclidean distance is calculated for all the elements in the array. If the affinity (similarity measure) was set to two, the message in Table 2 would be classified as a match, i.e., spam. If the affinity measure was larger than two, it would be classified as a valid message.

The tolerance was calculated as follows: first the Euclidean distance is calculated between the corresponding elements of the set. In one dimension, the distance between two points on the real line is the absolute value of their numerical difference. Thus, if x and y are two points on the real line, then the distance between them is given by:

$$\sqrt{(x-y)^2} = |x-y|$$

So the Euclidean distances for all the elements in our two arrays are as follows:

[60-60] = 0

[48-48] = 0

[2-2] = 0

[12-12] = 0

[1-5] = 4

[12-12] = 0

[0-0] = 0

[0-3] = 3

[0-5] = 5

If the affinity tolerance was not exceeded (in this case to elements are out by a factor of one), that means that the antibody recognises the antigen as a self (valid message). If it is exceeded it is recognised as a pathogen. Depending on the algorithm chosen, this message would be classified as spam or non-spam. The prototype implemented here uses a negative selection algorithm (discussed in detail further in the paper). In this case, the antibody binds to (matches) the message and is thus a non-self (spam) message.

#### *6.4.1.4 Algorithm Selection*

The detection of SMS spam is a typical classification problem where patterns (signatures) need to be classified as legitimate or not. It is a simple binary classification problem in which the data set (messages) are classified as either spam or non-spam. The challenge faced, however, was that initially the implementation would only be able to store legal signatures and would not actually know what to expect or what the signature of an illegitimate message looks like. AISs, however, are better suited for this kind of problem where only patterns of one class are known and, hence, the complement of that class simply needs to be identified.

In the context of the natural immune system, the prototype implements the censoring process on the signatures in the signature library, i.e., matches the antibodies against the antigen to determine the validity of the message.

This is known as negative selection, where random signature detectors are generated and "matched" against the repository of known legitimate signatures. If a randomly generated signature detector "matches" a legitimate signature (to a certain degree) it is replaced (or mutated) until the signature detector does not match the legitimate signature. In other words, we attempt to create a signature library that contains possible matches for spam messages by discarding any signatures in the library that match a valid message.

This is different from positive selection where one would compile a list of valid message signatures in the signature library and train the system to learn those patterns.

The result is a set of legitimate signature-tolerant detectors which are unable to detect legitimate signatures (because of the censoring process) but are able to detect anything else that does not "look" like the legitimate signatures.

These detectors are then used to classify a message as illegitimate or not. The problem with this approach is that it may take a long time to actually get a signature detector which does not detect any of the legitimate signatures. The following section explains how this was solved by implementing a negative selection artificial immune system using a mini-affinity measure.

##### *6.4.1.5 Negative selection algorithm using a mini-affinity measure.*

This section discusses the algorithm selected for implementing the prototype and discusses the logic behind the algorithm, as well as its shortcomings. Instead of having a global affinity threshold (i.e., a user-defined affinity threshold), each detector was assigned its own affinity threshold (64). An affinity is defined as the degree of matching between a signature in the signature library (antibody) and a message (antigen).

The latter satisfies the relation implied by the former if this degree is greater than an affinity threshold (65). This threshold could then be set to the minimum calculated Euclidean distance between the specific detector and all signatures randomly generated by the spam detecting AIS signatures A.

This means that the closest signature, in A, to the detector will determine the affinity threshold of that detector. The end result is a set of detectors where each has its own affinity threshold. This cuts out the indefinite process of generating random detectors until one is actually found that is signature tolerant, i.e., tolerates self (non-spam) and also removes one of the user-specified parameters (otherwise the affinity measure would have to be defined as some constant).

Algorithm 6.1 below, which will be explained in detail, was the algorithm selected for this implementation.

```
Algorithm 6.1 : Negative selection AIS with self calculated affinity
1: Given **A** a set of valid SMS signatures
2:          n the user-defined number of antibodies
3: Initialize the set of patterns **B** to empty
4: while |**B**| < n do
5:     Randomly generate pattern **D**
6:     $De = \min_{a \in A} dist(a,D)$
7:     Add **D** to **B**
8: end while
```

**Algorithm 6-1: Negative selection AIS with self-calculated affinity**

When an SMS is sent, the signature of the new message is then measured against all detectors in the detector (signature library) list B. This means that as soon as there is a detector (i.e., an antibody) in B with an affinity (Euclidean distance) to the new signature equal or less than minaffinity, the new message signature is detected as illegitimate (non-self). If the user indicates that the message was, in fact, legitimate, then the new signature needs to be added to A and all detectors in B needs to go through the censoring process with the new signature. If a detector in list B detects the new signature (which is now part of A), the detector should be removed and replaced with a randomly-generated detector.

The affinity is calculated between the new message signature and each detector in B. The affinity threshold is local to each detector, therefore, when a message is legitimate, only the affinity thresholds of those detectors that detected the new message need to be updated.  The following section describes the data selection process.

#### *6.4.1.6 Data selection*

The data that was used to train the AIS implementation was selected by using SMS messages sent by the author over the last six months. In total, 1,000 of these valid SMSs were randomly selected and used to train the AIS Botnet Detector.

The second set of valid SMSs (80 in total) used to test the efficiency of the Botnet detector were selected in a similar manner. The only difference being that the user prompt and learning of the AIS was switched off to accurately measure how well the device learnt the valid SMS signatures. The invalid SMSs (20 in total) were selected by the author from unsolicited SMSs received by the author during the same time period, usually advertising some product or other.

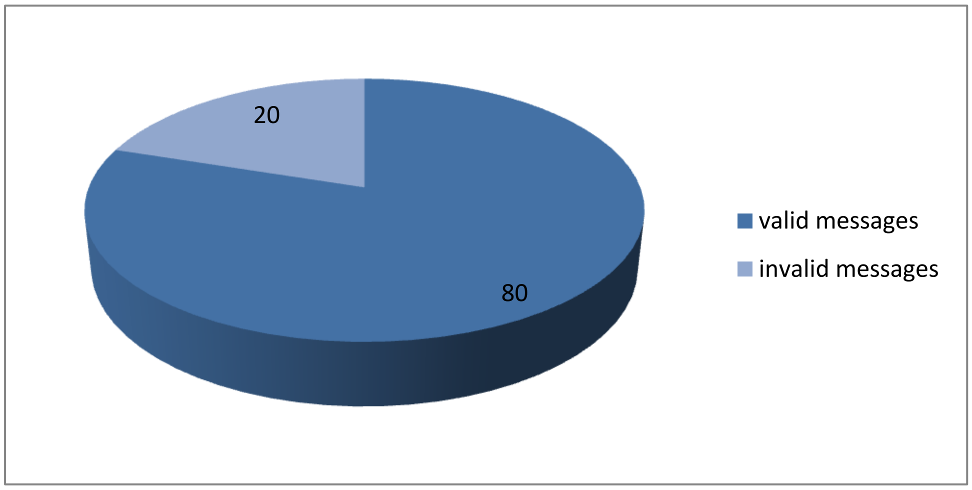


**Chart 6-1: Valid and invalid message breakdown**

Chart 6.1 provides a visual breakdown of the valid and invalid messages used in the experiment. The following section explains how our second prototype, the artificial neural network, was developed.

#### *6.4.1.7 Summary – AIS*

This section showed how the prototype AIS, installed on an Android emulator, was set up. The author discussed the message characteristics chosen to extract and train the AIS module with, the algorithm selected to train the AIS, the message signature representation and also how an affinity measure, which is used to match the message signatures to our signature library, is calculated. The next section will describe how the prototype using a Bayesian filter was setup.

### 6.4.2 Development and Setup of a Mobile Botnet Detector using a Bayesian filter

This section describes how the second prototype was implemented using a Bayesian filter. The author first discusses how the algorithm used to train the Bayesian Botnet Detector (BBD) was selected, how the dataset used to train the BBD module was chosen, and how the Bayesian spam filtering was implemented.

#### *6.4.2.1 Algorithm selection*

As discussed in Chapter 4, the detection of SMS spam is a typical classification problem where patterns need to be classified as legitimate or illegitimate. It is a simple binary classification problem in which the data set (messages) are classified as either spam or non-spam. As stated in Chapter 4, Bayesian filtering is ideal for this sort of classification problem as we can train the spam filter on legitimate (white list) and illegitimate (black list) messages and from this data the filter should be able to deduce whether messages are spam or not.

The algorithm selected is an implementation of Paul Graham’s original Naïve Bayesian Spam filtering algorithm. The statistical approach to detecting spam used by the BBD does not attempt to identify the individual properties of spam, such as the words “click on the link”. Rather, we leave it up to the algorithm to determine the probability of a message being spam or not based on the datasets used to train the filter. The BBD, which uses Bayesian filtering, assigns a probability to a message as an indication of it being spam or not.

As it is measuring probabilities, the BBD considers all the evidence in the SMS message, both good and bad. Words that occur disproportionately rarely in spam, like "later" or "morning", contribute as much to decreasing the probability as possible spam words like "click" and "opt-in" do to increasing it. Therefore, an otherwise innocent SMS message that happens to include the word "Viagra" is not going to be misidentified as spam.

#### *6.4.2.2 Data selection*

The data that was used to train the BBD implementation was selected by using SMS messages collected by the SMS Spam Collection project (44) and from the Grumbletext website. Grumbletext is a UK forum in which cell phone users make public claims about SMS spam messages, most of them without reporting the spam message received. In total, 4,815 of these valid SMSs were used to create a white list of legitimate SMS messages. The second set, i.e., the black list of invalid SMSs, totalled 746 (including blacklisted URLs).

#### *6.4.2.3 Prototype setup*

When the BBD initialises, it reads through the white and black lists and builds a list of words. It then iterates through every word in the list and looks up its individual spam probability.

Figure 6.6 shows an example of a set of words and their spam probabilities. The closer the probability value is to 1, the greater the probability of it being spam. The closer the probability value is to 0, the greater the probability of it being a legitimate SMS message.

```
.0218,would
.4691,Would
.3293,wow
.9861,www
.9998,x150p
.9998,XCHAT
.5956,Xmas
.9998,XMAS
.2866,xx
.5956,XX
.3616,xxx
```

```
.4010,XXX
.8154,xxxxxxx
.0427,ya
.0256,Yeah
.2164,year
.2903,years
.9999,YES
.0686,yesterday
.0602,yet
.1971,yo
.0655,Yo
.1402,you
.3673,You
```

**Figure 6.6: List of random words and their spam probabilities**

After the BBD was built, we could start testing it by feeding it valid and invalid SMS messages. Using Paul Graham's original Naïve Bayesian Spam filtering algorithm we process the SMS message body by iterating through every word in the SMS body and looking up its individual spam probability. We then sort this list of spam probabilities in descending order based on how far the probability is from 50%.

Once the list is compiled and sorted we then combine the most "interesting" probabilities together to give us a single probability, which is used to determine whether the whole message is spam or not. In this case "interesting" is defined as how far our score is from 50%, i.e., how close the score is to indicating a word is spam or valid. Equation 1 gives us the equation used to combine the individual probabilities. This is explained further with a real example in the next chapter.

$$\frac{abc\ldots n}{abc\ldots n + (1-a)(1-b)(1-c)\ldots(1-n)}$$

*Equation 6-1: Formula to combine probabilities [15]*

#### 6.4.2.4 Summary – BBD

In this section the author described how the prototype using a Bayesian filter was setup. The author first discussed how the algorithm used to train the BBD was selected, how the dataset used to train the BBD module was chosen, and how the Bayesian spam filtering was implemented. The next section will describe how the prototype, using an artificial neural network, was setup.

### 6.4.3 Development and Setup of a Mobile Botnet Detector using an artificial neural network

This section describes how the prototype was implemented on a neural network. The author first discusses the message characteristics chosen to extract and train the ANN prototype with. Next, the section describes the process of training the neural network. This is followed by a section documenting the testing of the neural network's accuracy, as well as a section describing how the data was selected for this implementation.

#### 6.4.3.1 Message signatures

The prototype creates a signature (pattern) for each message sent by the mobile device. The signature consists of the following characteristics that are analysed by the neural network to determine the validity of the message:

- Does the SMS message contain links?
- Does the SMS message contain telephone numbers?
- The number of words in a message.
- The ratio of punctuation characters to words.
- The ratio of links to words in the message.
- The ratio of capital letters to words in the message.

- The ratio of misspelt words in the SMS message.
- Bayesian spam probability.

The specific characteristics mentioned above were chosen by the author to define the message signature as they allow the implementation to build a profile of the user's sending behaviour. The characteristics chosen are simple to capture, yet indicative of sending behaviour. Use of punctuation, capital letters and their ratios to words in a message may reveal much about an SMS message.

The majority of spam emails contain a link to a URL, thus it makes sense to mark the presence of URLs in the SMS message (these links might lead to a website which sells a product that the spammer is attempting to advertise). The presence of telephone numbers is also a useful bit of information to mark as the spammer might include a telephone number for the recipient of the spam message to call in order to enquire about the product or service advertised. Quite possibly this call may be charged at a premium rate.

The ratio of misspelt words is also a useful characteristic to monitor as spammers will often hide words. They will often send an SMS with phrases like "v1agr@a" instead of Viagra in order to thwart a spam filter. The Bayesian spam probability, which calculates the probability of a message being spam, will become the dominant feature if the other characteristics fail to identify the message as being spam.  Additional characteristics may be added in future to the prototype to increase the accuracy of the implementation. This would enable us to build a better profile of the valid messages.

##### *6.4.3.2 Training the neural network*

The training of our neural network is a three stage approach. First, we add the messages we are going to train the neural network on to a spam list and a ham list (a ham list is a list of valid SMS messages).

Once we have done this, we can then generate the message signatures for each message, which we save to a list. Finally, we pass the message signatures from our saved list into the training procedure of our neural network, which, in turn, causes the network to learn what message signatures represent spam.

#### *6.4.3.3 Data selection*

The data that was used to train the ANN implementation was selected by using SMS messages collected by the SMS Spam Collection project [5]. In total 4,815 of these valid SMSs were used to create a white list of legitimate SMS messages. The second set, i.e., the black list of invalid SMSs, totalled 746 (including blacklisted URLs). In the following section we tabulate the experimental results. The section following that concludes the chapter by providing a brief overview of the prototypes.

#### *6.4.3.4 Summary – ANN*

In this section the author described how the prototype using an ANN was set up. The author first discussed how the algorithm used to train the ANN was selected, how the dataset used to train the ANN module was chosen, and how it was implemented. The next section concludes the chapter.

## 6.5 Conclusion

Chapter 6 was devoted to introducing the reader to the prototype setup. It discussed the different demonstration environments, as well as the development and setup of the three prototypes implemented, in detail. This chapter covered topics such as data selection and algorithm selection. Now that we have seen how the three prototypes have been set up the following chapter, Chapter 7, gives us a demonstration of the three prototypes.

# 7 Prototype Demonstration

## 7.1 Introduction

The three prototypes proposed in this dissertation have a number of benefits for spamming Botnet detection. The subsections that follow discuss three demonstrations that have been completed in order to focus on various features of the prototypes.

These demonstrations show the running of each of the three prototypes implemented in Section 7.2. The chapter is then concluded in Section 7.3.

## 7.2 Running the Prototypes

The subsections that follow discuss three demonstrations that have been completed in order to focus on various features of the prototype. These demonstrations show the running of the prototypes under simulated conditions. The first subsection demonstrates the running of the AIS prototype on an Android emulator. This is followed by a demonstration of the Bayesian filter prototype and, finally, the third prototype, the artificial neural network implementation of a spamming Botnet detector.

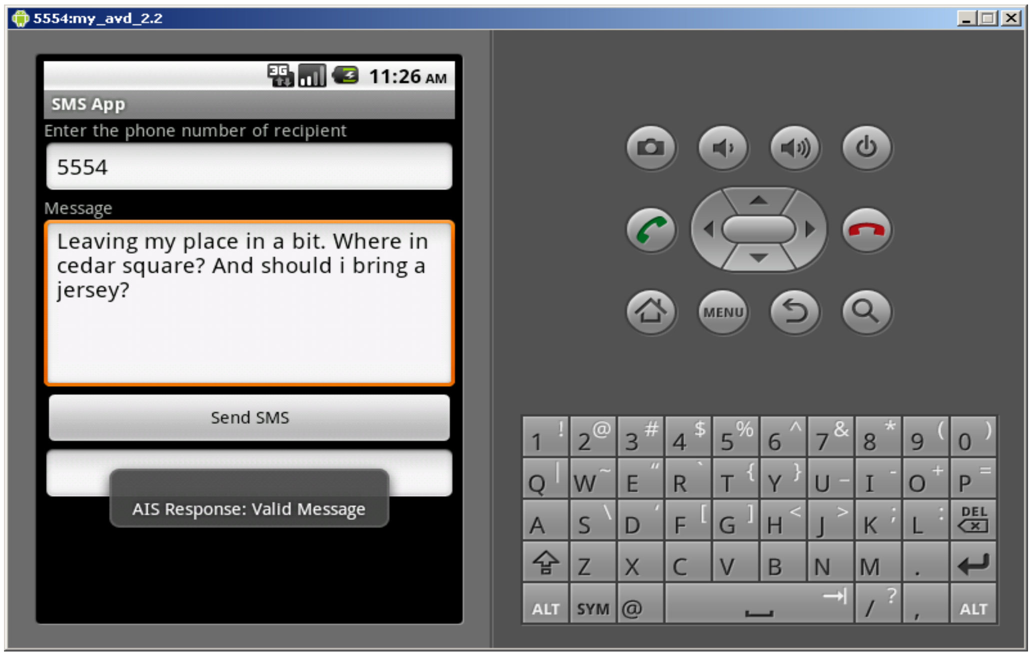
### 7.2.1 Demonstration I – AIS

This section details the demonstration of the AIS prototype, the first subsection details the running of the prototype, the second subsection tabulates the results of the demonstration and the third subsection provides a summary of this section.

#### *7.2.1.1 Running the artificial immune system Prototype on an Android emulator*

The AIS algorithm was implemented on an android mobile smart phone emulator. The Botnet

Detector captured all outgoing messages and extracted the message characteristics from the message body. These were saved in a SQLite3 database. The AIS then processed this data to determine if the message was valid (Figure 7.1) or not (Figure 7.2). Figure 7.1 and Figure 7.2 show responses to valid and invalid SMSs during the evaluation phase, respectively.


5554:my_avd_2.2

SMS App

Enter the phone number of recipient

5554

Message

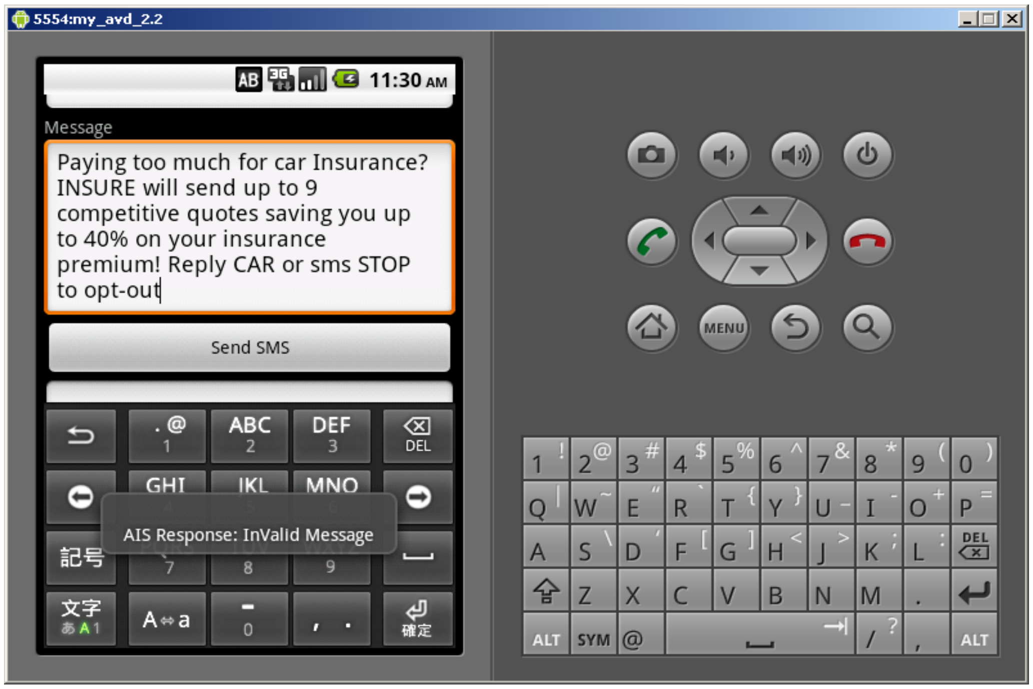
Leaving my place in a bit. Where in cedar square? And should i bring a jersey?

Send SMS

AIS Response: Valid Message

*Figure 7.1: Response to a valid message sent by Author*

The message in Figure 7.1 is a sample message collected from one of the author's mobile device. As discussed, these and other messages were used to train the AIS. When this message was picked up by the AIS, the message characteristics listed in the previous chapter were extracted and compared to all the signatures in the signature library (detectors). As this message does not match any detectors in the signature library it is classified as legitimate.



Paying too much for car Insurance? INSURE will send up to 9 competitive quotes saving you up to 40% on your insurance premium! Reply CAR or sms STOP to opt-out

AIS Response: InValid Message

*Figure 7.2: Response to an Invalid message*

The message in Figure 7.2 undergoes the same process as the message in Figure 7.1, except in this case the message signature is matched to a detector in the signature library, and is thus classified as invalid. The user will then be alerted. The next section details the results of this experiment.

### 7.2.1.2 Results and Discussion

The results of the experiment are tabulated as follows and show the accuracy in detecting spam SMSs.

*Table 7-1 : Results*

| Valid (non-spam) Message (Self) | Invalid (spam) Message (non - Self) | Total error |
|---|---|---|
| 84% | 65% | 20% |

The results tabled in Table 4 show that an AIS implementation can effectively detect invalid SMS messages. The results are now briefly discussed. The AIS correctly identifies 84% of non-spam (self) SMSs (i.e., 67 out of 80 valid messages). The AIS correctly identifies 65% of spam (non-self) SMSs (i.e., 13 out of 20 invalid messages). Its total error (incorrectly identified messages) is 20%.

Chart 7.1 shows the accuracy of the AIS in correctly identifying valid messages.

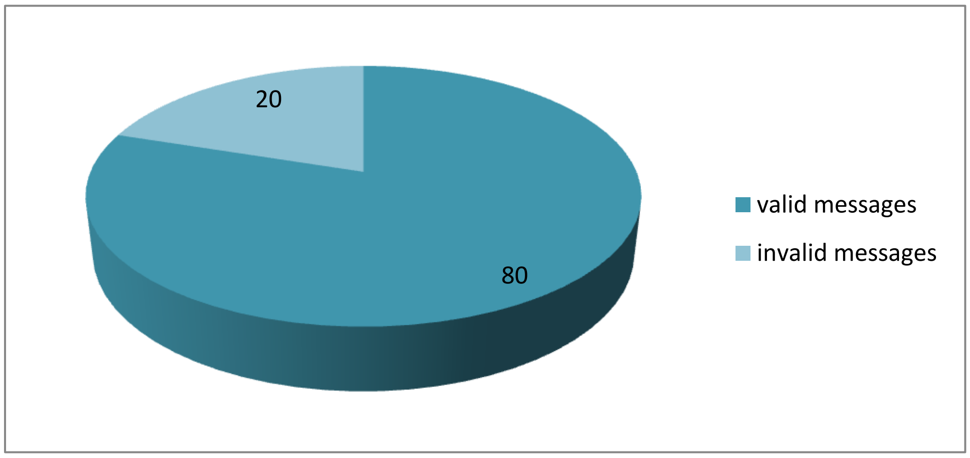


*Chart 7-1: Accuracy of AIS in identifying valid messages*

Chart 7.2 shows the accuracy of the AIS in correctly identifying invalid messages as spam.

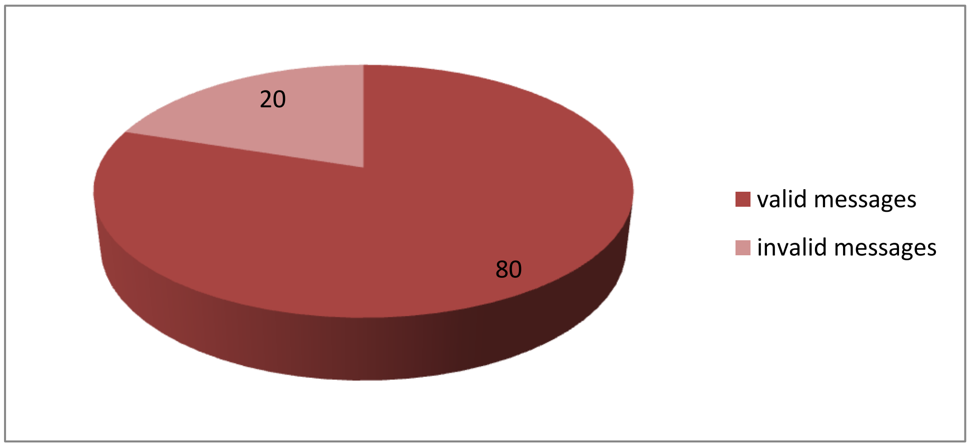


*Chart 7-2: Accuracy of AIS in identifying invalid messages*

This shows us that the AIS is better at accurately identifying valid messages than it is at identifying invalid messages. The accuracy of the AIS could be enhanced by adding more metrics to be analysed, thus, increasing the accuracy of the antigen binding. The size of the antibody list (which is used to match non-self messages) could be expanded.

The accuracy of the AIS implementation should improve with more learning and a larger antibody list size. The larger antibody size, would allow the AIS to better identify non-self cells and thus increase the accuracy of its responses. In the following section, the author will summarise the demonstration results.

##### *7.2.1.3 Summary – AIS*

The author started this research with the aim of combating potential threats to mobile devices, primarily the use of mobile devices to send spam SMSs to persons on their phone book, as well as others. As was shown in this demonstration, the AIS chosen was relatively successful.

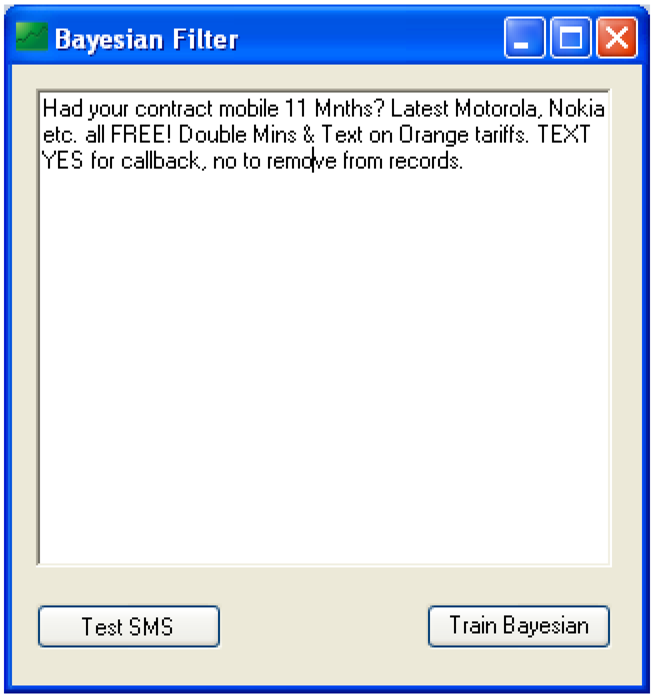
The author believes that this implementation would serve as a useful tool in alerting a user of possible Bots on their mobile device. The next section details the second prototype implemented, namely, the Bayesian filter Botnet detector.

#### 7.2.2 Demonstration II – Bayesian filter

This section details the demonstration of the Bayesian filter prototype, the first subsection details the running of the prototype, the second subsection tabulates the results of the demonstration and the third subsection concludes this section.

##### *7.2.2.1 Running the Bayesian filter prototype*

To test the Bayesian Botnet Detector (BBD) the authors extracted 86 spam SMSs to test the BBD from the Grumbletext website. Figure 7.3 is an example of a spam message being tested on the Bayesian Filter.



Had your contract mobile 11 Mnths? Latest Motorola, Nokia etc. all FREE! Double Mins & Text on Orange tariffs. TEXT YES for callback, no to remove from records.

*Figure 7.3: Screenshot of Bayesian filter*

The authors then extracted another 86 valid messages from one of the author's own phones to test for false positives. False positives refer to legitimate SMS messages that are mistakenly identified as spam.

As discussed in Chapter 3, the Bayesian spam filter works by first computing the spam probabilities of each individual word in a message and then combines the most "interesting" probabilities together to give us a single probability for the entire message. To determine the best number of "interesting" probabilities to combine, we tried various combinations. The combinations we experimented with were 15, 10, 5, and 3. These combinations are explained further in the next section.

#### *7.2.2.2 Results and Discussion*

The results of the demonstration are tabulated as follows and show the accuracy in detecting spam SMSs in Table 7.2.

*Table 7-2: Results of running BBD on spam messages.*

| Number of Combinations | Correctly identified as spam | Incorrectly not identified as spam |
|---|---|---|
| 15 | (74%) | (26%) |
| 10 | (74%) | (26%) |
| 5 | (82%) | (18%) |
| 3 | (87%) | (13%) |

As can be seen from the results in Table 7.2, reducing the number of probabilities we combine allows us to better identify a message as spam (true positive). This is intuitively what we would be expecting since the number of words in an SMS message usually ranges between 15 and 25 words. This is because SMS messages are limited to 160 characters (these numbers were obtained by counting the average number of words in a random selection of SMS messages sent by one of the authors over the last six months).

Most email spam filters would combine the top 15-20 most interesting probabilities (17). As SMS messages are generally much shorter than emails, it would make sense to combine a smaller set of probabilities to evaluate whether or not a message is spam. Figure 7.4 shows the BBDs performance in accurately identifying spam based on the number of combinations.

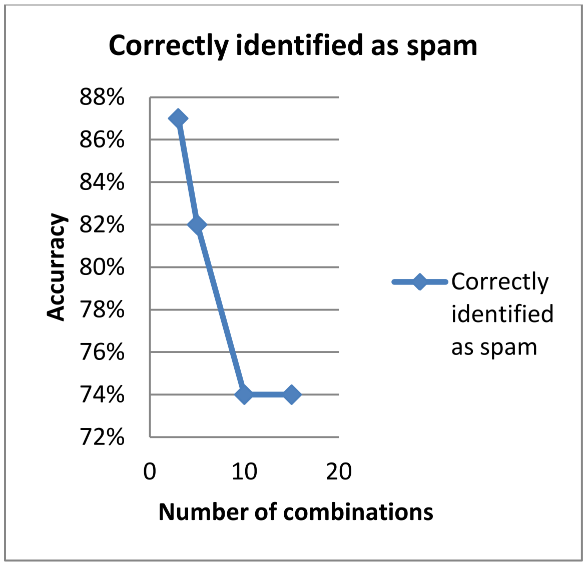


*Figure 7.4: Visualises the effect reducing the number of combinations has on the BBD's performance*

It should, however, be noted that most of the spam messages that were not correctly identified as spam were loaded with words that are very similar to SMSs that one of the authors would normally send. Figure 7.5 is an example of a message that was not correctly identified by the BBD as being spam.

your secret admirer is very caring about his friends and holds them dear to his heart. He will do anything for a close friend

*Figure 7.5: an example spam message not picked up by the BBD*

To expand this point further, let us consider the spam probabilities of the words that occur in this SMS in closer detail.

Figure 7.6 is an example spam SMS message with each word's spam probability.

```
your:0.422130102351602
secret:0.702064402663205
admirer:0.9998
is:0.245045552028142
very:0.040947132430149
caring:0.011
about:0.063779200737555
his:0.0275424076854587
friends:0.158827296392993
and:0.187233877650676
holds:0.187233877650676
them:0.0355117066206902
dear:0.011
to:0.368514614315756
his:0.0275424076854587
heart:0.0756322233148982
He:0.011
will:0.162777015101
do:0.0605925536861735
anything:0.011
for:0.342251411689513
close:0.184724884863709
friend:0.304517249692058
```

*Figure 7.6: An example spam SMS message with each word's spam probability*

As was discussed in Section 4.4, the spam probabilities of a selection of “interesting” words are combined to give us a single probability. The top three most “interesting” words in this example are admirer: 0.9998, caring: 0.011 and He: 0.011. These three probabilities combined, as per Equation 1 [11], are given in Equation 2.

$$\frac{(0.9998)(0.011)(0.011)}{(0.9998)(0.011)(0.011)+(1-0.9998)(1-0.011)(1-0.011)}$$

*Equation 7-1: Formula to combine probabilities*

The result of Equation 2 is 0.0068311747673944. A number close to one indicates that the message is spam and a number close to zero indicates that the message is valid. Thus, the message is incorrectly identified as non-spam. This example shows us that it is possible for some messages to get through the spam filter but, as shown by the example, it is not obvious even to a human filter that the message is spam.

We will now look at how well the BBD works in identifying legitimate messages by measuring the number of false positives, i.e. the number of legitimate messages, incorrectly identified as spam. Table 2 shows the results of the experiment when testing for legitimate messages.

*Table 7-3: BBD results after testing on legitimate messages.*

| Number of Combinations | Correctly identified as legitimate | Incorrectly identified as spam |
|---|---|---|
| 15 | (100%) | (0%) |
| 10 | (100%) | (0%) |
| 5 | (100%) | (0%) |
| 3 | (100%) | (0%) |

As can be seen from the results in Table 2, the number of false positives is zero. The next section summarises the demonstration of the Bayesian filter prototype.

#### *7.2.2.3 Summary – BBD*

Spamming Botnets have the potential to become more common place. With the increasing processing power of mobile phones, they are becoming more attractive for exploitation by malware. The aim of this research is to provide a tool that is not only capable of identifying spam SMSs being sent from a user's mobile device, but also to allow the spam filter to learn new spam features that are continuously changed by spammers to confuse spam filters.

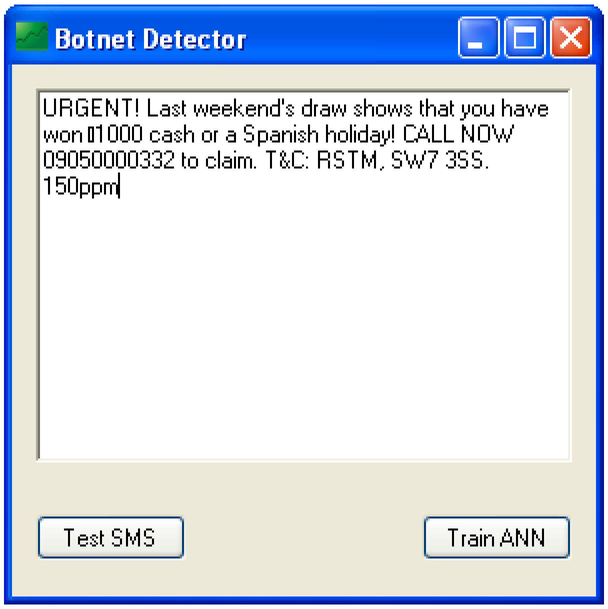
The BBD was capable of correctly identifying 87% of all spam messages. The author believe that over time, with more spam messages added to the black list, the accuracy of this implementation would improve. The next section will detail our third prototype, the artificial neural network.

### 7.2.3 Demonstration II – ANN

This section details the demonstration of the artificial neural network prototype. The first subsection details the running of the prototype, the second subsection tabulates the results of the demonstration and the third subsection concludes this section.

#### *7.2.3.1 Running the ANN*

To test the ANN we run the network against a collection of spam and ham messages extracted from one of the authors' mobile phones.  The following figure, Figure 7.7, is an example of a spam SMS message being tested on our ANN.



URGENT! Last weekend's draw shows that you have won £1000 cash or a Spanish holiday! CALL NOW 09050000332 to claim. T&C: RSTM, SW7 3SS. 150ppm

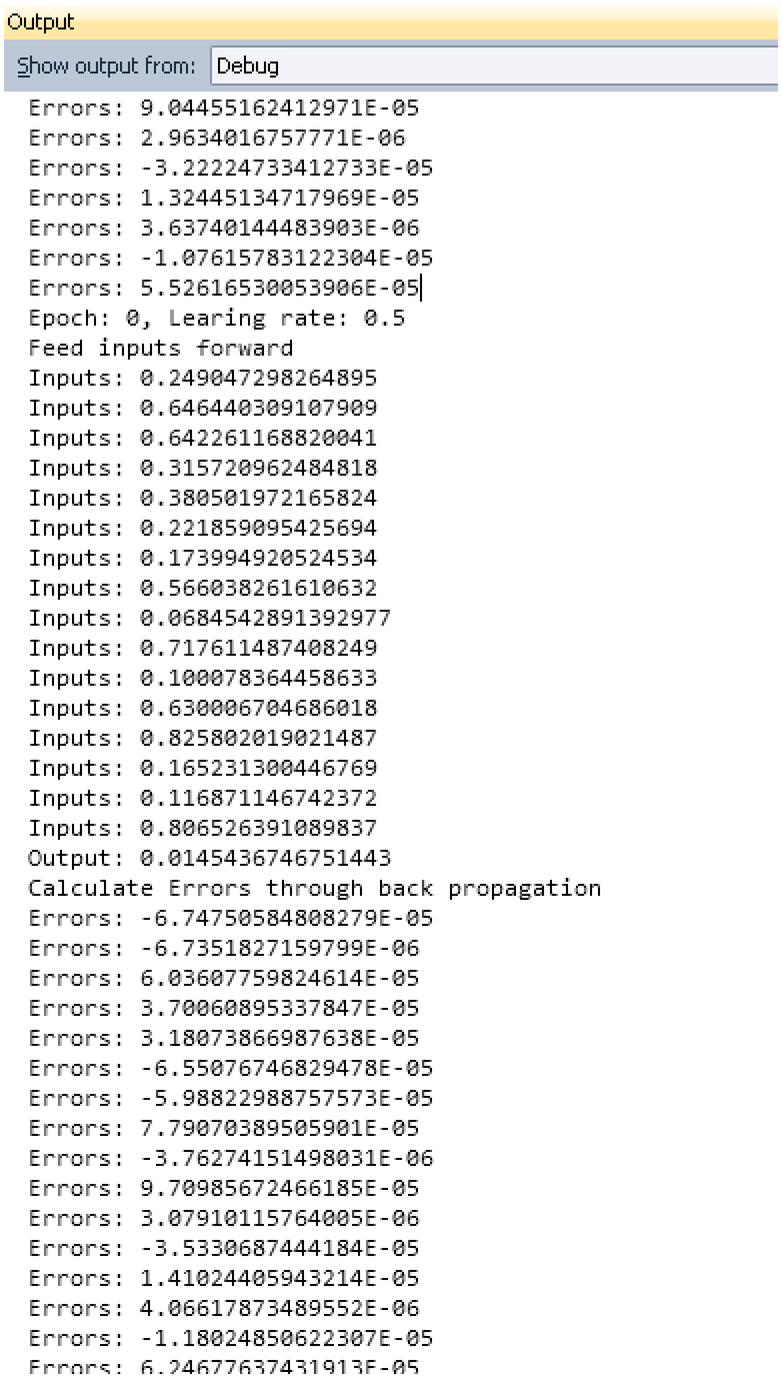
*Figure 7.7: Screenshot of our ANN processing a spam SMS*

The neural network looks at the message signatures as identified previously in the previous chapter. These inputs are the following:

- Does the SMS message contain links?
- Does the SMS message contain telephone numbers?
- The number of words in a message.
- The ratio of punctuation characters to words.
- The ratio of links to words in the message.
- The ratio of capital letters to words in the message.
- The ratio of misspelt words in the SMS message.
- Bayesian spam probability.

The ANN then generates a numerical statistic for each signature between 0 and 1.

The following figure, Figure 7.8, is a screenshot of the inputs and outputs generated by the ANN during training.


Output

Show output from: Debug

```
  Errors: 9.04455162412971E-05
  Errors: 2.9634016757771E-06
  Errors: -3.22224733412733E-05
  Errors: 1.32445134717969E-05
  Errors: 3.63740144483903E-06
  Errors: -1.07615783122304E-05
  Errors: 5.52616530053906E-05
  Epoch: 0, Learing rate: 0.5
  Feed inputs forward
  Inputs: 0.249047298264895
  Inputs: 0.646440309107909
  Inputs: 0.642261168820041
  Inputs: 0.315720962484818
  Inputs: 0.380501972165824
  Inputs: 0.221859095425694
  Inputs: 0.173994920524534
  Inputs: 0.566038261610632
  Inputs: 0.0684542891392977
  Inputs: 0.717611487408249
  Inputs: 0.100078364458633
  Inputs: 0.630006704686018
  Inputs: 0.825802019021487
  Inputs: 0.165231300446769
  Inputs: 0.116871146742372
  Inputs: 0.806526391089837
  Output: 0.0145436746751443
  Calculate Errors through back propagation
  Errors: -6.74750584808279E-05
  Errors: -6.7351827159799E-06
  Errors: 6.03607759824614E-05
  Errors: 3.70060895337847E-05
  Errors: 3.18073866987638E-05
  Errors: -6.55076746829478E-05
  Errors: -5.98822988757573E-05
  Errors: 7.79070389505901E-05
  Errors: -3.76274151498031E-06
  Errors: 9.70985672466185E-05
  Errors: 3.07910115764005E-06
  Errors: -3.5330687444184E-05
  Errors: 1.41024405943214E-05
  Errors: 4.06617873489552E-06
  Errors: -1.18024850622307E-05
  Errors: 6.24677637431913E-05
```

*Figure 7.8: Screen shot of ANN being trained*

Once these inputs have been fed into the ANN and the statistics are generated, the ANN will then take the collection of statistics generated and feed them as input vectors into the neural network. The neural network will have one output, which will be a probability between {0, 1} on whether the SMS is spam or ham. The neural network will then use back propagation to adjust the network to cater for errors.

Figure 7.9 shows how the ANN is trained. The Epoch refers to the iteration process of providing the ANN with inputs and updating the network weights. The ANN was trained on 4,000 epochs. A better method would be to look for an acceptable MSE but, for simplicities sake, the author trained the ANN for a set amount of epochs.

```
Epoch: 0
Feed inputs forward
Inputs: 0.182471694907655
Inputs: 0.652668916874137
Inputs: 0.522135609211864
Inputs: 0.228474132104753
Inputs: 0.302488233737601
Inputs: 0.318965437901895
Inputs: 0.125809140577694
Inputs: 0.282260473093567
Inputs: 0.0713620773399616
Inputs: 0.399295321810301
Inputs: 0.1066266900388
Inputs: 0.630810390007151
Inputs: 0.810849718052133
Inputs: 0.159386642325069
Inputs: 0.0872980852354105
Inputs: 0.719031425636609
Output: 0.132123222447607
Calculate Errors through back propagation
Errors: -0.00170063895063053
Errors: -0.000633659983175537
Errors: 0.00266652634776394
```

```
   Errors: 0.0013530538480361
   Errors: 0.000829718002416703
   Errors: -0.00250302803337286
   Errors: -0.00170490374559201
   Errors: 0.00261249437420277
   Errors: -0.000296725187280312
   Errors: 0.00378432228079181
   Errors: 0.000315559182914681
   Errors: -0.00146532077341316
   Errors: 0.00164839067338557
   Errors: 3.88493740721934E-05
   Errors: 0.00011717570702581
Errors: 0.00246104428359079
```

*Figure 7.9: ANN being trained.*

As can be seen in Figure 7.9, the inputs are fed into the ANN. The ANN then modifies the weights of the connections according to the input patterns that it is presented with. The author used back propagation (discussed in Chapter 3) to adjust these weights. The next section details the results of this demonstration.

#### 7.2.3.2 Results and Discussion

The results of the experiment are tabulated as follows and show the accuracy in detecting spam SMSs in Table 1. For this study the author compared the results from the ANN against a previous prototype that used a Bayesian filter to detect spamming Botnets.

*Table 7-4: Results*

| | Valid Message | Invalid Message | Total error |
|---|---|---|---|
| Neural Network | 100% | 95% | 2.5% |
| Bayesian Filter | 100% | 90% | 5% |

As can be seen in Table 7.4, the ANN has an accuracy of 100% in identifying valid messages, and an accuracy of 95% in identifying spam messages giving us a total error of 2.5%. The total error is calculated by determining how many of the spam and ham messages (40 in total, 50% spam and 50% ham) were incorrectly identified.

This compares well to the prototype that uses a Bayesian probability to detect spam, as shown in the second line of Table 7.4. The next section concludes the demonstration of this prototype.

#### *7.2.3.3 Summary - ANN*

Spam cannot be eliminated solely with technological solutions. To reduce spam, people must ideally stop responding to spam messages, for example, by actually purchasing items advertised in spam messages. Nevertheless, technical solutions play a strong role in combating spam. By allowing less spam to get through, we can reduce the incentive for spammers to spam and increase the cost of sending spam for spammers, thereby reducing spam.

The author believes that this neural network, when implemented, could reduce spam significantly. The advantage of situating the ANN on a network service provider’s server is that processing can be done by dedicated servers. The disadvantage, of course, is that the user cannot be prompted for feedback and confirmation, which might result in certain messages being incorrectly classified.

## 7.3 Conclusion

The author detailed and discussed the results of the three demonstrations that have been completed in order to focus on various features of the prototypes that were developed. These demonstrations showed the running of the prototypes under simulated conditions. The first demonstration demonstrated the running of the AIS prototype on an Android emulator. This was followed by a demonstration of the Bayesian filter prototype and, finally, by the third prototype, the ANN implementation of a spamming Botnet detector.

# Part V: Conclusion

# 8 Conclusion

## 8.1 Introduction

This chapter highlights the three main contributions that this study has made. It also discusses why these items should be considered contributions to the field of computer security. For this reason, this chapter is divided into four components, the first of which summarises the work detailed in this study, followed by the next three components, which focus on the three contributions, namely the Artificial Immune System (AIS) Botnet detector, the Bayesian filter Botnet detector and the Artificial Neural Network (ANN) prototype.

Section 8.2 starts off with a summary of what was detailed in this study. Section 8.3 discusses the contribution made by the AIS prototype, Section 8.4 details the contribution made by the Bayesian Filter prototype, followed by Section 8.5 which details the contribution made by the ANN. The chapter is then concluded with Section 8.6.

## 8.2 Summary

This study started off by providing the reader with background information on the subjects of spam and machine learning techniques in Chapters 2 and 3. The author discussed the threats posed by spam, the use of Botnets by spammers to conceal their identities and then the migration of spamming Botnets to mobile devices. The author then introduced the reader to three machine learning techniques demonstrated in this study. These techniques are artificial immune systems, Bayesian filters and artificial neural networks.

After the reader was provided with the necessary background, the dissertation then provided the reader with a list of requirements for the construction of a model to detect spamming Botnets in Chapter 4. Chapter 5 detailed the three models proposed by the author, each one using a machine learning technique detailed in Chapter 3, while adhering to the requirements set out in Chapter 4.

Chapter 6 discussed in detail how the prototype was set up. The chapter also detailed how the training and test data was selected. Chapter 7 then detailed how the three prototypes were tested and how their performance during testing. The following three sections will discuss the contribution made by the three prototypes in greater detail.

## 8.3 Artificial Immune System

The author started this research with the aim of combating potential threats to mobile devices, primarily the use of mobile devices to send spam SMSs to persons on their phone book as well as others.  An artificial immune system was one of the machine learning techniques chosen to accomplish this because AISs are unique in the sense that training only requires positive examples. This approach to detecting spamming Botnets was thought to be the most practical solution for the commercial application of the prototype.

The author believes that this implementation would serve as a useful tool for alerting a user of possible Bots on their mobile device. This would allow them to remove the malware from their device.

The author plans to improve upon the accuracy of the Bot Detector by adding to the list of metrics being extracted from the SMS and used to train the AIS implementation. Among the metrics the author plan on extracting, is the time of day that the messages are sent and perhaps the number of recipients per SMS. It is hoped that this will improve the accuracy of this implementation by adding a non-message metric to the analysis of  user behaviour.

The author hopes to apply these ideas out on an instant messaging platform, as well as on SMS messaging. The authors also hope to test the performance by testing the AIS on a mobile phone. As the current experiment was conducted on an Android emulator, the processing power and data storage capabilities of the AIS do not truly reflect those that would be found on a mobile device, as the mobile device would have limited storage and processing capabilities. The performance of the AIS would be expected to suffer as a result and there could be a noticeable slowdown in user performance.

## 8.4 Bayesian Filter

Spamming Botnets have the potential to become more common place. With the increasing processing power of mobile phones, they are becoming more attractive for exploitation by malware. The aim of this research is to provide a tool that is not only capable of identifying spam SMSs being sent from a user's mobile device, but also to allow the spam filter to learn new spam features as spammers continuously change the spam features to confuse spam filters.

The Bayesian Filter Botnet Detector (BBD) was capable of correctly identifying 87% of all spam messages. The author believes that, over time, with more spam messages added to the black list, the accuracy of this implementation would improve. The author will attempt to apply these ideas on an instant messaging platform, as well as on SMS messaging in future research.

As stated, this BBD  could be implemented on a network provider’s server. The advantage of situating the BBD on a network service provider’s server is that processing can be done by dedicated servers. Of course the disadvantage of installing it on a network server and not on a user’s mobile device is that the user cannot be prompted for feedback and confirmation, which might result in certain messages being incorrectly classified.

## 8.5 Artificial Neural Network

Spamming Botnets have the potential to become more common with the increasing processing power of mobile phones make mobile phones more attractive for exploitation by malware. The aim of this research is to provide a tool that is not only capable of identifying spam SMSs being sent from a user's mobile device, but also to allow the spam filter to learn new spam features as spammers continuously change the spam features to confuse spam filters.

The Artificial Neural Network Botnet detector (ANN) has been shown to be capable of correctly identifying 95% of all spam messages with a zero false positive rate. This shows that the ANN can be used to detect spamming mobile Botnets that have infected mobile devices.

The author believes that, over time, with more spam messages added to the black list, the accuracy of this implementation would improve. The author hopes to apply these ideas out on an instant messaging platform, as well as on SMS messaging.

## 8.6 Conclusion

This chapter was devoted to discussing the three contributions made in this research. The contributions built on, and thus reinforced one another. Each of the prototypes showed that they could be viable contributions to the computer security domain. All of the research above was conducted to answer the main research question: Is it possible to combat the migration of spamming Botnets onto mobile devices? This question was answered by providing three prototypes that were able to detect spamming Botnets.

The author believes that these prototypes could be further improved by tweaking the detection algorithms and improving the training set. The author hopes to apply these ideas to an instant messaging platform in the future.

## 9 References

1. *Integrating botnet simulations with network centric warfare simulations.* **Stytz, Martin R and Banks, Sheila B.** Orlando, Florida : SPIE, 2010. doi:10.1117/12.849183.

2. *Considerations and foundations for Botnet simulation.* **Stytz, Martin R. and Banks, Sheila B.** 2009. Data Mining, Intrusion Detection, Information Security and Assurance, and Data Networks Security 2009.

3. **The Economist.** Mobile operating systems. [Online] 27 January 2013. http://www.economist.com/blogs/babbage/2013/01/mobile-operating-systems.

4. *Mobile phones as computing devices: The viruses are coming!* **Dagon, David, Tom Martin, and Thad Starner.** s.l. : IEEE, 2004. pp. 11-15.

5. *When cell phones become computers.* **Want, Roy.** s.l. : IEEE, 2009, Pervasive computing, Vol. 8.2.

6. *SMS spam filtering: Methods and data.* **Delany, Sarah Jane, Buckley, Mark and Greene, Derek.** 10, s.l. : Expert Systems with Applications, August 2012, Vol. Volume 39, pp. Pages 9899–9908. ISSN 0957-4174, 10.1016/j.eswa.2012.02.053.

7. **Online, Acts.** Electronic Communications and Transactions Act. [Online] 2002. http://www.acts.co.za/ect_act/.

8. *An Inside Look at Botnets.* **Paul Barford, Vinod Yegneswaran.** s.l. : Springer, 2007, Malware Detection: Advances in Information Security, Vol. 27, pp. 171-191.

9. *Spam filter analysis.* **Garcia, Flavio D., Jaap-Henk Hoepman, and Jeroen Van Nieuwenhuizen.** Security and Protection in Information Processing Systems, s.l. : Springer , 2004, pp. 395-410.

10. **Magazine, Information Security.** Network Security. [Online] http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci85 9579,00.html.

11. *Intelligent optimisation techniques.* **Pham, D. T., and D. Karaboga.** New York : Springer, , 2000, Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks.

12. *In Defense of Spam.* **Neville, H.** 4, s.l. : IEEE Computer, Vol. 38.

13. **Association, Internet Service Providers'.** What is Spam? [Online] 2008. http://ispa.org.za/spam/.

14. **US-CERT Control Systems Security Center, Department of Homeland Security.** An Undirected Attack Against Critical Infrastructure. [Online] US-CERT Control Systems Security Center, Department of Homeland Security, September 2005. http://www.us-cert.gov/control_systems/pdf/undirected_attack0905.pdf.

15. Android Malware Up 472 Percent, Study Finds. *PC Magazine.* [Online] http://www.pcmag.com/article2/0,2817,2396558,00.asp.

16. **Beware, Spambot.** What is a spambot. [Online] http://www.turnstep.com/Spambot/info.html#whatspambot.

17. **Institute, SANS.** Bots & Botnet: An Overview. *SANS Institute.* [Online] http://www.sans.org/reading_room/whitepapers/malicious/bots-botnet-overview_1299.

18. *Acceptance and adoption of the innovative use of smartphone.* **Park, Yangil and Chen, Jengchung V.** 9, s.l. : Emerald Group Publishing Limited, 2007, Industrial Management & Data Systems, Vol. 107, pp. 1349 - 1365.

19. **Engelbrecht, Andries P.** *Computational Intelligence: An Introduction.* s.l. : John Wiley & Sons, 2007.

20. **de Castro, Leandro Nunes and Timmis, Jonathan.** *Artificial Immune Systems: A New Computational Intelligence Approach.* s.l. : Springer, 2002.

21. *A Bayesian approach to filtering junk e-mail.* **M. Sahami, S. Dumais, D. Heckerman, E. Horvitz.** s.l. : AAAI'98 Workshop on Learning for Text Categorization, 1998.

22. *Bayesian Inference in Statistical Analysis.* **Box, G.E.P. and Tiao, G.C.** s.l. : Wiley, 1973. ISBN 0-471-57428-7.

23. *Artificial Neural Network Modeling of the Rainfall-Runoff Process.* **Hsu, Kuo-lin, Gupta, Hoshin Vijai and Sorooshian, Soroosh.** 10, s.l. : John Wiley & Sons, 2010, Water Resources Research, Vol. 31, pp. 2517–2530.

24. —. **Kuo-lin Hsu, Hoshin Vijai Gupta, Soroosh Sorooshian.** 10, s.l. : WATER RESOURCES RESEARCH, 1995, Vol. 31.

25. **Spamhaus.** The Definition of Spam. [Online] http://www.spamhaus.org/definition.html.

26. **Group, Messaging Anti-Abuse Working.** Email Metrics Program: The Network Operators' Perspective. [Online] 2007. http://www.maawg.org/about/MAAWG20072Q_Metrics_Report.pdf.

27. *Better Bayesian filtering.* **Graham, Paul.** 2003.

28. **Sebastian Schrittwieser, Peter Fruhwirt, Peter Kieseberg, Manuel Leithner,.** *Guess Who's Texting You? Evaluating the Security of Smartphone Messaging Applications.* Vienna, Austria : SBA Research gGmbH.

29. *Detecting spam in voip networks.* **R. Dantu, P Kolan.** s.l. : SRUTI, 2005.

30. *Spotlight: the rise of the smart phone.* **Zheng, P. and Ni, L.M.** 3, s.l. : IEEE, 2006, Distributed Systems Online, Vol. 7.

31. *Intercepting mobile communications: the insecurity of 802.11.* **Borisov, Nikita, Goldberg, Ian and Wagner, David.** New York, USA : ACM Digital Library, 2001.

32. **McFee.** *McFee.* [Online] 2010. http://vil.mcafeesecurity.com/vil/content/v_138726.htm.

33. *Analysis of applicability of traditional spam regulations to VoIP spam.* **Park, So Young, Kim, Jeong-Tae and Kang, Shin-Gak.** [ed.] Advanced Communication Technology. s.l. : IEEE, 2006. pp. pp.3 pp.,1217.

34. *Developing a Legally Compliant Reachability Management System as a Countermeasure against SPIT.* **Hansen, Markus, et al.** Berlin : Third Annual VoIP Security Workshop, 2006.

35. *Characterizing comment spam in the blogosphere through content analysis.* **Bhattarai, A., Rus, V. and Dasgupta, D.** s.l. : IEEE, 2009. Computational Intelligence in Cyber Security. pp. 37,44. doi: 10.1109/CICYBS.2009.4925088.

36. Data protection: "Junk" e-mail costs Internet users 10 billion a year worldwide. *Europa Press Releases.* [Online] 2009. http://europa.eu/rapid/pressReleasesAction.do?reference=IP/01/154&format=HTML&aged=0&language=EN&guiLanguage=en.

37. *The impact that placing email addresses on the Internet has on the receipt of spam: An empirical analysis.* **Schryen, Guido.** s.l. : computers & security, 2007, Vol. 26, pp. 361-372.

38. *The "Social Engineering" of Internet Fraud.* **RUSCH, Jonathan J.** San Jose, Calif : Internet Society, 1999.

39. *IT outsourcing evolution---: past, present, and future.* **Lee, Jae-Nam et al.** s.l. : ACM, Vol. 46.5, pp. 84-89.

40. *Understanding how spammers steal your e-mail address: An analysis of the first six months of data from project honey pot.* **Prince, Matthew.** 2005. Second Conference on Email and Anti-Spam.

41. **Boneh, Dan.** *The Difficulties of Tracing Spam Email.* s.l. : Department of Computer Science Stanford University, 2004.

42. **Evan Cooke, Farnam Jahanian, Danny McPherson.** *The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets.* s.l. : The advanced computing systems association, 2005.

43. Big brother bosses. *The Economist.* [Online] 11 September 2009. http://www.economist.com/businessfinance/displaystory.cfm?story_id=14413380.

44. *Spam Filter Analysis.* **Flavio D. Garcia, Jaap-Henk Hoepman and Jeroen van Nieuwenhuizen.** s.l. : IFIP International Federation for Information Processing, 2004.

45. *Onion routing.* **Goldschlag, David, Michael Reed, and Paul Syverson.** 1999. Communications of the ACM. Vol. 42.2, pp. 39-41.

46. *Emerging Cyber Threats Report for 2009.* s.l. : Georgia Tech Information Security Center, 2008.

47. **Edmund K. Burke, Graham Kendall.** Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques. [Online] http://eprints.nottingham.ac.uk/621/1/03intros_ais_tutorial.pdf.

48. *Self-nonself discrimination in a computer.* **Forrest, Stephanie, et al.** s.l. : IEEE, 1994. Research in Security and Privacy.

49. *Increasing the accuracy of a spam-detecting artificial immune system.* **Oda, Terri, and Tony White.** s.l. : IEEE. Evolutionary Computation. Vol. 1.

50. **Deza, Elena Deza & Michel Marie.** Encyclopedia of Distances. s.l. : Springer, 2009.

51. **U. Aickelin, D. Dasgupta.** *Artificial Immune Systems.* s.l. : Springer, 2005.

52. *An artificial immune system approach to misbehavior detection in mobile ad hoc networks.* **Le Boudec, Jean-Yves, and Slaviša Sarafijanović.** s.l. : Springer Berlin Heidelberg, 2004. Biologically Inspired Approaches to Advanced Information Technology. pp. 396-411.

53. *An artificial immune system for data analysis.* **Timmis, Jon, Mark Neal, and John Hunt.** s.l. : Biosystems , 2000. Vol. 55.1, pp. 143-150.

54. **Cohen, Lee A. Segel and Irun R.** *Design Principles for the Immune System and Other.*

55. *A bayesian approach to filtering junk E-mail.* **Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz.** Madison : In Learning for Text Categorization, 1998.

56. *Bayesian Inference in Statistical Analysis.* **Box, G.E.P. and Tiao, G.C.** s.l. : Wiley, 1973. ISBN 0-471-57428-7.

57. *Content based SMS spam filtering.* **José María Gómez Hidalgo, Guillermo Cajigas Bringas , Enrique Puertas Sánz , Francisco Carrero García.** Amsterdam : ACM symposium on Document engineering, 2006. 10.1145/1166160.1166191.

58. *Feature engineering for mobile (SMS) spam filtering.* **Gordon V. Cormack, José María Gómez Hidalgo , Enrique Puertas Sánz.** Amsterdam : international ACM SIGIR conference on Research and development in information retrieval. 10.1145/1277741.1277951.

59. *Contributions of Artificial Neural Networks and Other Statistical Methods.* **Judith E. Dayhoff Ph.D, *, James M. DeLeo.** 8, s.l. : Conference on Prognostic Factors and Staging in Cancer Management, 2001, Vol. 91.

60. *A simple method for anzlyzing multifactorial data.* **Dugdale, A. E.** 1975, The American journal of clinical nutrition, pp. 788-792.

61. **COOK, NIGEL P.** *Introductory Digital Electronics.* s.l. : Prentice Hall, 1997.

62. *A survey of Decision Tree Classifier Methodology.* **Safavian, SR., and D.Langrebe.** s.l. : IEEE Transactions on Systems, Man and Cybernetics, 1991.

63. **Deza, Elena Deza & Michel Marie.** *Encyclopedia of Distances.* s.l. : Springer, 2009.

64. **Zaknich, Anthony.** *Artificial Neural Networks :An Introductionary Course.*

65. *Neural network design.* **Hagan, Martin T., Howard B. Demuth, and Mark H. Beale.** s.l. : Boston London, 1996.

66. *Prediction of corrosion behavior using neural network as a data mining tool.* **Kamrunnahar, Mst, and Mirna Urquidi-Macdonald.** s.l. : Corrosion Science, 2010, Vol. 52.3, pp. 669-677.

67. *Artificial neural network prediction of material removal rate in electro discharge machining.* **Panda, Deepak Kumar, and Rajat Kumar Bhoi.** 2005. Materials and Manufacturing Processes. Vol. 20.4, pp. 645-672.

68. *Applications of chemometric tools in corrosion studies.* **Luciano, Giorgio, Pierluigi Traverso, and Paola Letardi.** 2010. Corrosion Science. Vol. 52.9.

69. *Differential evolution training algorithm for feed-forward neural networks.* **Ilonen, Jarmo, Joni-Kristian Kamarainen, and Jouni Lampinen.** 2003. Neural Processing Letters . pp. 93-105.

70. *Recurrent neural networks for prediction: Learning algorithms, architectures and stability.* **Mandic, Danilo P., and Jonathon Chambers.** s.l. : John Wiley & Sons, Inc., 2001.

71. *Two new learning procedures for recurrent networks.* **Pearlmutter, Barak A.** s.l. : Neural Network Review, 1990.

72. *Neural smithing: supervised learning in feedforward artificial neural networks.* **Reed, Russell D., and Robert J. Marks.** s.l. : Mit Press, 1998.

73. *Adaptive pattern recognition and neural networks.* **Pao, Yohhan.** 1989.

74. *Theoretical neuroscience: Computational and mathematical modeling of neural systems.* **Dayan, Peter, Laurence F. Abbott, and L. Abbott.** 2001.

75. *Investigating Identity Concealing and Email Tracing Techniques.* **VURAL, I. and VENTER, H.S.** Johannesburg : Information Security South Africa, 2009. 978-1-86854-740-1.

76. *Using Network Forensics and Artificial Intelligence Techniques to Detect Bot-nets on an Organizational Network.* **VURAL, I. and VENTER, H.S.** Las Vegas : International Conference on Information Technology : New Generations (ITNG) 2010, IEEE Computer Society Washington, 2010. 978-1-4244-6270-4.

77. *Mobile Botnet Detection using Network Forensics.* **VURAL, I. and VENTER, H.S.** Berlin : 3rd Future Internet Symposium, Springer; Lecture Notes in Computer Science, 2010. 10.1007/978-3-642-15877-3_7.

78. *A Network Forensic Implementation for Detecting Mobile Botnets using Artificial Immune Systems.* **VURAL, I. and VENTER, H.S.** Orlando : International Conference on Digital Forensics, 2011.

79. *Combating Mobile Spam through Botnet Detection using Artificial Immune Systems.* **VURAL, I. and VENTER, H.S.** 6, s.l. : Journal of Universal Computer Science, 2012, Vol. 18.

80. *Spamming Botnet Detection using Neural Networks.* **VURAL, I. and VENTER, H.S.** Wroclaw : 9th International Workshop on Security in Information Systems, 2012.

81. *Combating Spamming Mobile Botnets through Bayesian Spam Filtering.* **VURAL, I. and VENTER, H.S.** George : outhern African Telecommunication Networks and Applications Conference, 2012.

82. *Identifying significant features for network forensic analysis using artificial intelligent techniques.* **Mukkamala S, Sung AH.** s.l. : Int'l Journal of Digital Evidence, 2003.

83. **Garfinkel, Simson.** *Web Security, Privacy & Commerce.* s.l. : Web Security, Privacy & Commerce, 2002.

84. Network Security. *Information Security Magazine.* [Online] http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci85 9579,00.html.

85. Android Edges RIM, Apple as Most Popular Smartphone OS. *PCWorld.* [Online] March 2011. http://www.pcworld.com/article/221358/android_edges_rim_apple_as_most_popular_smartphone_os .html.

86. **Shankland, Stephen.** Google'sAndroid parts ways with Java industry group. *CNET News.* [Online] http://www.news.com/8301-13580_3-9815495-39.html.

87. *Optimised Coverage of Non-self with Evolved Lymphocytes in an Artificial Immune System.* **Engelbrecht, A. J. Graaff and A. P.** 2, s.l. : International Journal of Computational Intelligence Research, 2006, Vol. 2.

88. *Artificial immune systems.* **Leandro N. De Castro, Fernando J. Von Zuben, Helder Kni.** s.l. : ICARIS, 2007.

89. *Immunology for physicists.* **A.S. Perelson, G. Weisbuch.** 4, s.l. : Reviews of Modern Physics, 1997, Vol. 69.

90. *Artificial immune systems: a new computational intelligence approach.* **De Castro, Leandro Nunes, and Jonathan Timmis.** s.l. : Springer Verlag, 2002.

91. *Innate immune recognition.* **Janeway Jr, Charles A., and Ruslan Medzhitov.** s.l. : Science Signaling, 2002, Vol. 20.1.

92. *Innate immune recognition and control of adaptive immune responses.* **Medzhitov, Ruslan, and Charles A. Janeway Jr.** s.l. : Academic Press, 1998, Seminars in immunology.

93. *An artificial immune system for data analysis.* **Timmis, Jon, Mark Neal, and John Hunt.** s.l. : Biosystems, 2000, Vol. 55.1, pp. 143-150.

94. *How your immune system works.* **Brain, Marshall.** s.l. : HowStuffWorks, 2004.

95. *An artificial immune system for data analysis.* **Timmis, Jon, Mark Neal, and John Hunt.** s.l. : Biosystems, 2000, Vol. 55.1, pp. 143-150.

96. *Architecture for an artificial immune system.* **Hofmeyr, Steven A., and Stephanie Forrest.** s.l. : Evolutionary computation, 2000, Vol. 8.4, pp. 443-473.

97. *John Holland's invisible hand: An artificial immune system.* **Forrest, Stephanie, and Steven A. Hofmeyr.** 1999, Festschrift held in honor of John Holland.

98. **Microsoft.** Windows Phone. [Online] Microsoft, February 2012. http://www.microsoft.com/windowsphone/en-us/default.aspx.

# Appendix A: Natural Immune Systems

## A.1 Introduction

This appendix gives the reader an overview of natural immune systems. The natural immune system is detailed here in order to improve the readers understanding of artificial immune systems which are discussed in Chapter 3. Section A.2 describes the goals of the natural immune system. Section A.3 details the adaptive immune system. A.4 describes the creation of antibodies. A.5 discusses auto immune reactions. A.6 discusses detection binding. The appendix is then concluded in Section A.7

## A.2 The Goal of an Immune System

The goal of an immune system is to differentiate between self and potentially dangerous non-self elements. In an anti-spam system we would want to differentiate between self (legitimate messages) and non-self (spam). Theories of how the biological immune system works can serve as starting point for creating computer systems. This section describes the components of the immune system.

The human body has many different mechanisms to defend itself. The natural immune system (NIS) detects foreign material inside the body that could be harmful to the body. The NIS differentiates between normal cells (self cells) and foreign cells (non-self cells). The immune system works on the principle of a pattern recognition system, recognising unwanted patterns (non-self cells) from the normal patterns (self cells) (66).

It is this classification of self and non-self that makes the immune system an appealing model for spam detection, which also requires a classification between the legitimate messages (the self) and spam (non-self).

As we have seen in the discussion of spam, most commercial-grade solutions use multiple techniques to achieve higher accuracy. Similarly, the immune system is not reliant upon a single technique, but instead consists of many layers which all help to protect the body (76).

The skin and mucous membranes form the outer layer of defence, a physical barrier against attack. Physiological defences, such as high pH or temperature, make it harder for pathogens to survive in the body.

The innate immune system (77) is a non-adaptive system available at birth. Its quick reaction to potential infections can stop many attacks before they can get underway, and it also serves to activate the adaptive immune system.

The adaptive or acquired immune system (78) can handle invaders which have been missed by the innate immune system. It takes longer to mount a response the first time something is encountered, but after that can detect the same thing or similar things very quickly. The adaptive immune system is what was used as a model for this work and it is described in greater detail in the next section.

## A.3 The Adaptive Immune System

The central component to the adaptive immune system is specialised white blood cells called lymphocytes. These serve to identify anything in your body, and act upon those things which are not part of self. Rather than attacking everything, the body uses a system called the major histocompatibility complex (MHC) (79), which marks the cells of the body as self. Anything not carrying these markings may be attacked by the immune system (67).

When a cell is infected, the MHC molecules present fragments called antigens to the surface of the cell. The immune system checks the antigens using a specialised detector called an antibody. Each lymphocyte actually has many copies of the same antibody on its surface, and it detects pathogens when antibodies on the lymphocyte cell surface bind to antigens. This binding does not have to be perfect. If the antibody and antigen are “close enough” to a perfect match, binding will still occur, although not as strongly as it would for a perfect match.

To apply these ideas to spam, we treat spam as a pathogen, and the complete message is used as an antigen. The lymphocytes are digital bits of information, and each one includes a pattern which is used as an antibody.

## A.4 Creation of Antibodies

Antibodies are constantly being created by the body (81) and each associated lymphocyte can live for as little as days or as long as years. The body has a library of gene fragments, which represent all the necessary information to create detectors for all the possible pathogen types. In order to create the repertoire or population of lymphocytes in the body at any given time, elements from this library are randomly recombined to produce a diverse population of receptors. This same concept is applied to constructing an anti-spam immune system digital library

## A.5 Autoimmune reactions

If the antibodies are created through random recombination, we can't be sure that these antibodies won't detect self cells. When antibodies detect self and a reaction occurs, this is called an autoimmune reaction (82). These are avoided through a fairly complex process.

There are actually classes of lymphocyte, the B-cells and the T-cells. The B-cells originate and mature largely in the bone marrow of humans. T-cells mature in the thymus (although they also originate in the bone marrow). The thymus contains many different types of self-proteins, and as the T-cells mature, those which match self are killed off or are not selected to reproduce.

Before the body actually mounts any attack on non-self cells, B-cells must detect a cell as being foreign, and a T-cell must activate the B-cell before any further action is taken. Since T-cells cannot detect self, this confirms that the B-cell is not mistakenly detecting self. In this way, the body avoids autoimmune reactions.

## A.6 Detection/Binding

Detection is done through binding (83). One lymphocyte's antibodies may bind to many different antigens, although some will bind more closely than others. The antigens to which a given antibody will bind must be similar in shape, but do not need to be exactly the same. A given detector will bind to many cells, and a given cell might have multiple detectors which can match it. The strength of the binding depends on how closely the two shapes can match.

There are approximately $10^{16}$ different foreign proteins which the natural immune system must recognize, yet the repertoire of the immune system contains a much smaller number of actual receptor types, closer to $10^{8}$ (41). This is accomplished by using approximate binding, thus the immune system can use a smaller number of antibodies to detect a large number of potential pathogens, as long as pathogens have similar shapes.

## A.7 Conclusion

This appendix introduced the reader to the natural immune system of the human body. The author thought it was important to add this appendix to assist the reader in understanding artificial immune systems, which were discussed in Chapter 3.

## Appendix B: Published Papers

The papers are presented in a chronological order of date published.

Investigating identity concealing and email tracing techniques. (Ickin Vural, Hein Venter) Information Security South Africa 2009.

Spammer detection using honeypots and digital forensics (Ickin Vural, Hein Venter) Southern African Telecommunication Networks and Applications Conference 2009 (WIP)

Using Network Forensics and Artificial Intelligence Techniques to Detect Bot-nets on an Organizational Network (Ickin Vural, Hein Venter) International Conference on Information Technology : New Generations (ITNG) 2010, IEEE Computer Society Washington, DC, USA , ISBN: 978-1-4244-6270-4

Mobile Botnet Detection Using Network Forensics (Ickin Vural, Hein Venter) Future Internet Symposium 2010, Lecture Notes in Computer Science, Springer Berlin/Heidelberg, 6369/2010, 57-67, ISBN: 978-3-642-15876-6

Detecting Mobile Spam Botnets Using Artificial immune Systems (Ickin Vural, Hein Venter) Advances in Digital Forensics VII, IFIP Advances in Information and Communication Technology, 2011, Springer Boston, Volume 361/2011, 183-192, ISBN: 978-3-642-24211-3

Combating Mobile Spam through Botnet Detection using Artificial Immune Systems (Ickin Vural, Hein Venter) Journal of Universal Computer Science, Vol. 18, No. 6, pp. 750-774.

Spamming Botnet Detection using Neural Networks (Ickin Vural, Hein Venter) , 9th International Workshop on Security in Information Systems - WOSIS 2012

Combating Spamming Mobile Botnets through Bayesian Spam Filtering, (Ickin Vural, Hein Venter) Southern African Telecommunication Networks and Applications Conference 2012

# INVESTIGATING IDENTITY CONCEALING AND EMAIL TRACING TECHNIQUES

**[1]Ickin Vural, [2]HS Venter**

Information and Computer Security Architectures Research Group (ICSA)
Department of Computer Science, University of Pretoria

[1]ickin@tuks.co.za
[2]hventer@cs.up.ac.za

## ABSTRACT

At present it is very difficult to trace the identity of spammers who use identity concealment techniques. It is difficult to determine the identity of the spammer by just analysing the electronic trail.

This paper will look at standard email tracing techniques and how email senders try and hide their electronic trail. The identity concealing techniques that that are discussed are: Spoofing, Bot-Networks, Open proxies, Open mail relays and untraceable Internet connections. The techniques used to trace spam that we discuss are: Header analysis and honeypot computers.

The paper will also Investigate advanced digital forensics techniques for email tracing namely Investigating residual data on servers and investigating network devices.

## KEY WORDS

Digital Forensics, Electronic tracing, identity concealment techniques, Spoofing, Bot-Networks, Open proxies, Open mail relays, untraceable Internet connections, Header analysis, honeypot computers.

# INVESTIGATING IDENTITY CONCEALING AND EMAIL TRACING TECHNIQUES

## 1 INTRODUCTION

Unsolicited bulk communication also known as spam is the practise of sending unwanted e-mail messages, frequently with commercial content, in large quantities to an indiscriminate set of recipients *(*Spamhaus, 2009).

The sending of unsolicited bulk communications with the intention to advertise products and generate sales is economically viable because senders have no operating costs beyond the management of their mailing lists. Because the cost of setting up a spamming operation is low spammers are numerous. Thus the volume of unsolicited bulk communications has increased dramatically over the past few years *(Messaging Anti-Abuse working group, 2007).*

The costs of spam, involve lost productivity and fraud, these costs are borne by the general public, institutions that store and retrieve mail for their employees and by Internet service providers. Institutions and Internet service providers have been forced to add extra capacity to cope with the high volumes of unsolicited bulk communications *(Europa Press Releases, 2009)*.

Anti spamming legislation has been introduced in many jurisdictions. The problem faced by law enforcement is that spammers move their operations to jurisdictions that have no or weak anti spamming laws.

At present it is very difficult to trace the identity of spammers who use identity concealment techniques. It is difficult to determine the identity of the spammer by just analysing the electronic trail using standard email tracing techniques.

This paper focuses on current email tracing techniques and how email senders try and hide their electronic trail. The objective is to present the current state of tracing the origin of unsolicited bulk communications and then suggest techniques utilising digital forensics in an attempt to trace spam.

The remainder of the paper is structured as follows. The background section defines spam in more detail and also defines its cost and causes. . The next two sections are devoted to the state of the art of spamming techniques and how to trace spammers. More specifically, these two sections contrast each other in the sense that the third section looks at techniques that spammers use to conceal their identities, whereas the fourth section looks techniques for tracing the origins of spam so that the spammers can be identified. The paper's main contribution is purported in the next section, which discusses advanced digital forensics techniques for email tracing.

## 2 BACKGROUND

Unsolicited bulk email otherwise known as spam is an email sent to a large number of email addresses, where the owners of those addresses have not asked for or consented to receive the mail *(Internet Service Providers' Association, 2008).* Spam is used to advertise a service or a product. An example of spam is an unsolicited email message from an unknown or forged address advertising Viagra.

Spam is one of the most significant threats to the Internet, accounting for around 60% of all email traffic (Internet Service Providers' Association, 2008). Spam costs consumers and ISPs lots of money in bandwidth charges. Despite the growing number of technological means for combating spam, the spammers somehow manage to stay one step ahead and the deluge shows little sign of abating. .

Spammers generally do not pay much for the sending of spam. They accomplish this by exploiting open mail servers to do their task for them. The spammer need only send one email message to an incorrectly configured mail server to reach thousands of email addresses, with the bulk of the transfer being handled by the mis-configured mail server. Recipients in turn need to pay access costs or telephone costs in order to receive content they didn't ask for.

ISPs have to bear the bulk of the cost for bandwidth overuse by spammers, this cost is often passed onto the consumer through increased Internet access fees or a degraded service level.

With the introduction of the "Electronic Communications and Transactions Act, 2002" unsolicited emails now have a legal definition and the sending of spam is illegal *(Acts Online, 2002).* Spammers if identified are liable for a fine and prosecution. Thus spammers will attempt to cover their trail to prevent identification.

Spammers are able to send email and cover their trail because Emails use Standard Mail Transfer Protocol (SMTP) which is not a secure protocol and can be tampered with *(Tzerefos, P. Smythe, C. Stergiou, I. Cvetkovic, S. 1997).*

Emails consist of two main parts a message header and a message body. The message header contains information about the destination network address and the source network address as well as routing information. The email headers are not secure and can be easily forged to add false routing data and to hide the source network address. This paper will discuss both email concealment and email tracking techniques respectively in the following two sections.

The following section will describe in detail techniques used by spammers to conceal their identities from persons who would attempt to identify the source of spam mail.

## 3 HOW SPAMMERS CONCEAL THEIR IDENTITIES

Spammers conceal their identities for a number of reasons. If they are based in a jurisdiction which has strict anti-spamming laws they do not want to be traced for fear of prosecution. If they are based in a jurisdiction which has weak anti-spamming laws then the primary motive is not to be traced and blacklisted. As many ISP’s will block any mail from blacklisted sites. The techniques studied are Spoofing, Bot-Networks, Open proxies and untraceable Internet connections.

### 3.1 Spoofing

Spoofing is the process whereby a spammer would insert fictitious headers into the email address to hide the network address of their computer. The spammer will usually insert fake “From” and “Reply-To” headers into the email, these headers would point to a non-existent network address or more commonly an innocent third parties’ network address *(Boneh, Dan, 2004).*

## 3.2 Bot- Networks

A Bot-Network consists of a set of machines that have been taken over by a spammer using Bot software sent over the Internet. This Bot software hides itself on its host machine and periodically checks for instructions from its human Bot-Network administrator. Botnets today are often controlled using Internet Relay Chat *(Evan Cooke, Farnam Jahanian, and Danny McPherson. 2005).* The owner of the computer usually has no idea that his machine has been compromised until its Internet connection is shut down by an ISP. As most ISP’s block bulk mail if they suspect it is spam the spammers who control these Bot-Networks typically send low volumes of mail at any one time so as not to arouse suspicions. Thus the spam mail can be traced to an innocent individuals network address and not the spammers network address.

While the number of Botnets appears to be increasing, the number of bots in each Botnet is actually dropping. In the past Botnets with over 80 000 machines were common *(Evan Cooke, Farnam Jahanian, and Danny McPherson. 2005).* Currently Botnets with a few hundred to a few thousands infected machines are common. One reason for this is that smaller Botnets are more difficult to detect and may be easier to sell or rent.

## 3.3 Open Proxies

An open proxy is a machine that allows computers to connect through it to other computers on the Internet. Open proxies exist because they enable unhindered Internet usage in countries that restrict access to certain sites for political or social reasons. An Internet user in a country that restricts Internet access can access blocked sites by using an open proxy in a country that does not restrict Internet access.

Spammers use open proxies to hide their network addresses. The recipient of a spammers email will not see the spammers’ network address on the email but the open proxy's network address. It is estimated that sixty percent of all spam is sent using an open proxy *(Boneh, Dan, 2004)*.

### 3.4 Open mail relays

Emails sent over the Internet pass through a number of gateways on their way from the sender to the receiver, these gateways are called mail relays. Each time an email passes through a mail relay it has a Received header inserted, this will have the network address of the computer that connected to the mail relay.

An open mail relay is a mis-configured mail relay that accepts mail from any computer on the Internet and forwards it to any other computer on the Internet as opposed to a normal mail relay that accepts mail from a limited number of computers on the Internet and forwards it to a limited number of computers *(Flavio D. Garcia , Jaap-Henk Hoepman and Jeroen van Nieuwenhuizen. 2004).*

This helps the spammer conceal his identity as it appears that the mail is from the open relay and not from the spammer. However as the spammers network address is still found in the emails headers the spammer would insert fake headers into the email. Open mail relays are usually used together with open proxies to conceal the network address of the spammer.

### 3.5 Untraceable Internet connections

Spammers can also conceal their identities by accessing the Internet from Internet cafes, university computer labs and by using stolen 3G cards. There is thus no way of tracing spammers who access the Internet using these methods. Even if the network address of the computer used is identified this cannot be connected to the identity of the spammer.

## 4 HOW DIGITAL FORENSIC INVESTIGATORS CAN TRACE THE IDENTITY OF SPAMMERS

The two primary methods for tracing the origins of spam are header analysis and honeypot computers. The following section studies methods for email tracing and their limitations. The methods studied are header analysis and honeypot computers.

## 4.1 Header analysis

By studying the email headers in a spam email we should be able to identify the senders' network address. Spammers know this and try and divert us from the trail by inserting fake headers. In addition as mentioned previously by using open proxies or Bot-networks the network address of the spammer is not even on the headers. Thus the use of header analysis to trace spammers is highly unlikely.

The only header that cannot be easily forged is the first received header, as all the others may be faked. Spammers will fake their headers to conceal their network addresses. This means that header analysis is not a time and cost effective method to use when tracing spam. The following figure shows an email with the various header tags.

```
Microsoft Mail Internet Headers Version 2.0

Received:       from       s058eml004004.ds1.ad.absa.co.za[1]
([10.6.50.91])  by  V058EMLFFF004.ds1.ad.absa.co.za[2]  with
Microsoft SMTPSVC(6.0.3790.3959);

        Thu, 5 Mar 2009 11:47:49 +0200

Received:   from   S200INT006001   ([169.202.65.146])   by
s058eml004004.ds1.ad.absa.co.za       with       Microsoft
SMTPSVC(6.0.3790.3959);

        Thu, 5 Mar 2009 11:47:49 +0200

Received: from relayin-at1.absa.co.za ([169.202.65.20]) by
S200INT006001 with InterScan Message Security Suite; Thu, 05
Mar 2009 12:03:31 +0200

Received:   from   kendy.up.ac.za   ([137.215.101.101])   by
relayin-at1.absa.co.za            with            Microsoft
SMTPSVC(6.0.3790.3959);

        Thu, 5 Mar 2009 11:45:59 +0200

Received:   from   b040pc181.up.ac.za[3]   ([137.215.40.181]
helo=notebook)

       by kendy.up.ac.za with esmtp (Exim 4.63)

       (envelope-from <hventer@cs.up.ac.za>)

       id 1LfAB1-0001RS-AW

       for   Ickin.Vural@absa.co.za;   Thu,   05   Mar   2009
11:47:39 +0200
```

```
From: "Prof. Hein Venter" <hventer@cs.up.ac.za>
To: <Ickin.Vural@absa.co.za>
Subject: Meeting next week
Date: Thu, 5 Mar 2009 11:45:02 +0200
Message-ID: <FC414216270A45DEAEB887C3BF3C8A18@UP>
MIME-Version: 1.0
Content-Type: text/plain;
       charset="us-ascii"
Content-Transfer-Encoding: 7bit
X-Mailer: Microsoft Office Outlook 11
Thread-Index: Acmddw5GmQGAim2STqiIW8+jHkG6AQ==
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.5579
X-Scan-Signature: 1241d9d45ce102941afa91f8ab9dc533
Return-Path: hventer@cs.up.ac.za
X-OriginalArrivalTime:  05  Mar  2009  09:46:03.0937  (UTC)
FILETIME=[33047910:01C99D77]
```

*Figure 4.1 An Email Header*

The above email is sent by the University of Pretoria's email sever b040pc181.up.ac.za as highlighted at number [3] in the figure. This is sent to the relay server relayin-at1.absa.co.za [2] which, in turn, sends it to Absa's email server s058eml004004.ds1.ad.absa.co.za [1]. This shows us how we can find the identity of the email sender by tracing the route the email took by analysing the header. But as mentioned earlier a spammer can tamper with these headers so as to confuse an investigator.

## 4.2 Honeypot computers

A honeypo*t* is a closely monitored computing resource that is intended to be compromised *(Niels Provos. 2004)*. A honeypot computer can be applied to Bot-networks, open proxies and open relays. Thus by setting up a computer to imitate an open proxy or a Bot-network, investigators can attempt to trap the spammers into revealing their network addresses.

### 4.2.1 Honeypots on Bot-Networks

One way of identifying spammers is to set up a computer to pretend that it is part of a Bot-network *(Boneh, Dan, 2004)*. By allowing the honeypot computer to become part of the Bot-network we can obtain the Bot-network software used by the spammer. Once this has been done the honeypot waits for the spammer to send new instructions and then identifies the network address of the sender. The problem with this approach is that spammers could send the instructions to the Bot-networks under their control over open relays and open proxies thus it may be impossible to discover the identity of the spammer's network address.

### 4.2.2 Honeypots on open proxies

By setting up a honeypot on an open proxy and waiting for spammers to use it in order to send their spam, we can attempt to identify the spammer's network address. This could be done by keeping records of all connections made by the proxy to locate the source of the spam.

The fake open proxies emulate a subset of the HTTP protocol. Requests made with methods other than GET and CONNECT are answered with an error message. GET requests are answered with a randomly generated page. CONNECT requests to port 25 are internally redirected to an emulated open relay. The reason for this redirection is that the spammer may think nothing went wrong and he is connected to the SMTP server he requested, while he actually is connected to a honeypot. CONNECT requests to ports other than 25 are served with a "Request timeout" message (*Mauro Andreolini,Alessandro Bulgarelli, Michele Colajanni and Francesca Mazzoni. 2005)*.

To identify spammers, it is necessary to encourage them to use honeypot services to their advantages. This is done through the deployment of fake servers, such as open proxies. To ensure traceability of their actions, logging must be enabled for the honey pot open proxy *(Mauro Andreolini,Alessandro Bulgarelli, Michele Colajanni and Francesca Mazzoni. 2005)*.

Spammers try and get around this by using a proxy chain. A proxy chain is when a spammer sends spam mail through a chain of open proxies and, thus, reducing the chances of their network addresses being compromised *(Boneh, Dan, 2004)*.

### 4.2.3 Honeypots on open relays

This works by setting up a honeypot on an open relay and waiting for spammers to use it. We would then be able to trace the network address of the spammer using the open relay to send spam.

The fake open relays emulate a SMTP server. All the main commands of the SMTP protocol are implemented, so that spammers cannot notice the difference with a real server. When an e-mail is sent through the open relay, it actually does not reach destination, since all messaged are logged but not forwarded, except the very first one. This is done in order to fool a spammer who sends a first probe message to himself to see if the service is properly running *(Mauro Andreolini,Alessandro Bulgarelli, Michele Colajanni and Francesca Mazzoni. 2005)*.

This technique is similar to that used on open proxies. Thus the spammer would attempt to get around this in the same manner by using a relay chain. The following section describes some advanced techniques used in identifying spammers.

# 5 ADVANCED EMAIL TRACING TECHNIQUES USING DIGITAL FORENSICS

As discussed previously it is very difficult to obtain the network addresses of spammers. This paper discussed techniques such as header analysis and honeypot computers used to discover the network addresses of spammers. This section discusses digital forensic techniques used to determine the network addresses of spammers. The techniques studied are investigating network devices such as routers, investigating residual data on servers and using bait tactics to identify spammers.

## 5.1 Investigating Network devices

If logs are unattainable from the servers used by the spammer a routers log can be used instead to obtain information about the spammers network address. *(Patryk Szewczyk, 2007)* If say no access was given to the server logs of the ISP or proxy server that sent the email, the investigator can analyse the log files of the router or switch that routed the email. This should enable an investigator to determine where the email was sent from.

## 5.2 Investigating Residual data on servers

SMTP servers keep a copy of emails even after they have been delivered. By using this information we can trace the address of the computer that made the connection. By analysing these in a proxy server the identity of the computer making the connection could be obtained. This would require access to the servers which might not always be given as the proxy server might be located in a jurisdiction that does not have anti spamming laws *(Al-Zarouni Marwan, 2004)*.

## 5.3 Using Bait tactics to identify spammers

If the email address of the spam message is genuine, forensic investigators can e-mail a message to the sender containing an http "<img src>" tag where the source of the picture is placed on an http server. As soon as the person receiving the e-mail opens it, a log entry with his IP address is recorded on the http server holding the image. This tracks down the sender of the e-mail and establishes his ownership of the e-mail account (Al-Zarouni Marwan, 2004). However this technique is not always useful as some browsers automatically block the downloading of images by default (Microsoft Office online, 2009).

If the person receiving the e-mail is using a proxy server, his IP address will not show in the HTTP logs but rather, the IP of the Proxy server he/she used. In this case the proxy logs can be checked for persons accessing that picture at that time.

If the person in question is using an open proxy server that does not cooperate with law enforcement, one of the following two tactics can be used to track him/her down:

1. Java Applet: The investigator sends an e-mail with an "embedded" Java applet that runs on the receiver's machine and extracts his IP address and e-mails it to the investigator.

2. Active X Control: The investigator sends an e-mail address containing an HTML page with Active X that extracts the receiver's IP address and other information from his machine and sends it to the investigator.

# 6 SPAMMER IDENTIFICATION

One of the issues with identifying spammers is that SMTP is not a secure protocol and can be tampered with. Some researchers have advocated the adoption of a secure email protocol. *(A. Herzberg, 2005)* But until such time that this technology is widely adopted, and its usefulness would be limited if spammers make use of bot-networks and open proxies to send spam, other means of discouraging spammers must be found.

Spammers on the other hand are in the business of spamming because they want to make a profit. Spammers send spam advertising a product that they hope to sell and a bank account number to which payment should be made. By tracing the information in the email message body an investigator should be able to identify the source of some spam. This however is not enough as the enterprise can deny having sent the spam mail and the investigators may not be able to conclusively prove ownership.

Thus a proposal to identify spammers would be to create an implementation that identifies computers which are sending spam. These computers should then be added to a spam list that could be blocked by an ISP. This framework would need to identify bot-networks as well as open proxies sending spam. Once on a spam list, it is up to the individual or organisation concerned to have their network address removed from the spam list.

The detection of spamming computers can be done by analysing the network layer traffic and determining patterns that match bot-networks and open proxies sending spam.

The implementation would detect spam by analysing data provided by an ISP to identify abnormal behaviour on the network to identify spammers. This system will enable ISP's to proactively locate open proxies and bot-networks.

This would require analysing large data sets which would require a large amount of computing power. However the computing power of computers has increased and computers such as blade servers that can be programmed to analyse data using parallel computing techniques are now available. Thus it is possible for ISP's to analyse large datasets to detect abnormal behaviour in a way that was not possible a number of years ago.

## 7 DISCUSSION

The implementation of a system to detect spammers by analysing network traffic for abnormal behaviour has some shortcomings, mainly that spammers do not usually send out mail in bulk but in smaller packets so as to avoid detection.

The implementation would have to take into account spam email sending patterns to effectively identify spammers. The implementation could either make use of artificial intelligence to learn behaviour by feeding it training patterns or by using graph analysis which is perhaps better suited for large scale data analysis.

The authors of this paper, however, still need to explore their ideas mentioned in this paper in future work so as to produce a proof-of-concept prototype.

## 8 CONCLUSION

This paper outlines the challenges facing digital forensic investigators when attempting to identify spammers. Servers that contain forensic data such as log files showing the network addresses of the computers that have connected to it, that would enable a digital forensic investigation, are not made available by the servers' owners for various reasons. The usual reason for the refusal is that court orders requesting this data to be made available, may not apply to that jurisdiction.

The use of bot-networks means that even if the source of the machine sending the spam is identified the person owning the machine is not the one responsible for sending spam. The use of untraceable Internet connections and open proxies to communicate instructions to bot-networks makes the use of Honeypots unlikely to succeed.

Thus any success in tracing spammers will be matched by spammers using increasingly sophisticated techniques to evade detection. Greater responsibility will have to fall to ISP's in monitoring connections to open proxies as well as attempting to shut down open relays. Nevertheless an arms race between spammers and forensic investigators will continue for the foreseeable future.

## 9 REFERENCES

Al-Zarouni Marwan. 2004. Tracing E-mail Headers. We-B Centre & Edith Cowan University.

Boneh, Dan. 2004. The Difficulties of Tracing Spam Email. Department of Computer Science Stanford University.

A. Herzberg, Controlling Spam by Secure Internet Content Selection, Proceedings of Secure Communication Networks (SCN) 2004, LNCS vol. 3352, Springer-Verlag.

Acts Online, 2002. Electronic Communications and Transactions Act, 2002. Available: http://www.acts.co.za/ect_act/. [April 2009]

Europa Press Releases, 2009. Data protection: "Junk" e-mail costs Internet users 10 billion a year worldwide. Available: http://europa.eu/rapid/pressReleasesAction.do?reference=IP/01/154&format=HTML&aged=0&language=EN&guiLanguage=en. [April 2009]

Messaging Anti-Abuse Working Group, 2007.Email Metrics Program: The Network Operators' Perspective. Available: http://www.maawg.org/about/MAAWG20072Q_Metrics_Report.pdf . [April 2009]

Evan Cooke, Farnam Jahanian, Danny McPherson. 2005 . The advanced computing systems association. [Online] The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets. Available: http://www.usenix.org/events/sruti05/tech/full_papers/cooke/cooke_html/ [April 2009]

Flavio D. Garcia , Jaap-Henk Hoepman and Jeroen van Nieuwenhuizen, 2004. Spam Filter Analysis: IFIP International Federation for Information Processing. Springer Boston

Internet Service Providers' Association, 2008. 'What is Spam?' Available: http://www.ispa.org.za/spam/whatisspam.shtml. [April 2009]

Microsoft Research. S-GPS: Spammer Global Positioning System. Available: http://research.microsoft.com/en-us/projects/S-GPS/. [April 2009]

Mauro Andreolini, Alessandro Bulgarelli, Michele Colajanni and Francesca Mazzoni. 2005. The advanced computing systems association. [Online] HoneySpam: Honeypots fighting spam at the source. Available: http://www.usenix.org/event/sruti05/tech/full_papers/andreolini/andreolini_html/. [April 2009]

Microsoft Office online, 2009. About protecting your privacy by blocking automatic picture downloads. [Online] Available: http://office.microsoft.com/en-us/outlook/HP010440221033.aspx. [April 2009]

Niels Provos. 2004. The advanced computing systems association. [Online]. A Virtual Honeypot Framework. Available: http://www.usenix.org/event/sec04/tech/full_papers/provos/provos_html/. [April 2009]

Patryk Szewczyk, 2007. ADSL Router Forensics Part 1: An introduction to a new source of electronic evidence We-B Centre & Edith Cowan University.

Spamhaus, 2009. The Definition of spam. Available: http://www.spamhaus.org/definition.html [April 2009]

Tzerefos, P. Smythe, C. Stergiou, I. Cvetkovic, S. 1997. A comparative study of Simple Mail Transfer Protocol (SMTP), PostOffice

Protocol (POP) and X.400 Electronic Mail Protocols. Proceedings., 22nd Annual Conference on Publication Date: 2-5 Nov1997.

Yao Zhaoy , Yinglian Xie, Fang Yu, Qifa Ke, Yuan Yu, Yan Cheny, and Eliot Gillumz BotGraph: Large Scale Spamming Botnet Detection. Microsoft Research

# Combating Spamming Mobile Botnets through Bayesian Spam Filtering

Ickin Vural, H.S Venter
Department of Computer Science
University of Pretoria
email: Ickin@tuks.co.za,hventer.cs.up.ac.za

**Abstract- Malicious software (malware) infects large numbers of mobile devices. Once infected these mobile devices may be involved in many kinds of online criminal activity, including identity theft, unsolicited commercial SMS messages, scams and massive coordinated attacks. Until recently, mobile networks have been relatively isolated from the Internet, so there has been little need to protect them against Botnets. Mobile networks are now well integrated with the internet, so threats on the internet, such as Botnets, have started to migrate to mobile networks. This paper studies the potential threat of Botnets based on mobile networks, and proposes the use of Bayesian spam filtering techniques to detect Spamming Botnets. We then simulate mobile Bot detection by detecting mobile spam originating from a mobile device.**

**Index Terms**—Botnet, mobile, malware, Bayesian spam filtering

## I. INTRODUCTION

The field of computer security, which for many years focused on paradigms such as network security, information security and workstation security, is facing a paradigm shift with the ever-increasing gain in popularity of mobile devices such as smart phones and tablets. As many of the current threats to mobile devices (also known as cell phones or mobile phones) are similar to those that threaten desktop machines connected to the internet, many of the same solutions can be adapted to deal with mobile devices. Nevertheless, mobile devices present their own unique challenges such as a fragmented operating system market (such as Apple Os, Android, Windows mobile, RIM etc.), a proliferation of manufactures building devices on different standards, as well as the more limited processing and data storage capabilities of mobile devices. Security solutions have to be programmed with these limitations in mind.

This migration of computing from desktop devices to smart phones and tablets has lead to the appearance of threats that initially only affected desktop devices. The threat that this paper addresses is the migration of spamming Botnets onto mobile devices. Botnets are now capable of infecting mobile devices and using them to send mobile spam.

The paper is structured as follows: the background section introduces the topics of spam, Botnets and Bayesian spam filtering. The following section introduces a model on combating mobile spam through Botnet detection using Bayesian spam filtering. This is followed by a description of the prototype, the actual implementation of the prototype, a section where experimental results are tabulated. The paper is then concluded.

## II. BACKGROUND

### A. *Spam*

Unsolicited bulk mail, otherwise known as spam, is an email (electronic message) sent to a large number of email addresses, where the owners of those addresses have not asked for or consented to receive the email [1]. Spam is used to advertise a service or a product. One of the most well-known examples of spam is an unsolicited email message from an unknown or forged address advertising Viagra [2]. The lack of a universally recognized definition for spam is one of the major obstacles to creating solutions designed to minimize its harmful effects. The definition of spam could range from any unsolicited emails [3], to unsolicited commercial emails sent by any source [4], to unsolicited commercial emails from sources the recipient has never had contact with [5], to simply emails or postings transmitted in bulk quantities [6].

Spam is not limited to e-mail as is usually thought to be the case. Spam also exists in text messaging services (SMS). In the case of an SMS, spam can cost even more that it would when received through email. For example, assume that a user has subscribed to receive a notification via SMS when he/she receives an email. Depending on the particular cellular network, the user might have to pay for every SMS received regardless if it is a spam or a valid email.

### B. *Botnets*

A Botnet is a network that consists of a set of machines that have been taken over by a spammer using Bot software Bot software (or Bots for short) is a kind of malware that is often distributed in the form of a Trojan horse [7]. The challenge for businesses and banks in the near future will be to produce secure mobile applications while ensuring ease of use at the same time [8]. The motivation for installing a Botnet that sends spam on a mobile device and installing one on a desktop differ. In the case of Botnets on desktops the motivation is to prevent the spammer's identity being revealed, in the case of mobile Botnets that is not the only motive the cost of sending SMS messages is exponentially higher than the cost of sending email messages. Thus another motivation for mobile Botnets is to pass on the cost of sending the message to someone else. Malware writers can also use Bots to send SMSs to premium rate numbers. The following section discusses Bayesian filtering.

## III. 2.3 Bayesian Filtering Techniques

This section gives an overview of Bayesian spam filtering techniques. Firstly an overview of Bayesian filtering is given. This is then followed by an explanation of Bayesian filtering applied to mobile spam.

### A. *Bayesian filtering*

Bayesian spam filtering is a statistical technique that makes use of probabilities to classify email as being spam or non spam [9]. This classification is done by correlating the message contents with spam and non spam messages and then using Bayesian inference [10] to calculate the probability of it being a spam message or not. The process works as follows. Some words will have a particular probability of occurring in spam messages and legitimate messages. The spam filter will not know the probabilities of these words occurring in advance, but is trained to learn the probabilities by use of both black and white lists. Black lists contain message samples that are definitely spam and white lists contain message samples that are definitely legitimate. These black and white lists are built up manually by the user. This is accomplished by adding available spam messages to the black list and available legitimate messages to the white list. The size of the dataset used to train the filter is important as filter performance climbs with increased data to train upon [11]. The size of the dataset used to train the filter in this experiment was limited by the number of legitimate and spam SMSs the authors were able to collect. The data selection process is expanded upon in section 4.2. The following section discusses the application of Bayesian filtering to mobile spam.

### B. *Bayesian filtering applied to mobile spam*

Having a good term representation is one of the most important parts for getting a good classifier [12]. The spam filter should be designed with the SMS message characteristics in mind, namely only 160 characters are allowed on a standard SMS message [12]. The problem with this limitation is that fewer words means less information that can be transferred. Also, due to this constraint, people tend to use acronyms when writing SMSs [12]. SMS messages lack structured fields and their text is rife with abbreviations and idioms [13]. Moreover, the abbreviations used by SMS users are not standard for a language, but they depend on the user communities. Such language variability provides more terms to the word collection. For example, the sender of an SMS could replace the word 'tomorrow' with the abbreviation 'tmw'. This would usually be understandable to the SMS recipient, but there would now be two permutations of the word 'tomorrow' to analyse, hence, expanding the collection of words for which their spam probability need to be calculated.

## IV. A Model for a Botnet Detector using Bayesian Filtering

In this section we introduce the model for our spamming Botnet detector and explain the reasons for it being modelled in this manner as well as its advantages and disadvantages. We then round this section off by explaining how the prototype would be used in a real-world situation to combat Botnet spam.

### A. *Remote analyses vs. Analysis on the device*

There are two ways in which one could analyse a mobile device user's SMS data. The first possibility would be to analyse the SMSs sent on the ISP's servers and use this to build a profile of the user's SMS sending behaviour. There are, however, several drawbacks to this approach. Firstly, there are privacy implications of analysing SMS data on an ISPs server, but even if these concerns were addressed, the classification algorithm would have to determine whether or not a message is valid or invalid without the user's input. Thus, it would not be able to learn based on user feedback.
The second solution is to implement the detection algorithm on the device, thus, taking care of the privacy implications as well as allowing for user feedback in the learning process. The major disadvantage of this approach is that the prototype needs to be installed on a mobile device which has limited processing power and storage space, especially when compared to an ISP's server. Thus, the prototype had to be programmed to use minimum storage and be optimised to make use of as little memory as possible.

### B. *Flow Diagram for Botnet Detector*

The detection of SMS spam is a typical classification problem where patterns, also known as signatures, need to be classified as legitimate or not. Thus, it is a binary classification problem in which the data set, consisting of messages, are classified as either spam or non spam. Bayesian filtering is ideal for this sort of classification problem as we can train the spam filter on legitimate and non-legitimate (spam) messages. From this data the filter should be able to deduce whether messages are spam or not. Figure 2 visualizes how the Botnet detector is designed to work. The mobile user enters a text message and sends it to a recipient. This message is intercepted and analysed by the BBD, which then determines whether the message is valid. If the BBD can determine that the message is valid, the message is sent onwards. Else the network provider will be alerted by the BBD so that the service provider can investigate the malware that is sending these spam messages and remove it from the mobile device.

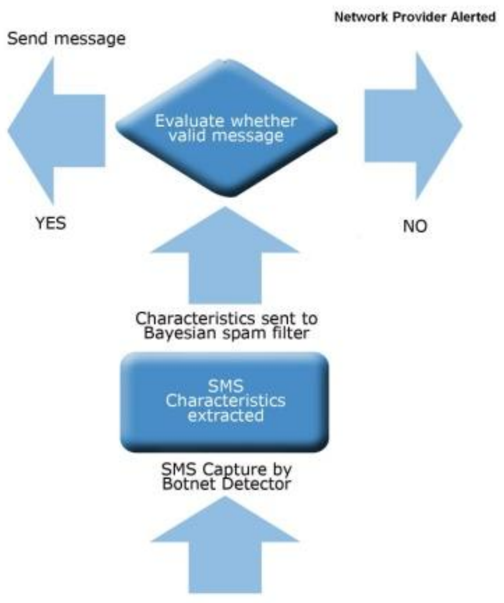

Network Provider Alerted
Send message
Evaluate whether valid message
YES
NO
Characteristics sent to Bayesian spam filter
SMS Characteristics extracted
SMS Capture by Botnet Detector

*Figure 2: Model for the Botnet detector*

The premise behind the BBD implementation is that the authors believe that a spamming Botnet that has installed itself on a user's mobile device, and is sending out spam SMSs without the knowledge of the mobile devices user, can be detected by the BBD. Thus, if these spam messages are blocked, the BBD would have succeeded in preventing the sending of spam as well as saving the mobile device owner the cost of the spam SMSs being sent. Additionally, the BBD would alert the mobile device user to the presence of Botnets on their device so that the malware, that has installed itself on their device, can be removed.

## V. IMPLEMENTATION OF SPAMMING BOTNET DETECTOR – A PROTOTYPE

This section describes how the prototype was implemented. The authors discuss how the algorithm used to train the BBD was selected, how the dataset used to train the BBD module was chosen, and how the Bayesian spam filtering was implemented.

### *A. Algorithm selection*

As discussed in section 3, the detection of SMS spam is a typical classification problem where patterns need to be classified as legitimate or not. Thus, it is a simple binary classification problem in which the data set (messages) are classified as either spam or non spam. As stated in section 3, Bayesian filtering is ideal for this sort of classification problem as we can train the spam filter on legitimate (white list) and non legitimate (black list) messages and from this data the filter should be able to deduce whether messages are spam or not.
The algorithm selected is an implementation of Paul Graham's original Naïve Bayesian Spam filtering algorithm [11]. The statistical approach to detecting spam used by the BBD does not attempt to identify the individual properties of spam, such as the words "click on the link". Rather, we leave it up to the algorithm to determine the probability of a message being spam or not based on the datasets used to train the filter on. Unlike feature-recognising spam filters like SpamAssassin [11], which assign a score to messages, the BBD, which uses Bayesian filtering, assigns a probability to a message as an indication of it being spam or not. Because it is measuring probabilities, the BBD considers all the evidence in the SMS message, both good and bad. Words that occur disproportionately rarely in spam, like "later" or "morning", contribute as much to decreasing the probability as possible spam words like "click" and "opt-in" do to increasing it. Therefore, an otherwise innocent SMS message that happens to include the word "Viagra" is not going to be misidentified as spam.

### *B. Data selection*

The data that was used to train the BBD implementation was selected by using SMS messages collected by the SMS Spam Collection project [15] and from the Grumbletext website [14]. Grumbletext is a UK forum in which cell phone users make public claims about SMS spam messages, most of them without reporting the spam message received.. In total 4815 of these valid SMSs were used to create a white list of legitimate SMS messages. The second set, i.e. the black list of invalid SMSs, totalled 746 (including blacklisted URLs).

### *C. Prototype setup*

When the BBD initialises, it reads through the white and black lists and builds a list of words. It then iterates through every word in the list and looks up its individual spam probability. Figure 3 shows an example set of words and their spam probabilities. The closer the probability value is to 1, the greater the probability of it being spam. The closer the probability value is to 0, the greater the probability of it being a legitimate SMS message.

```
.0218,would
.4691,Would
.3293,wow
.9861,www
.9998,x150p
.9998,XCHAT
.5956,Xmas
.9998,XMAS
.2866,xx
.5956,XX
.3616,xxx
.4010,XXX
.8154,xxxxxxx
.0427,ya
.0256,Yeah
.2164,year
.2903,years
.9999,YES
.0686,yesterday
.0602,yet
.1971,yo
.0655,Yo
.1402,you
.3673,You
```

*Figure 3: List of random words and their spam probabilities*

After the BBD was built, we could start testing it by feeding it valid and invalid SMS messages. Using Paul Graham's original Naïve Bayesian Spam filtering algorithm [11] we process the SMS message body by iterating through every word in the SMS body, and looking up its individual spam probability. We then sort this list of spam probabilities in descending order based on how far our probability is from 50%.

Once our list is compiled and sorted we then combine the most "interesting" probabilities together into to give us a single probability which is used to determine whether the whole message is spam or not. In this case "interesting" is defined as how far our score is from 50%, i.e. how close the score is to indicating a word is spam or valid. Equation 1 gives us the equation used to combine the individual probabilities. In section 5 this is explained further with a real example.

$$\frac{abc \ldots n}{abc \ldots n + (1-a)(1-b)(1-c) \ldots (1-n)}$$

*Equation 1: Formula to combine probabilities*

### *D. Running the BBD Prototype*

To test the BBD the authors extracted 86 spam SMSs to test the BBD from the Grumbletext website [14]. The authors then extracted another 86 valid messages from one of the author's own phones to test for false positives. False positives refer to legitimate SMS messages that are mistakenly identified as spam. As discussed in the previous section, the spam filter works by first computing the spam probabilities of each individual word in a message and then combines the most "interesting" probabilities together to give us a single probability for the entire message. To determine the best number of "interesting" probabilities to combine, we tried various combinations. The combinations we experimented with were 15, 10, 5, and 3. These combinations are explained further in section 5

## VI. Tabulation of Results and Discussion

The results of the experiment are tabulated as follows and show the accuracy in detecting spam SMSs in Table 1.

| **Number of Combinations** | **Correctly identified as spam** | **Incorrectly not identified as spam** |
|---|---|---|
| 15 | (74%) | (26%) |
| 10 | (74%) | (26%) |
| 5 | (82%) | (18%) |
| 3 | (87%) | (13%) |

*Table 1: Results of running BBD on spam messages.*

As can be seen from the results in Table 1, reducing the number of probabilities we combine allows us to better identify a message as spam (true positive). This is intuitively what we would be expecting since the number of words in an SMS message usually ranges between 15 and 25 words, because SMS messages are limited to 160 characters (these numbers were obtained by counting the average number of words in a random selection of SMS messages sent by one of the authors over the last six months). Most email spam filters would combine the top 15 - 20 most interesting probabilities [11]. As SMS messages are generally much shorter than emails, it would make sense to combine a smaller set of probabilities to evaluate whether or not a message is spam. Figure 4 shows us the BBD's performance in accurately identifying spam based on the number of combinations.

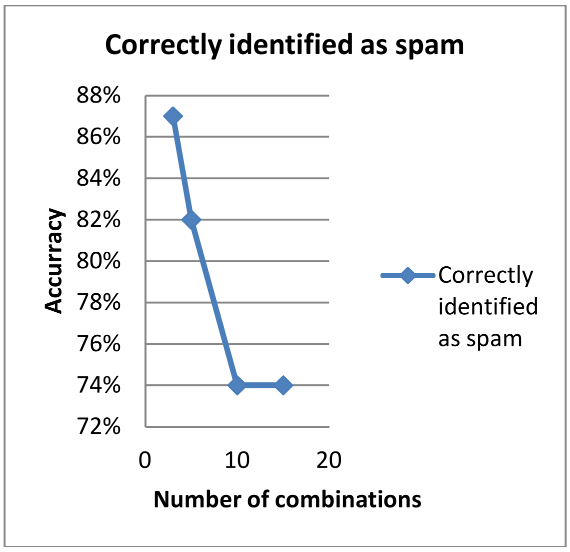


*Figure 4: Visualises the effect reducing the number of combinations has on the BBD's performance*

It should, however, be noted that most of the spam messages that were not correctly identified as spam were loaded with words that are very similar to SMSs that one of the authors would normally send. Figure 5 is an example of a message that was not correctly identified by the BBD as being spam.

```
your secret admirer is very caring
about his friends and holds them
dear to his heart. He will do
anything for a close friend
```

*Figure 5: An example spam message not picked up by the BBD*

To expand this point further, let us consider the spam probabilities of the words that occur in this SMS in closer detail. Figure 6 is an example spam SMS message with each word's spam probability.

```
your:0.422130102351602
secret:0.702064402663205
admirer:0.9998
is:0.245045552028142
very:0.0409412132430149
caring:0.011
about:0.0631279200737555
his:0.0275424076854587
friends:0.158827296392993
and:0.187233877650676
holds:0.187233877650676
them:0.0355117066206902
dear:0.011
to:0.368514614315756
his:0.0275424076854587
heart:0.0756322233148982
He:0.011
will:0.162777015101
do:0.0605925536861735
anything:0.011
for:0.342251411689513
close:0.184724884863709
friend:0.304517249692058
```

*Figure 6: An example spam SMS message with each word's spam probability*

As was discussed in section 4.4, the spam probabilities of a selection of "interesting" words are combined to give us a single probability. The top three most "interesting" words in this example are admirer: 0.9998, caring: 0.011 and He: 0.011. These three probabilities combined as per Equation 1 are given in equation 2.

$$\frac{(0.9998)(0.011)(0.011)}{(0.9998)(0.011)(0.011) + (1 - 0.9998)(1 - 0.011)(1 - 0.011)}$$

*Equation 2: Formula to combine probabilities*

The result of equation 2 is 0.0068311747673944. A number close to one indicates that the message is spam and a number close to zero indicates that the message is valid. Thus, the message is incorrectly identified as non spam. This example shows us that it is possible for some messages to get through the spam filter, but as shown by the example it is not obvious even to a human filter that the message is spam.

We will now look at how well the BBD works in identifying legitimate messages by measuring the number of false positives, i.e. the number of legitimate messages, incorrectly identified as spam. Table 2 shows the results of the experiment when testing for legitimate messages.

| **Number of Combinations** | **Correctly identified as legitimate** | **Incorrectly identified as spam** |
|---|---|---|
| 15 | (100%) | (0%) |
| 10 | (100%) | (0%) |

| | | |
|---|---|---|
| 5 | (100%) | (0%) |
| 3 | (100%) | (0%) |

*Table 2: BBD results after testing on legitimate messages.*

As can be seen from the results in Table 2, the number of false positives is zero.

## VII. Conclusion and Further Work

Spamming Botnets have the potential to become more common place. With the increasing processing power of mobile phones, they are becoming more attractive for exploitation by malware. The aim of this research is to provide a tool that is not only capable of identifying spam SMSs being sent from a user's mobile device, but also to allow the spam filter to learn new spam features as spammers continuously change the spam features to confuse spam filters.

The BBD was capable of correctly identifying 87% of all Spam messages. The authors believe that over time, with more spam messages added to the black list, the accuracy of this implementation would improve. The authors will attempt to apply these ideas on an Instant Messaging platform as well as on SMS messaging in future research.

This BBD, as stated in the paper, could be implemented either on a mobile device or on a network provider's server. The advantage of installing it on a mobile device is that the user can be prompted to confirm whether a particular message is indeed spam or not in the case that the neural network is unsure. The disadvantage of installing the BBD on a user's mobile device is the degradation of performance that could be experienced when the BBD calculates its spam probability. The advantage of situating the BBD on a network service provider's server is that processing can be done by dedicated servers. The disadvantage, of course, is that the user cannot be prompted for feedback and confirmation, which might result in certain messages being incorrectly classified.

## VIII. Acknowledgments

This work is based on research supported by the National Research Foundation of South Africa (NRF) as part of a SA/Germany Research cooperation programme. Any opinion, findings and conclusions or recommendations expressed in this material are those of the author(s) and therefore the NRF does not accept any liability in regard thereto

## IX. References

[1] Internet Service Providers' Association, 2008. 'What is Spam?' Available: http://www.ispa.org.za/spam/whatisspam.shtml. [April 2009]

[2] Spam-site "Samples of Spam" http://www.spam-site.com/spam-sample.shtml [September 2011]

[3] B.G. Kutais, "Spam and Internet Privacy", 'Journal of High Technology Law Suffolk University Law School'

[4] Consumer fraud reporting, "Spam Emails and Spamming", http://www.consumerfraudreporting.org/spam.php [September 2011]

[5] Federal Communication Commissio, "Spam: Unwanted Text Messages and Email", http://www.fcc.gov/guides/spam-unwanted-text-messages-and-email [September 2011]

[6] Spamhaus "The Definition of Spam", http://www.spamhaus.org/definition.html [September 2011]

[7] Seach Security.com "botnet (zombie army)". http://searchsecurity.techtarget.com/definition/botnet [September 2011]

[8] Emerging Cyber Threats Report for 2009, Georgia Tech Information Security Center, October 15, 2008

[9] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz (1998). "A Bayesian approach to filtering junk e-mail". AAAI'98 Workshop on Learning for Text Categorization.

[10] Box, G.E.P. and Tiao, G.C. (1973) Bayesian Inference in Statistical Analysis, Wiley, ISBN 0-471-57428-7

[11] Paul Graham (2003), Better Bayesian filtering

[12] José María Gómez Hidalgo , Guillermo Cajigas Bringas , Enrique Puertas Sánz , Francisco Carrero García, Content based SMS spam filtering, Proceedings of the 2006 ACM symposium on Document engineering, October 10-13, 2006, Amsterdam, The Netherlands [doi>10.1145/1166160.1166191]

[13] Gordon V. Cormack , José María Gómez Hidalgo , Enrique Puertas Sánz, Feature engineering for mobile (SMS) spam filtering, Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, July 23-27, 2007, Amsterdam, The Netherlands [doi>10.1145/1277741.1277951]

[14] Grumbletext, Available: http://www.grumbletext.co.uk/vt.php?t=333&subj=complaints++SMS+Spam+%28General%29+complaint, accessed 24/02/2012

[15] SMS Spam Collection, Available: http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/, accessed09/05/2

# Spamming Botnet Detection using Neural Networks

Ickin Vural and Hein Venter

Department of Computer Science, University of Pretoria, Pretoria, South Africa

**Abstract.** The dramatic revolution in the way that we can share information has come about both through the Internet and through the dramatic increase in the use of mobile phones, especially in developing nations. Mobile phones are now found everywhere in the developing world. In 2002, the total number of mobile phones in use worldwide exceeded the number of landlines and these mobile devices are becoming increasingly sophisticated. For many people in developing countries their primary access point to the internet is a mobile device. Malicious software (malware) currently infects large numbers of mobile devices. Once infected, these mobile devices may be used to send spam SMSs. Mobile networks are now infected by malicious software such as Botnets. This paper studies the potential threat of Botnets based on mobile networks, and proposes the use of computational intelligence techniques to detect Botnets. We then simulate mobile Bot detection by use of a neural network.

## 1 Introduction

The field of computer security, which for many years focused on paradigms such as network security, information security and workstation security, is facing a paradigm shift with the ever-increasing gain in popularity of mobile devices such as smart phones and tablets. As many of the current threats to mobile devices (also known as cell phones or mobile phones) are similar to those that threaten desktop machines connected to the internet, many of the same solutions can be adapted to deal with mobile devices. Nevertheless, mobile devices present their own unique challenges such as a fragmented operating system market (such as Apple Os, Android, Windows mobile, RIM etc.), a proliferation of manufactures building devices on different standards, as well as the more limited processing and data storage capabilities of mobile devices. Security solutions have to be programmed with these limitations in mind.

This migration of computing from desktop devices to smart phones and tablets has lead to the appearance of those threats that initially only affected desktop devices. The threat that this paper addresses is the migration of spamming Botnets onto mobile devices. Botnets are now capable of infecting mobile devices and using them to send mobile spam.

The paper is structured as follows: the background section introduces the topics of spam, Botnets and Neural Networks. The following section introduces a model on combating mobile spam through Botnet detection using Neural Networks. This is followed by a description of the prototype, the actual implementation of the

Vural I. and Venter H. (2012).
Spamming Botnet Detection using Neural Networks.
In *Proceedings of the 9th International Workshop on Security in Information Systems*, pages 27-38
DOI: 10.5220/0004089800270038
Copyright © SciTePress

prototype, and a section where experimental results are tabulated. The paper is then concluded.

## 2 Background

### 2.1 Spam

This section gives an overview of spam, mobile spam, spamming Botnets, Botnets on mobile devices and Neural networks. The definition of spam is discussed followed by a discussion on the different types of spam, a section on mobile spam followed by a section on the economics of spam.

#### 2.1.1 The Definition of Spam

Unsolicited bulk mail, otherwise known as spam, is an email (electronic message) sent to a large number of email addresses, where the owners of those addresses have not asked for or consented to receive the email [1]. Spam is used to advertise a service or a product. One of the most well-known examples of spam is an unsolicited email message from an unknown or forged address advertising Viagra [2]. The lack of a universally recognized definition for spam is one of the major obstacles to creating solutions designed to minimize its harmful effects. The definition of spam could range from any unsolicited emails [3], to unsolicited commercial emails sent by any source [4], to unsolicited commercial emails from sources the recipient has never had contact with [5], to simply emails or postings transmitted in bulk quantities [6].

#### 2.1.2 Types of Spam

Figure 1 shows the different types of spam that are commonly encountered today. Email spam is the most common form of spam and the one that most people are most familiar with. Comment spam is of the kind that inflicts the comments section of newspaper websites, where adverts are inserted in the comments section. Messaging spam, also known as spam over instant messaging (SPIM) is of the kind of spam that one would receive over an instant messenger application such as Google Talk [7]. Mobile spam, which is discussed in more detail later in this paper, is spam received on one's mobile device in the form of SMSs. Voice over internet protocol (VOIP) spam is the kind of spam that one receives through automated voice messages over a VOIP phone [8].

Mobile spam is the focus of this paper and therefore the next section is devoted to discuss mobile spam in more detail.

#### 2.1.3 Mobile Spam

Spam is not limited to e-mail as is usually thought to be the case. Spam also exists in text messaging services (SMS). In the case of an SMS, spam can cost even more that it would when received through email. For example, assume that a user has subscribed to receive a notification via SMS when he/she receives an email.

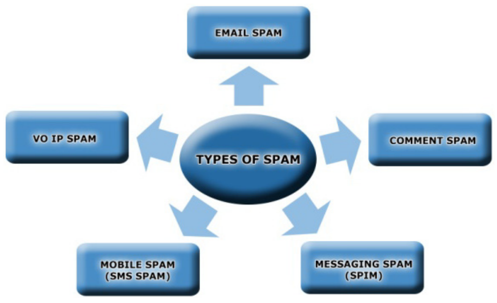


**Fig. 1.** Types of Spam.

Depending on the particular cellular network, the user might have to pay for every SMS received regardless if it is a spam or a valid email.

Until recently mobile networks have been relatively isolated from the Internet. Mobile networks are now well integrated with the Internet and with the proliferation of smart phones running on operating systems such as Android and Windows Mobile, threats to the Internet have started to migrate to mobile networks.

Conveniences that people access today on a desktop computer are available on mobile computing devices. Greater adaptation of these devices will encourage more users to access personal and financial data on their phones. This means that the threats to the internet such as spyware, phishing and spam are migrating to mobile devices.

Desktop operating systems are getting harder to exploit, but mobile devices have code bases that are largely unexplored, and updates to new versions with security flaws occurring frequently [9].

The increasing processing power of mobile phones, and growing features and applications included with them, make mobile phones an ideal candidate for exploitation by malware.

Malware writers are attacking commercial programs for mobile devices. A malicious program such as J2ME/RedBrowser [10], which is a Trojan horse program, pretends to access WAP web pages via SMS messages. In reality, instead of retrieving WAP pages, it sends SMS messages to premium rate numbers, thus costing the user more than intended. The next section discusses the motivation behind the sending of spam.

### 2.1.4 Economics of Spam

Spam makes money for those who send it, as its cost versus benefit ratio for email spam is so low. If even a small percentage of spam recipients respond to the advertised product, spammers will still make money.

Spammers generally pay nothing or very little for the sending of email spam. They accomplish this by exploiting open mail servers to send spam. The spammer need only send one email message using an open or exploited email server in a bid to reach thousands of email addresses, with the bulk of the transfer being handled by the open or exploited email server. Recipients, in turn, need to pay access costs or telephone costs in order to receive content they didn't ask for.

ISPs have to bear the bulk of the cost for bandwidth overuse by spammers; this cost is often passed onto the consumer through increased Internet access fees or a degraded service level.

With the introduction of the "Electronic Communications and Transactions Act, 2002" unsolicited emails now have a legal definition and the sending of spam is illegal in South Africa [11]. Similar legislation exists for many other countries [12], [13]. Spammers, if identified, are liable for a fine and prosecution. Thus, spammers attempt to cover their trail to prevent identification.

The economics of SMS spam differ from email spam as network service providers have to be paid to deliver messages. Also, unlike email spam, the filters in place to filter SMS spam are not as prevalent. Thus, there is a huge incentive for SMS spammers to send SMSs from other people's devices and, thus, not pay for the SMSs being sent. Sending one million SMSs, for example, is exponentially more costly than sending one million emails. Thus, there is an incentive for SMS spammers to hijack mobile devices in a bid to send SMS spam. The following section expands on this topic by discussing Botnets.

## 2.2 Botnets

This section gives an overview of Botnets, the first part gives a definition of Botnets, and this is then followed by a description of mobile Botnets

### 2.2.1 Definition of a Botnet

A Botnet is a network that consists of a set of machines that have been taken over by a spammer using Bot software Bot software (or Bots for short) is a kind of malware that is often distributed in the form of a Trojan horse [14]. A Bot hides itself on its host machine and periodically checks for instructions from its human Botnet administrator. Botnets today are often controlled using Internet Relay Chat [15]. The owner of the computer usually has no idea that his machine has been compromised until the user's Internet connection is shut down by an ISP. Most ISPs block bulk email if they suspect it is spam. The spammers who control these Botnets typically send low volumes of mail at any one time so as not to arouse suspicions. Thus, the spam email can often be traced to an innocent individuals network address and not the spammer's actual network address. Botnets are a prized commodity on the internet and hackers are often willing to rent their hard-earned bots for money.

While the number of Botnets appears to be increasing, the number of bots in each Botnet is actually dropping. In the past Botnets with over 80 000 infected machines were common [15]. Currently Botnets with a few hundred to a few thousands infected machines are common. One reason for this decline in Bot numbers per Botnet is that smaller Botnets are more difficult to detect. Someone is more likely to notice a big Botnet and take steps to dismantle it [16]. It has also been suggested that the wider availability of broadband access makes smaller Botnets as capable as the larger Botnets of old [15].

When Procter & Gamble ran a security check of its 80,000 PCs, it found 3,000 were infected with Bots [17]. The following section elaborates on the spread of Botnets to mobile devices.

### 2.2.2 Botnets on Mobile Networks

Mobile devices are capable of accessing the internet through technologies such as High Speed Downlink Packet Access (HSDPA) and General Packet Radio Service (GPRS) [18]. The connection between the internet and mobile devices acts as a gateway for malware to move from the internet to mobile networks. More and more financial transactions will take place over mobile devices; this puts valuable information at risk.

The challenge for businesses and banks in the near future will be to produce secure mobile applications while ensuring ease of use at the same time [19]. The motivation for installing a Botnet that sends spam on a mobile device and installing one on a desktop differ. In the case of Botnets on desktops the motivation is to prevent the spammer's identity being revealed, in the case of mobile Botnets that is not the only motive. As was discussed in section 2.1.4 the cost of sending SMS messages is exponentially higher than the cost of sending email messages. Thus another motivation for mobile Botnets is to pass on the cost of sending the message to someone else. An implementation that would enable users to identify Botnets on their mobile devices would slow the emergence of SMS spam. The following section discusses anomaly detection; a technique that has been used in many security applications such as intrusion detection and that the authors believe can also be used to combat SMS spam.

## 2.3 Artificial Neural Networks

Artificial neural networks are computational methodologies that perform multifactorial analyses. Inspired by networks of biological neurons, artificial neural network models contain layers of simple computing nodes that operate as nonlinear summing devices [20]. These nodes are richly interconnected by weighted connection lines, and the weights are adjusted when data are presented to the network during a "training" process. Successful training can result in artificial neural networks that perform tasks such as predicting an output value, classifying an object, approximating a function, recognizing a pattern in multifactorial data, and completing a known pattern.

### 2.3.1 Learning

Learning is a process in which different events are associated with different outcomes, i.e. substantiating the cause and effect principle. This section will examine the different types of learning mechanisms.

### 2.3.2 Learning through Association

Learning through association is simply learning through the cause and effect principle. One example of learning by association is Boolean algebra [21]. Boolean algebra is the logical association between truth and falsehood. The logic of truth and false can be represented by AND, OR and NOT operators. Almost all arithmetic expressions can be represented by Boolean algebra and modern computers use Boolean logic for their operation. For example consider the statement a person must be at least 18 years old to drive a car, from this we can rationally deduce that someone driving a car is at least 18 years old.

Another example of association is decision tree algorithms [22]. Every node on the tree is evaluated and a decision is made as to which child node to proceed to until a solution to the problem is found. This method will fail when the parameters on the node being evaluated do not produce a sharp distinction so as to allow the algorithm to know which child node to evaluate. In cases where it is some correlation between the parameters themselves that means it is better to not just evaluate them but their combinations as well. A Neural network is designed to handle such situations. In general decision making is an attempt to find the best association between known features and known outcomes, by assigning a certain weight to each association we can select the association that is most likely.

### 2.3.3 Feature Identification

Identification of useful features that will be used to build associations is a pertinent issue in machine learning. A good feature is one that produces proximity to some unique region in the feature space, leading to the formation of a cluster of similar objects in the region. The greater the separation between the clusters of objects in the feature space the better the parameter. This separation is usually measured as a distance measure. A commonly used distance measure is the Euclidean distance formula. . The Euclidian distance formula, in mathematical terms, is the ordinary distance between two points that one would measure with a ruler [23].

### 2.3.4 Artificial Neural Network Learning

An artificial neural network models a system based on the information fed into it. If we build a good model it should be possible for the network to predict the correct output from the inputs. Real systems can be very complex and thus difficult to duplicate, neural networks can if given enough data correctly model the system that produced the data. Neural networks can be one of two types those that use supervised learning and those that use unsupervised learning [24]. Supervised learning networks learn from training data that contains many examples of possible inputs and their corresponding outputs from a real system. Thus the network attempts to mimic the training data. For unsupervised learning the training data consists of a collection of patterns with no distinction between inputs and outputs from this data the network attempts to group the patterns into different clusters. Thus the network makes a self evaluation of the possible sources of the variants in the data.

## 3 A Model for a Botnet Detector using Neural Networks

In this section we introduce the model for our spamming Botnet detector and explain the reasons for it being modelled in this manner as well as its advantages and disadvantages. We discuss the data that will be used to train our algorithm, the population and selection of the data. We then round this section off by explaining how the prototype would be used in a real-world situation to combat Botnet spam.

## 3.1 Learning with Negative and Positive Data

The detection of SMS spam is a typical classification problem where patterns, also known as signatures, need to be classified as legitimate or not. Thus, it can be regarded as a binary classification problem in which the data set, consisting of SMS messages, are classified as either spam or non spam. Neural Networks are ideal for this sort of classification problem as we can train the spam filter on legitimate (positive) and non-legitimate (negative - spam) messages. From this data, the neural network should be able to deduce whether messages are spam or not. The Neural Network Botnet Detector (NNBD) was implemented using the .Net framework.

## 3.2 Data Classification

As discussed in section 2.3.4 supervised learning networks learn from training data that contains many examples of possible inputs and their corresponding outputs from a real system. It is thus possible to train a neural network on a data set consisting of only spam SMSs. The neural network trained on this data should be able to accurately identify SMSs as spam or ham. To increase the accuracy of the neural network it can be trained on an additional data set. This additional data set consisting of ham messages can be used to improve the accuracy of the implementation and reduce the number of false positives (valid SMSs incorrectly classified as spam). Section 4.4 further describes the data selection process.

## 3.3 Remote Analyses Vs. Analysis on the Device

There are two ways in which one could analyse a mobile device user's SMS data. The first possibility would be to analyse the SMSs sent on the ISP's servers and use this to build a profile of the user's SMS sending behaviour. There are several drawbacks to this approach. Firstly, there are privacy implications of analysing SMS data on an ISPs server, but even if these concerns were addressed, the classification algorithm would have to determine whether or not a message is valid or invalid without the user's input. Thus, it would not be able to learn based on user feedback. The second solution is to implement the detection algorithm on the device, thus, taking care of the privacy implications as well as allowing for user feedback in the learning process. The major disadvantage of this approach is that the prototype needs to be installed on a mobile device which has limited processing power and storage space especially when compared to an ISP's server. Thus, the prototype had to be programmed to use minimum storage and be optimised to make use of as little memory as possible.

## 3.4 Flow Diagram for Botnet Detector

Figure 2 visualizes how the Botnet detector is designed to work. The mobile user enters a text message and sends it to a recipient. This message is intercepted and certain message characteristics such as the number of capital letters (the full list of characteristics is defined in section 4.1) are also extracted for analysis by the Botnet

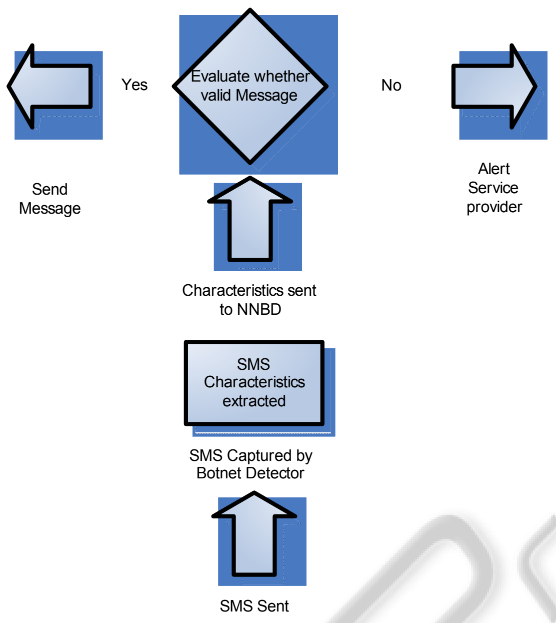

Yes
Evaluate whether valid Message
No
Send Message
Alert Service provider
Characteristics sent to NNBD
SMS Characteristics extracted
SMS Captured by Botnet Detector
SMS Sent

**Fig. 2.** Model of the Botnet detector.

Detector before the message is sent (the neural network should not send out messages identified as spam messages). The characteristics are sent to the neural network which then determines whether the message is valid or not by inputting the SMS message into the neural network (this is explained further in the next section). If the NNBD can determine that the message is valid, the message is sent onwards. Else the network provider will be alerted by the NNBD so that the service provider can investigate the malware that is sending these spam messages and remove it from the mobile device

The premise behind the NNBD implementation is that the authors believe that a spamming Botnet that has installed itself on a user's mobile device, and is sending out spam SMSs without the knowledge of the mobile devices user, can be detected by the NNBD. Thus, if these spam messages are blocked, the NNBD would have succeeded in preventing the sending of spam as well as saving the mobile device owner the cost of the spam SMSs being sent. Additionally, the NNBD would alert the mobile device user to the presence of Botnets on their device so that the malware, that has installed itself on their device, can be removed.

## 4 Implementation of Mobile Botnet Detector – A Prototype

This section describes how the prototype was implemented. The authors first discuss the message characteristics chosen to extract and train the Neural Network module with. Next the section describes the process of training the neural network. This is followed by a section documenting the testing of the neural networks accuracy as well as a section describing how the data was selected for this implementation.

## 4.1 Message Signatures

The prototype creates a signature (pattern) for each message sent by the mobile device. The signature consists of the following characteristics that are analysed by the neural network to determine the validity of the message:

- Does the SMS message contain links?
- Does the SMS message contain telephone numbers?
- The number of words in a message
- The ratio of punctuation characters to words
- The ratio of links to words in the message
- The ratio of capital letters to words in the message
- The ratio of misspelt words in the SMS message
- Bayesian spam probability

The specific characteristics mentioned above were chosen by the authors to define the message signature as they allow the implementation to build a profile of the user's sending behaviour. The characteristics chosen are simple to capture, yet indicative of sending behaviour. Use of punctuation, capital letters and their ratios to word in a message may reveal much about an SMS message.

The majority of spam emails contain a link to a URL, thus it makes sense to mark the presence of URLs in the SMS message (these links might lead to a website which sells a product that the spammer is attempting to advertise). The presence of telephone numbers is also be a useful bit of information to mark as the spammer might include a telephone number for the recipient of the spam message to call in order to enquire about the product or service advertised, quite possibly this call may be charged at a premium rate. The ratio of misspelt words is also a useful characteristic to monitor as spammers will often hide words. They will often send an SMS with phrases like "v1agr@a" instead of Viagra in order to thwart a spam filter. The Bayesian spam probability, which calculates the probability of a message being spam, will become the dominant feature if the other characteristics fail to identify the message as being spam. To calculate The Bayesian spam probability examines all string tokens within a message and calculates whether the token appears often in spam message. This Bayesian probability is used to calculate the spam probability of the message. Additional characteristics may be added in future to the prototype to increase the accuracy of the implementation. This would enable us to build a better profile of the users messages.

## 4.2 Training the Neural Network

The training of our neural network is a three stage approach. First, we add the messages we are going to train the neural network on, to a spam list and a ham list (a ham list is a list of valid SMS messages). Once we have done this, we can then generate the message signatures for each message, which we save to a list. Finally, we pass the message signatures from our saved list into the training procedure of our neural network, which, in turn, causes the network to learn what message signatures represent spam.

### 4.3 Testing the Neural Network

To test the neural network we run the network against a collection of spam and ham messages extracted from one of the authors' mobile phones. The neural network looks at the message signatures as identified previously in section 4.1 and generate a numerical statistic for each signature between 0 and 1. The NNBD then takes the collection of statistics generated, and feed them as input vectors into our neural network. The neural network will have one output, which will be a probability between {0, 1} on whether the SMS is spam or ham..

### 4.4 Data Selection

The data that was used to train the NNBD implementation was selected by using SMS messages collected by the SMS Spam Collection project [25]. In total 4815 of these valid SMSs were used to create a white list of legitimate SMS messages. The second set, i.e. the black list of invalid SMSs, totalled 746 (including blacklisted URLs). In the following section we tabulate the experimental results.

## 5 Tabulation of Results

The results of the experiment are tabulated as follows and show the accuracy in detecting spam SMSs in Table 1. For this paper the authors compared the results from the NNBD against a previous prototype they had built that used a Bayesian filter to detect spamming Botnets.

**Table 1.** Results.

| | Valid Message | Invalid Message | Total error |
|---|---|---|---|
| Neural Network | 100% | 95% | 2.5% |
| Bayesian Filter | 100% | 90% | 5% |

As can be seen in Table 1, the NNBD has an accuracy of 100% in identifying valid messages, and an accuracy of 95% in identifying spam messages giving us a total error of 2.5%. The total error is calculated by determining how many of the spam and ham messages (40 in total, 50% spam and 50% ham) were incorrectly identified. This compares well to the prototype that uses a Bayesian probability to detect spam as shown in the second line of Table 1.

## 6 Discussion

Spam cannot be eliminated solely with technological solutions. To reduce spam, people must ideally stop responding to spam messages by actually purchasing items advertised in spam messages. Never the less, technical solutions play a strong role in combating spam. By allowing less spam to get through, we can reduce the incentive for spammers to spam and increase the cost of sending spam for spammers, thereby reducing spam.

The authors believe that this neural network, when implemented, could reduce spam significantly. This NNBD could be implemented either on a mobile device or on a network provider's server. The advantage of installing it on a mobile device is that the user can be prompted to confirm whether a particular message is indeed spam or not if the neural network is unsure. The disadvantage of installing the NNBD on a user's mobile device is the degradation of performance that could be experienced when the NNBD calculates its spam probability. The advantage of situating the NNBD on a network service provider's server is that processing can be done by dedicated servers. The disadvantage, of course, is that the user cannot be prompted for feedback and confirmation, which might result in certain messages being incorrectly classified.

## 7 Conclusions and Further Work

Spamming Botnets have the potential to become more common with the increasing processing power of mobile phones make mobile phones more attractive for exploitation by malware. The aim of this research is to provide a tool that is not only capable of identifying spam SMSs being sent from a user's mobile device, but also to allow the spam filter to learn new spam features as spammers continuously change the spam features to confuse spam filters. The NNBD is has been shown to be capable of correctly identifying 95% of all Spam messages with a zero false positive rate. This shows that the NNBD can be used to detect spamming mobile Botnets that have infected mobile devices.

The authors believe that over time, with more spam messages added to the black list, the accuracy of this implementation would improve. The authors hope to apply these ideas out on an Instant Messaging platform as well as on SMS messaging.

## References

1. Internet Service Providers' Association, 2008. 'What is Spam?' Available: http://www.ispa.org.za/spam/whatisspam.shtml. [April 2009]
2. Spam-site "Samples of Spam" http://www.spam-site.com/spam-sample.shtml [September 2011]
3. B. G. Kutais, "Spam and Internet Privacy", 'Journal of High Technology Law Suffolk University Law School'
4. Consumer fraud reporting, "Spam Emails and Spamming", http://www.consumerfraudreporting.org/spam.php [September 2011]
5. Federal Communication Commissio, "Spam: Unwanted Text Messages and Email", http://www.fcc.gov/guides/spam-unwanted-text-messages-and-email [September 2011]
6. Spamhaus "The Definition of Spam", http://www.spamhaus.org/definition.html [September 2011]
7. Earth Web. "Think Spam is tough? Try Fighting Spim" http://itmanagement.earthweb.com/secu/article.php/3365931 [September 2011]

8. R. Dantu and P. Kolan. Detecting Spam in VoIP Networks. In Proceedings of USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI), July 2005.
9. Security Vision from McFee Avert Labs, 2007 The Future of Security McFee, 2010 Available: http://vil.mcafeesecurity.com/vil/content/v_138726.htm
10. McFee, 2010 Available: http://vil.mcafeesecurity.com/vil/content/v_138726.htm
11. Acts Online, 2002. Electronic Communications and Transactions Act ,2002. Available: http://www.acts.co.za/ect_act/. [April 2009]
12. Australian Government Department of Broad Band Communications and the Digital Economy. “Spam”. http://www.dbcde.gov.au/online_safety_and_security/spam [September 2011]
13. Industry Canada. “Government of Canada Introduces Anti-Spam Legislation” http://www.ic.gc.ca/eic/site/ecic-ceac.nsf/eng/gv00521.html#Q1 [September 2011]
14. Seach Security.com “botnet (zombie army)”. http://searchsecurity.techtarget.com/definition/botnet [September 2011]
15. E. Cooke, F. Jahanian, and D. McPherson. The zombie roundup: Understanding, detecting, and disrupting botnets. In USENIX SRUTI Workshop, pages 39–44,2005.
16. Ryan Vogt, John Aycock, and Michael J. Jacobson, Jr. “Army of Botnets”, Proceedings of the 2007 Network and Distributed System Security Symposium, pp. 111–123, 2007
17. The Economist “Big brother bosses” September 11 2009 Available: http://www.economist.com/businessfinance/displaystory.cfm?story_id=14413380 [September 2009].
18. Sumit Kasera and Nishit Narang. , 2005, 3G Mobile Networks. Architecture, Protocols and Procedure. Tata MCGraw-Hill Publishing Company, limited edition.
19. Emerging Cyber Threats Report for 2009, Georgia Tech Information Security Center, October 15, 2008
20. Judith E. Dayhoff Ph.D,, James M. DeLeo Supplement: Conference on Prognostic Factors and Staging in Cancer Management: Contributions of Artificial Neural Networks and Other Statistical Methods, Volume 91, Issue Supplement 8, pages 1615–1635, 15 April 2001
21. NIGEL P. COOK, Introductory Digital Electronics, Prentice Hall (1997)
22. Safavian, SR., and D.Langrebe, A survey of Decision Tree Classifier Methodology, IEEE Transactions on Systems, Man and Cybernetics (1991)
23. Elena Deza & Michel Marie Deza (2009) Encyclopedia of Distances, page 94, Springer.
24. Anthony Zaknich, Artificial Neural Networks :An Introductionary Course available 'http://www.maths.uwaedu.au/~rkealley/ann_all/index.html
25. José María Gómez Hidalgo , Guillermo Cajigas Bringas , Enrique Puertas Sánz , Francisco Carrero García, Content based SMS spam filtering, Proceedings of the 2006 ACM symposium on Document engineering, October 10-13, 2006, Amsterdam, The Netherlands [doi>10.1145/1166160.1166191]

*Journal of Universal Computer Science, vol. 18, no. 6 (2012), 750-774*
*submitted: 15/10/11, accepted: 15/2/12, appeared: 28/3/12 © J.UCS*

# Combating Mobile Spam through Botnet Detection using Artificial Immune Systems

**Ickin Vural**
(Department of Computer Science
University of Pretoria, Pretoria, Republic of South Africa
Ickin@tuks.co.za)

**Hein S. Venter**
(Department of Computer Science
University of Pretoria, Pretoria, Republic of South Africa
hventer@cs.up.ac.za)

**Abstract:** Malicious software (malware) infects large numbers of mobile devices. Once infected these mobile devices may be involved in many kinds of online criminal activity, including identity theft, unsolicited commercial SMS messages, scams and massive coordinated attacks. Until recently, mobile networks have been relatively isolated from the Internet, so there has been little need to protect them against Botnets. Mobile networks are now well integrated with the internet, so threats on the internet, such as Botnets, have started to migrate to mobile networks. This paper studies the potential threat of Botnets based on mobile networks, and proposes the use of computational intelligence techniques to detect Botnets. We then simulate mobile Bot detection by detecting anomalies using an artificial immune system implementation on an Android device.

**Keywords:** Botnet, mobile, malware, computational intelligence, artificial immune system
**Categories:** J.0

## 1 Introduction

Digital security can in many ways be likened to an arms race; the belligerents who are security experts defending systems from attack and criminals and adventurers targeting digital systems are locked in a battle in which the development of security countermeasures is matched by new more sophisticated attacks. The field of digital security is, thus, a fast evolving one that finds its ecosystem in a constant state of evolution. Threats of yesteryear are replaced with new and unforeseen threats, and every now and then an old threat reappears to wreak havoc. Every successful attack is copied by others, and evolved to combat countermeasures put in place by defenders of digital systems. This paper explores a new challenge to digital systems posed by the adaptation of mobile devices and proposes a countermeasure to secure systems against threats to this new digital ecosystem.

The field of computer security, which for many years focused on paradigms such as network security, information security and workstation security, is facing a paradigm shift with the ever-increasing popularity of mobile devices such as smart phones and tablet devices. As many of the current threats to mobile devices (also known as cell phones or mobile phones) are similar to those that threaten desktop machines connected to the internet, many of the same solutions can be adapted to deal with mobile devices. Nevertheless, mobile devices present their own unique challenges such as a fragmented operating system market (such as Apple Os, Android, Windows mobile, BlackBerry OS etc.), a proliferation of manufactures building devices on different standards, as well as the more limited processing and data storage capabilities of mobile devices - security solutions have to be programmed with these limitations in mind. Several years ago, the word "spam" was used colloquially to represent unwanted junk email sent from desktop machines. The migration of computing from desktop devices to smart phones and tablets has lead to the appearance of threats such as spam that initially only affected computers. Spam is also no longer just limited to email. Various types of spam are encountered today, for example, Sms spam and instant messaging spam (SPIM). Various jurisdictions have implemented legislation to control spam. One particular example is the introduction of the Electronic Communications and Transactions Act, 2002 *(Acts Online, 2002)*. Unsolicited mail now has a legal definition and the sending of spam is prohibited. Spammers, if identified, are liable for a fine and/or prosecution. Thus, spammers will attempt to cover their trail to prevent identification.

There are a number of identity concealment techniques used by spammers. The most common is the use of Botnets, as the use of Botnets allows the spammer to send spam from devices that cannot be linked to him. The owner of the device usually has no idea that their machine has been compromised until their Internet connection is shut down by an Internet service provider (ISP). As most ISPs block bulk mail if they suspect it might be spam, the spammers who control these Botnets typically send low volumes of mail from numerous infected computers. It is done in this fashion so as not to arouse suspicions of bulk mail originating from a single computer.

The threat that this paper addresses is the migration of spamming Botnets onto mobile devices. Botnets are now capable of infecting mobile devices and using them to send mobile spam. Thus, a significant cause of spam email sent by Botnets could be a significant generator of spam SMSs in the very near future. Network forensics is the capture, recording, and analysis of network events in order to discover the source of problem incidents *(Garfinkel, S)*. The use of network forensics to detect Botnets has two primary drawbacks; firstly the analysis, which leads to the detection of Botnets, only happens after the spam message has already been sent. This is not ideal as people do not like receiving spam, and every spam Sms sent costs the device owner money. The second drawback is that the capture of packet information including user data raises privacy issues and in terms of the Electronic Communications Privacy Act (ECPA) ISPs are forbidden from eavesdropping or disclosing intercepted contents except with user permission, for limited operations monitoring, or under a court order *(Garfinkel, S)*. Their two drawbacks are the main focus of this paper. Thus, in order to address these problems, an approach for detecting spamming Botnets on mobile devices is proposed in order to detect Botnets with software residing on the mobile device and being capable of intercepting spam SMSs before they are sent.

The aim of this research is to introduce a novel way of combating a not much researched threat to mobile devices, namely Sms spam. The technique to be used is in the form of a software tool employing an artificial immune system that is installed on a mobile device and that detects spam SMSs being sent by malware or Botnets installed on the device. The application must be capable of classifying Sms messages as spam and non-spam. The application should only stop those messages classified as being spam and allow those that are not classified as spam to be sent through. When starting this research we started of by asking what spam is and what the prevalence of Spam is in the mobile ecosystem. Questions such as the economic impact of spam were discussed; we studied how spam was sent. How spam was traced, anti spam techniques, spammer identification techniques and an in-depth study of Botnets. Once we identified mobile spam as the target of our research we needed to identify how mobile spam is sent, this required an understanding of how mobile Botnets functioned. We devised an application to monitor anomaly in Sms sending behaviour at a device level. This required choosing a computational intelligence technique to identify infected devices sending Spam. The implementation was programmed using an android emulator using the Java programming language, thus, knowledge of androids SDK had to be acquired.

More specifically, the paper is structured as follows: the background section introduces the topics of spam, Botnets, immune systems, anomaly detection and artificial immune systems. The following section introduces a model on combating mobile spam through Botnet detection using artificial immune systems. This is followed by a description of an artificial immune system prototype, the actual implementation of the prototype, a section where experimental results are tabulated and, finally, by a conclusion.

## 2 Background

### 2.1 Spam

Unsolicited bulk email, otherwise known as spam, is an email sent to a large number of email addresses, where the owners of those addresses have not asked for or consented to receive the email *(Internet Service Providers)*. Spam is commonly used to advertise a service or a product. One of the most well-known examples of spam is an unsolicited email message from an unknown or forged address advertising Viagra *(Samples of Spam)*.

Figure 1 shows the different types of spam that are commonly encountered today. Email spam is the most common form of spam and the one that most people are most familiar with. Comment spam is of the kind that inflicts the comments section of newspaper websites, where adverts are inserted in the comments section. Messaging spam, also known as spam over instant messaging (SPIM) is of the kind of spam that one would receive over an instant messenger application such as Google Talk *(Earth Web)*. Mobile spam, which is discussed in more detail later in this paper, is spam received on one's mobile device in the form of SMSs. Voice over internet protocol (VOIP) spam is the kind of spam that one receives through automated voice messages over a VOIP phone *(R. Dantu and P. Kolan)*.

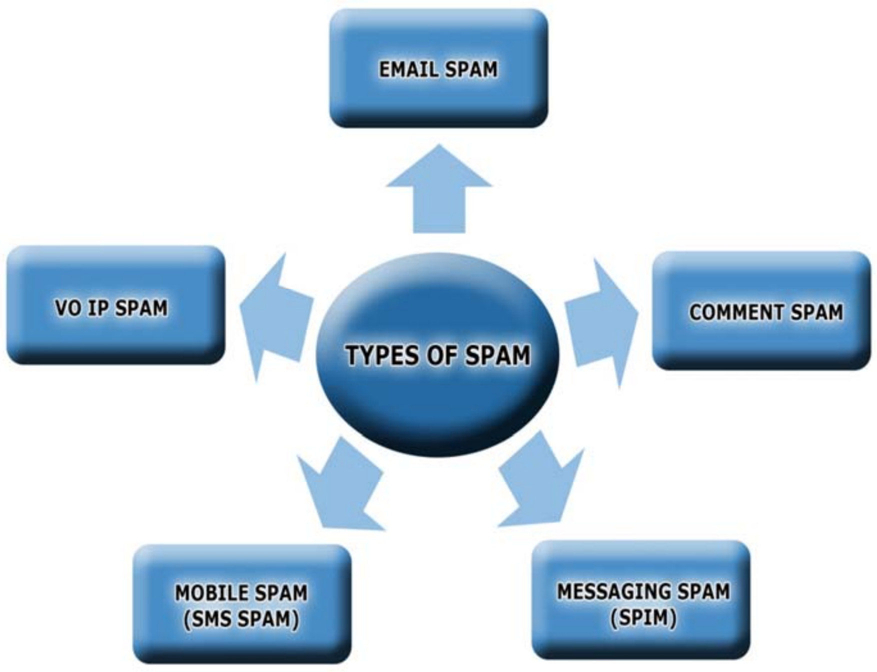


*Figure 1: Types of Spam*

## 2.2 Spamming Botnets

A Botnet is a network that consists of a set of machines that have been taken over by a spammer using Bot software Bot software (or Bots for short) is a kind of malware that is often distributed in the form of a Trojan horse *(Security.com)*. A Bot hides itself on its host machine and periodically checks for instructions from its human Botnet administrator. Botnets today are often controlled using Internet Relay Chat *(E. Cooke, F. Jahanian, and D. McPherson)*. The owner of the computer usually has no idea that his machine has been compromised until the user's Internet connection is shut down by an ISP. Most ISPs block bulk email if they suspect it is spam. The spammers who control these Botnets typically send low volumes of mail at any one time so as not to arouse suspicions. Thus, the spam email can often be traced to an innocent individual’s network address and not the spammer's actual network address. Botnets are a prized commodity on the internet and hackers are often willing to rent their hard-earned bots for money.

While the number of Botnets appears to be increasing, the number of bots in each Botnet is actually dropping. In the past Botnets with over 80 000 infected machines were common *(E. Cooke, F. Jahanian, and D. McPherson)*. Currently Botnets with a few hundred to a few thousands infected machines are common. One reason for this decline in Bot numbers per Botnet is that smaller Botnets are more difficult to detect. Someone is more likely to notice a big Botnet and take steps to dismantle it *(Ryan Vogt, John Aycock, and Michael J. Jacobson)*. It has also been suggested that the wider availability of broadband access makes smaller Botnets as capable as the larger

Botnets of old *(E. Cooke, F. Jahanian, and D. McPherson)*. When Procter & Gamble ran a security check of its 80,000 PCs, it found 3,000 were infected with Bots *(The Economist)*.

Mobile devices are capable of accessing the internet through technologies such as High Speed Downlink Packet Access (HSDPA) and General Packet Radio Service (GPRS) *(Sumit Kasera and Nishit Narang)*. The connection between the internet and mobile devices acts as a gateway for malware to move from the internet to mobile networks. More and more financial transactions will take place over mobile devices; this puts valuable information at risk.

The challenge for businesses and banks in the near future will be to produce secure mobile applications while ensuring ease of use at the same time *(Georgia Tech Information Security Center)*. An implementation that would enable users to identify Botnets on their mobile devices would slow the emergence of SMS spam. The following section discusses anomaly detection; a technique that has been used in many security applications such as intrusion detection and that the authors believe can also be used to combat SMS spam.

## 3 Anomaly detection

This section discusses anomaly detection and how anomaly detection techniques can be applied to SMS traffic to detect Botnets that send spam (also referred to as spamming Botnets). It should be stressed that the purpose of this research is to identify mobile devices that may have been infected with Botnet software. The purpose is, thus, not to identify the identity of the human Botnet controllers, otherwise known as spammers.

Using statistical analysis techniques, it is possible to build behaviour models of individual mobile phone users as this is similar to building a behavioural model of people's email sending behaviour *(Michael Negnevitsky, Mark Jyn-Huey Lim, Jacky Hartnett, Leon Reznik)*. These behavioural models are used to establish the normal or expected behaviour of the mobile users. Users’ behaviour on the mobile phone is then monitored and compared with current or recent usage data to detect abnormal changes in communication behaviour. This, of course, raises some privacy issues, but as discussed later in the paper, the prototype implemented here stores this information on the mobile device and the data is not stored as it is represented in the message body, but in an encoded format. This should allay any privacy concerns a user might have about this technique.

The shortcoming of using an abnormal behavioural model is deciding on what constitutes abnormal behaviour. One way of detecting abnormal behaviour is by using mathematical techniques. The problem with using mathematical techniques, however, is that they impose strict boundaries around the profiles of what is considered normal and abnormal behaviour *(Michael Negnevitsky, Mark Jyn-Huey Lim, Jacky Hartnett, Leon Reznik)*. Thus, in a bid to find a better solution, the authors have implemented a solution using an artificial immune system. Artificial immune systems can learn continuously and are, thus, adaptive as explained in more detail later in the paper. This will allow the system to adapt to changes in “what is normal or not” over time.

The following section introduces artificial immune systems and how they can be modelled to detect spam. The authors apply artificial immune systems as the basis to their research.

## 4 Modelling the Artificial Immune System

This section explains the components of the "spam immune system" and how they work together to learn and unlearn message signatures in order to classify them as spam or non spam. The artificial immune system (AIS) is modelled on the natural immune system (NIS). The goal of a NIS is to differentiate between self cells and potentially dangerous non-self cells. Self cells are cells naturally belonging to the body and non-self cells are foreign cells such as viruses and bacteria which might threaten the body. In an anti-spam system one needs to differentiate between self (legitimate) messages and non-self (spam) messages. Theories of how the NIS works, can serve as a starting point for creating computer systems as many of the techniques used in artificial intelligence are inspired by the principles and processes found in nature.

It is the classification of self and non-self cells that makes the NIS an appealing model for spam detection, which also requires a classification between the legitimate (self) messages and spam (non-self) messages.

The aim of this artificial immune system implementation is to alert the user or network provider if the mobile device is being used to send SMS spam. The first section describes how the SMS signature is modelled as an antigen. This is followed by a section explaining how the artificial immune System distinguishes between Spam and non Spam SMS messages. Before we continue, the authors thought it would be useful to include Table 1 to explain some of the terms used in this paper.

| **Term** | **Explanation** |
|---|---|
| Self | Refers to a valid message (i.e. not spam). |
| Non self | Refers to an invalid message (i.e. spam). |
| Antigen | Is a message signature that is not recognised as being self and is thus possibly spam. |
| Pathogen | Is a message positively identified as being spam. |
| Signature | The digital representation of an SMS message. |
| Signature library | Storage module in which signatures are stored. |

*Table 1: Glossary of terms*

### 4.1 Signature: SMS Message and Signature library

In the context of this domain, an antigen refers to the target or solution that is potentially bad, e.g. whether the message we need to check is spam *(Edmund K. Burke , Graham Kendall ).* Antigens have to be encoded in some way to be represented in digital form; this could be in binary or real number format. For the implementation of the artificial immune system in this paper, the SMS message was converted to a digital representation (signature) for the system to process. This was

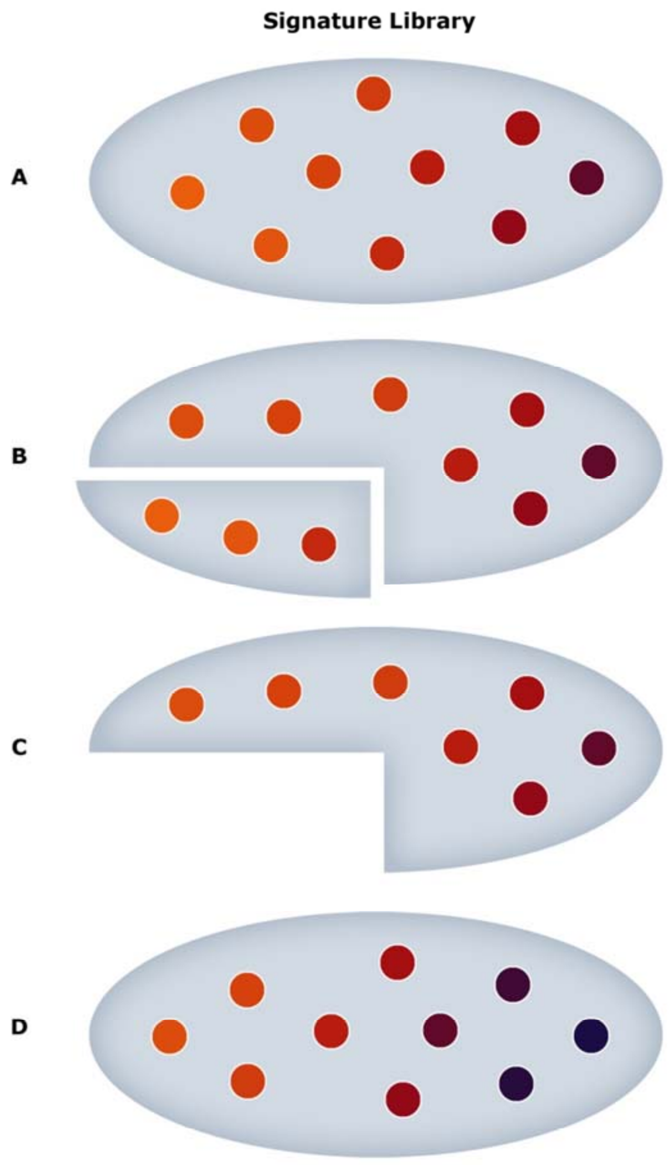


*Figure 2: Signature Library*

done by reading in the message and identifying characteristics such as the number of capital letters used, the length of the message, use of punctuation and presence of URLs. It is by storing and analysing these messages that a profile of the user's SMS sending behaviour can be built. The selection of the characteristics to be stored is critical in building a valid profile of the user. The prototype, discussed later in the paper, also has a signature storage module in which randomly-generated signatures are stored; this is called the signature library. The library is initially populated with randomly generated signatures as a starting point. This practice is common with many artificial intelligence techniques. The library changes with each successive generation by a process of selection until we have a set of signatures that represent an optimum solution. This process is explained in Figure 2. As shown in Figure 2.A the signature library is first populated by randomly generated signatures, each represented by a red dot. This list is then pruned to remove all signatures that match self (non spam) signatures as shown in Figure 2.B with randomly generated new signatures, being generated in order to replace the signatures that were removed (See D in Figure 2). The newly-generated signatures, as well as those retained from the previous iteration, will form the next generation of the signature library. This process is repeated until the signature library contains a set of signatures that match non self (spam) signatures, indicating that the signature library has now reached an optimum.

The following section describes the self and non self messages.

### 4.2 Self and non Self

A message that is identified as non self, i.e. sent by someone else, should be classified as spam by our system. The user of our system have the opportunity to provide input to our system, indicating whether a message constitutes spam or not. Thus, the system learns what spam is by initially basing its decisions on the user's input. The system should then be able to identify a message to be spam when a message is encountered that is not usually sent by the user. The system should be adaptive so as to cater for changes in users' sending behaviour and learn to relearn what constitutes spam in the case that a user's sending behaviour might change due to, for example, social influences. The AIS implementation should then still be able to identify an SMS sent by the user as a valid SMS, even though the user's sending behaviour might be changing. It is essential that the user correctly identifies to the system that a legitimate message is a non-spam message, as opposed to identifying a legitimate message as a spam message. For example, a user may tolerate messages incorrectly identified as spam (which the system will then learn to identify as non spam), but the user will be less tolerant of spam messages being incorrectly marked as non spam. The following section describes how the affinity measure is calculated as well as how the antibodies are represented digitally in our AIS.

### 4.3 Affinity measure and Digital Antibodies

The affinity measure or similarity measure is used to determine how similar a message signature is to a signature library of legitimate (self) message signatures. Such libraries of message signatures are also known as digital antibodies. As explained in section 4.1, these antibodies are randomly generated by the system and matched to antigens using an Euclidian distance formula. The Euclidian distance formula is, in this case, acting as our fitness function, determining how closely our antibodies bind to their target. These digital antibodies, thus, are used to determine whether or not an SMS message is a self (non spam) or non-self (spam) message. The Euclidian distance formula, in mathematical terms, is the ordinary distance between two points that one would measure with a ruler *(Elena Deza & Michel Marie Deza )*. The Euclidian distance was selected for this prototype, but other methods such as the Pearson Correlation Coefficient *(U. Aickelin, D. Dasgupta)* can be used instead. The aim of the latter is, similar to using the Euclidian Distance, to match the digital representation of the message signature with the digital signatures in our signature library.

## 4.4 Detection binding

Detection of foreign objects in the NIS of the body is done through binding. This means that antigens are the target which antibodies attack (bind to). The immune system's antibodies may bind to many different antigens, although some will bind more closely than others - the binding process does not always happen due to an exact match, but can also bind due to a partial match known as approximate binding. The antigens to which a given antibody will bind must, therefore, be similar in shape, but do not need to be exactly compatible. A given detector (antibody) can bind to many cells, and a given cell might have multiple detectors which can bind to it. The strength of the binding depends on how closely the two shapes can match. In a nutshell, the antibodies will attack the antigens (in the case of spam, the message is "attacked") and determine if it is a pathogen (non self). In the case of spam, the spam message that needs to be destroyed, constitutes the pathogen. In the case that it is not a pathogen, i.e. in the case that the message is not spam, the message is left alone.

There are approximately $10^{16}$ different foreign proteins which the NIS must recognize, yet the repertoire of the NIS contains a much smaller number of actual receptor (in our case, signatures in the signature library) types, closer to $10^{8}$ *(Lee A. Segel and Irun R. Cohen)*. This is accomplished by using approximate binding. Thus, the immune system can use a smaller number of antibodies to detect a large number of potential pathogens, as long as pathogens have similar shapes. Figure 3 shows an example of binding and non binding. In the first example the antibody on the left binds to the antigen on the right (i.e. in the case of spam, identifies it as spam) In the second example the antibody on the left does not bind to the antigen (thus, in the case of spam, it is not identified as spam). Just because an antibody does not bind to an antigen does not mean the antigen is not a pathogen (actual spam), but that the immune system has not learnt to recognise it as such yet.

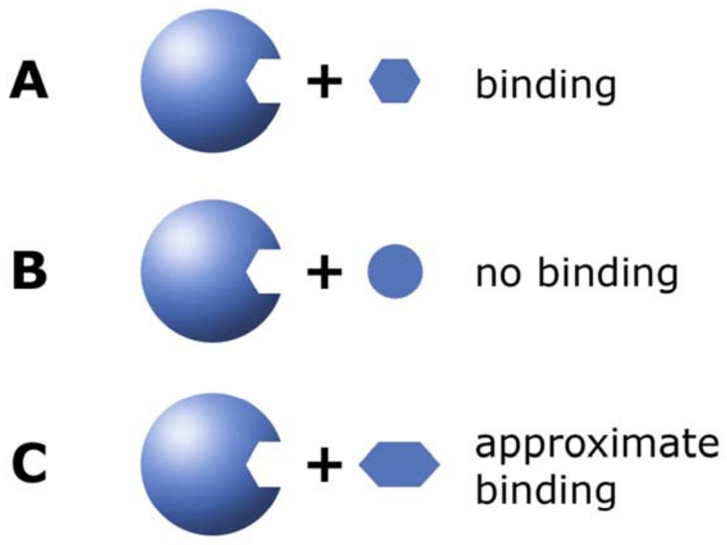


*Figure 3: Detection binding*

The AIS should, as is the case with the NIS, use approximate binding to identify spam messages as this would reduce the number of antigens that need to be stored in the signature library. The following section provides a model for the artificial immune system used to detect spamming Botnets on a mobile device.

## 5 A Model for a Botnet Detector using an Artificial Immune System

In this section we introduce the model for our spamming Botnet detector and explain the reasons for it being modelled in this manner as well as its advantages and disadvantages. We discuss the data that will be used to train our algorithm, the population and evolution of the signature library as well as the reasons for implementing the prototype on a device and not on a network provider's server (as was initially considered). We round this section off by explaining how the prototype would be used in a real-world situation to combat Botnet spam.

### 5.1 Learning with positive data

A unique characteristic of Artificial Immune Systems (AISs) is that training only requires positive examples *(A. J. Graaff and A. P. Engelbrecht)*. What we mean by this is that the AIS implementation only needs to be trained on positive data (valid messages), i.e the training set only needs to consist of valid SMS messages and not negative data (spam messages). Thus, there is no need to train the AIS implementation to learn what a spam message will look like, as the AIS will deduce this by figuring out what constitutes valid SMS signatures. This is ideal when a profile of what the spam SMSs will look like does not exist and, thus, the AIS cannot be provided with spam SMSs to train on. After training, the AIS should have the ability to classify non-self (spam) patterns from the self (non-spam) patterns. This makes the AIS an ideal candidate for classification problems where only one class of pattern is available for training.

## 5.2 Building the signature library

When building the signature library, which stores the antigens, the choice was between targeting valid SMSs and spam SMSs, as one cannot possibly build a profile of all the different types of SMS Spam. This is sp due to the fact that, what constitutes Spam is forever changing. Also, a list of all possible Spam profiles would be very large. In addition, there is a question of where SMS spam would be obtained from.
The approach followed by the authors in this implementation was to build a profile of the user's SMS-sending behaviour as this could be done by storing and analyzing message signatures. The prototype would be trained on the user's messages, learning to identify messages as self. Thus, everything else becomes non self, i.e. spam.

## 5.3 Remote analyses vs. Analysis on the device

There are two ways in which one could analyse a mobile device user's SMS data. The first possibility would be to analyse the SMSs sent on the ISP's servers and use this to build a profile of the user's SMS sending behaviour. There are several drawbacks to this approach. Firstly, there are privacy implications of analysing SMS data on an ISPs server, but even if these concerns were addressed, the classification algorithm would have to determine whether or not a message is valid or invalid without the user's input. Thus, it would not be able to learn based on user feedback. The second solution – the one selected by the authors – was to implement the detection algorithm on the device, thus, taking care of the privacy implications as well as allowing for user feedback in the learning process. The major disadvantage of this approach is that the prototype needed to be installed on a mobile device which has limited processing power and storage space especially when compared to an ISP's server. Thus, the prototype had to be programmed to use minimum storage and be optimised to make use of as little memory as possible.

The authors implemented an anomaly detector prototype using an Artificial Immune System (AIS) on an Android mobile device emulator to detect Botnets. The model developed – on which the AIS implementation is based – states that a software implementation used to detect Botnets, called Botnet Detector, is deployed to a mobile user's device. This application captures all outgoing SMSs and feed them to the AIS. The AIS implementation would learn to classify valid (self) SMSs. When the device encounters an SMS that it suspects to be invalid (non self) and, thus, possibly constitute spam, it would alert the user and ask them to confirm whether the message is indeed valid or otherwise intended. If the user indicates that the message is not valid, the system could perhaps prompt the user to contact their network provider then a clean up of the device can begin with the network provider perhaps installing an anti virus. If the user indicates the message to be valid, then the AIS implementation learns to recognize the new pattern as a valid SMS thus it will update its signature library and remove those signatures that bind to the message.

### 5.4 Flow Diagram for Botnet Detector

Figure 4 visualizes how the Botnet detector is designed to work. The mobile user enters a text message and sends it to a recipient (figure 4.3). This message is intercepted and certain message characteristics such as the number of capital letters (the full list of characteristics is defined in the next section) are also extracted for analysis by the Botnet Detector before the message is sent (the AIS should not send out messages identified as spam messages). The characteristics are sent to the AIS which then determines whether the message is valid or not by matching the signature of the message to the signatures in its signature library (this is explained further in the next section). If the AIS can determine that the message is valid, the message is sent onwards (figure 4.1). Else, if the AIS suspects that the message is spam, it prompts the user to confirm whether the message is valid or not (figure 4.2). If the user confirms the sms is valid, the message is released and sent onwards to the recipient and the AIS learns to recognize that type of message as valid. Else, if the user indicates that the message is invalid, the AIS does not learn the new pattern and the message is not sent. The next section describes how this model was implemented by means of a prototype.

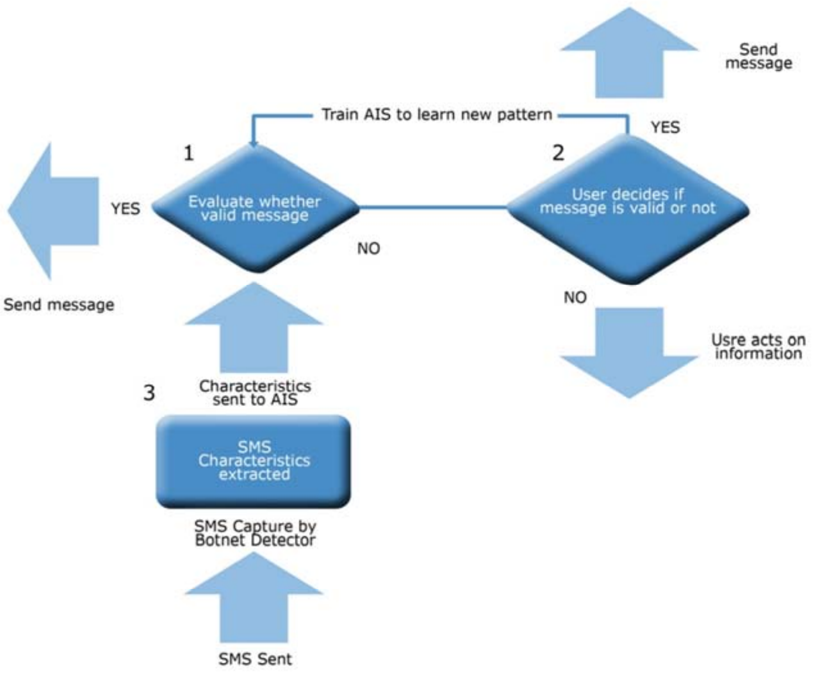

Send message
Train AIS to learn new pattern
YES
1
2
YES
Evaluate whether valid message
User decides if message is valid or not
NO
Send message
NO
Usre acts on information
3
Characteristics sent to AIS
SMS Characteristics extracted
SMS Capture by Botnet Detector
SMS Sent

*Figure 4: Model for a Botnet Detector using an Artificial Immune System to detect Spamming Botnets*

## 6 Implementation of Mobile Botnet Detector on an Android Device – Prototype

This section describes how the prototype was implemented. The authors discuss the message characteristics chosen to extract and train the AIS module with, the algorithm selected to train the AIS, the message signature representation and also how an affinity measure, which is used to match the message signatures to our signature library, is calculated. Included here is also a section describing the Android operating system which was chosen as the operating system onto which this model is implemented.

### 6.1 Message Signature

The prototype creates a signature (pattern) for each message sent by the mobile device. The signature consists of the following characteristics that are analysed by the AIS to determine the validity of the message:

- The total number of characters in a message including white spaces
- The total number of characters excluding white spaces
- The number of capital letters
- The number of white spaces
- The number of punctuation characters
- The number of digits
- The number of words
- The presence of URL’s
- The presence of telephone numbers

The specific characteristics mentioned above were chosen by the authors to define the message signature as they allow the implementation to build a profile of the user’s sending behaviour. The characteristics chosen are simple to capture, yet indicative of sending behaviour. Use of punctuation, capital letters as well as message length may reveal much about a user’s SMS sending behaviour as people have different messaging styles with some individuals being more attentive to grammar and thus more likely to use punctuation than others.

The majority of spam emails contain a link to a URL, thus it makes sense to mark the presence of URL’s in the SMS message (these links might lead to a website which sells a product the spammer is attempting to advertise). The presence of telephone numbers is also be a useful bit of information to mark as the spammer might include a telephone number to call (quite possibly at a premium rate).

Additional characteristics may be added in future to the prototype to increase the accuracy of the implementation, this would enable us to build a better profile of the users messages. The expansion of the list of characteristics would require more processing and storage, but would better define the message signature and, thus, the AIS’s ability to distinguish between spam and non-spam.

The authors felt that the current implementation though storing and analysing a limited number of message characteristic should nevertheless be able to detect invalid messages as the marking of URL’s and telephone numbers (Needed for individual being spammed to be able to purchase the product advertised), combined with the characteristics that define the user (such as message length) should enable us to accurately build a profile. When deciding what metrics to measure we made a number of assumptions, namely those individuals that normally send short messages with liberal use of Capital letters would not generally send long messages with lots of punctuation and no capital letters. This alone would not be enough to mark the message as spam (He could be typing a message to his mother where normally he messages his mates) but if this change was accompanied with a message containing URL's and telephone numbers this could be enough to warrant asking the user for a confirmation before sending the message.

Figure 5 shows a sample message that could be sent by the user to a friend.

hey bud wuu2?

*Figure 5: Sample message sent by user to a friend*

Figure 6 shows a sample message sent by the user to his mother, this message is very different from the previous sample, but should nevertheless not be identified as spam.

Good morning mommy, trust you are well. Please tell dad I'll be over to fix the blinds later this afternoon, I just want to pop round to Mary's place on my way to the hardware shop. Bye, love you!

*Figure 6: Sample message sent by user to their mother*

Figure 7 shows a sample message that is advertising a product and thus unlikely to be sent by the user.

INSURE now and SAVE! Up to 50% OFF car and household insurance!! Don't miss out CALL 0005556677 NOW. This offer ends 01/05/2012… www.carandhouseholdinsurance.com

*Figure 7: Sample message that is unlikely to be sent by user, possible spam*

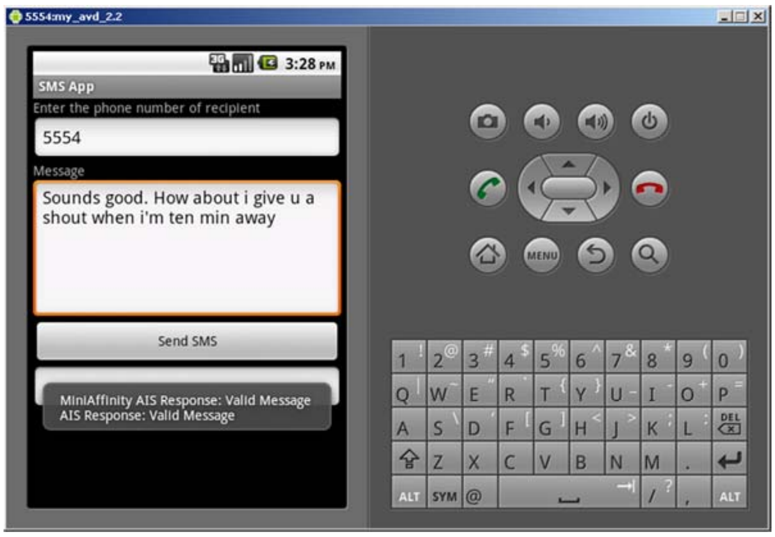
The sample messages above give us an indication of what the AIS should be looking for, it is however important to note that we do not tell the botnet detector what to look for, but rather allow the AIS to learn what to look for through trial and error. The next section discusses how the message is digitally represented in the AIS.

### 6.2 Signature Library

The initial population of the signature library is generated randomly by the AIS as discussed previously in section 4.1; this library is digitally represented as a relational database and stored in a database installed on the device. During each successive generation (training is continuous) a proportion of the initial population that matches a valid message pattern is selected for deletion, these antigens are replaced by randomly generated new antigens or by mutated clones of existing antigens. The selection of signatures in the signature library for deletion is determined by its affinity measure, this is explained in further detail in the following section.

### 6.3 Digital representation of the SMS signature and affinity measure

The SMS signature (pattern) composed of the characteristics listed above, needs to be represented in a form that the algorithm can process. The attribute values are represented as real numbers and a Euclidean distance function is used as an affinity measure matching the patterns to the signature. In mathematics, the Euclidean distance is the ordinary distance between two points that one would measure with a ruler. This affinity measure is used to fit the signatures in the signature library to the message signature. The use of a Euclidean function serves the desired purpose for the model, as it approximates how closely the patterns fit.


5554:my_avd_2.2

3:28 PM

SMS App

Enter the phone number of recipient

5554

Message

Sounds good. How about i give u a shout when i'm ten min away

Send SMS

MiniAffinity AIS Response: Valid Message
AIS Response: Valid Message

*Figure 8: Sample message*

For example, the message in figure 8 can be represented in the message signature library as follows:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 60 | 48 | 2 | 12 | 1 | 0 | 0 | 0 |

*Table 2: Antigen- Array 1*

Table 2 is the digital representation of a message sent by a user; the message characteristics are stored as real numbers in a database table. This signature can be represented by an integer array. The values in the array are real number representations of the message body. For example the first column of this array is a count of the total number of characters including white spaces in the message body. The value of each element in the signature array is then matched to the corresponding element for all the signatures in the signature library if, say, it is compared against the following message signature from our signature library.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 60 | 48 | 2 | 12 | 5 | 0 | 3 | 5 |

*Table 3: Antibody – Array 2*

Table 3 is the digital representation of an antibody in the signature library. The Euclidean distance between the first element in array 1 and the first element of array 2 is 0. The Euclidean distance is calculated for all the elements in the array. If, say, the affinity (similarity measure) was set to two, the message in Table 2 would be classified as a match, i.e. spam. If the affinity measure was larger than two, it would be classified as a valid message.

The tolerance was calculated as follows: first the Euclidean distance is calculated between the corresponding elements of the set. In one dimension, the distance between two points on the real line is the absolute value of their numerical difference. Thus if x and y are two points on the real line, then the distance between them is given by:

$$\sqrt{(x-y)^2} = |x - y|$$

So the Euclidean distances for all the elements in our two arrays are as follows,

[60-60] = 0
[48-48] = 0
[2-2] = 0
[12-12] = 0
[1-5] = 4
[12-12] = 0
[0-0] = 0
[0-3] = 3
[0-5] = 5

If the affinity tolerance was not exceeded (in this case to elements are out by a factor of one), that means that the antibody recognises the antigen as a self (valid message), if it is exceeded it is recognised as a pathogen. Depending on the algorithm chosen this message would be classified as spam or non spam. As is discussed later the prototype implemented here uses a negative selection algorithm (discussed in detail further in the paper), In this case the antibody binds to (matches) the message and is thus a non self (spam) message.

### 6.4 Algorithm Selection

The detection of SMS spam is a typical classification problem where patterns (signatures) need to be classified as legitimate or not. Thus, it is a simple binary classification problem in which the data set (messages) are classified as either spam or non spam. The challenge faced, however, was that, initially, the implementation would only be able to store legal signatures and would not actually know what to expect or how the signature of an illegitimate message looks like. AISs, however, are better suited for this kind of problem where only patterns of one class are known and, hence, the complement of that class simply needs to be identified.

In the context of the natural immune system, the prototype implements the censoring process on the signatures in the signature library, i.e. matches the antibodies against the antigen to determine the validity of the message. This is known as negative selection, where random signature detectors are generated and "matched" against the repository of known legitimate signatures. If a randomly generated signature detector "matches" a legitimate signature (to a certain degree), it is replaced (or mutated) until the signature detector does not match the legitimate signature. In other words we attempt to create a signature library that contains possible matches for spam messages by discarding any signatures in the library that match a valid message. This is different from positive selection where one would compile a list of valid message signatures in the signature library and train the AI system to learn those patterns.

The result is a set of legitimate signature-tolerant detectors which are unable to detect legitimate signatures (because of the censoring process) but are able to detect anything else that does not "look" like the legitimate signatures. These detectors are then used to classify a message as illegitimate or not.

The problem with this approach is that it may take a long time to actually get a signature detector which does not detect any of the legitimate signatures. The following section explains how this was solved by implementing a negative selection artificial immune system using a mini-affinity measure.

### 6.5 Negative selection algorithm using a mini-affinity measure.

This section discusses the algorithm selected for implementing the prototype and discusses the logic behind the algorithm as well as its shortcomings. Instead of having a global affinity threshold (i.e. a user-defined affinity threshold), each detector was assigned its own affinity threshold *(A. J. Graaff and A. P. Engelbrecht)*. An affinity is defined as the degree of matching between a signature in the signature library (antibody) and a message (antigen). The latter satisfies the relation implied by the former if this degree is greater than an affinity threshold *(Leandro N. De Castro, Fernando J. Von Zuben, Helder Kni)*. This threshold could then be set to the minimum calculated Euclidean distance between the specific detector and all signatures randomly generated by the Spam Detecting AIS (SDAIS) signatures **A**. This means that the closest signature in **A** to the detector will determine the affinity threshold of that detector. Thus, the end result is a set of detectors where each has its own affinity threshold. This cuts out the indefinite process of generating random detectors until one is actually found that is signature tolerant, i.e. tolerates self (non spam) and also removes one of the user-specified parameters (otherwise the affinity measure would have to be defined as some constant). Algorithm 1 bellow, which will be explained in detail, was the algorithm selected for this implementation.

```
Algorithm 1 : Negative selection AIS with self calculated affinity
1: Given A a set of valid SMS signatures
2:          n the user-defined number of antibodies
3: Initialize the set of patterns B to empty
4: while |B| < n do
5:       Randomly generate pattern D
6:       $D_e = min_{a \in A}$ dist(a,D)
7:       Add D to B
8: end while
```

*Algorithm 1: Negative selection AIS with self calculated affinity*

When an SMS is sent, the signature of the new message is then measured against all detectors in the detector (signature library) list **B**. This means that as soon as there is a detector (i.e. an antibody) in **B** with an affinity (Euclidean distance) to the new signature equal or less than minaffinity, the new message signature is detected as illegitimate (non self). If the user indicates that the message was in fact legitimate, then the new signature needs to be added to **A** and all detectors in **B** needs to go through the censoring process with the new signature. If a detector in list **B** detects the new signature (which is now part of **A**), the detector should be removed and replaced with a randomly-generated detector. The affinity is calculated between the new message signature and each detector in **B** The affinity threshold is local to each detector, thus, when a message is legitimate, only the affinity thresholds of those detectors which detected the new message need to be updated. The following section describes the android operating system and mobile device emulator used to build and test the prototype with.

## 6.6 Android

Android is a mobile operating system based upon a modified version of the Linux kernel. Android has a large community of developers writing application programs (apps for short) that extend the functionality of the devices. Android is the most popular smart phone operating system in the world *(PCWorld)*. Developers write managed code in the Java language, controlling the device via Google-developed Java libraries *(Shankland, Stephen )*. The authors implemented the prototype with Android version 2.2 using an eclipse integrated development environment. The authors selected the Android operating system primarily because Android is an open source project with lots of developer support and also because of the authors familiarity with the Java programming language. The following section describes the how the prototype was set up as well as the experimental results.

## 6.7 Prototype Setup

This section presents and discusses the performance of the AIS model implemented on an android emulator. The responses - valid and invalid inputs - were tabulated. The first 500 inputs were used to train the AIS implementation. The next 100 inputs for the AIS implementation were divided into two data sets. The first set consisted of 80 valid SMS messages (the self set); the second set consisted of 20 invalid sms messages (non self set). The following section discusses the data selection process followed by a section discussing how the experiment was carried out, which, in turn, is followed by a section tabulating the results.

## 6.8 Data selection

The data that was used to train the AIS implementation was selected by using SMS messages sent by one of the authors over the last six months. In total 500 of these valid SMSs were randomly selected and used to train the AIS Botnet Detector. The second set of valid smses (80 in total) that were used to test the efficiency of the Botnet detector were selected in a similar manner, the only difference being that the user prompt and learning of the AIS was switched off to accurately measure how well the device learnt the valid SMS signatures. The Invalid SMSs (20 in total) were selected by the authors from unsolicited SMSs received by the authors during the same time period usually advertising some or other product.

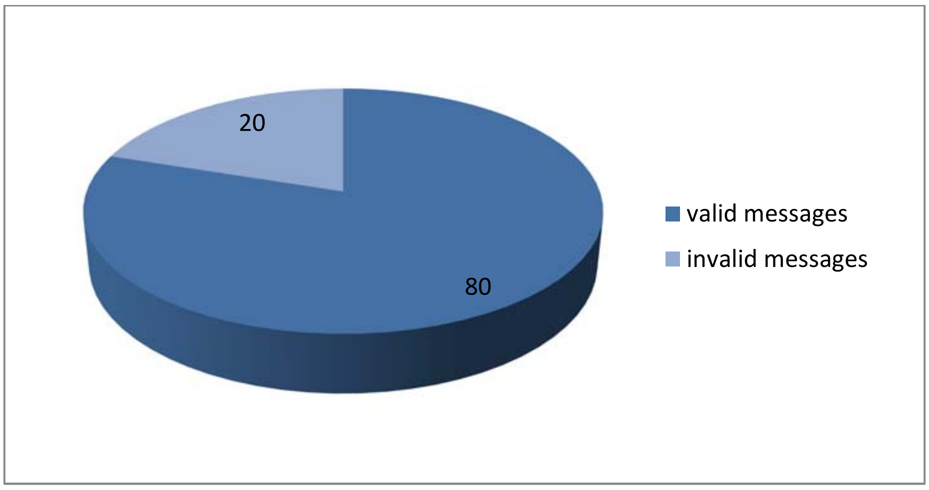


*Chart 1: Valid and invalid message breakdown*

Chart 1 provides a visual breakdown of the valid and invalid messages used in the experiment. The following section discusses the experiment.

## 6.9 Running the Prototype

The AIS algorithm was implemented on an android mobile smart phone emulator. The Botnet Detector captures all outgoing messages and extracts the message characteristics from the message body. These are saved in a SQLite3 database *(SQLite)*. The Artificial Immune System then processed this data to determine if the message was valid (Figure 9) or not (Figure 10). Figure 9 and Figure 10 show responses to valid and invalid SMSs during the evaluation phase respectively.

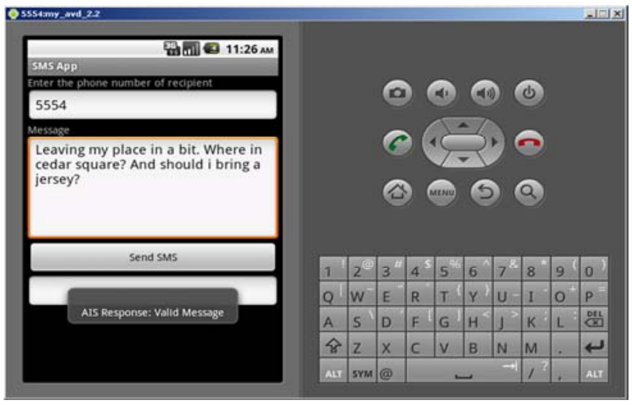

5554:my_avd_2.2
11:26 AM
SMS App
Enter the phone number of recipient
5554
Message
Leaving my place in a bit. Where in cedar square? And should i bring a jersey?
Send SMS
AIS Response: Valid Message

*Figure 9: Response to a valid message sent by Author*

The message in Figure 5 is a sample message collected from one of the author's mobile device. As discussed these and other messages were used to train the AIS, when this message was picked up by the AIS the message characteristics listed in section 6.1 were extracted and compared to all the signatures in the signature library (detectors), as this message does not match any detectors in the signature library it is classified as legitimate.

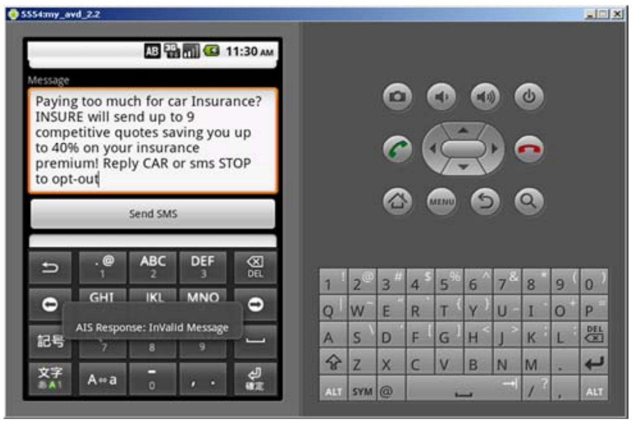

5554:my_avd_2.2
11:30 AM
Message
Paying too much for car Insurance? INSURE will send up to 9 competitive quotes saving you up to 40% on your insurance premium! Reply CAR or sms STOP to opt-out
Send SMS
AIS Response: InValid Message

*Figure 10: Response to an Invalid message*

The message in Figure 10 goes through the same process as the message in Figure 9, except in this case the message signature is matched to a detector in the signature library, and is thus classified as invalid. The user will then be alerted.

## 7 Tabulation of Results and Discussion

The results of the experiment are tabulated as follows and show the accuracy in detecting spam SMSs

| Valid (non-spam) Message (Self) | Invalid (spam) Message (non - Self) | Total error |
|---|---|---|
| 84% | 65% | 20% |

*Table 4: Results*

The results tabled in table 4 show that indeed an AIS implementation can effectively detect invalid SMS messages. The results are now briefly discussed. The AIS correctly identifies 84% of non-spam (self) SMSs (i.e. 67 out of 80 valid messages). The AIS correctly identifies 65% of spam (non-self) SMSs (i.e. 13 out of 20 invalid messages). Its total error (Incorrectly identified messages) is 20%.

Chart 2 shows the accuracy of the SDAIS in correctly identifying valid messages.

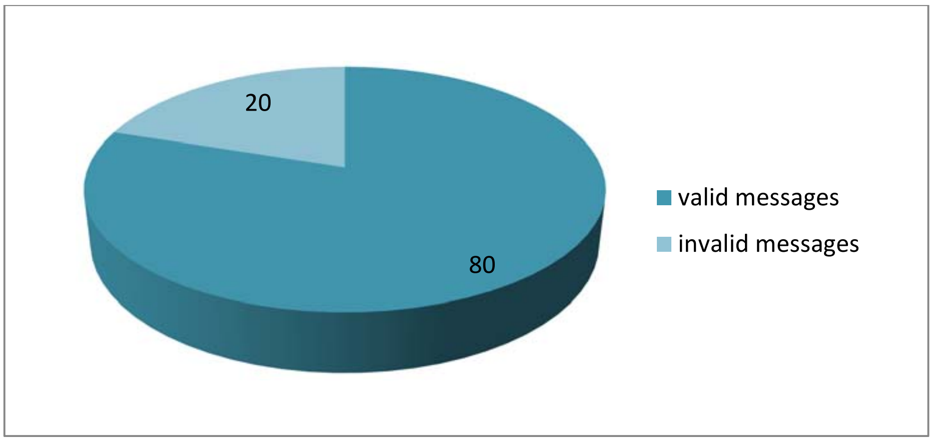


*Chart 2: Accuracy of SDAIS in identifying valid messages*

Chart 3 shows the accuracy of the SDAIS in correctly identifying invalid messages as spam.

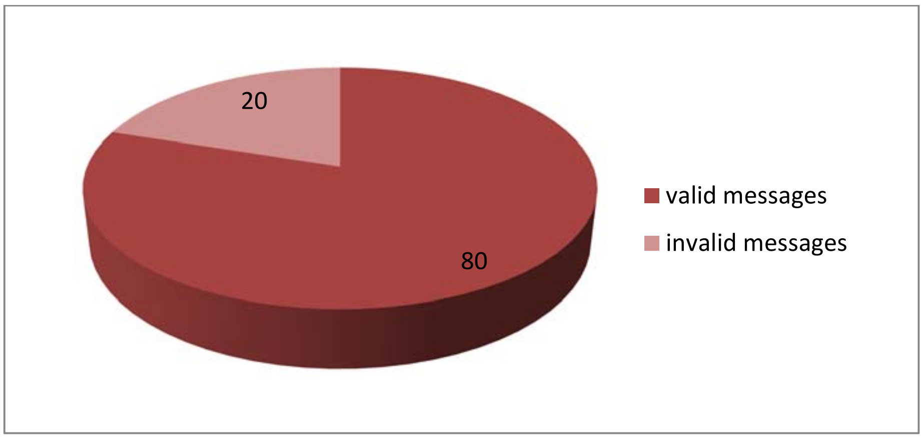


*Chart 3:Accuracy of SDAIS in identifying invalid messages*

This shows us that the SDAIS is better at accurately identifying valid messages then it is at identifying invalid messages. The accuracy of the SDAIS could be enhanced by adding more metrics to be analysed, thus, increasing the accuracy of the antigen binding. The size of the antibody list (which is used to match non-self messages) could be expanded. The accuracy of the AIS implementation should improve with more learning and a larger antibody list size. The larger antibody size, would allow the SDAIS to better identify non-self cells and thus increase the accuracy of its responses. The Authors conclude in the following section and also describe future work to be undertaken.

## 8 Conclusion and Further Work

The Authors started this research with the aim of combating potential threats to mobile devices, primarily the use of mobile devices to send spam SMSs to persons on their phone book as well as others. An Artificial Immune System was chosen to accomplish this because AISs are unique in the sense that training only requires positive examples. This approach to detecting spamming Botnets was thought to be the most practical solution for the commercial application of the prototype. The authors believe that this implementation would serve as a useful tool alerting a user of possible Bots on their mobile device. This would allow them to remove the malware from their device.

The Authors plan to improve upon the accuracy of the Bot Detector by adding to the list of metrics being extracted from the SMS and used to train the AIS implementation. Among the metrics the authors plan on extracting, is the time of day that the messages are sent and perhaps the number of recipients per SMS. It is hoped that this will improve the accuracy of this implementation by adding a non-message metric to our analysis of the user behaviour. The authors hope to apply these ideas out on an Instant Messaging platform as well as on SMS messaging. The authors also hope to test the performance by testing the SDAIS on a mobile phone, as the current experiment was conducted on an Android emulator the processing power and data storage capabilities of the SDAIS do not truly reflect those found that would be found on a mobile device, as the mobile device would have more limited storage and processing capabilities. The performance of the SDAIS would be expected to suffer as a result and their could be a noticeable slow down in user performance.

**Acknowledgments**

This work is based on research supported by the National Research Foundation of South Africa (NRF) as part of a SA/Germany Research cooperation programme. Any opinion, findings and conclusions or recommendations expressed in this material are those of the author(s) and therefore the NRF does not accept any liability in regard thereto.

## References

[Australian Government] Australian Government Department of Broad Band Communications and the Digital Economy. “Spam”. http://www.dbcde.gov.au/online_safety_and_security/spam [September 2011]

[Acts Online, 2002] Acts Online, 2002. Electronic Communications and Transactions Act ,2002. Available: http://www.acts.co.za/ect_act/. [April 2009]

[A. J. Graaff and A. P. Engelbrecht] A. J. Graaff and A. P. Engelbrecht, “Optimised Coverage of Non-self with Evolved Lymphocytes in an Artificial Immune System,” International Journal of Computational Intelligence Research, vol. 2, no. 2, pp. 127-150, 2006.

[A.S. Perelson, G. Weisbuch] A.S. Perelson, G. Weisbuch, “Immunology for physicists”, Reviews of Modern Physics, vol. 69, no. 4, October 1997.

[B.G. Kutais] B.G. Kutais, “Spam and Internet Privacy”, ‘Journal of High Technology Law Suffolk University Law School’

[Earth Web] Earth Web. ”Think Spam is tough? Try Fighting Spim” http://itmanagement.earthweb.com/secu/article.php/3365931 [September 2011]

[Edmund K. Burke , Graham Kendal] Edmund K. Burke , Graham Kendall , ‘Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques’ http://eprints.nottingham.ac.uk/621/1/03intros_ais_tutorial.pdf [September 2011]

[E. Cooke, F. Jahanian, and D. McPherson] E. Cooke, F. Jahanian, and D. McPherson. The zombie roundup: Understanding, detecting, and disrupting botnets. In USENIX SRUTI Workshop, pages 39–44,2005.

[Elena Deza & Michel Marie Deza] Elena Deza & Michel Marie Deza (2009) Encyclopedia of Distances, page 94, Springer.

[Federal Communication Commission],“Spam: Unwanted Text Messages and Email”, http://www.fcc.gov/guides/spam-unwanted-text-messages-and-email [September 2011]

[Garfinkel, S] Garfinkel, S. Network Forensics: Tapping the Internet. Web Security, Privacy & Commerce, 2nd Edition.http://www.oreillynet.com/pub/a/network/2002/04/26/ nettap.html

[Georgia Tech Information Security Center] Emerging Cyber Threats Report for 2009, Georgia Tech Information Security Center, October 15, 2008

[Industry Canada] Industry Canada. “Government of Canada Introduces Anti-Spam Legislation” http://www.ic.gc.ca/eic/site/ecic-ceac.nsf/eng/gv00521.html#Q1 [September 2011]

[Internet Service Providers] Internet Service Providers' Association, 2008. ‘What is Spam?’ Available: http://www.ispa.org.za/spam/whatisspam.shtml. [April 2009]

[Leandro N. De Castro, Fernando J. Von Zuben, Helder Kni] Leandro N. De Castro, Fernando J. Von Zuben, Helder Kni, “Artificial immune systems”, 6th international conference, ICARIS 2007 pg 124

[Lee A. Segel and Irun R. Cohen] Lee A. Segel and Irun R. Cohen, editors. Design Principles for the Immune System and Other

[Marshall Brain] Marshall Brain. How your immune system works. HowStuffWorks, 2004.

Distributed Autonomous Systems. Santa Fe Institute Studies in the Sciences of Complexity. Oxford University Press, 2001.

[McFee] McFee, 2010 Available: http://vil.mcafeesecurity.com/vil/content/v_138726.htm

[Michael Negnevitsky, Mark Jyn-Huey Lim, Jacky Hartnett, Leon Reznik] Michael Negnevitsky, Mark Jyn-Huey Lim, Jacky Hartnett, Leon Reznik Email Communications Analysis: How to Use Computational Intelligence Methods and Tools? Computational Intelligence for Homeland Security and Personal Safety, 2005. CIHSPS 2005. Proceedings of the 2005 IEEE International Conference. Orlando Florida, March 31 2005-April 1 2005, pp. 16-23.

[PCWorld] PCWorld “Android Edges RIM, Apple as Most Popular Smartphone OS” March 4 2011 Available: http://www.pcworld.com/article/221358/android_edges_rim_apple_as_most_popular_smartphone_os.html [September 2011].

[U. Aickelin, D. Dasgupta] U. Aickelin, D. Dasgupta, “Artificial Immune Systems”, 2005, Springer US

[Ryan Vogt, John Aycock, and Michael J. Jacobson] Ryan Vogt, John Aycock, and Michael J. Jacobson, Jr. “Army of Botnets”, Proceedings of the 2007 Network and Distributed System Security Symposium, pp. 111–123, 2007

[R. Dantu and P. Kolan] R. Dantu and P. Kolan. Detecting Spam in VoIP Networks. In Proceedings of USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI), July 2005.

[Samples of Spam] Spam-site “Samples of Spam” http://www.spam-site.com/spam-sample.shtml [September 2011]

[Security Vision from McFee Avert Labs] Security Vision from McFee Avert Labs, 2007 The Future of Security McFee, 2010 Available: http://vil.mcafeesecurity.com/vil/content/v_138726.htm

[Security.com] Seach Security.com “botnet (zombie army)”. http://searchsecurity.techtarget.com/definition/botnet [September 2011]

[Spam Emails and Spamming] Consumer fraud reporting, “Spam Emails and Spamming”, http://www.consumerfraudreporting.org/spam.php [September 2011]

[Spamhaus] Spamhaus “The Definition of Spam”, http://www.spamhaus.org/definition.html [September 2011]

[Seth Thigpen] Seth Thigpen, “Investigating Botnets, Zombies, and IRC Security”

[Shankland, Stephen] Shankland, Stephen (12 November 2007). "Google'sAndroid parts ways with Java industry group". CNET News. http://www.news.com/8301-13580_3-9815495-39.html.

[Sumit Kasera and Nishit Narang] Sumit Kasera and Nishit Narang. , 2005, 3G Mobile Networks. Architecture, Protocols and Procedure. Tata MCGraw-Hill Publishing Company, limited edition.

[SQLite] SQLite Copyright. sqlite.org. http://www.sqlite.org/copyright.html. [September 2011]

[Terri Oda] Terri Oda ‘A Spam-Detecting Artificial Immune System’ [2005]

[The Economist] The Economist “Big brother bosses” September 11 2009 Available: http://www.economist.com/businessfinance/displaystory.cfm?story_id=14413380 [September 2009].

# Spammer detection using honeypots and digital forensics

Ickin Vural, HS Venter

**Abstract— At present it is very difficult to trace the identity of spammers who use identity concealment techniques. It is difficult to determine the identity of the spammer by just analysing the electronic trail.**

**This paper proposes the design and implementation of a spammer detection system that uses honeypot techniques to detect abnormal behaviour on a network so as to identify potential spammers.**

## I. INTRODUCTION

Unsolicited bulk communication also known as spam is the practise of sending unwanted email messages, frequently with commercial content, in large quantities to an indiscriminate set of recipients [1].

The sending of unsolicited bulk communications with the intention to advertise products and generate sales is economically viable because senders have no operating costs beyond the management of their mailing lists. Because the cost of setting up a spamming operation is low spammers are numerous. Thus the volume of unsolicited bulk communications has increased dramatically over the past few years [2].

The costs of spam which involve lost productivity and fraud, these costs are borne by the general public, institutions that store and retrieve email for their employees and by Internet service providers (ISP's). Institutions and ISP's have been forced to add extra capacity to cope with the high volumes of unsolicited bulk communications [3]. Anti spamming legislation has been introduced in many jurisdictions. The problem faced by law enforcement is that spammers move their operations to jurisdictions that have no or weak anti spamming laws. At present it is very difficult to trace the identity of spammers who use identity concealment techniques. It is difficult to determine the identity of the spammer by just analysing the electronic trail using standard email tracing techniques. This paper line proposes the design and implementation of a spammer detection system that uses artificial intelligence techniques to detect abnormal behaviour on a network.

The remainder of the paper is structured as follows. The background section defines spam in more detail and also defines its cost and causes.

The next two sections are devoted to the state of the art of spamming techniques and how to trace spammers. More specifically, these two sections contrast each other in the sense that the third section looks at how spammers use bot-networks to conceal their identities, whereas the fourth section looks at techniques for tracing the identity of spammers using bot-networks. The paper then proposes the design of a spammer detection system to detect bot-networks in section V followed by the conclusion.

.

## II. BACKGROUND

Unsolicited bulk email otherwise known as spam is an email sent to a large number of email addresses, where the owners of those addresses have not asked for or consented to receive the mail [4]. Spam is used to advertise a service or a product. An example of spam is an unsolicited email message from an unknown or forged address advertising Viagra.

Spam is one of the most significant threats to the Internet, accounting for around 60% of all email traffic [4]. Spam costs consumers and ISPs vast amount of money in bandwidth charges.

Spammers generally do not pay much for the sending of spam. They accomplish this by exploiting open mail servers to do their task for them. The spammer need only send one email message to an incorrectly configured mail server to reach a vast number of email addresses. Recipients in turn need to pay access costs or telephone costs in order to receive content they did not ask for.

ISPs have to bear the bulk of the cost for bandwidth overuse by spammers. This cost is often passed onto the consumer through increased Internet access fees or a degraded service level.

The following section describes how spammers use bot-nets to send spam and how they conceal their identities from persons who would attempt to identify the source of spam mail.

## III. HOW SPAMMERS CONCEAL THEIR IDENTITIES USING BOT-NETWORKS

A Bot-Network consists of a set of machines that have been taken over by a spammer using Bot software sent over the internet. This Bot software hides itself on its host machine and periodically checks for instructions from its human Bot-Network administrator. Bot-nets today are often controlled using Internet Relay [5]. The owner of the computer usually has no idea that his machine has been compromised until its internet connection is shut down by an ISP. As most ISP's block bulk mail if they suspect it is spam the spammers who control these Bot-Networks typically send low volumes of mail at any one time so as not to arouse suspicions. Thus the spam mail can be traced to an innocent individuals network address and not the spammers network address.

Manuscript received June 7, 2009. Ickin Vural is with the Information and Security Architectures Research Group, Department of Computer science, University of Pretoria, Pretoria, South Africa (e-mail: ickin@tuks.co.za).

Prof Hein Venter is the leader of the Information and Security Architectures Research Group, Department of Computer science, University of Pretoria, Pretoria, South Africa (e-mail: hventer@cs.up.ac.za).

While the number of Bot-nets appears to be increasing, the number of bots in each Bot-net is actually dropping. In the past Bot-nets with over 80 000 machines were common [5]. Currently Bot-nets with a few hundred to a few thousands infected machines are common. One reason for this is that smaller Bot-nets are more difficult to detect.

## IV. Identifying the Identity of Spammers by Using Honeypots on Bot-Networks

A honeypot is a closely monitored computing resource that is intended to be compromised [6]. A honeypot computer can be applied to Bot-networks, open proxies and open proxies. Thus by setting up a computer to imitate a Bot-network, investigators can attempt to trap the spammers into revealing their network addresses.

One way of identifying spammers is to set up a computer to pretend that it is part of a Bot-network [7]. By allowing the honeypot computer to become part of the Bot-network we can obtain the Bot-network software used by the spammer. Once this has been done the honeypot waits for the spammer to send new instructions and then identifies the network address of the sender. The problem with this approach is that spammers send the instructions over open relays and open proxies thus it may be impossible to discover the identity of the spammer's network address in this way.

An open proxy is a machine that allows computers to connect through it to other computers on the internet. Open proxies exist because they enable unhindered internet usage in countries that restrict access to certain sites for political or social reasons. An internet user in a country that restricts internet access can access blocked sites by using an open proxy in a country that does not restrict internet access.

Spammers use open proxies to hide their network addresses. The recipient of a spammers email will not see the spammers' network address on the email but the open proxy's network address. It is estimated that sixty percent of all spam is sent using an open proxy [7]. Thus the spammer will use an open proxy to send instructions to the machines on their bot-network to avoid detection.

## V. Discussion

### A. *Is Spammer identification possible?*

This paper outlines the challenges facing digital forensic investigators when attempting to identify spammers using bot-networks in conjunction with open proxies. The use of bot-networks means that even if the source of the machine sending the spam is identified the person owning the machine may not be the one responsible for sending spam. The use of untraceable internet connections and open proxies to communicate instructions to bot-networks makes the use of Honeypots unlikely to succeed.

Thus any success in tracing spammers will be matched by spammers using increasingly sophisticated techniques to evade detection. Greater responsibility will have to shift to ISP's in monitoring connections to open proxies as well as attempting to shut down open relays. Nevertheless an arms race between spammers and digital forensic investigators will continue for the foreseeable future.

### B. *Bot-net identification*

This paper proposes the design and implementation of a system to detect spammers by analysing network traffic for abnormal behaviour. The implementation would have to take into account spam email sending patterns to effectively identify spammers. The implementation could make use of artificial intelligence to learn behaviour and thus detect abnormal behaviour.

The proposal would be to model a network as a graph and then train an artificial intelligence agent to learn expected and unexpected behaviour so as to detect a machine that could possibly have been taken over by a bot-network.

## VI. Conclusion

This paper outlines the challenges facing digital forensic investigators when attempting to identify spammers. The paper promotes the idea of Spammer identification as opposed to Spam identification system to halt spam.

## Acknowledgment

The Authors would like to thank the ICSA research group members for their input.

## References

[1] Spamhaus. 2009 The Definition of spam. Available: http://www.spamhaus.org/definition.html [April 2009].

[2] Email Metrics Program. 2007. 'The Network operators perspective' , messaging Anti-Abuse working group. Available: http://www.maawg.org/about/MAAWG20072Q_Metrics_Report.pdf. [April 2009]

[3] Europa. 2009 Data protection: "Junk" e-mail costs internet users 10 billion a year worldwide.Available: h'ttp://europa.eu/rapid/pressReleasesAction.do?reference=IP/01/154&format=HTML&aged=0&language=EN&guiLanguage=en' [April 2009].

[4] Internet Service Providers' Association, 2008 Available: http://www.ispa.org.za/spam/whatisspam.shtml. [April 2009].

[5] Evan Cooke, Farnam Jahanian, Danny McPherson. 2005 . The advanced computing systems association. [Online] The Zombie Roundup Understanding, Detecting, and Disrupting Botnets. Available: http://www.usenix.org/events/sruti05/tech/full_papers/cooke/cooke_html/, [April 2009].

[6] Niels Provos. 2004. The advanced computing systems association. [Online]. A Virtual Honeypot Framework. Available: http://www.usenix.org/event/sec04/tech/full_papers/provos/provos_html/. [April 2009]

[7] Boneh, Dan. 2004. The Difficulties of Tracing Spam Email. Department of Computer Science Stanford University.