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)*.

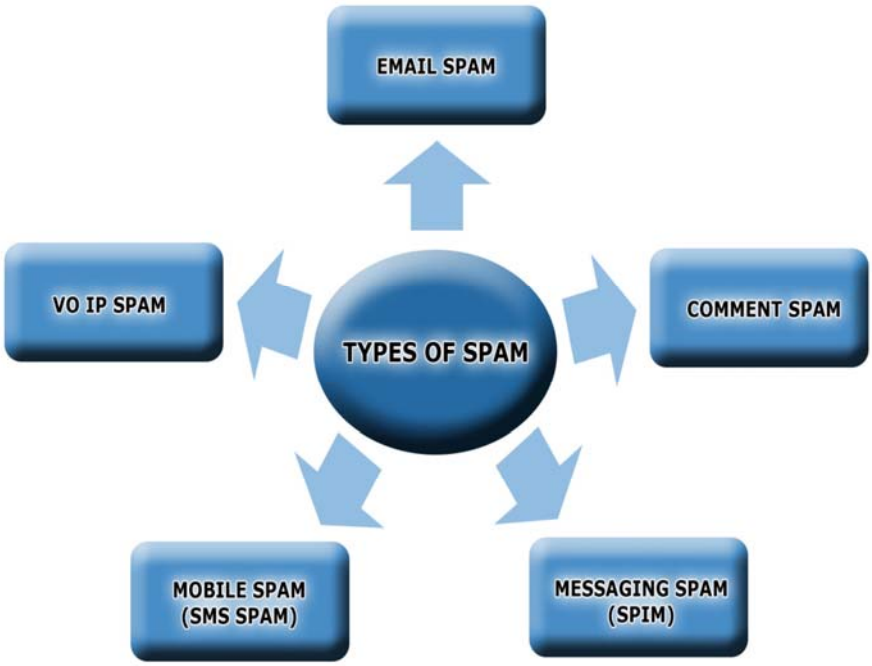

Figure 1: Types of Spam

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

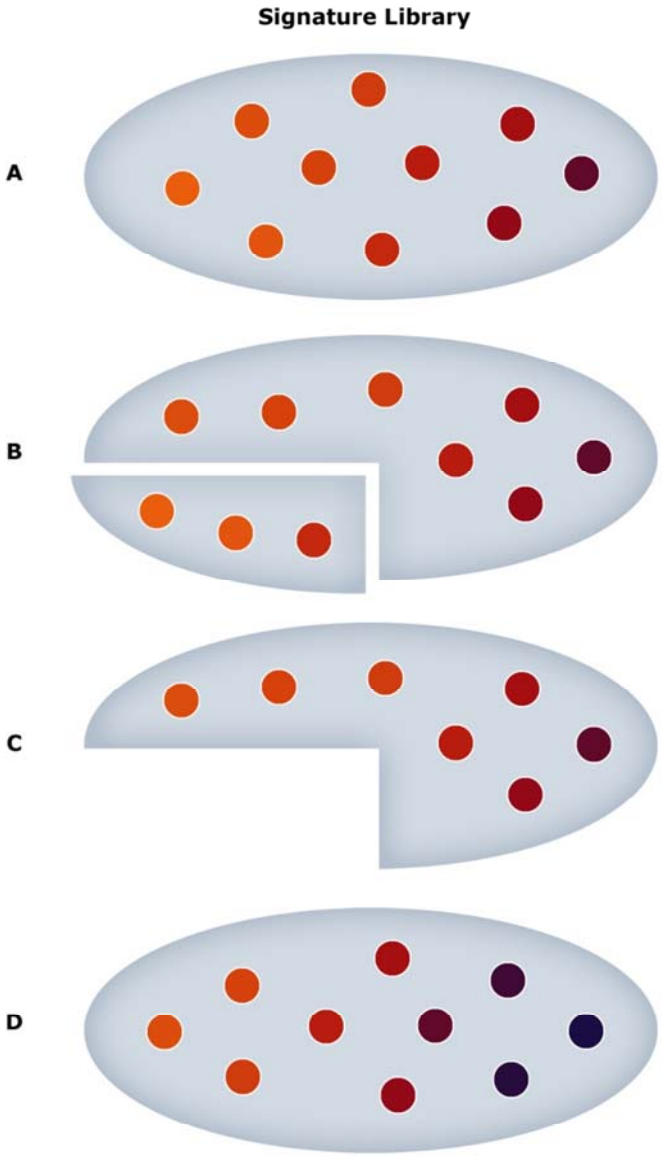

Figure 2: Signature Library

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.

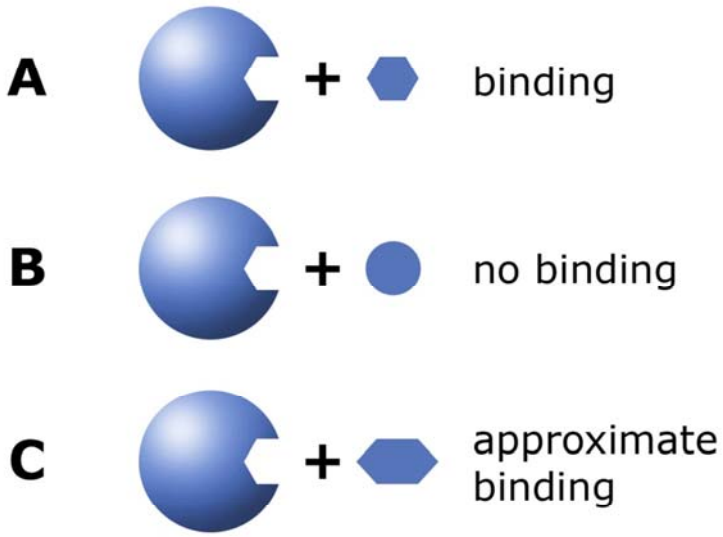

Figure 3: Detection binding

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.

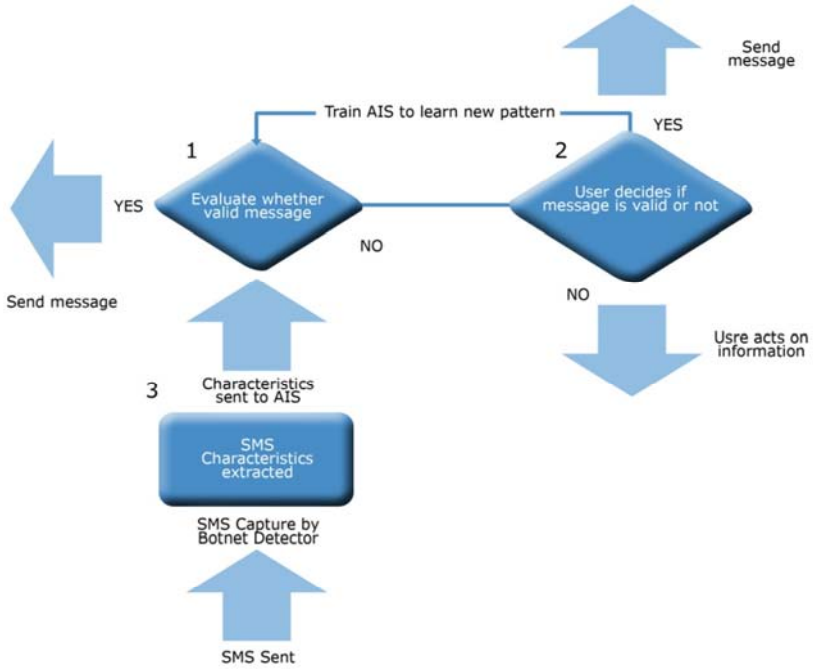

Figure 4: Model for a Botnet Detector using an Artificial Immune System to detect Spamming Botnets

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.

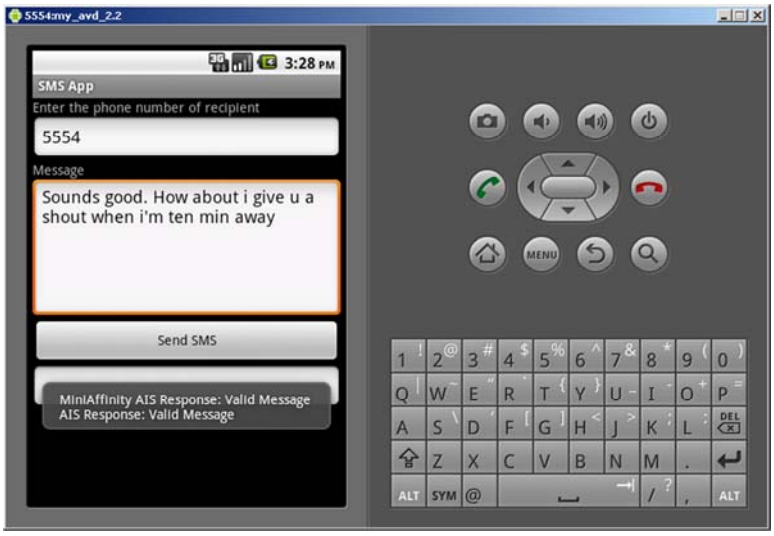

Figure 8: Sample message

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: De = $min_{a \in A}$ 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.

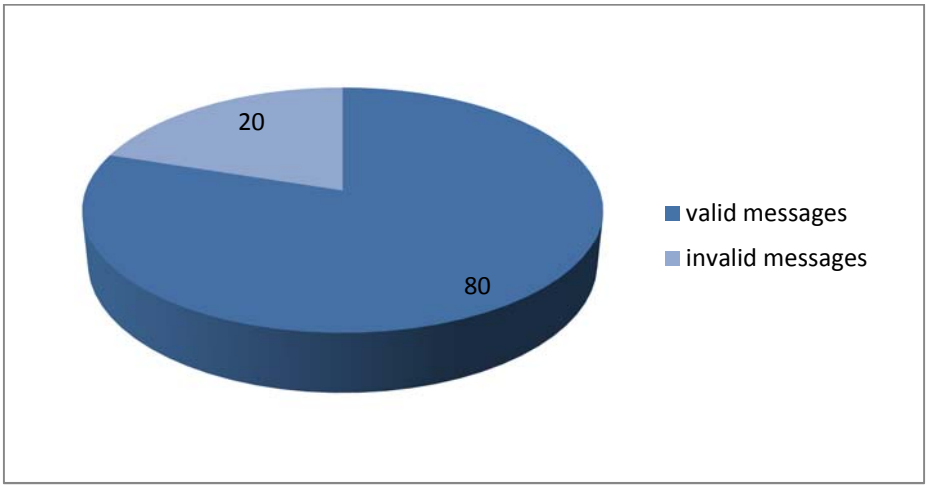

Chart 1: Valid and invalid message breakdown

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.

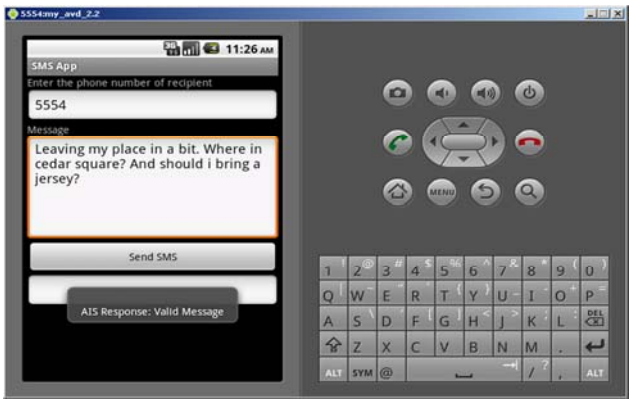

Figure 9: Response to a valid message sent by Author

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.

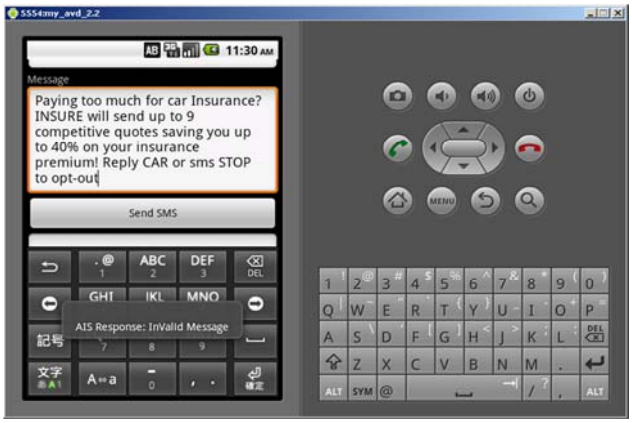

Figure 10: Response to an Invalid message

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.

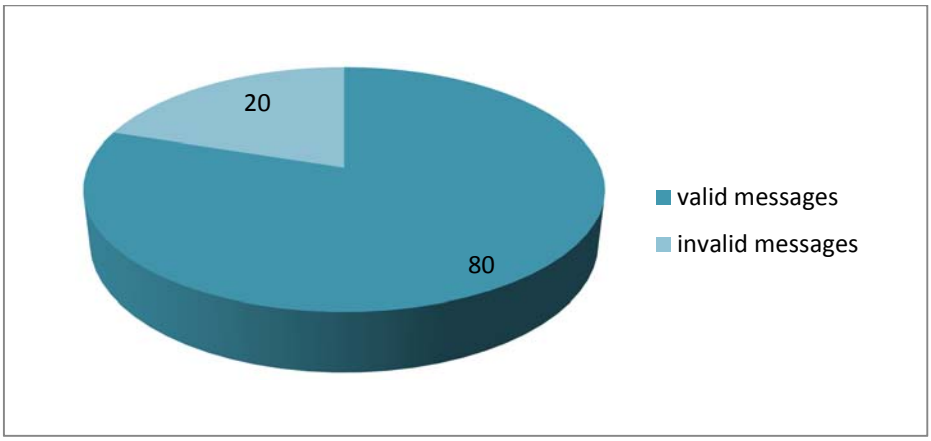

Chart 2: Accuracy of SDAIS in identifying valid messages

Chart 3 shows the accuracy of the SDAIS in correctly identifying invalid messages as spam.

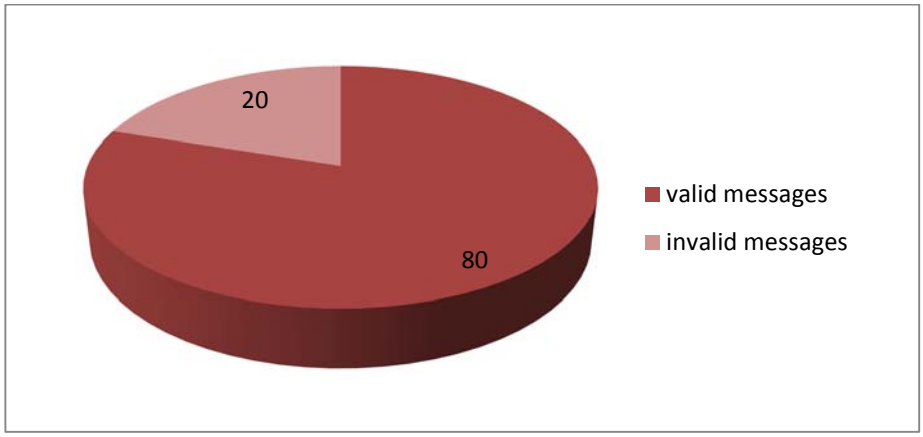

Chart 3:Accuracy of SDAIS in identifying invalid messages

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].