# Increasing the Accuracy of a Spam-Detecting Artificial Immune System

**Terri Oda**
Carleton University
1125 Colonel By Drive
Ottawa, ON K1S 5B6
terri@zone12.com

**Tony White**
Carleton University
1125 Colonel By Drive
Ottawa, ON K1S 5B6
arpwhite@scs.carleton.ca

**Abstract-**

**Spam, the electronic equivalent of junk mail, affects over 600 million users worldwide. Even as anti-spam solutions change to limit the amount of spam sent to users, the senders adapt to make sure their messages are seen. This paper looks at application of the artificial immune system model to protect email users effectively from spam. In particular, it tests the spam immune system against the publicly available SpamAssassin corpus of spam and non-spam, and extends the original system by looking at several methods of classifying email messages with the detectors produced by the immune system. The resulting system classifies the messages with similar accuracy to other spam filters, but uses fewer detectors to do so, making it an attractive solution for circumstances where processing time is at a premium.**

## 1 Introduction

The word “spam” is used to indicate the electronic equivalent of junk email. Exact definitions will vary, but it typically covers a range of unsolicited and undesired advertisements and bulk email messages.

Humans are very good at finding and handling spam messages, but as the quantity of spam and the ratio of spam to legitimate messages increases, it becomes more difficult as well as more time-consuming and costly to have email filtered manually.

SpamCon Foundation estimates a cost of $1-2 per spam in lost productivity, wasted resources, and anti-spam software and support [1]. When you consider the volume of spam sent and received daily, the costs become quite significant. International Data Corp estimates that 7.3 billion spam messages are sent daily (doubling from last year), and AOL users alone reported 5.5 million spam messages on March 5th 2003 (up from 4 million per day in late February) [2].

It seems logical to have a spam detector that adapts as spam changes, and an adaptive system based on Bayes rules was proposed in 1998 [3] [4]. This model was more recently popularized by Graham [5], who reported extraordinarily good results with his Bayesian classifier. This and several other popular anti-spam solutions are listed in Section 2.

The immune system model lends itself reasonably well to creation of another adaptive system [6]. Section 3 describes the immune system model and how it can be applied to spam detection.

Like the Bayesian systems, the Artificial Immune System (AIS) produced contains a unique set of detectors, making it harder for spam senders to create messages that will go through many such systems. It can detect not only repeat messages, but also materials that have not already been seen by the system. An AIS has many of the advantages seen in the Bayesian systems, and in addition it can use fairly complex heuristics as detectors, that should lead to more accurate matching. For example, the word “free” may not be a good spam detector by itself, but when your system can distinguish between “free software” and “absolutely FREE!!!!” you can sort mail more accurately.

Section 3.2 describes these detectors in more detail. Sections 4.1 and 4.2 describe how the system can be seeded with information that gives it the potential to become more accurate with less training. Section 4.3 describes the fully public data set that has been used for testing and training. Section 5 describes how the detectors are created, trained, and used to score messages.

The original system [6] performed well, but the weighting scheme used to score messages in the original system had a potential flaw. It was not bounded, so there was the potential for a highly-weighted detector to overwhelm the effect of any others on the final score. This paper compares that approach to a percentage-based weighting system for lymphocytes and a score based on a weighted average, with the aim to find a weighting system that achieves better accuracy. The scoring system is described in Section 5.4, and the results follow in Section 6.

## 2 Anti-spam solutions

The solutions available for avoiding spam are fairly diverse. Some popular methods include:

- legislation prohibiting the sending of spam (These laws are becoming more common in the United States [7].)
- disguised email addresses (For example, some people will insert extra strings such as “NOSPAM” into their real email addresses)
- challenge-response systems (where users are challenged and must authenticate themselves before the message is accepted. Typically, they are asked a question that is easy for a human to answer but difficult to create an automated responder for, such as “What is the sum of two and three?”)
- filters such as the spam immune system, which clas-

sify messages based on patterns or words found in them

Filtering solutions often rely on humans to create detectors based on the spam they've received. Humans can create good heuristic detectors, but with spam senders intelligently adapting to anti-spam solutions, it is hard to keep up. A dedicated spam sender can use the frequently publicly available information about such heuristics and their weightings to evade detection.

SpamAssassin [8] relies on a genetic algorithm to weight its heuristics, but does not do this while the program is in use; the weights are updated with each release. To limit the effectiveness of attacks against these heuristics, it uses some that are updated continually. One of these is a Bayesian solution (based on Graham's paper [5]) which uses Bayes Rule to do a probabilistic weighting of individual *tokens*, which are usually single words. Another is the Realtime Blackhole List, which blocks mail servers that have been used to send spam or are known for their spam-friendly policies [9].

At the 2003 Spam Conference, Michael Salib made a presentation that suggested that combining automated learning with more complex heuristics would yield more effective spam classifying results [10]. Most of the learning systems, like the Bayesian systems, work with a fairly narrowly-defined set of detectors. The spam immune system has the potential to do automated weighting of much more complex detectors, which may make it more robust to attacks designed to disrupt the Bayesian systems (for example, it would be possible for spam senders to begin appending large unrelated pieces of text to the spam message so that the Bayesian system must weight more tokens and the final scoring will be thrown off by the unrelated text.)

## 3 The Spam AIS

### 3.1 Self and non-self

An immune system's main goal is to distinguish between self and potentially dangerous non-self elements. In a biological system, these non-self elements (known collectively as *pathogens*) include bacteria and viruses. In a spam immune system, we want to distinguish legitimate messages from spam. Like biological pathogens, spam comes in a variety of forms and some pathogens will only be slight variations (mutations) of others.

A biological immune system has an advantage when it comes to distinguishing self from non-self, however, since a biological self does not change in ways that matter to the immune system. The surface proteins used by the immune system to distinguish self do not change over time. Unfortunately, the spam immune system has the same problem as a computer security immune system [11]: the self changes over time. The content of a person's legitimate mails will change over time as they meet new friends and business contacts, develop new interests, discuss current issues, maybe even learn new languages, etc.

This does not mean the immune system model cannot be used to build a spam detector, only that the model must be used with some caution. It does indicate a need for the system to forget as well as learn things, since features that previously indicated spam could begin to indicate non-spam. For example, a monolingual English speaker might get spam in other languages and be able to easily filter based on the character set of the message, but if that person then learns Japanese and begins communicating in that language, the system must be able to adapt accordingly to avoid an *auto-immune* reaction (where the immune system reacts to self instead of non-self).

### 3.2 Detectors

In the biological system, specialized white blood cells called *lymphocytes* are created to detect and destroy pathogens. Each lymphocyte has a detector (called an *antibody*), or rather a set of copies of the same antibody. The antibodies are created through random recombination of a library of genes.

Lymphocytes detect pathogens by binding to their surface proteins (called *antigens*). This binding is approximate: one lymphocyte's antibodies may bind to many different antigens, although some will bind more closely than others.

We consider the text of the email (both the headers and the body) as the antigen of a spam message. Approximate binding is then simulated by the spam system by using *regular expressions* (patterns that can match a variety of strings) as antibodies.

One benefit to an immune system based spam detector is that each system will have different detectors, making it much more difficult to create a message that will defeat many systems at once. A system such as SpamAssassin [8], which uses a set of pre-weighted heuristics (Does the message contain a "click here" link? Was the message sent by a known mail program?), can be circumvented by a dedicated sender who carefully avoids the heuristics being used by all copies of that version of the software. This sort of attack is known to be used by senders who want to make sure that their ads come through. In a simple example, a spam message could use the string "enl4rge" instead of "enlarge" to avoid filters checking for the word. However, if new and different heuristic detectors are being created all the time, then it becomes more difficult for a sender to find a way to consistently avoid detection.

By using regular expression antibodies, the system can weight many possible strings identically. "Enlarge" and "enl4rge" and "enlarg3" are all read the same way by the human recipient of a message, so it makes sense to allow the immune system to treat them as the same string. Further examples can be found in [6].

## 4 Input to the AIS

### 4.1 The library

Although it would be possible to make an electronic library containing every character that could possibly be used in an email message, doing so would waste valuable knowledge we have about the structure of messages. We know that messages (legitimate or spam) usually contain more words than randomly-concatenated letters. And the words used in spam messages usually represent only a subset of written language.

Using a smaller library has advantages for speed, but there are also drawbacks. One of the most significant problems for learning occurs when a message is found that no detector matches. With a library that is not utterly comprehensive, it may be possible that no gene combination could even produce such a detector.

Researchers working with Bayesian-type spam systems have circumvented this problem by creating detectors based upon messages the system sees, and in the future, we hope to take a similar approach.

Initial gene libraries for the spam immune system could be taken from a variety of sources, including:

- Words from one or many languages
- Words found in a collection of messages (spam, non-spam, or both)
- Phrases found in a collection of messages
- Contact information in spam messages. Since many spam messages are attempting to sell something, the telephone numbers and web addresses are often constant even if the rest of the message changes.
- Header information
- Bits of Javascript and HTML code often used in spam.

More accurate filters typically use information from a variety of sources, so it makes sense to use genes from a variety

- Message matching:
    - updating of lymphocytes
    - scoring of messages
- Expiry of old lymphocytes

These phases represent a cycle that is repeated many times over the life span of the artificial immune system.

### 5.1 Generation of antibodies

As in the biological immune system, antibodies are created by random recombination of the genes in the library as described in [6], except that identical antibodies are not allowed in this spam immune system implementation.

### 5.2 Creation of lymphocytes

When a new, unique antibody has been created, it can be inserted into a new lymphocyte. A lymphocyte consists of the information listed in Table **??**.

The completed lymphocyte is then written to a data store. In this case, a database was used.

### 5.3 Training and Application of Lymphocytes

To train the database, a collection of known spam and known non-spam must be obtained. Each lymphocyte is then applied to each message and is updated each time the antibody matches against the message.

When the message being weighted is not already confirmed to be spam or non-spam, we can use weights between zero and one to increment the spam_matched field. For example, a message tagged as spam by another spam detection system might be given a weight increment of 0.8 instead of 1.

The two numbers spam_matched and msg_matched can be used to show what percentage of the time an antibody detects spam. The field msg_matched gives an indication of how often this antibody has been used, which helps determine how important it should be in the final weighting. An antibody that matches with a rate of 100% over a sample of 2 messages is probably not as useful as one that matches with accuracy 80% over a sample of 1000 messages.

Training can be done either with messages known to be spam or non-spam, messages classified by other spam detecting methods, or even messages that have been marked as spam by the system. Lymphocytes do not have to be updated if nothing is known about the message.

### 5.4 Final spam scores

We have investigated two types of final score given to each message: a *straight sum* which is simply a sum of the spam_matched values from all matching lymphocytes, and a *weighted average* which is the sum of the spam_matched values from all matching lymphocytes divided by the sum of all the msg_matched values from all matching lymphocytes.

$$\text{Straight sum} = \sum_{matching\ lymphocytes} spam\_matched$$

$$\text{Weighted average} = \frac{\sum_{matching\ lymphocytes} spam\_matched}{\sum_{matching\ lymphocytes} msg\_matched}$$

The straight sum scoring technique is similar to the method used in [6], only the messages have been weighted with a value of 1 for a spam and 0 for a non-spam rather than positive and negative values. The weighted average allows lymphocytes that have matched more often to have more effect on the final score than those that only match occasionally. It is also worth noting that the weighted sum has bounded results (all results are between 0 and 1, inclusive).

### 5.5 Expiry of old lymphocytes

When each lymphocyte is created, it is given an expiry date. This is typically the current date plus a set increment such as two days. It could also be done based on the number of messages passed through the system, but because the average user will have busier and lighter days for their personal mail, but a fairly constant rate of spam, this gives the lymphocytes a chance to match spam messages even if the user becomes involved in a discussion that increases their average number of messages per day for a short time.

Expiry allows the system to remove old lymphocytes that have never matched any messages so that they do not waste processing time of the system. In addition, this expiry is what allows the system to forget lymphocytes that are no longer in active use. Once old lymphocytes have been removed, new lymphocytes can be generated to replace them, starting the cycle again.

## 6 Results

### 6.1 Initial setup and training

1000 lymphocytes were generated from a library of less than 200 genes. The genes were fairly complex, based on heuristic phrases used for spam detection.

Initial training was done with 1500 spam and 1500 non-spam messages. Once the training was done, only 156 lymphocytes had any weight. Of these, 127 matched only spam, while the others had also matched legitimate messages.

Messages were not expired and new lymphocytes were not generated in these tests, which were only intended to give a comparison between the two final scoring systems for messages.

With more lymphocytes generated initially or with more genes in the original library, the system would be more accurate, but these numbers were chosen because they seemed to give reasonable results with a fairly lightweight system [6].

### 6.2 Detection

The trained lymphocytes were then tested against a collection of 501 non-spam and 401 spam messages. (These num-

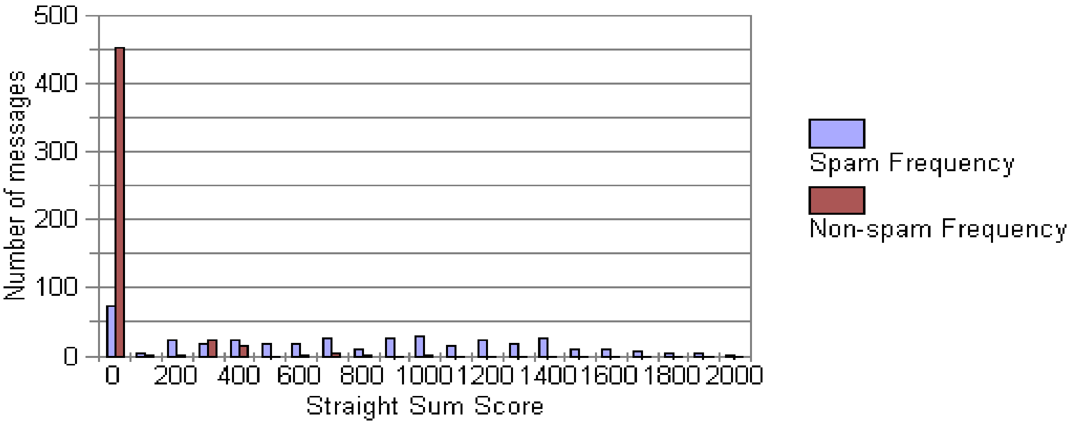

Figure 1: Straight sum scores

bers were chosen to give a fairly large sample and to reflect the percentage of overall mail that is currently estimated to be spam [1].) These trained lymphocytes were then used to score the messages. No further training took place as the messages were scored.

With many detection systems, you can either have the system detect most spam messages or have the system be accurate in its detection, not both. The idea behind many systems, including this one, is that a threshold must be set, with messages on one side of the threshold (typically above) classified as spam, and messages on the other side of this threshold classified as non-spam.

### 6.2.1 Straight Sum

With the threshold set at a 500 for the straight sum, we can correctly classify nearly 99% of non-spam, but then only classify 70% of spam correctly with an overall rate of 86%. Since most people prefer not to risk missing legitimate mail (See Section 6.3), this would probably be the best solution even though better overall spam detection rates could be achieved. More detailed results are given in Figure 1.

The table shows that while the non-spam results are clustered at 0 and no non-spam messages achieve a score of 1000 or higher, the scores for spam are fairly evenly spaced over a very wide range. There is no clear score at which most messages below this value are non-spam and most messages above this value are spam.

### 6.2.2 Weighted Average

The weighted average seems to provide a better balance. With the threshold set at .7, the immune system correctly classifies 90% of the messages correctly. (More specifically, it classifies 84% of spam and 98% of non-spam). More detailed results are given in Figure 2

Unlike in the straight sum, there is a fairly clear threshold at 0.7 where most spam scores are above that value *and* most non-spam scores are below. There is the notable exception of the 49 messages for which no detectors existed. If no detectors exist which match these messages, then the system cannot learn from theses messages. Section 6.4 discusses the relevance of these messages, as well as strategies for handling them in the future.

### 6.3 False positives

False positives (non-spam messages that have been classified incorrectly as spam) are generally considered to be more harmful than false negatives (spam messages incorrectly tagged as non-spam). The reasoning behind this is it is much easier to just delete an extra message than to remember to check your spam filters regularly to make sure no messages were missed.

With the weighted average scoring, fewer than 2% of the non-spam messages were mis-classified. The straight sum approach mis-labeled a slightly smaller percentage, closer to 1% than 2%. SpamAssassin achieves a less than 1% false positive rate on their corpus when using a threshold of 5.0 (their default threshold) [8], so these are not the best rates achieved, but as spam rates soar, they are definitely comparable to the rate achieved by a human detector, who may accidentally delete legitimate messages along with spam messages.

Because neither method is not completely free of false positives, they should not be used in conjunction with something that deletes messages without giving the user time to check them. This is a learning system and functions best if the user can occasionally provide input when a message is mis-classified. As many users would prefer to have the option to verify that the messages have been classified properly, just in case something does get caught, this should not present a problem.

It might also be wise to combine the spam immune system with a whitelisting system to further avoid false positives. A whitelisting system automatically allows known senders (for example, the user's friends, family, business contacts, mailing lists to which the user is subscribed, etc.) to send mail to the user. This could be done separately from the spam immune system, or the system could be "vaccinated" (see Section 4.2) to adjust scores for trusted senders. The disadvantage to this is that if a spam sender can deter-

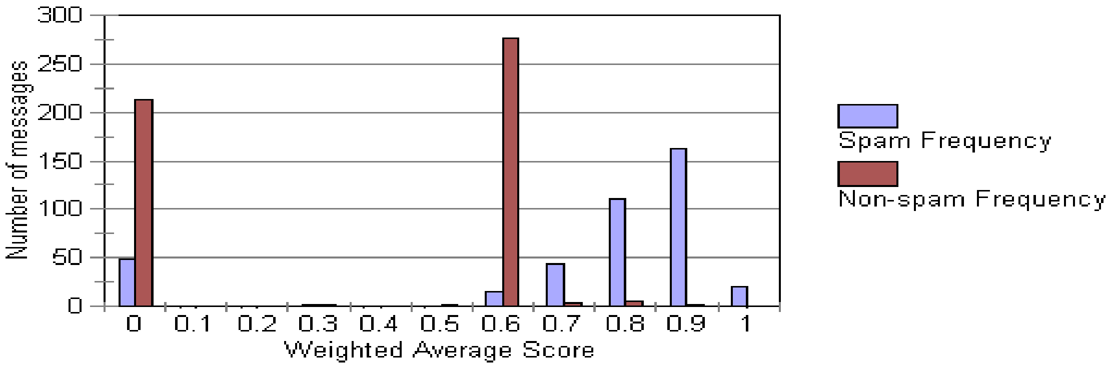

Figure 2: Weighted average scores

mine your whitelist, then they can circumvent the system, but since this is difficult to do and would need to be done for each user, this form of attack is not very common.

### 6.4 False negatives

While the false negative rate is higher than the false positive, for many users it is less essential that this rate be low, since they are able to deal with small quantities of spam and only need an automated filter to make that number more manageable.

Most of these mis-classified messages are actually simply not matched by any lymphocyte in the system. With more genes and more lymphocytes, this rate could probably be reduced considerably for either scoring system, although the effect would be more impressive with the weighted average system, since these unmatched messages are the largest group of the mis-classified messages.

It could be possible to let users know specifically about these unmatched messages and let them create some genes or lymphocytes that could be added to the system so that repeat and similar messages would not be missed in the future. Alternatively, an automated solution could be found to deal with these messages – perhaps the words appearing in this message could be added to a special library and extra lymphocytes could be generated until several were found that matched the original message.

## 7 Conclusions

The weighted average scoring system, with overall accuracy of 90% versus the 86% of the straight sum, is the more accurate of the two weighting systems tried. In addition, the weighted sum approach does not suffer from the problems that occur in a system with unbounded lymphocyte weights.

Although, like the Bayesian systems, the immune system will probably perform best when used with personal mail for one user (because the characteristics of the legitimate mail "self" will vary less wildly), the system did achieve these results with a more wildly varying spam corpus, so it may be possible to run it for an entire mail server as a whole rather than needing to store weightings for each user separately.

The spam immune system model classifes the SpamAssassin spam corpus well, and the overall accuracy of 90% is comparable to the rate achieved by SpamAssassin itself. (SpamAssassin achieves an overall rate of 95% when using a threshold of 5.) Of particular note is that the spam AIS does not need a huge number of detectors. Our immune system created 1000 lymphocytes, less than 200 of which were weighted and used. SpamAssassin uses over 700 complex heuristic tests to achieve similar results. Graham's Bayesian filter boasts over 99% accuracy, but it recognizes over 20, 000 tokens, and the accuracy reflects the classification of the mail of one individual, rather than the more difficult collection found in the SpamAssassin public corpus.

This lighter-weight approach will be attractive in situations where processing time is at a premium, such as large mail servers. SpamAssassin can be difficult to run on a mail server that processes a large volume of mail. A spam immune system like the one described in this paper, with its adaptive/learning capabilities, unique set of detectors and decent accuracy will be an attractive anti-spam solution for organizations seeking a solution that will not over-tax their servers.

## References

[1] S. Atkins, "Size and cost of the problem," in *Proceedings of the Fifty-sixth Internet Engineering Task Force (IETF) Meeting*, (San Francisco, CA), SpamCon Foundation, March 16-21 2003.

[2] J. Weaver, "AOL escalates spam warfare," *MSNBC*, March 5 2003.

[3] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A bayesian approach to filtering junk E-mail," in *Learning for Text Categorization: Papers from the 1998 Workshop*, (Madison, Wisconsin), AAAI Technical Report WS-98-05, 1998.

[4] P. Pantel and D. Lin, "Spamcop: A spam classification & organization program," in *Learning for Text Categorization: Papers from the 1998 Workshop*, (Madison, Wisconsin), AAAI Technical Report WS-98-05, 1998.

[5] P. Graham, “A plan for spam,” August 2002. `http://www.paulgraham.com/spam.html`.

[6] T. Oda and T. White, “Developing an immunity to spam,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2003)*, (Chicago), July 2003.

[7] C. A. U. Email, “Pending legislation,” August 2002. `http://www.cauce.org/legislation`.

[8] SpamAssassin, “SpamAssassin website,” 2002. `http://spamassassin.org/`.

[9] P. Vixie, “MAPS RBL rationale,” July 19 2000. `http://mail-abuse.org/rbl/rationale.html`.

[10] M. Salib, “Heuristics in the blender,” in *Proceedings of the 2003 Spam Conference*, (Cambridge, US), 2003.

[11] S. Forrest, S. A. Hofmeyr, and A. Somayaji, “Computer immunology,” *Communications of the ACM*, vol. 40, no. 10, pp. 88–96, 1997.

[12] SpamAssassin, “SpamAssassin public corpus,” February 28 2003. `http://spamassassin.org/publiccorpus/`.