ISSN 2315-5027; Volume 1, Issue 2, pp. 22-29; June 2012

Online Journal of Physical and Environmental Science Research

©2012 Online Research Journals

Full Length Research

Available Online at http://www.onlineresearchjournals.org/JPESR

Spam Detection Using Artificial Neural Networks (Perceptron Learning Rule)

Owen Kufandirimbwa*[1] and Richard Gotora[2]

[1,2]Department of Computer Science, Faculty of Sciences, University of Zimbabwe, Harare, Zimbabwe.

[1]E-mail: kufandirimbwa@gmail.com; Tel: +263712784287.
[2]Email: rgotora@gmail.com

Downloaded 30 April, 2012 Accepted 30 May, 2012

Spam is email sent in bulk where there is no direct agreement in place between the recipient and the sender to receive email solicitation. To prevent the delivery of this so called spam, an automated tool called a spam filter is used to recognize spam. The circular nature of these definitions along with their appeal to the intent of sender and recipient make them difficult to formalize. The spam problem seems to persist and the current state of the art techniques in fighting this problem seems not to provide full proof. There are several approaches which try to stop or reduce the huge amount of spam on individuals. These approaches include legislative measures such as anti-spam laws over world-wide. Other techniques are known as Origin-Based filters which are based on using network information and IP addresses in order to detect whether a message is spam or not. The most common techniques are the filtering techniques attempting to identify whether a message is spam or not based on the content and other characteristics of the message. In this paper, we present a technique to spam filtering using Artificial Neural Networks, and the perceptron learning rule.

Keywords: Artificial neural network, spam filtering, perceptron learning rule, training algorithm, learning rate.

INTRODUCTION

E-mail is an efficient form of communication that has become widely adopted by both individuals and organizations. Today, more and more people are relying on e-mail to connect them with their friends, family, colleagues, customers and business partners. Unfortunately, as e-mail usage has evolved, so too has its threats, in particular spam, which is also known as unsolicited bulk e-mail or junk mail, has become an increasingly difficult threat to detect and is being delivered in incredibly high volumes. For example according to Message Labs [1], “spam accounted for 67% of all e-mail traffic in October 2006 up from 57% the same time a year before”.

Spam is a serious problem that potentially threatens the existence of e-mail services. In particular, it is now a non-trivial task to find legitimate e-mails in an e-mail inbox cluttered with spam. According to Shiels [2], “spam is also an expensive problem that costs service providers and organizations billions of dollars per year in lost bandwidth”. Further to the bandwidth cost, “it is also estimated that each piece of spam costs an organization one dollar in lost employee productivity” [2]. There are also published reports, which suggest that spam has resulted in lost opportunity costs of several billions of dollars [3] because of organizations that have lost faith in the security industry’s ability to fight this problem.

It is impossible to tell exactly who first come upon a simple idea that if you send out an advertisement to millions of people, then at least one person will react to it no matter what the proposal is. E-mail provides a perfect way to send these millions of advertisements at no cost for the sender, and this unfortunate fact is nowadays extensively exploited by several organizations. As a result, the e-mailboxes of millions of people get cluttered with all this so-called unsolicited bulk e-mail also known as “spam” or “junk mail”. Being incredibly cheap to send, spam causes a lot of trouble to the Internet community: large amounts of spam-traffic between servers cause delays in delivery of legitimate email, people with dial-up Internet access have to spend bandwidth downloading junk mail. Sorting out the unwanted messages takes time and introduces a risk of deleting normal mail by mistake.

Finally, there is quite an amount of pornographic spam that should not be exposed to children.

Spam has the potential ability to become a very serious problem for the internet community. The introduction of new technologies, such as Bayesian filtering, SVM, Artificial Neural Network, Artificial Immune system… etc is improving filter accuracy [4].

Problem Statement

Email is not just text; it has structure. Spam filtering is not just classification; because false positives are so much worse than false negatives that you should treat them as a different kind of error. And the source of error is not just random variation, but a live human spammer working actively to defeat your filter [5].

Spam is a huge problem for all email users, from the casual user, who loses time deleting all the junk mails before reading the legitimate ones, to the large companies which spend millions of dollars yearly trying to combat it. According to Hunt and Carpinter [6], “millions of spam email messages are sent every day, advertising pornographic web sites, drugs or software products, or of fraud (phishing)”. Spam emails have an important economic impact on end users and service providers. The increasing importance of this problem has motivated the development of a set of techniques to fight it, or at least, providing some relief. “Conservative estimates indicate that the total cost of spam on users (worldwide) in 2001 was ₤10 billion a year” [5,6]. Ironport [7], estimate that spam cost US companies alone $10 billion in lost productivity. Summarized below are potential problems that spam brought to users.

It cost you time

By the mere fact that the e-mailboxes of millions of people get cluttered with all this so-called unsolicited bulk e-mail also known as “spam” or “junk mail”, it is now a non-trivial task to find legitimate e-mails in an e-mail inbox cluttered with spam. Deleting such spam messages takes time and it even poses a problem of deleting the legitimate messages by mistake. According to Brad [8], “it doesn't take that long to delete a message, of course, but day in and day out, deleting 30 starts to grind”.

Spam interrupts

For those who use the Internet in business, there is an e-mail window open all the time on the screen and mail comes in and is dealt with all day long, often every few minutes. Many people want to have their terminal beep if they have some new e-mail, because many companies run on their employees being able to quickly send e-mail around, or correspond quickly with customers and suppliers. According to Brad [8], “for those who use e-mail all day, it interrupts them 30 times or more a day. It's almost like a phone ringing randomly 4 times an hour to hear a sales-droid or a hang-up: just a few moments of your time, but very annoying”.

Spam Invades Privacy

Privacy is the right to be left alone when minding your own business. But this is not the case if you go out on the public Internet. Do almost anything in public under your e-mail address and you will quickly get inundated with unwanted e-mail.

“The saddest thing is that ordinary net users have come to realize what is going on, and as a result they start to fear public participation in the net. Many people now don't want to post to a newsgroup or mailing list because they know they will get Spam” [8].

Damage to the "end to end" principle

Another great evil of Spam is that it interferes with the "end to end" principle, an ethos of network design near and dear to the hearts of many of those involved in internet design. This principle says that internet applications should work directly, from user to user, without requiring special smart gateways in the center of the network. End-to-end systems are important in that they scale and foster innovation [8].

Stopping spam, however, requires some central control, a violation of the principle. If everybody can email everybody directly, we get the abuse of spam. Given this, we must try not to let spam make us abandon our principles altogether [8].

The ISPs (Internet Service Providers)

According to Shiels [2], “spam is also an expensive problem that costs service providers and organizations billions of dollars per year in lost bandwidth”. Further to the bandwidth cost, “it is also estimated that each piece of spam costs an organization one dollar in lost employee productivity” [2]. This is because they have to devote staff, in some cases full-time; to dealing with customer complaints over spam and tuning mail software and anti-spam software to deal with the problem. There are also published reports, which suggest that spam has resulted in lost opportunity costs of several billions of dollars [3] because of organizations that have lost faith in the security industry’s ability to fight this problem.

The Research

Current State of the Art Spam Prevention Techniques

There are many available techniques to stop the arrival of spam or junk e-mail. The techniques available generally evolve around using of spam filters. Generally, filters examine various parts of an email message to determine if it is spam or ham. On the basis of the parts of the email messages; filtering systems can be further classified and used for spam detection. Origin or address-based filters typically use network information for spam classification, while content filters examine the actual contents of email messages.

Most of the techniques applied to the problem of spam are useful but the key role in reducing spam email is the content-based filtering. Its success has forced spammers to periodically change their practices, behaviours, and to trick their messages, in order to bypass these kinds of filters. Outlined below are the commonly used techniques used for spam detection:

Legislation: According to Weiss [9],"the goal of legislation is to create an enforceable law, which would make it illegal to send spam and impose heavy penalties for those caught doing so". We consider legislation as an anti-spam solution because imposing legal consequences to spamming should discourage people from spamming and there are those who believe that no technological solution will solve the spam problem.

In the United States, anti-spam legislation called the "CAN-SPAM" act [10] was introduced to help combat the spam problem. According to the CAN-SPAM act, "in order to send unsolicited commercial e-mails, the e-mail must include, among other things, a means for the end user to opt-out of the distribution, a postal address where the user can contact the sender, true memo headers and a subject line header that does not mislead the end user as to what the content of the e-mail contains". In addition, "all sexually explicit email content needs to be sent with a label indicating such content in the subject line". However, the CAN-SPAM act and other legislation methods may have promoted spam by legalizing certain types of spam. Tracing spam to the originating source will often lead to the innocent hi-jacked owner and not the spammer themselves. Furthermore, the distributed nature of the Internet makes it easy for people to send spam from anywhere around the world. Therefore, anti-spam laws have the effect of forcing spammers send spam from jurisdictions where laws are not in force.

Pre-emption: The goal of pre-emption is to prevent non-dedicated mail servers from becoming spam sources. "According to Sauver [11], "this is done using mass educational campaigns with the goal of educating end users on how to secure their computers and prevent them from becoming spam sources". The goal of such campaign is to make end users aware of the risks involved in not applying security patches, updating anti-virus signatures, opening suspicious ***e-mail*** attachments, downloading spyware infested peer-to-peer programs, and clicking on malicious links obtained via ***e-mail*** or instant messaging. Unfortunately, "user education can only go so far and no matter how much effort and money is invested in such campaigns; the method is not fool proof "[12]. So while there is value in user education, the battle against spam zombies is unlikely to be fully resolved by this method alone. Port 25 filtering is another pre-emption technique that aims at many Internet Service Providers and corporations blocking outbound port 25 accesses on all hosts on their network except those that are explicitly allowed to perform SMTP relay functions. Although this is a simple yet effective means of stopping spam, it has its shortcomings. Specifically, not all service providers or corporations have adopted this policy since their users do not wish to have such blocks implemented. It should also be noted that the port 25 blocks does not prevent computers from becoming zombies. Rather it prevents zombie computers from being able to send spam. As such, an attacker will still be able to control the zombie computer and use it for other purposes such as storing illegal software or scanning the network for other machines to infect.

Filtering: The goal of anti-spam filtering is to filter messages by identifying which messages are likely to be spam and which are not. Anti-spam filtering typically occurs towards the end of the SMTP transaction. According to Ahmed and Mithun [13] and Crawford *et al*. [14], "content filtering was one of the first types of anti-spam filters to be used". "Such filters use hard coded rules where each rule has an associated score and is periodically updated". [13,14] An example of such filter is Spam Assassin [15], which works by scanning the textual content of the e-mail against each rule and adds the scores for all matching rules. According to Stern [16], "if the total score of the e-mail exceeds some set threshold score then the message is considered spam. In order to generate these scores, a single perceptron is used where the inputs to the perceptron indicate whether a particular rule was matched and the weight for the corresponding input indicates the score for each rule". There are several popular content filters such as Bayesian filters, Rule Based Filters, Support Vector Machines (SVM) and Artificial Neural Network (ANN).

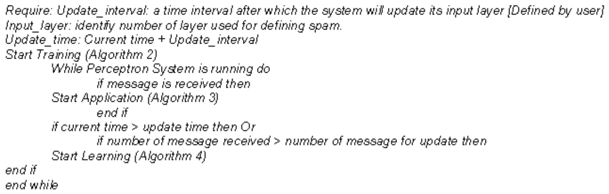
An alternate approach to content based filtering is network level filtering, which typically comprises of a blacklist. A blacklist comprises of known bad IP addresses that have been used to send spam. A mail server uses a blacklist when it receives a connection from a sending server. Specifically, the receiving server checks the sending server's IP address against the blacklist to determine if the sending server's IP address is listed. If it is listed then the mail is rejected. Machine learning techniques are also used for spam filtering. Machine learning is the development of algorithms and techniques, which has the capability to learn and adapt. It is a wide area of Artificial Intelligence (AI). There are several machine learning and text classification techniques which are currently available and under research; Bayesian classification, Support Vector Machine (SVM), digest-based filters, Artificial Neural Network and Artificial Immune System. One of the most interesting new techniques is artificial immune system and artificial neural network (ANN).

RESEARCH METHODS

The Artificial Neural Network Algorithm

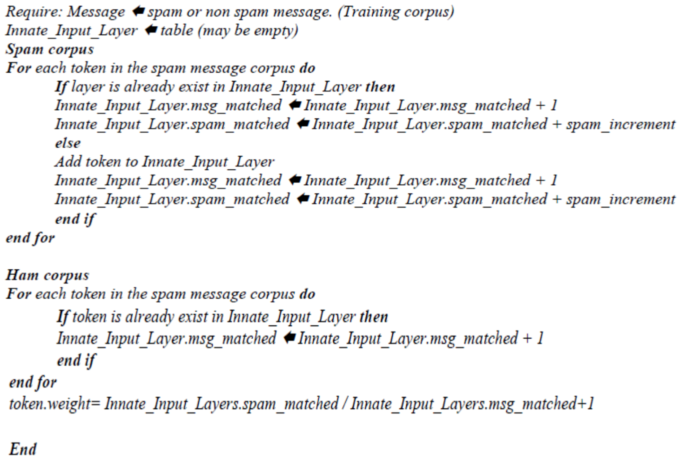
We designed the artificial neural network spam detector using the perceptron learning rule with the algorithm design below. The algorithms were adapted from Alia et al. [4].

Algorithm A: Perceptron Algorithm



```
Require: Update_interval: a time interval after which the system will update its input layer [Defined by user]
Input_layer: identify number of layer used for defining spam.
Update_time: Current time + Update_interval
Start Training (Algorithm 2)
        While Perceptron System is running do
                if message is received then
        Start Application (Algorithm 3)
                end if
        if current time > update time then Or
                if number of message received > number of message for update then
        Start Learning (Algorithm 4)
end if
end while
```

Algorithm B: Training Algorithm



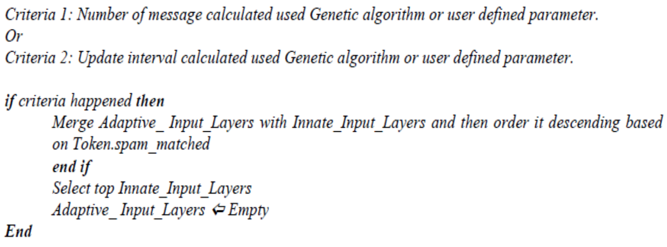
```
Require: Message ⬅ spam or non spam message. (Training corpus)
Innate_Input_Layer ⬅ table (may be empty)
Spam corpus
For each token in the spam message corpus do
        If layer is already exist in Innate_Input_Layer then
        Innate_Input_Layer.msg_matched ⬅ Innate_Input_Layer.msg_matched + 1
        Innate_Input_Layer.spam_matched ⬅ Innate_Input_Layer.spam_matched + spam_increment
        else
        Add token to Innate_Input_Layer
        Innate_Input_Layer.msg_matched ⬅ Innate_Input_Layer.msg_matched + 1
        Innate_Input_Layer.spam_matched ⬅ Innate_Input_Layer.spam_matched + spam_increment
        end if
end for

Ham corpus
For each token in the spam message corpus do
        If token is already exist in Innate_Input_Layer then
        Innate_Input_Layer.msg_matched ⬅ Innate_Input_Layer.msg_matched + 1
        end if
end for
token.weight= Innate_Input_Layers.spam_matched / Innate_Input_Layers.msg_matched+1

End
```

Algorithm C: Learning Algorithm

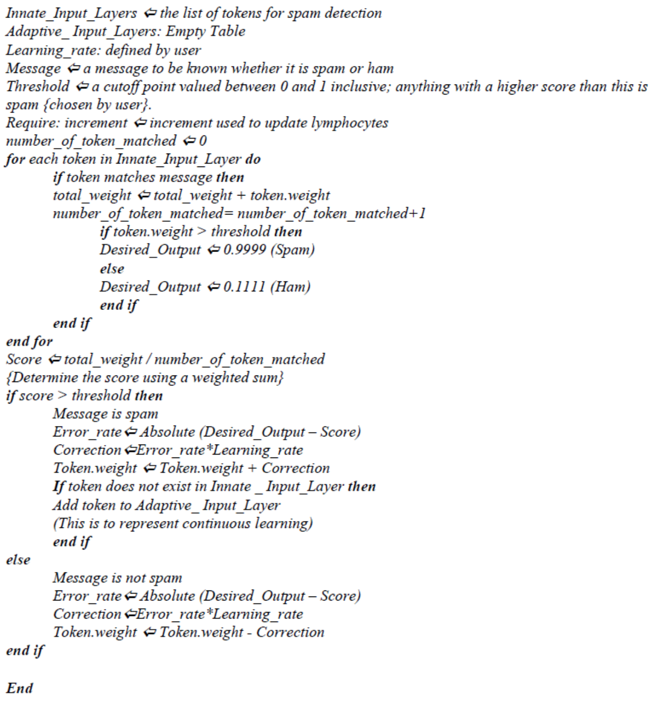
This algorithm is used to delete old input layer and replace it with a new promising input layer. Delete can happen based on the following criterion:



```
Criteria 1: Number of message calculated used Genetic algorithm or user defined parameter.
Or
Criteria 2: Update interval calculated used Genetic algorithm or user defined parameter.

if criteria happened then
        Merge Adaptive_ Input_Layers with Innate_Input_Layers and then order it descending based
        on Token.spam_matched
        end if
        Select top Innate_Input_Layers
        Adaptive_ Input_Layers ⇦ Empty
End
```

Algorithm D: Application Algorithm



```
Innate_Input_Layers ⇦ the list of tokens for spam detection
Adaptive_ Input_Layers: Empty Table
Learning_rate: defined by user
Message ⇦ a message to be known whether it is spam or ham
Threshold ⇦ a cutoff point valued between 0 and 1 inclusive; anything with a higher score than this is
spam {chosen by user}.
Require: increment ⇦ increment used to update lymphocytes
number_of_token_matched ⇦ 0
for each token in Innate_Input_Layer do
        if token matches message then
        total_weight ⇦ total_weight + token.weight
        number_of_token_matched= number_of_token_matched+1
                if token.weight > threshold then
                Desired_Output ⇦ 0.9999 (Spam)
                else
                Desired_Output ⇦ 0.1111 (Ham)
                end if
        end if
end for
Score ⇦ total_weight / number_of_token_matched
{Determine the score using a weighted sum}
if score > threshold then
        Message is spam
        Error_rate ⇦ Absolute (Desired_Output – Score)
        Correction ⇦Error_rate*Learning_rate
        Token.weight ⇦ Token.weight + Correction
        If token does not exist in Innate _ Input_Layer then
        Add token to Adaptive_ Input_Layer
        (This is to represent continuous learning)
        end if
else
        Message is not spam
        Error_rate ⇦ Absolute (Desired_Output – Score)
        Correction ⇦Error_rate*Learning_rate
        Token.weight ⇦ Token.weight - Correction
end if

End
```

All algorithms were adapted from Sabri et al. [4]

The Experiment

This part of the research outlines and examines the proper experimentation, training and testing that was performed on our perceptron.

Training Phase

Our perceptron employs a stochastic gradient method for training, where the true gradient is evaluated on a single training example and the weights adjusted accordingly until a stopping condition is met. In other words, for each training value, the perceptron continuously uses said value as its input until it either generates the desired output or reaches a pre-specified maximum number of iterations. Such pre-specified maximum number of iterations were determined by experimentation and described in greater detail later.

At each iteration, an error and weight adjustment value are computed by comparing the actual output value with the expected output value. Once the weight adjustment value is computed, the actual weight values are then adjusted. Upon updating the weights, the same parsed message is used as input to the perceptron and the corresponding output is computed. If the actual output and the expected output do not equal then the weights are adjusted again. This process continuously iterates until the algorithm is able to generate an output equal to the expected output or the maximum number of iterations is reached. An error value is subsequently computed by

Table 1. Determining Perceptron Learning Parameters

Perceptron Learning Rate	Perceptron Maximum Iterations	Emails Blocked
0.8	1 000	7
0.5	1 000	4
0.2	1 000	6
0.8	10 000	8
0.5	10 000	5
0.2	10 000	4
0.8	100 000	3
0.5	100 000	2
0.2	100 000	1

Table 2. Spam Precision and Recall Results

Number of Iterations	Spam Precision	Spam Recall
200	95.325%	69.355%
300	94.022%	77.859%
400	96.673%	65.396%
500	96.969%	70.381%
600	96.586%	71.554%
700	96.918%	67.009%
800	96.569%	70.694%
900	97.149%	68.328%
1000	96.765%	70.381%

using the sum of squares error equation (Mean Square Output Error), which is presented as equation 1 [17]:

$$E(\omega)=\frac{1}{2}\sum_{i=1}^{n} \Sigma_{e}(y_{e}-g(x_{i}))^{2} \qquad (1)$$

Adapted from Duda and Hart [17].

In training phase each e-mail message was treated as a text file, and then parsed to identify each header information (such as From: Received: Subject: or To:) to distinguish them from the body of the message. After that, every substring within the subject header and the message body delimited by white space was considered to be a token. In training phase we use (45) spam and (50) ham messages from the Spam Assassin public corpus.

Determining Parameter Values

We conducted experiments in our test environment to determine the preferred learning rate and maximum iteration number that we should use for our perceptron. We conducted our experiment using a combination of three distinct learning rates and three distinct maximum iteration counts.

Testing Phase

This was done by subjecting the ANN to messages that were not used in training without adjusting the weights. Dataset, used for testing, consists of (25) spam and (20) ham.

RESULTS AND DISCUSSION

This section outlines and discusses the results of our experiment. The data given in tables 1 to 4 shows what come out of the experiment as the iterations were varied. From table 1, it can be observed that a larger number of iterations resulted in better overall blocking rates. Using a high number of maximum iteration counts requires further processing for each message parsed, which means less processing for additional transactions. With regard to the learning rate it was observed that a higher learning rate combined with a smaller maximum iteration count

Table 3. Ham Precision and Recall Result

Number of Iterations	Ham Precision	Ham Recall
200	94.244%	98.622%
300	95.763%	98.360%
400	93.561%	98.825%
500	94.438%	98.854%
600	94.644%	98.796%
700	93.844%	98.854%
800	94.487%	98.796%
900	94.077%	98.884%
1000	94.437%	98.825%

Table 4. False Positive, False Negative, Total Error

No of Iterations	False Positive	False Negative	Total Error
200	0.316%	5.077%	5.392%
300	0.534%	3.668%	4.202%
400	0.146%	5.732%	5.878%
500	0.121%	4.906%	5.028%
600	0.170%	4.712%	4.712%
700	0.121%	5.465%	5.587%
800	0.170%	4.858%	5.028%
900	0.097%.	5.247%	5.344%
1000	0.146%	4,906%	5.052%

resulted in our perceptron module blocking more connections. A learning rate of 0.8 with a maximum iteration count of 1,000 produced 3 more perceptron-based blocks than a learning rate of 0.5 with the same maximum iteration count. Opted to use a perceptron with a learning rate of 0.8 and a maximum iteration count of 10,000.

Tables 2 to 4 shows that our perceptron give promising results that could be used in the process of fighting against spam. We have an accepted false positive value when the number of the iterations is 300. The best false positive value is found when the number of iterations is 900 (table 2).

Analysis of Results

Table 2 and 3 shows that our perceptron gives promising results that could be used in the process of fighting against spam. We have an accepted false positive value when the number of iterations is 300. The best false positive value is found when the number of iterations is 900.

Depending on the results in table 4, we find that we get a very low false positive rate which is very acceptable. On the other hand we get tolerable false negative rates. According to figures 1 to 3, we can see that we have the best results when the number of iterations used for spam detection is 300. Also, we get a good result when the number of iterations is 600. The modification on ANN gives excellent results. We get promising values when the number of iterations is 300. We know that if you have a system which can achieve such results with this low number of iterations this means that we will have a perfect performance which is amazing since we always look for a high performance with a minimum CPU usage.

CONCLUSION AND FUTURE WORK

In this research, we presented a new technique for filtering spam that cannot be easily overcome by spammers. The technique consisted of a single perceptron that was designed to learn and distinguish legitimate and illegitimate sending server parameter values and messages. The perceptron algorithm due to the incorporation of a continuous learning feature also produces favourable detection rates.

As future work, the researchers intend to:

. Implement the perceptron algorithm in a filtering server so as to enable the perceptron to block server identification

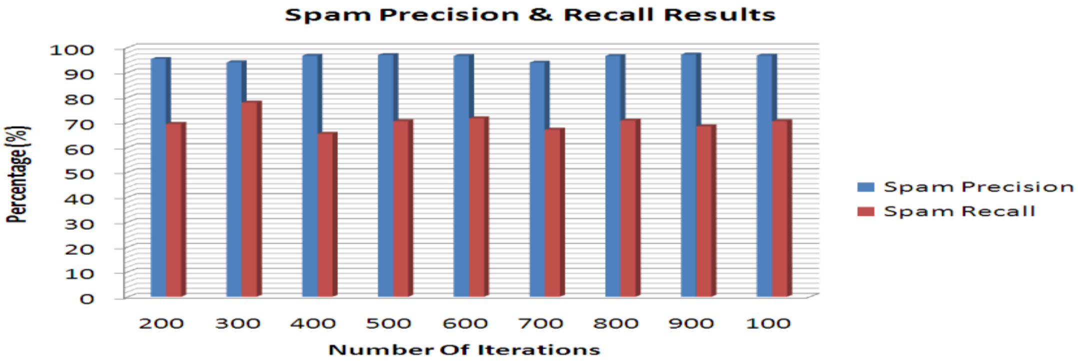


Figure 1. Spam Precision and Recall (Perceptron).

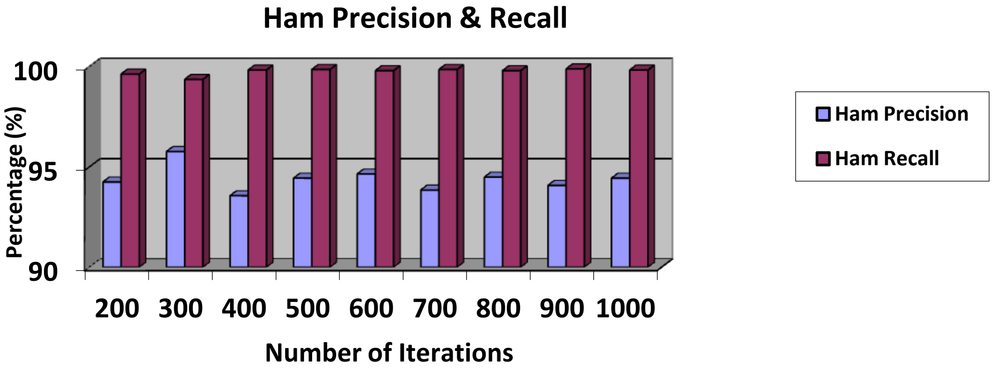


Figure 2. Ham Precision and Recall Result (Perceptron).

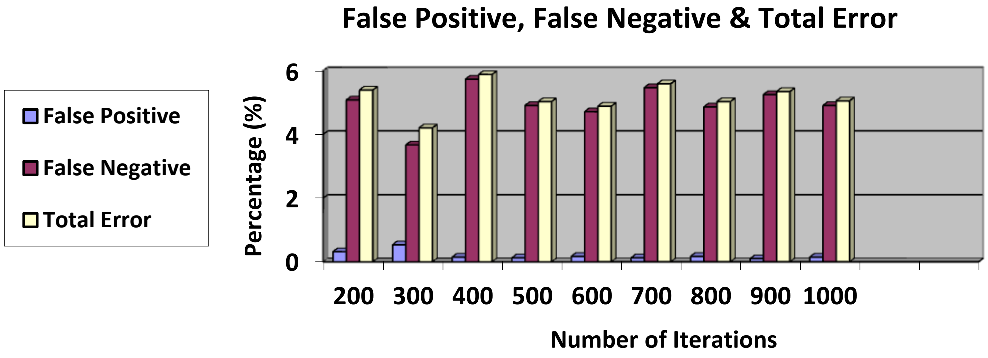


Figure 3. False Positive, False Negative, Total Error (Perceptron).

values such as IP addresses that will be listed in the Spamhaus' blacklist

. Conduct further performance tests on my filtering technique to determine the preferred activation function, learning rate and maximum iteration count required for my technique to achieve higher blocking statistics.

. Observe the filtering technique over a longer period of time

. Incorporate the use of Mail filtering server logs to track and record various events on each mail-filtering server at each filtering level

REFERENCES

[1] MessageLabs. Threat Statistics. December 2006, http://www.messagelabs.com/Threat_Watch/Threat_Statistics/.

[2] Shiels M. Why one spam could cost $50. BBC News, April 2002. http://news.bbc.co.uk/1/hi/sci/tech/1917458.stm.

[3] Coalition against Unsolicited Bulk Email, Australia. Microeconomics of Spam and Direct Marketing. December 2006. http://www.caube.org.au/microec.htm.

[4] Sabri AT, Adel. Mohammads H, Al-Shargabi B, Hamdeh MA. Developing New Continuous Learning Approach for Spam, Detection using Artificial Neural Network (CLA_ANN). Eur. J. Sci. Res. 2010; 42(3): 525-535

[5] Paul G. Better Bayesian Filtering, http://www.paulgraham.com/better.html.

[6] Hunt R, Carpinter J. Current and New Developments in Spam Filtering. in Proceedings of 14th IEEE International Conference on Networks. 2006; 2: 1-6.

[7] Iron port. Spammers Continue Innovation: Iron Port Study Shows Image-based Spam, Hit and Run, and Increased Volumes Latest Threat to Your Inbox", 2006. http://www.ironport.com/company/ironport_pr_2006-06-28.html.

[8] Brad T. The Insidious Evil of Spam. http://www.templetons.com/brad/spam/evil.html.

[9] Weiss A. Ending spam's free ride, networker. 2003; 7: 18-24.

[10] U.S. Senate and House of Representatives, Controlling the assault of non-solicited pornography and marketing act of 2003. S. 877.

[11] St. Sauver J. Spam Zombies and Inbound Flows to Compromised Customer Systems," MAAWG General Meeting, 2005. http://www.uoregon.edu/~joe/zombies.pdf.

[12] Gorling S. The myth of user education. In Proceedings of the 16th Virus Bulletin International Conference, 2006; pp. 11-13.

[13] Ahmed S, Mithun F. Word stemming to enhance spam filtering," in Proceedings of the First Conference on Email and Anti-Spam (CEAS), 2004. http://www.ceas.cc/papers-2004/167.pdf.

[14] Crawford E, Kay J, McCreath E. "Automatic induction of rules for e-mail classification," in Sixth Australian Document Computing Symposium, Coffs Harbour, Australia, 2001.

[15] SpamAssassin. "The Apache SpamAssassin Project," 2006, http://spamassassin.apache.org/.

[16] Stern H. Fast Spam Assassin Score Learning Tool", 2004. http://search.cpan.org/src/PARKER/Mail-SpamAssassin-3.0.3/masses/README.perceptron.

[17] Duda R and Hart P. Pattern classification and scene analysis. New York: John Wiley and Sons. 1973