European Journal of Scientific Research
ISSN 1450-216X Vol.42 No.3 (2010), pp.511-521
© EuroJournals Publishing, Inc. 2010
http://www.eurojournals.com/ejsr.htm

Developing New Continuous Learning Approach for Spam Detection using Artificial Neural Network (CLA_ANN)

Alia Taha Sabri
Computer Science Department, Jerash Private University, Amman-Jordan
E-mail: aliasabri@hotmail.com

Adel Hamdan Mohammads
Computer Information System Department, Applied Science University, Amman-Jordan
E-mail: a_hamdan@asu.edu.jo or Adel_hamdan@yahoo.com
Tel: +962-785557100

Bassam Al-Shargabi
Computer Information System Department, Middle East University, Amman-Jordan
E-mail: bshargabi@meu.edu.jo

Maher Abu Hamdeh
Computer Information System Department, the Arab Academy for Banking & Financial Sciences
E-mail: maher.abu.hamdeh@orange.jo

Abstract

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 apply core odifications on ANN in the input layers which allow the input layers to be changed over time and to replace useless layers with new promising layers which give us promising results. We call our approach Continuous Learning Approach Artificial Neural Network CLA_ANN. This work is evaluated using SpamAssassin Corpus which is rarely used with ANN.

Keywords: Artificial Neural Network, Spam detection, Email classification.

1. Introduction

Nowadays; Spam has the potential ability to become a very serious problem for the internet community; Anti-spam vendors offer a wide array of products designed to help us to keep spam out; They are implemented in various ways (software, hardware) , several techniques (content, rule-based) and at various levels (server and user). The introduction of new technologies, such as Bayesian filtering, SVM, Artificial Neural Network, Artificial Immune system… etc is improving filter accuracy. The implementation of machine learning algorithms is likely to represent the next step in the continuous fight to reclaim our inboxes [1]. This paper presents core modifications on ANN by enabling the system to replace old layers with new ones; This is done by introducing a new concept which we call adaptive input layers; These adaptive input layers have the ability to learn from any new spam messages entered the system. The system will add all new tokens (layers) to the adaptive input layers, and the system will then replace old and useless input layers with promising ones from adaptive input layers.

2. Impact of Spam

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 euros yearly trying to combat it. Millions of spam email messages are sent every day, advertising pornographic web sites, drugs or software products, or of fraud (phishing) [2]. 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 [3, 4]. Other reports estimate that spam cost US companies alone $10 billion in lost productivity [3, 5].

3. Related Works

James Clark [6] in his paper presents a neural network based system for automated e-mail classification. Also, He presents LINGER; Linger is a NN-based system used for automatic e-mail categorization problem. Although LINGER was tested in the domain of email classification, Linger is a generic architecture for all kinds of Text Categorization (TC). LINGER is flexible, adaptable and uses configurable options for most of its operations. He has shown that NNs can successfully be used for automated e-mail filing into mailboxes and spam mail filtering.

Levent Ozgur [7]; proposes anti-spam filtering methods for agglutinative languages in general and for Turkish in particular. His methods are dynamic and based on Artificial Neural Networks (ANN) and Bayesian Networks. The developed algorithms used by Levent are user-specific and they adjust themselves with the characteristics of the incoming e-mails. . In the experiments, a total of 750 e-mails (410 spam and 340 normal) were used and a success rate of about 90% was achieved.

Ian Stuart [8] in his research uses a neural network approach on a corpus of e-mail messages from a one user. The feature set used to define spam messages is descriptive characteristics of words and messages similar to those that a reader would use to identify spam. The project used a corpus of 1654 of e-mails received by one of the authors over a period of several months. He notes that the NN required fewer features to achieve results similar to the Naïve Bayesian Approach.

D. Puniškis [9] in his research applied the neural network approach to the classification of spam. His method employs attributes composed of descriptive characteristics of the evasive patterns that spammers employ rather than using the context or frequency of keywords in the message. The data used is corpus of 2788 legitimate and 1812 spam emails received during a period of several months. The result shows that ANN is good and ANN is not suitable for using alone as a spam filtering tool.

Duhong Chen [10] in his research implements the back propagation ANN algorithm. The Back propagation algorithm is the most commonly used ANN machine learning technique. They create a neural network with 3 layers: The first layer is the input layer with a node number which is equal to the number of frequent words plus one. The second layer is the hidden layer with half number of the number of the input node. And the third layer is the output layer with a single node. The project makes use of the test corpus from http://spamassassin.org/publiccorpus. This archive contains 1,896 spam and 4,149 legitimate messages.

- 1028 legitimate emails are used for training; 297 legitimate emails are used for testing

- 1223 Spam mail are used for training; 319 Spam Emails are used for testing.

Number of frequent words used =50.The result is 99.68 Spam Recall, 93.25 Spam Precision, and 96.1 Accuracy. Number of frequent words used =100.The result is 99.69 Spam Recall, 96.65 Spam Precision, and 98.05 Accuracy. Number of frequent words used =150.The result is 99.69 Spam Recall, 97.55 Spam Precision, and 98.54 Accuracy.

Julia Itskevitch [11] in her master thesis demonstrates the problem of term dependency by building an associative classifier called Classification using Cohesion and Multiple Association Rules (COMAR). The main advantages of the COMAR classifier are using multiple association rules to classify each new case and employing a deep rule pruning that results in much lower running time. The studies show that the hierarchical associative classifier that utilizes phrases, multiple rules and deep rule pruning and uses biased confidence or rule cohesion for rule ranking achieves higher accuracy and is more efficient than other associative classifiers and is also more accurate than Naive Bayes.

Sarah Jane Delany [12] in her PhD thesis talks about Using Case-Based Reasoning for Spam Filtering. In her thesis she presents Email Classification Using Examples (ECUE). Her contribution in the PhD thesis was a content based approach to spam filtering that can handle the concept drift inherent in spam email. She used the machine learning method of case-based reasoning which models the emails as cases in a knowledge-base or case-base. Her approach used in ECUE involves two components; A case-base editing stage and a case-base update policy. She presents a new method for a case-base editing named Competence-Based Editing which uses the competence properties of the cases in the case-base to determine which cases are harmful to the predictive power of the case-base and should be removed. The update policy allows new examples of spam and legitimate emails to be added to the case-base as they are encountered allowing ECUE to track the concept drift. Evaluations showed that ECUE was successful at filtering mail at a personal level, identifying between 92% and 94% of spam correctly with less than 1.0% false positives.

Gabriel R. Weinberg [13] in his master thesis shows that the Spam problem is a complex problem. To deal with this complex problem we should develop different strategies. Such strategies must contain both technical and legal realities simultaneously in order to be successful. In his thesis, he builds a model of the system surrounding the spam problem in the form of a casual loop diagram. His diagram shows the casual interactions between the various technical, legal, social, and economic forces presented in the spam problem system. Based on his diagram, he identifies a number of places that solutions could interact with this system. These places comprise a set of possible levers that could be pulled to lighten the spam problem.

This set of levers, developed in his thesis, is used to make sense of the attempted and suggested solutions to date. Various solutions are grouped according to the way in which they interact with the system. These solutions categories are then presented in details by showing, diagrammatically, how they positively and negatively affect the spam system through their interactions with it.

4. Current Solutions

There are many available techniques to stop the arrival of spam or junk e-mail. The techniques available generally move around using of spam filters. Generally, filters examine various parts of an email message to determine if it is spam or not. On the bases 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 [12, 13, 14, 15, 16, 17].

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, behaviors, and to trick their messages, in order to bypass these kinds of filters [18].

4.1. Origin-Based Filters

Origin based filters are methods which based on using network information in order to detect whether it is spam or not. IP and the email address are the most important pieces of network information used. There are several major types of origin-Based filters such as Blacklists, Whitelists, and Challenge/Response systems [3].

4.2. Content Filtering

While Origin-based filters such as: Blacklists and Whitelists examine network information and email headers to determine whether a message is spam or not. Content filters examine the message contents to determine whether it is spam or ham (legitimate). Content based filters try to read the text in order to examine its content. Filters which use this technique are called Keyword-Based Filters. There are several popular content filters such as: Bayesian filters, Rule Based Filters, Support Vector Machines (SVM) and Artificial Neural Network (ANN) [3, 12].

4.2.1. Bayesian Filters

The most well known commercial machine learning approaches used to spam filtering is the use of Naive Bayes classifiers. Naive Bayes classifiers are a probabilistic classifier. Briefly, it calculates and uses the probability of certain words/phrases occurring in the known examples (messages) to categorize new examples (messages). Naive Bayes has been shown to be very successful at categorizing text documents [12].

4.2.2. Support Vector Machine SVM

Support Vector Machines (SVM) have successes at using as classifying text documents [12, 19, 20, 21]. SVM has prompted significant researches into applying them to spam filtering [12, 22, 23]. SVMs are kernel methods whose central idea is to embed the data representing the text documents into a vector space where linear algebra and geometry can be performed [12, 24]. SVMs attempt to construct a linear separation between two classes in this vector space.

4.2.3. Rule- Based Filters

Rule-Based uses a set of rules on the words included in the entire message (Header, Subject, and Body) to determine whether the message is spam or not. Rule based filters were the most common method for spam detection until 2002, when Bayesian filters became more popular [25]. The limitation of Rule-Based filtering is a rule set which is very large and static and causes less performance. The spammers can easily defeat these filters by word obfuscation, for example, the word “Free” could be modified to be F*R*E*E so it will pass the filters [3, 12].

4.3. Machine Learning 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 [22] artificial Neural Network [27] and Artificial Immune System. One of the most interesting new techniques is artificial immune system and artificial neural network ANN (section 4.3.1).

4.3.1. Artificial Neural Network

By definition, a “neural network” is a collection of interconnected nodes or neurons. The best-known example is the human brain; The most complex and sophisticated neural network [28].

The term neural network has moved round a large class of models and learning methods. The main idea is to extract linear combinations of the inputs and derived features from input and then model the target as a nonlinear function of these features. Neural networks find applications in many different fields [29].

Artificial Neural Network (ANN) is a large class of algorithms that has the capability of classification, regression and density estimation [30]. Neural network is composed of a complex set of functions that have the ability to be decomposed into smaller parts (neurons, processing unit) and represent graphically as a network of these neurons. There are two classical types of neural networks that are most often used when the term ANN is used. The Perceptron and the Multilayer Perceptron. This paper presents the Perceptron Algorithm with core modifications which let the system replace the input layers with new ones.

5. CLA Artificial Neural Network

In CLA_ANN we develop the following perceptron learning algorithm approach.

```
Algorithm 1 CLA_ANN:
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 CLA_ANN 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 2 (Training): creation of input layer
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 3 Application:
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
```

```
Algorithm 4 (Learning) Delete Old input layer and replace it with anew promising input layer
Delete can happened based on
```

```
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
```

5.1. How ANN Works

The CLA_ANN works as the following:

- When a message is received, the system treats it as a text file. Then, the system will look in the innate input layers to serach for any matched token. Depending on the weight of each token , the system will calculate the score for each received message.
- The system will compare the score with a thresold value , if the score is greater than the thresold value then the message will be considered as spam. Otherwise, it will be considered as ham.
- If the message is spam, then the system will add each new token to the adaptive input layers to be used in future (learning).
- The system will use a user define value to determine when to remove old layers and replace them with new adaptive input layers. This method will guarantee that there is a continuous learning.
- When the system delete useless innate input layers ,they will be replaced with new adaptive input layers which means that there is a continous learning and the system will remain effective all the time

5.1.2. The Training Phase

We start preparing the input layers where we create the library which will be used then to generate the input layers. This library contains thousands of input layers but we notice that most of these input layers are useless since there are rubbish words(more than 20 characters, address of sites,….etc) So, in order to decrease the huge number of input layers in the library; All layers with spam_matched=1 and Message_matched=1 are deleted. The useful input layer is one which has matched a large number of spam messages.

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 (1075) spam and (710) ham for training. Also, In the training phase we evaluate the system several time each one with different input layers (see section 5.3.1).

5.1.3. Testing ANN

This is done by subjecting ANN to input layers that were not used in training without adjusting the weights. Dataset, used for testing, consists of (682) spam and (3435) ham.

Table 1, 2, 3 shows that our modifications on CLA_ANN 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 input layers is 300. The best false positive value is found when the number of input layers is 700.

Table 1: Spam Precision, Recall Result

No Of Input Layers	Spam Precision	Spam Recall
200	97.325%	69.355%
300	96.022%	77.859%
400	98.673%	65.396%
500	98.969%	70.381%
600	98.586%	71.554%
700	98.918%	67.009%
800	98.569%	70.674%
900	99.149%	68.328%
1000	98.765%	70.381%

Figure 1: Spam Precision, Recall Result (CLA_ANN)

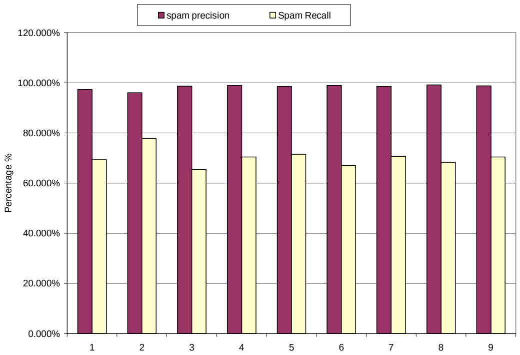


Table 2: Ham Precision, Recall Result

No Of Input Layers	Ham precision	Ham Recall
200	94.244%	99.622%
300	95.763%	99.360%
400	93.561%	99.825%
500	94.438%	99.854%
600	94.644%	99.796%
700	93.844%	99.854%
800	94.487%	99.796%
900	94.077%	99.884%
1000	94.437%	99.825%

Figure 2: Ham Precision, Recall Result (CLA_ANN)

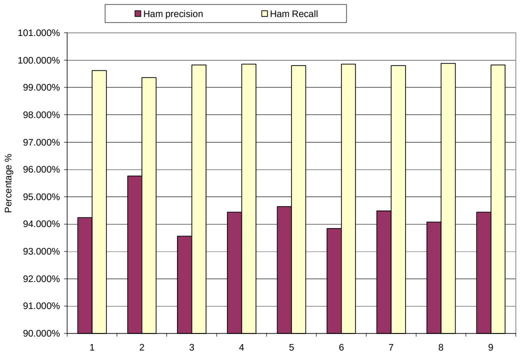
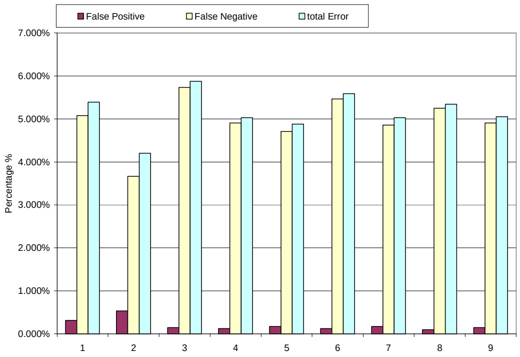


Table 1 and Table 2 show Spam precision and recall, Ham Precision and recall respectively. We notice that with fewer input layers, we have excellent results for Spam precision, Ham precision, and ham recall (see figure 1 and 2), however, this is not true with Spam recall rate. As shown in this section; using the generated input layers from library in batches of 200, 300, 400, 500, 600, 700, 800, 900, and 1000. Each one was tested against all the messages in the testing dataset. Table 3 summarizes false positive and false negative results using CLA_ANN.

Table 3: False Positive, False Negative, Total Error Result

No Of Input Layers	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%	4.882%
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%

Figure 3: False Positive, False Negative, Total Error Result (ANN)



F

Depending on the results in table 3 we find that we get a very low false positive rate which is very acceptable as told before. On the other hand we get tolerable false negative rates. According to Figure 3 we can see that we have the best results when the number of the input layers used for spam detection is 300 input layers. Also, we get a good result when the number of input layers is 600.

The modifications on ANN give excellent results. We get promising values when the number of input layers is 300. We know that if you have a system which can achieve such results with this low number of input layers 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.

6. Conclusion & Future Work

The modifications on ANN give excellent results. We get promising values when the number of input layers is 300. We know that if you have a system which can achieve such results, the system will be considered a successful system. The ANN_CLA also achieve promising results using Spam Assassin public corpus with 0.534 % false positive and 3.668% false negative using only 300 input layers. Modifications of future could use a machine learning method to delete old and useless input layers and replace them with new adapted layers.

- The ANN_CLA achieves promising results using Spam Assassin public corpus with 0.534 % false positive and 3.668% false negative using only 300 input layers.

In summary:

- Spam detection using CLA_ANN represents the following:

- It shows that it can give a promising result even with a small number of input layers.
- It has an excellent opportunity to be used as an anti-spam fighter even if it is not combined with another approach.
- Modification of future could be using a machine learning method to perform deleting old and useless input layers parameter and replace it with new adapted layers.

References

[1] **James Carpinter, Ray Hunt**, Tightening the net: A review of current and next generation spam filtering tools, , Department of Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand computers & security 25(2006) 566 – 578, journal homepage: www.elsevier.com/locate/cose, ELSEVIER.

[2] **Christine E. Drakeand Jonathan J. Oliver, Eugene J. Koontz**. Anatomy of a Phishing Email. Proceedings of the First Conference on Email and Anti-spam (CEAS), 2004.

[3] **Duncan Cook**, Catching Spam before it arrives: Domain Specific Dynamic Blacklists, Australian Computer Society, 2006, ACM

[4] **Gauthronet, S & Drouard**, E 2001, Unsolicited Commercial Communications and Data Protection, Commission of the European Communities.

[5] **Bekker**, S 2003, Spam to Cost U.S. Companies $10 Billion in 2003, ENT News, viewed May 11 2005, ttp://www.entmag.com/news/article.asp?EditorialsID =5651>.

[6] **James Clark, Irena Koprinska, Josiah Poon**, A Neural Network Based Approach to Automated E-mail Classification

[7] **Levent Ozgur, Tunga Gungor, Fikert Gurgen**, Adaptive anti-spam filtering for agglutinative languages: a special case for Turkish, Elsevier, 2004

[8] **Ian Stuart, Sung-Hyuk Cha, and Charles Tappert**, A Neural Network Classifier for Junk E-Mail. Proceedings of Student/Faculty Research Day, CSIS, Pace University, May 7th, 2004

[9] **D. Puniškis, R. Laurutis, R. Dirmeikis**, An Artificial Neural Nets for Spam e-mail Recognition, *electronics and electrical engineering ISSN 1392 – 1215 2006. Nr. 5(69)*

[10] **Duhong Chen, Tongjie Chen, and Hua Ming**, Spam Email Filter Using Naïve Bayesian, Decision Tree, Neural Network, and AdaBoost

[11] **Julia Itskevitch**. Master Thesis, A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in the School of Computing Science, Simon Fraser University, "Automatic Hierarchical E-Mail Classification Using Association Rules", July 2001

[12] **Sarah Jane Delany**, PhD Thesis, Using Case-Based Reasoning for Spam Filtering, A thesis submitted to the Dublin Institute of Technology in fulfillment of the requirements for the degree of Doctor of Philosophy School of Computing, Dublin Institute of Technology ,2006.

[13] **Gabriel R.Weinberg**, Master Thesis, Submitted To The Engineering Systems Division in Partial Fulfillment Of The Requirements For The Degree Of Master Of Science In Technology And Policyat The Massachusetts Institute Of Technology ,A System Analysis Of The Spam Problem, Massachusetts Institute Of Technology, 2005.

[14] **Csaba Gulyás**, Master Thesis, Creation of a Bayesian network-based meta spam filter, using the analysis of different spam filters, Budapest, 16th May 2006

[15] http://spam.abuse.net

[16] **Shlomo Hershkop**, PhD Thesis, Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate School of Arts and Sciences, COLUMBIA UNIVERSITY , Behavior-based Email Analysis with Application to Spam Detection, 2006

[17] **Anders Wiehes**, Master Thesis, "Comparing Anti Spam Methods", Master of Science in Information Security, Department of Computer Science and Media Technology, Gjøvik University College, 2005

[18] **José María, Guillermo Cajigas, Enrique Puertas**, Content Based SMS Spam Filtering, DocEng'06, October 10–13, 2006, Amsterdam, The Netherlands. 2006 ACM 1-59593-515-0/06/0010

[19] **Joachims**, T.: 1998, Text categorization with support vector machines: learning with many relevant features, in C. Nedellec and C. Rouveirol (eds), Proceedings of ECML-98, 10th European Conference on Machine Learning, number 1398 in LNCS, Springer Verlag, Heidelberg, DE, pp. 137-142.

[20] **Dumais, S., Platt, J., Heckerman, D. and Sahami, M**.: 1998, Inductive learning algorithms and representations for text categorisation, Procs of ACM 7th Internation Conference on information and Knowledge Management (CIKM 98), ACM Press, pp. 148-155.

[21] **Cardoso-Cachopo, A. and Oliveira, A**.: 2003, An empirical comparison of text categorisation methods, in M. A. Nascimento, E. S. de Moura and A. L. Oliveira (eds), Proceedings of Conference on String Processing and Information Retrieval, Springer Verlag, pp. 183-196

[22] **Drucker, H., Wu, D. and Vapnik, V**.: 1999, Support vector machines for spam categorisation, IEEE Transactions on Neural Networks 10(5), 1048-1055.

[23] **Kolcz, A. and Alspector, J**: 2001, SVM-based filtering of email spam with content-specific misclassification costs, TextDM'2001 (IEEE ICDM-2001 Workshop on Text Mining), IEEE, pp. 123-130.

[24] **Cristiani, N. and Scholkopf, B**.: 2002, Support vector machines and kernel methods: The new generation of learning machines, AI Magazine 23(3), 31{41.

[25] **Graham**, P2003 ,"better Bayesian Filtering", paper presented to 2003 spam conference.

[26] **Damiani, E, Vimercati, Paraboschi, S & Samarati**, P 2004, 'An Open Digest-based Technique for Spam Detection', paper presented to The 2004 International Workshop on Security in Parallel and Distributed Systems, San Francisco, CA USA.

[27] **Drewes**, R 2002, An artificial neural network spam classifier, Rich Drewes, viewed May 8 2005, <http://www.interstice.com/drewes/cs676/spamnn/ spam-nn.html>.

[28] **Chris Miller**, Group Product Manager, Enterprise Email Security, Neural Network-based Antispam Heuristics

[29] **Alessio Pascucci**, October 31, 2006, Toward a PhD Thesis on Pattern Recognition

[30] **Konstantin Tretyakov**, Machine Learning Techniques in Spam Filtering, Institute of Computer Science, University of Tartu ,Data Mining Problem-oriented Seminar, MTAT.03.177,May 2004, pp. 60-79.