

An Efficient Three-phase Email Spam Filtering Technique

Abstract –Email spam is one of the major problems of the today’s Internet, bringing financial damage to companies and annoying individual users. Many spam filtering techniques based on supervised machine learning algorithms have been proposed to automatically classify messages as spam or legitimate (ham). Bayesian spam filtering is a popular mechanism used to distinguish spam email from ham email. In this paper, we propose an efficient three-phase email spam filtering technique: Bayesian, Clonal Selection and Negative Selection. The experimental results applied on 10,000 email messages taken from the TREC 2007 corpus shows that when we apply the Clonal selection and Negative selection algorithms with the Bayesian spam filtering technique the accuracy rate is increased than applying each technique alone.

Keywords - Email spam, Machine Learning, Bayesian Classifier, Artificial Immune System, Negative selection and Clonal Selection.

1. INTRODUCTION

The problem of spam (unwanted) electronic mails is nowadays a serious issue. The average of spams sent per day is 94 billion in 2012 representing more than 70% of all incoming e-mail [1], it turns out they add up to a \$20 billion cost to society , according to a new paper called “The Economics of spam [2],”. Spam causes misuse of traffic, storage space and computational power. It causes also legal problems by advertising for a variety of products and services, such as pharmaceuticals, jewellery, electronics, software, loans, stocks, gambling, weight loss, and pornography, as well as phishing (identity theft) and malware distribution attempts [3], [4]. Many spam filtering techniques have been proposed to automatically classify messages as spam or legitimate (ham). Spam filtering in its simple form can be considered as a text categorization problem where the classes to be predicted are spam and legitimate [5]. A variety of supervised machine-learning algorithms have been successfully applied to email filtering task. A nonexhaustive list includes: naive Bayes classifiers [6], support vector machines [7] memory-based learning [8] AdaBoost [9], and a maximum entropy model [10].

White, black and grey listings of domain names, keyword-based filtering, heuristic-based filtering, etc are other filtering techniques that can be used to filter spam emails. All of these techniques, however, require heavy maintenance and cannot achieve very high overall accuracy. Goodman et al. [11] presented an overview of the field of anti-spam protection, giving a brief history of spam and anti-spam and describing major directions of development.

Artificial Immune System (AIS) is an area of research that bridges the disciplines of immunology, computer science and engineering [12][13][14]. The immune system has drawn significant attention as a potential source of inspiration for novel approaches to solving complex computational problems. The AIS concepts can be adapted to solve the email spam problem [15].

In this paper, a spam identification system that identifies and segregates spam messages from legitimate ones is proposed. The proposed system is based on both the classical Naive Bayesian approach and two of the artificial immune system models (Clonal Selection and Negative Selection). The paper is organized as follows: Section 2 describes Naive Bayesian approach to spam detection; The Artificial Immune System (AIS) approach to spam filtering is described in section 3; Section 4 describes the steps of the proposed spam filtering technique; Section 5 discusses the results of using the proposed spam filtering technique; and Section 6 draws conclusions from the work presented here.

2. NAÏVE BAYESIAN CLASSIFIER

Bayesian spam filtering is a popular mechanism used to distinguish spam email from ham email and it is based on machine-learning algorithms. Sahami et al. built a Naïve Bayesian classifier for the domain of spam filtering [6]. In this classifier a probabilistic method is used to train a model of classification by using features (keywords) extracted from messages. Give two classes $C = \{C1 = \text{spam}, C2 = \text{ham}\}$ and features $f1, f2, \dots, fn$ the probability that these features belong to a certain class using Naïve Bayesian can be expressed as follows:

$$P(C|f1, f2, \dots, fn) = \frac{P(f1, f2, \dots, fn|C)P(C)}{P(f1, f2, \dots, fn)} \quad (1)$$

Assuming conditional independence, one can compute $P(f1, f2, \dots, fn|C)$ as follows:

$$P(f_1, f_2, \dots, f_n | C) = \prod_{i=1}^n p(f_i | C) \quad (2)$$

To classify an email message as spam, one can check if it exceeds a specific threshold:

$$\frac{P(C1 = spam | f_1, f_2, \dots, f_n)}{P(C2 = non - spam | f_1, f_2, \dots, f_n)} \geq \beta, \quad 0 < \beta < 1 \quad (3)$$

3. ARTIFICIAL IMMUNE SYSTEM

Biological Immune System (BIS) has been successful at protecting the human body against a vast variety of foreign pathogens. A role of the immune system is to protect our bodies from infectious agents such as viruses, bacteria, fungi and other parasites. BIS based around a set of immune cells called lymphocytes comprised of B and T cells. On the surface of each lymphocyte is a receptor and the binding of this receptor by chemical interactions to patterns presented on antigens which may activate this immune cell. Subsets of the antigens are the pathogens, which are biological agents capable of harming the host (e.g. bacteria). Lymphocytes created in the bone marrow and the shape of the receptor determined by the use of gene libraries [16].

AIS is a paradigm of soft computing which motivated by BIS. It based on the principles of the human immune system, which defends, as discussed above, the body against harmful diseases and infections. To do this, pattern recognition tasks are performed to distinguish molecules and cells of the body (self) from foreign ones (non-self). AIS inspire the production of new ideas that could be used to solve various problems in computer science, especially in security field.

The main role of a lymphocyte in AIS is encoding and storing a point in the solution space or shape space. The match between a receptor and an antigen may not be exact and so when a binding takes place it does so with strength called an affinity. If this affinity is high, the antigen included in the lymphocyte's recognition region [17][18].

AIS contains many algorithms such as Negative selection algorithms, artificial immune network, Clonal selection algorithm, Danger Theory inspired algorithms, dendritic cell algorithms. Throughout this paper we used Clonal and Negative Selection algorithms:

3.1 Clonal selection algorithm

Clonal selection and expansion is the most accepted theory used to explain how the immune system copes with the antigens. In brief, the Clonal selection theory states that when antigens invade an organism, a subset of the immune cells capable of recognizing these antigens proliferate and differentiate into active or memory cells. The fittest clones are those, which produce antibodies that bind to antigen best (with highest affinity). The main steps of Clonal selection algorithm can be summarized as follows [14].

Algorithm 1: Clonal selection

Step 1: For each antibody element

Step 2: Determine its affinity with the antigen presented

Step 3: Select a number of high affinity elements and reproduce (clone) them proportionally to their affinity.

3.2 Negative selection algorithm

Negative selection algorithm is one of the important techniques in this paradigm that is widely applied to solve two-class (self and non-self) classification problems. Many advances to Negative Selection Algorithms (NSA) occurred over the last decade. This algorithm uses only one class (self) for training resulting in the production of detectors for the complement class (non-self). This paradigm is very useful for anomaly detection problems in which only one class is available for training, such as intrusive network traffic and its detection problem [20]. The main steps of Negative selection algorithm can be summarized as follows [19]:

Algorithm 2: Negative Selection

input : Sseen = set of seen known self elements

output : D = set of generated detectors

begin

repeat

Randomly generate potential detectors and place them in a set P

Determine the affinity of each member of P with each member of the self set Sseen

If at least one element in S recognizes a detector in P according to a recognition threshold,

Then

the detector is rejected, otherwise it is added to the set of available detectors D

until Stopping criteria has been met

end

4. THE PROPOSED EMAIL SPAM FILTERING TECHNIQUE

The proposed filtering technique consists of 3 phases in testing. The first phase uses the Naïve Bayesian classifier to train the system and filter it. The second phase filter spam emails by using the Clonal with Bayesian and Negative

selection algorithms with Bayesian. The third phase optimizes the results obtained from the first phase and second phase by merging the three techniques. Figure 1 illustrates the steps of the proposed spam filtering technique. The following subsections give a brief description of each step.

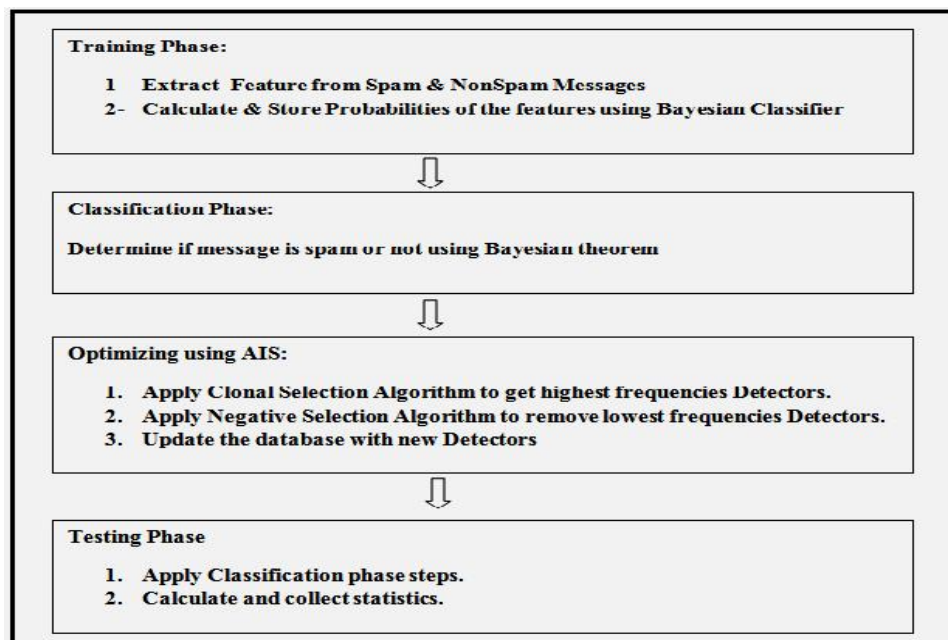


Figure 1: The steps of the proposed filtering technique

4.1 Training phase

The proposed filtering technique classifies an email message into one of two categories: spam or non-spam. A supervised learning approach (Bayesian classifier) is used to enable the filter to build a history of what spam and ham messages look like. Figure 2 describes the steps of the raining phase.

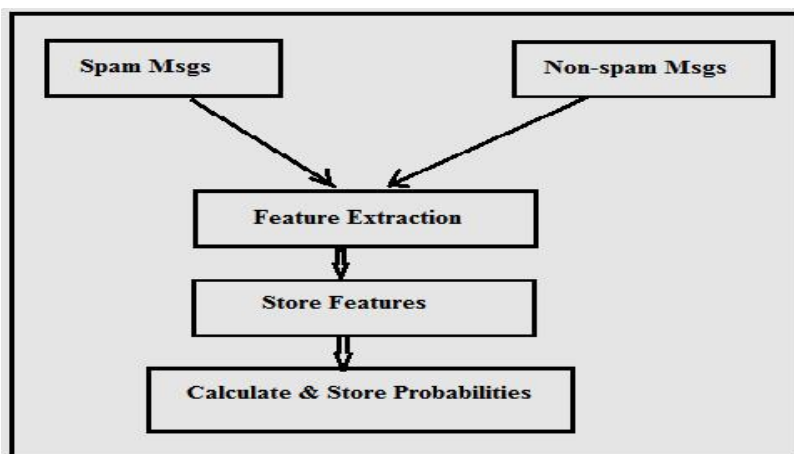


Figure 2: The Steps of Training Phase

4.1.1 Features Extraction

An email message contains two parts: a header and a body. The header consists of fields usually including at least the following:

- From: This field indicates the sender's email address.
- To: This field indicates the receiver's email address.
- Date: This field indicates the date and time of when the message was sent.
- Subject: This field indicates a brief summary of the message's content.
- Received: This field indicates the route the message took to go from the sender to the receiver.
- Message-ID: This field indicates the ID of the message.

Every email message has a unique ID at a given domain name: id@senders-domain-name. On the other hand, the body of an email message contains the actual content of the message. The header and the body of an email

message are separated by an empty line. To extract the features (words) from an incoming email message, the following steps will be done:

Step 1: Parse an email message and extract the header and the body parts.
Step 2: Remove the header fields described above from the messages since these fields appear in every email message.
Step 3: Extract features from the header by tokenizing the header using the following delimiter: \n\r\t\ /&%# {}[]! +=-() ""*?;:<>
Step 4: Extract features from the body by tokenizing the body using the following delimiter: \n\r\t\ /&%# {}[]! +=-() ""*?;:<>
Step 5: Ignore features of size strictly less than 3 and digits.

4.1.2 Store Features

To store the extracted features yield from previous step, a SQL Server 2008 database table is used. Each record stores the following information:

- Feature.
- Number of feature's occurrences in spam messages (initialized to 0).
- Number of feature's occurrences in ham messages (initialized to 0).
- Feature's probability (initialized to 0).

Every time a feature is extracted, the email message is checked if it is spam or ham. If it is spam then the number of spam occurrences is incremented by one. Otherwise, the number of ham occurrences is incremented by one.

4.1.3 Compute and Store Probabilities

After extracting all the features and filling the database table with such features, we start enumerating through the elements of the table to compute the probability for each feature. To compute the probability (P_f) for a feature, the following formula is used:

$$P_f = \frac{\frac{s}{t_s}}{\frac{s}{t_s} + \frac{kn}{t_n}} \quad (4)$$

Where s is the number of occurrences of feature f in spam messages, t_s is the total number of spam messages in the training set, n is the number of occurrences of feature f in ham messages, t_n is the total number of ham messages in the training set, and k is a number that can be tuned to reduce false positives by giving a higher weight to number of occurrences of ham features.

4.2 Classification phase

After the training step, the filtering system becomes capable of making decisions based on what it has seen in the training set. Every time an email message is presented to the filter the following steps are done:

Step 1: Parse the email message and extract both the header and the body parts.
Step 2: Extract features from the header and body using the feature extraction step described above.
Step 3: For each extracted feature retrieve the corresponding probability from the database DB. If the extracted feature does not exist in DB, then assign it a probability of 0.5. Compute the interestingness of the extracted features according to the formula:

$$I_f = |0.5 - P_f|$$

Step 4: Extract the most interesting features from the list of features.
Step 5: Calculate the total message probability (P) by combining the probabilities of the most interesting features using Bayes theorem based on Graham's assumptions [21].

$$P = \frac{P_{f1}P_{f2}P_{f3}...P_{fn}}{P_{f1}P_{f2}P_{f3}...P_{fn} + (1-P_{f1})(1-P_{f2})(1-P_{f3})...(1-P_{fn})} \quad (5)$$

The closer the P is to 0, the more likely the message is non-spam and the closer P is to 1, the more likely the message is spam. In our implementation the used threshold to determine whether a given email message is spam is 0.9, that is, if $P \geq 0.9$ then the email message is classified as spam otherwise it classified as non-spam. Figure 3 summarizes the classification phase steps.

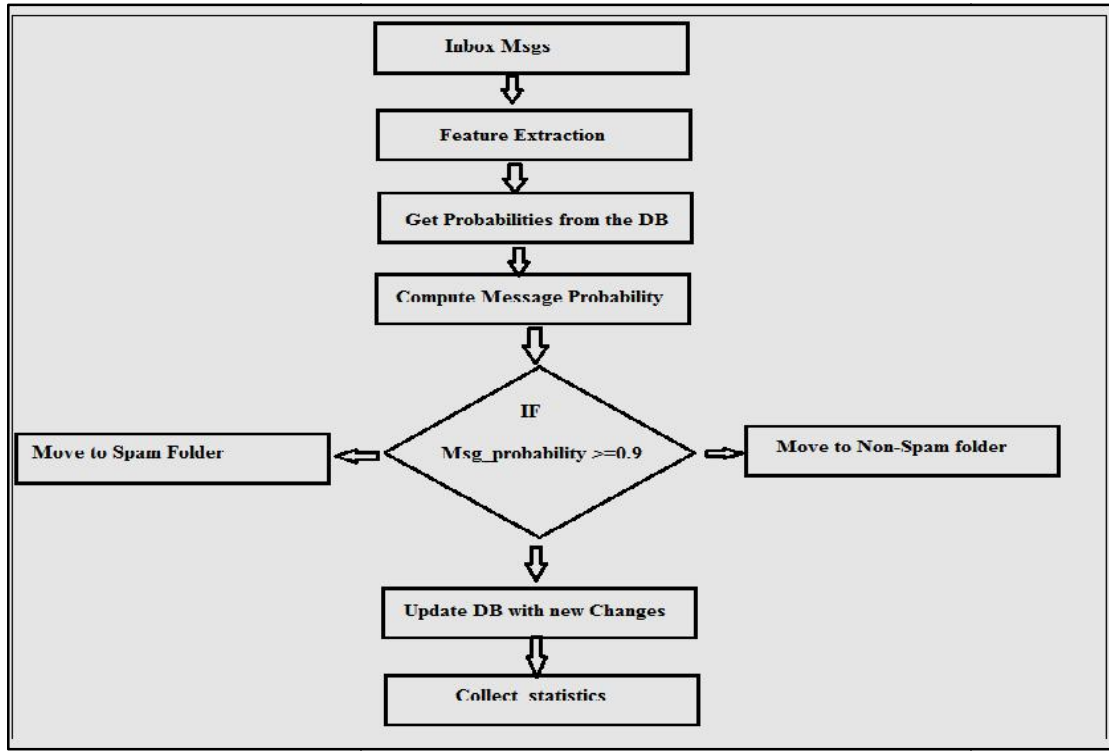


Figure 3: The Steps of Classification Phase

4.3 Optimizing phase using AIS

An immune system's main goal is to distinguish between self and potentially dangerous non-self elements. In a spam immune system, we want to distinguish legitimate messages (as self) from spam message (as nonself) like biological immune system. The central part of the AIS engine is its Detectors, which are regular expressions made by combining information from training process. These regular expressions match patterns in the entire message. Each Detector acts as an antibody and consists of three associated weights (initialized to zero) detailing what has been matched by that particular detector [22]:

- Spam Frequency: the cumulative weighted number of spams matched
- Ham Frequency: the cumulative weighted number of messages matched
- Affinity: is a measure that represents the strength of matching between antibody and message

The AIS engine applies on detectors (antibodies) dataset in two phases. Firstly, it determines the affinity ratio for all detectors with messages; secondly it rejects all detectors with low affinity value, so a clone of detectors with highest affinity is selected. The following algorithm illustrates the steps of the AIS engine:

Algorithm 4: AIS Engine

```

D input: set of detectors in the database
D' output: set of detectors have a highest affinity capable of classifying message.
begin
  Load a set of detectors D from database
  For all detectors in D do
    Calculate the affinity for each message according to Bayesian method.
  end
  For all detectors in D do
    Reject the detector with low affinity.
  end
  Select a clone of all detectors that have a highest affinity
End
  
```

179 Figure 4 illustrates the steps that implements in our system:

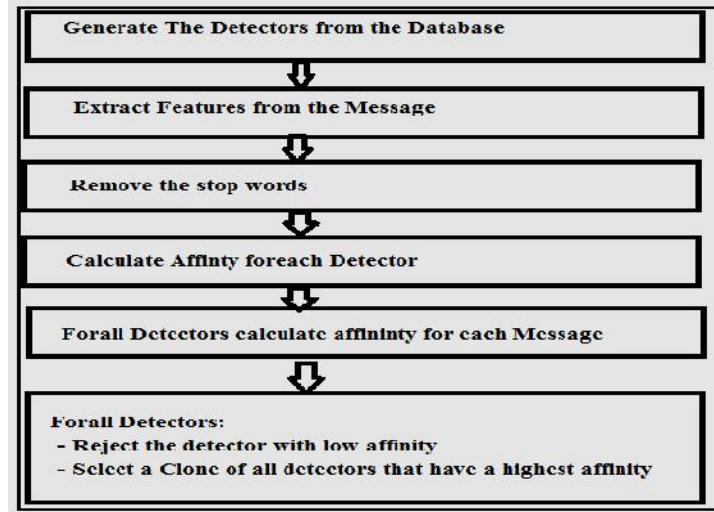


Figure 4: The Steps of AIS

4.4 Testing phase

To make the testing of the proposed filtering technique easier, a built-in testing functionality is included. As can be seen in Figure 5, the filter takes as input a directory specified by the user that contains all email messages (spam and non-spam). When the filter starts the classification, two directories are created: a spam directory and a non-spam directory. If the filter classifies a given email message as spam then the filter copies it from the input directory to the spam directory. Similarly, if the filter classifies a given email message as non-spam then the filter copies it from the input directory to the non-spam directory.

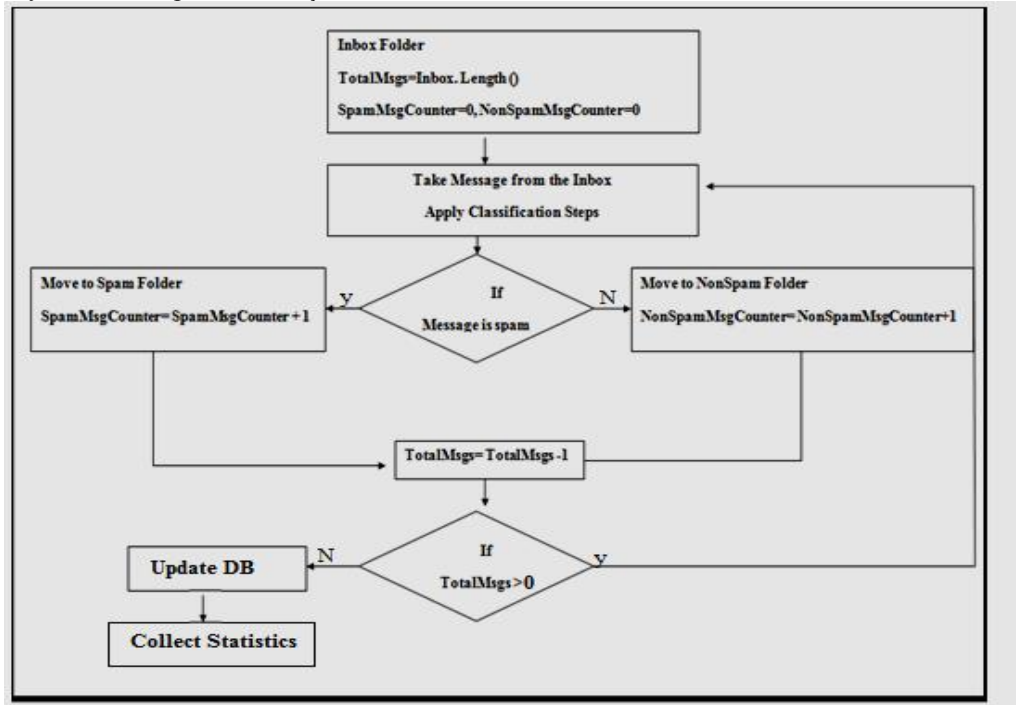


Figure 5: The Steps of Testing Phase

Every message in the dataset has a letter in its file name: s if the email message is spam and n otherwise. When the filter completes the classification process, it traverses the spam and non-spam directories to count:

True Positives (TP): The number of spam emails classified as spam.

True Negatives (TN): The number of ham emails classified as ham.

False Positives (FP): The number of ham falsely classified as spam.

False Negatives (FN): The number of spam falsely classified as ham.

After calculating the values of TP, TN, FP, and FN the Overall accuracy (OA) of the filter can be calculated as [24]:

Overall Accuracy (OA): $(TP + TN) / (TP + FP + FN + TN)$. (6)

5. EXPERIMENTAL RESULTS

To train and test the proposed spam filter the TREC 2007 Spam Track Public Corpora which contains a large number of email messages is used [23]. This corpus contains 75,419 email messages; 25,220 are non-spam messages and 50,199 are spam messages. The problem with the TREC corpus is that the non-spam and spam messages are not separated into two different directories. Both types of messages are placed in the same directory. The only way to know if a message is spam or non-spam is by looking at an index file which has the message number and its type. In our implementation we used C# code to separate the spam and non-spam messages into two directories.

5.1 Evaluation Strategy and Experimental Results

To determine the relative performance of the proposed filtering technique, it was necessary to test it against another continuous learning algorithm. The well-known Bayesian classifier was chosen as a suitable comparison algorithm. Our main goal is to analyze the detection capability of both the proposed filtering technique and Bayesian algorithm on actual email messages. We randomly choose dataset consists of 10000 messages from the TREC 2007. This data set consists of 5000 spam and 5000 non spam messages 50 % of them appears in the training set and the other 50 % does not appear before. We apply 9 Test cases for each technique. Each test case consists of 1000 message selected randomly from the Testing dataset. Our threshold is 0.9.

We employed the measures that are widely used in spam classification. The common evaluation measures include true positive, true negative, false positive, false negative, Detection rate, False positive rate and overall accuracy. Their corresponding definitions are as follows [24] :

Detection Rate (DR): $TP / (TP + FN)$. (7)

False positive Rate (FPR): $FP / (TN + FP)$.

(8)

Overall Accuracy (OA): $(TP+TN) / (TP+FP+FN+TN)$. (6)

5.1.1 Experiment 1: Testing using Bayesian Spam Filtering Technique

In this section we apply the Bayesian Method using dataset selected randomly from testing data set. We have 9 test cases every case have 1000 messages selected randomly from testing data set. At every test case we change the ratio between No. of spam messages and No. of non spam messages

Table 1: Testing using Bayesian

Test Case No.	Spam	Non-Spam	TP	TN	FP	FN	DR	FPR	Overall Accuracy
1	900	100	564	436	0	336	0.62	0	74.85%
2	800	200	401	599	0	399	0.50	0	71.48%
3	700	300	357	643	0	343	0.51	0	74.46%
4	600	400	303	697	0	297	0.50	0	77.10%
5	500	500	250	750	0	250	0.50	0	80.00%
6	400	600	200	800	0	200	0.50	0	83.33%
7	300	700	155	845	0	145	0.51	0	87.34%
8	200	800	100	900	0	100	0.5	0	90.91%
9	100	900	50	950	0	50	0.5	0	95.24%
Average of the measures							51.5%	0%	81.63%

As observed in Table (1) when we increase No. of non spam messages the Accuracy of the filter increase. Also we can observe that the false positive rate of our proposed system equal 0 this means that there is no Non Spam messages marked by mistake as Spam. This indicates higher better spam detection.

Based on Table (1) we can say that (on average) the detection rate, false positive rate, and overall accuracy are 51.5 %, 0%, 81.63%.

5.1.2 Experiment 2: Testing using AIS (Clonal Selection Algorithm)

In this section we apply the Clonal selection algorithm using two different dataset. The first one is the same dataset that we used in the Bayesian method. The second one is different dataset selected randomly from the testing dataset.

At first Table (2) we apply the Clonal selection algorithm on the same database resulted from the Training phase using the same dataset that used before in the Bayesian method. Then Table (3) we apply the Clonal selection algorithm on and the database resulted from applying the Bayesian method using the second dataset that selected randomly from the testing dataset.

Table 2: Testing using Clonal Selection only

Test Cases No.	Spam	Non-Spam	TP	TN	FP	FN	DR	FPR	Overall Accuracy
1	900	100	564	436	0	336	0.62	0	74.85
2	800	200	401	599	0	399	0.50	0	71.48
3	700	300	357	643	0	343	0.51	0	74.46
4	600	400	303	697	0	297	0.50	0	77.10
5	500	500	250	750	0	250	0.50	0	80.00
6	400	600	200	800	0	200	0.50	0	83.33
7	300	700	155	845	0	145	0.51	0	87.34
8	200	800	100	900	0	100	0.5	0	90.91
9	100	900	50	950	0	50	0.5	0	95.24
Average of the measures							55.13 %	0%	81.63%

Table 3: Testing using Clonal Selection with Bayesian

Test Cases No.	Spam	Non-Spam	TP	TN	FP	FN	DR	FPR	Overall Accuracy
1	900	100	739	261	0	161	0.82	0	86.13
2	800	200	740	260	0	60	0.92	0	94.34
3	700	300	660	340	0	40	0.94	0	96.15
4	600	400	564	436	0	36	0.94	0	96.53
5	500	500	500	500	0	0	1	0	100.00
6	400	600	388	612	0	12	0.97	0	98.81
7	300	700	297	703	0	3	0.99	0	99.70
8	200	800	80	820	0	20	0.8	0	97.83
9	100	900	88	912	0	12	0.88	0	98.81
Average of the measures							91.87%	0	96.48%

As can be seen in Table (2) we have 9 test cases every case have 1000 messages this is the same test cases we use it in the Bayesian. As observed the result is the same as when we use the Bayesian technique.

When we apply the Clonal selection algorithm after applying the Bayesian but with different dataset we obtain the results in table (3) and figure (3). It shows an improvement in all the measures than using Bayesian alone.

Based on Table (3) we can say that (on average) the detection rate, false positive rate, and overall accuracy are 91.87 %, 0%, 96.48%.

5.1.3 Experiment 3: Testing using AIS (Negative Selection Algorithm)

In this section we apply the Negative selection algorithm using two different dataset. The first one is the same dataset that we used in the Bayesian method. The second one is different dataset selected randomly from the testing dataset.

At first (Table 4) we apply the Negative selection algorithm on the same database resulted from the Training phase using the same dataset that used before in the Bayesian method. Then (Table 5) we apply the Negative selection algorithm on and the database resulted from applying the Bayesian method using the second dataset that selected randomly from the testing dataset

Table 4: Testing using Negative Selection only

Test Cases No.	Spam	Non-Spam	TP	TN	FP	FN	DR	FPR	Overall Accuracy
1	900	100	564	436	0	336	0.62	0	74.85
2	800	200	401	599	0	399	0.50	0	71.48
3	700	300	357	643	0	343	0.51	0	74.46
4	600	400	303	697	0	297	0.50	0	77.10
5	500	500	250	750	0	250	0.5	0	80.00
6	400	600	200	800	0	200	0.5	0	83.33
7	300	700	155	845	0	145	0.51	0	87.34
8	200	800	100	900	0	100	0.5	0	90.91
9	100	900	50	950	0	50	0.5	0	95.24
Average of the Measures							55.13 %	0%	81.63%

Table 5: Testing using Negative Selection with Bayesian

Test Case No.	Spam	Non-Spam	TP	TN	FP	FN	DR	FPR	Overall Accuracy
1	900	100	741	259	0	159	0.82	0	86.28
2	800	200	740	260	0	60	0.92	0	94.34
3	700	300	660	340	0	40	0.94	0	96.15
4	600	400	564	436	0	36	0.94	0	96.53
5	500	500	500	500	0	0	1	0	100.00
6	400	600	388	612	0	12	0.97	0	98.81
7	300	700	297	703	0	3	0.99	0	99.70
8	200	800	80	820	0	20	0.80	0	97.83
9	100	900	88	912	0	12	0.88	0	98.81
Average of the measures							91.90%	0%	96.50%

As can be seen in Table (4) we have 9 test cases every case have 1000 messages this is the same test cases we use it in the Bayesian and Clonal. As observed the result is the same as when we use the Bayesian technique.

When we apply the Negative selection algorithm after applying the Bayesian but with different dataset we obtain the results in table (5). It shows an improvement in all the measures than using Bayesian alone and with Clonal.

Based on Table (5) we can say that (on average) the detection rate, false positive rate, and overall accuracy are 91.9 %, 0%, 96.5%.

5.1.4 Experiment 4: Testing using AIS (Negative Selection & Clonal Algorithm) with Bayesian

In this section we apply the Negative selection algorithm using different dataset than we used before in the previous experiments. This dataset selected randomly from the testing dataset.

We apply the Negative selection algorithm on the database resulted from applying the Clonal Selection Algorithm before but with different dataset selected randomly from the testing dataset. Table (6) illustrates the improvements that occurred by applying this proposed technique.

Table 6: Testing using Negative Selection & Clonal Selection & Bayesian

Test Cases No.	Spam	Non-Spam	TP	TN	FP	FN	DR	FPR	Overall Accuracy
1	900	100	859	141	0	41	0.95	0	96.06
2	800	692	800	200	0	0	1	0	100.00
3	700	565	692	308	0	8	0.98	0	99.21
4	600	400	565	435	0	35	0.94	0	96.62
5	500	500	496	504	0	4	0.99	0	99.60
6	400	600	384	616	0	16	0.96	0	98.43
7	300	700	299	701	0	1	0.99	0	99.90
8	200	800	199	801	0	1	0.99	0	99.90
9	100	900	100	900	0	0	1	0	100.00
Average of the measures							98.09%	0%	98.86%

When we apply the Negative selection algorithm after applying the Clonal selection algorithm It shows an improvement in all the measures than using Bayesian alone and with Clonal and with negative.

Based on Table (5) we can say that (on average) the detection rate, false positive rate, and overall accuracy 98.09 %, 0%, 98.86%.

5.1.5 Some Statistics of the Proposed System

In this section we illustrate some evaluations about our proposed system. Table (7) gives a summary for the Overall Accuracy of the different techniques for each dataset. Table (8) gives a summary for the Time taken by applying the different techniques for each dataset. Table (9) gives a summery for all detection rates for each test case tested by different techniques. Table (10) and Table (11) evaluates the different techniques with dataset consists of All messages spam and all messages non spam.

300
301

Table 7: Summary of Overall Accuracy for each technique

Test cases	Bayesian	Clonal+Bayesian	Negative+Bayesian	Bayesian+Clonal+Negative
1	74.85%	86.13%	86.28%	96.06%
2	71.48%	94.34%	94.34%	100.00%
3	74.46%	96.15%	96.15%	99.21%
4	77.10%	96.53%	96.53%	96.62%
5	80.00%	100.00%	100.00%	99.60%
6	83.33%	98.81%	98.81%	98.43%
7	87.34%	99.70%	99.70%	99.90%
8	90.91%	97.83%	97.83%	99.90%
9	95.24%	98.81%	98.81%	100.00%
Average	81.63%	96.48%	96.49%	98.86%

302
303

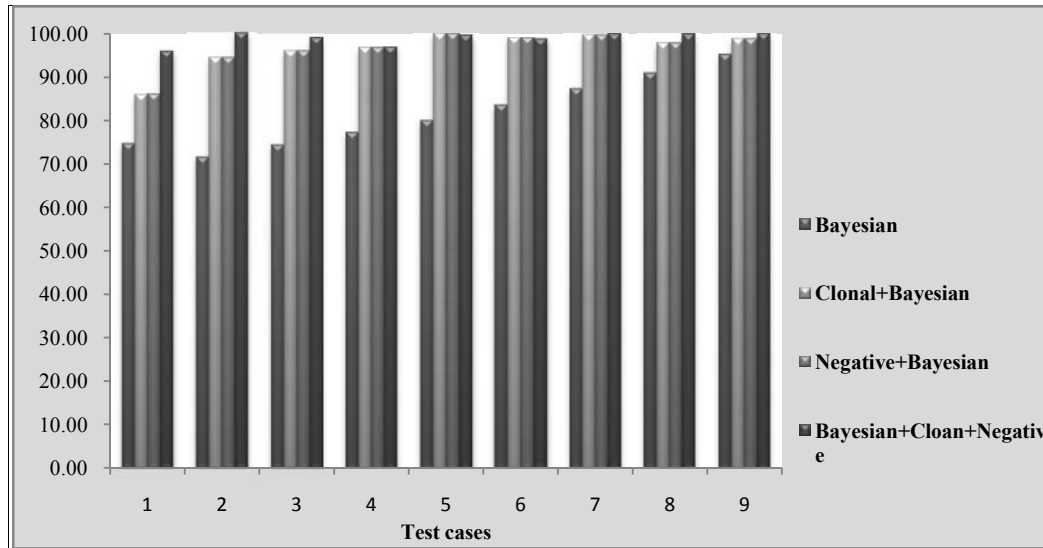


Figure 6: Overall Accuracy for each technique

304
305
306
307
308
309
310
311

Based on Table (7) and Figure (6) we can say that (on average) the overall accuracy for Bayesian, Clonal+Bayesian, Negative+Bayesian, Bayesian+Clonal+Negative are 81.63 %, 96.48%, 96.49%, 98.86%. This means that the performance of the proposed technique is better than Naive Bayesian technique.

Table8: A summary for the Time taken by applying the different techniques

Test cases	Bayesian	Clonal+Bayesian	Negative+Bayesian	Bayesian+Clonal+Negative
1	2:00	1:26	3:15	1:19
2	2:35	4:18	3:29	2:22
3	3:21	3:02	3:37	1:56
4	3:54	3:09	3:21	1:59
5	3:45	2:28	4:28	1:47
6	5:20	4:42	5:54	2:32
7	7:13	2:43	4:34	2:26
8	5:36	2:52	3:01	1:46
9	5:57	3:13	5:04	2:00
Average(minute)	4:24m	3:05 m	4:04 m	2:00 m

312
313

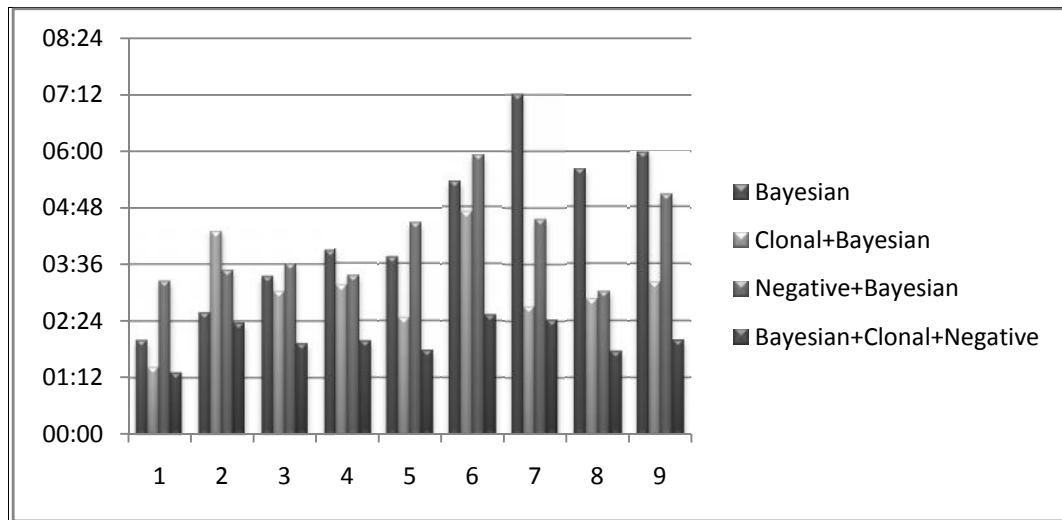


Figure 7: A summary for the Time taken by applying the different techniques

Based on Table (8) and Figure (7) we can say that (on average) the time taken by applying Bayesian, Clonal+Bayesian, Negative+Bayesian, Bayesian+Clonal+Negative are 4:24, 3:05, 4:04, 2:00. This means that the performance of the proposed technique is better than Naive Bayesian technique.

Table 9: Summary of Detection rate for each technique

Test cases	Bayesian	Clonal+Bayesian	Negative+Bayesian	Bayesian+Clonal+Negative
1	0.62	0.82	0.82333	0.95
2	0.5	0.92	0.925	1
3	0.51	0.94	0.94286	0.98
4	0.5	0.94	0.94	0.94
5	0.5	1	1	0.99
6	0.5	0.97	0.97	0.96
7	0.51	0.99	0.99	0.99
8	0.5	0.8	0.8	0.99
9	0.5	0.88	0.88	1
Average	52%	92%	92%	98%

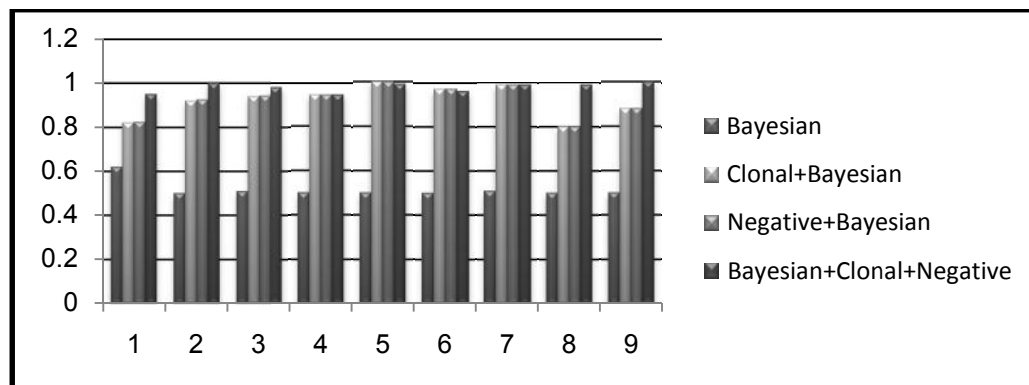


Figure 8: Detection rate for each technique

Based on Table (9) and Figure (8) we can say that (on average) the detection rate resulted by applying Bayesian, Clonal+Bayesian, Negative+Bayesian, Bayesian+Clonal+Negative 52%, 92%, 92%, 98%. This means that the performance of the proposed technique is better than Naive Bayesian technique.

Table10: Measures of performance using different techniques with dataset NonSpam.

Techniques	Spam	Non-Spam	TP	TN	FP	FN	DR	FPR	Overall Accuracy
Bayesian	0	1000	1000	0	0	0	1	0	100
Bayesian+Clonal	0	1000	1000	0	0	0	1	0	100
Bayesian+Negative	0	1000	1000	0	0	0	1	0	100
Bayesian+Clonal+Negative	0	1000	1000	0	0	0	1	0	100

Table11: Measures of performance using different techniques with dataset Spam.

Techniques	Spam	Non-Spam	TP	TN	FP	FN	DR	FPR	Overall Accuracy
Bayesian	1000	0	812	188	0	188	0.812	0	84.18
Bayesian+Clonal	1000	0	929	71	0	71	0.929	0	93.37
Bayesian+Negative	1000	0	929	71	0	71	0.929	0	93.37
Bayesian+Clonal+Negative	1000	0	971	29	0	29	0.971	0	97.18

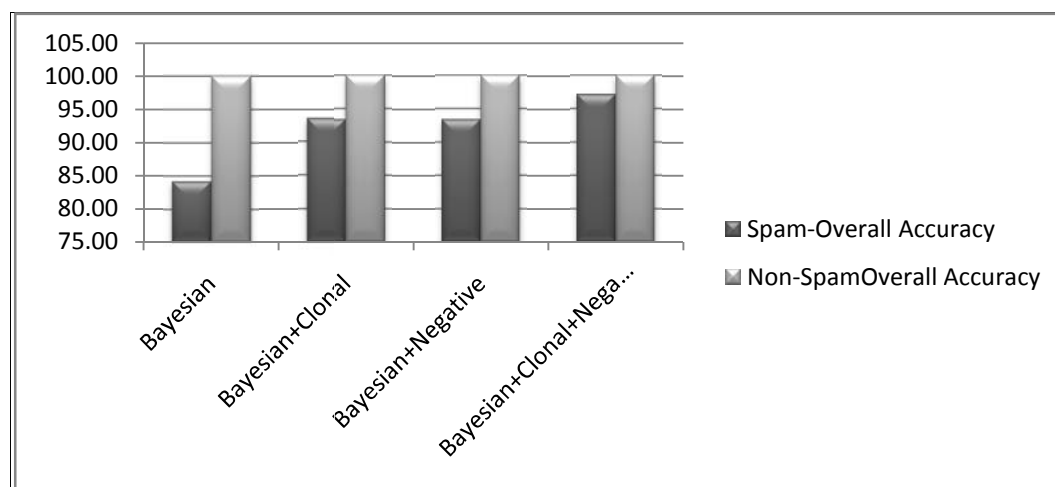


Figure9: The Overall accuracy for dataset with all spam and all non spam

As can be seen in Table (10), Table (11) and Figure (9) when we use a dataset contains Non-Spam Messages only we obtain overall accuracy of 100 % and false positive rate = 0 %. This means that our proposed system is accurate as the aim of any spam filtering technique is to decrease the number of false positives.

We can also observe that the Detection rate is increase when we use the AIS Clonal and Negative Selection algorithms with the Bayesian Classifier. This means that the performance of the proposed technique is better than Naive Bayesian technique.

6. CONCLUSION

An efficient email filtering approach which consists of three phases is presented. This new approach tries to increase the accuracy of a spam filtering via combine the well known Bayesian spam filter with the artificial immune system by using two of its algorithms the clonal selection algorithm and the negative selection algorithm. Experimental results showed an improvement in the performance of the new spam filtering than using each technique alone as it always seek to get the highest and fastest detectors to reduce the false positive rate and get highest accuracy. The experimental results applied on 10,000 email messages shows a high efficiency with the less number of false positives (on average) 0%, High detection rate (on average) 98.09% and the overall accuracy (on average) 98.86%.

360 7. REFERENCES

- 361
362 [1] alt-n. technologies and commtouch. (2012, Apr.) www.altn.com. [Online].
363 <http://www.altn.com/Literature/SecurityThreatTrendReports/EmailThreatTrendReport-2012-Q1.pdf>
364 Last Access Oct., 2013
365
- 366 [2] J. M. Rao and D. H. Reiley, "The Economics of Spam," *Journal of Economic Perspectives*, vol. 26, no. 3, pp. 87–
367 110, 2012.
- 368 [3] S. Chakraborty and B. Mondal, "Spam Mail Filtering Technique using Different Decision Tree Classifiers through
369 Data Mining Approach - A Comparative Performance Analysis," *International Journal of Computer Applications*, vol.
370 47, no. 16, pp. 26-31, Jun. 2012.
- 371 [4] D. DeBarr and H. Wechsler, "Spam detection using Random Boost," *Pattern Recognition Letters*, vol. 33, no. 10,
372 pp. 1237-1244, Jul. 2012.
- 373 [5] L. Zhang, J. Zhu, and T. Yao, "An Evaluation of Statistical Spam Filtering Techniques," in *ACM Transactions on*
374 *Asian Language Information Processing*, vol. 3, 2004, pp. 243-269.
- 375 [6] M. Saham, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian approach to Filtering junk e-mail. In Learning
376 for Text Categorization," in *AAAI Technical Report WS-98-05*, Madison, Wisconsin, 1998.
377 [Online].ftp://ftp.research.microsoft.com/pub/ejh/junkfilter.pdf , Last access Oct., 2013
- 378 [7] H. Drucker, D. Wu, and V. Vapnik, "Support vector machines for spam categorization," *IEEE transactions on*
379 *Neural Networks*, vol. 10, no. 5, pp. 1048-1055, Sep. 1999.
- 380 [8] L. Androutsopoulos, J. Koutsias, K. V. Chandrinou, G. Paliouras, and C. D. Spyropoulos, "An evaluation of naïve
381 Bayesian anti-spam filtering," in *Proceedings of the Workshop on Machine Learning in the New Information Age, 11th*
382 *European Conference on Machine Learning*, Barcelona, Spain, 2000a, 2000, pp. 9-17. [Online].
383 <http://arxiv.org/pdf/cs/0006013v1.pdf> , Last Access Oct., 2013
384
- 385 [9] X. Carreras and L. Ma'rquez, "Boosting trees for anti-spam email Filtering," in *Proceedings of RANLP-2001, 4th*
386 *International Conference on Recent Advances in Natural Language Processing*, 2001, pp. 58-64. [Online].
387 <http://www.eecs.wsu.edu/~holder/courses/cse6363/spr04/present/Carreras01.pdf> , Last access Oct, 2013
- 388 [10] L. Zhang and T. Yao, "Filtering junk mail with a maximum entropy model," in *Proceeding of 20th*
389 *International Conference on Computer Processing of Oriental Languages (ICCPOL03)*, 2003, pp. 446-453.
- 390 [11] J. J Goodman, G. Cormack, and D. Heckerman, "Spam and the ongoing battle for the inbox," in *Communication*
391 *of the ACM*, vol. 50, 2007, pp. 25-33.
- 392 [12] D. Dasgupta, "An overview of artificial immune systems," *Artificial Immune Systems and Their Applications*, pp.
393 3-19, 1998.
- 394 [13] J. Timmis, "Artificial immune systems: A novel data analysis technique inspired by the immune network theory,"
395 Ph.D Thesis, University of Wales, Aberystwyth., 2000.
- 396 [14] J. Timmis and L. N. D. Castro, *Artificial Immune Systems: A New Computational Intelligence Approach*. London,
397 London .uk: Springer Verlag, 2002.
- 398 [15] D. Dasgupta, S. Yu, and F. Nino, "Recent Advances in Artificial Immune Systems: Models and Applications,"
399 *Applied Soft Computing Journal*, vol. 11, no. 2, p. 1574–1587, Mar. 2011.
- 400 [16] I. Idris, "Model and Algorithm in Artificial Immune System for Spam Detection," *International Journal of*
401 *Artificial Intelligence & Applications (IJAI)*, vol. 3, no. 1, Jan. 2012.
- 402 [17] A. Somayaji, S. Hofmeyr, and S. Forrest, "Principles of a Computer Immune System," in *New Security*
403 *Paradigms Workshop*, 1998, p. 75–82.
- 404 [18] S. Sarafijanovic and J. -Y. L. Boudec, "Artificial Immune System For Collaborative Spam Filtering," in
405 *Proceedings of NICSO 2007, The Second Workshop on Nature Inspired Cooperative Strategies for Optimization*, vol.
406 129, Acireale, Italy, 2007, pp. 39-51.

- 407 [19] Z. Ji and D. Dasgupta, "Revisiting negative selection algorithms," *Evolutionary Computation*, vol. 15, no. 2, p.
408 223–251, 2007.
- 409 [20] S. Balachandran, "Multi-Shaped Detector Generation Using Real Valued Representation for Anomaly Detection,"
410 Master Thesis, University of Memphis, December 2005.
- 411 [21] P. Graham. (2003, Jan.) A plan for spam. [Online]. <http://www.paulgraham.com/spam.html> Last access
412 Oct.,2012
- 413 [22] T. O. a. T. White, "Immunity from spam: An analysis of an artificial immune system for junk email detection,"
414 Lecture Notes in Computer Science , 3627, 2005.
- 415 [23] TREC 2007 Spam Track Public Corpora. [Online]. <http://plg.uwaterloo.ca/~gvcormac/treccorpus07/>Last access
416 Oct., 2013
- 417 [24] D. Shih and C. ., H. Jhuan, "A study of mobile SpaSMS filterig," in *The XVIII ACME International Conference*
418 *on pacific RIM management*, Canada, July 24-July 26, 2008.
- 419
- 420
- 421