

# A Multi-layer Model to Detect Spam Email at Client Side

Kamini (Simi) Bajaj<sup>(✉)</sup>

School of Computing, Engineering and Mathematics,  
Western Sydney University, Parramatta, Australia  
k.bajaj@westernsydney.edu.au

**Abstract.** A solution to spam emails remains elusive despite over a decade long research efforts on spam filtering. Among different spam detection mechanisms that have been proposed, Naïve Bayesian Content Filtering has been very popular and has attained a reasonable level of success. SpamBayes is one such content filtering spam detection tool based on Naïve Bayesian classification using textual features. It is easy to deceive the learning techniques focusing only on textual attributes. Hence, in this paper we propose a multi-layer model that imposes, on top of SpamBayes, a second layer of non-textual filtering that exploits alternative machine learning techniques. This multi-layer model improves the accuracy of classification and eliminates the grey email into spam and ham emails. The experimental results of this model are quite encouraging.

**Keywords:** SpamBayes · Client based email filtering · Email spam · Content filtering · Supervised learning

## 1 Introduction

Spam exists in various forms such as spam email, web spam [1, 2], spam SMS [3, 4], and social spam [5]. Oxford dictionary defines spam as irrelevant messages sent on the internet to a large number of recipients.

The spam emails in any user's inbox has taken many forms such as phishing, image spam, DOS attacks, and malware distribution. It has impacted users and organizations from simple annoyance, loss of productivity, loss of personal information, system crashes to financial losses. Spam has varied from 36–95% [6–10] in more than a decade, the highest being 96% in 2010 [11] when this problem was at its peak and has reduced to about 53% in April–June 2015 [12, 13].

Though email is a form of communication for most these days, majority of the email traffic comes from business emails which account for over 116 billion emails sent and received per day [14] in 2015. This trend is going to continue and emails remain predominant form of communication in the business world [14]. Therefore, at least 58 billion emails sent and received daily are spam. For more than a decade efforts have been put into controlling the issue of spam. Various solutions such as blacklists, white lists, grey lists, content filtering, AIS (Artificial Immune Systems) filtering, reputation based filtering, content filtering (at mail server and email client) techniques have been

suggested [15–23], however the statistics above indicate that the issue is still ongoing and the area is open to further research.

Among the above mentioned techniques, machine learning techniques [24–26] have gained a reasonable amount of success and popularity in content filtering [27] both at mail server and client side. When applying filtering solutions at mail server, it is important to consider the following two points. Firstly, the filtering is being applied to all the emails being received on behalf of email users of the organization. Secondly, the same email may be spam to one user and not spam to some other user. Therefore, if the level of filtering at server side is very stringent, it would lead to a high number of false positives (FP: legitimate emails tagged/classified as spam by the filter). FP causes loss of important information. On the other hand if filtering at server side is too relaxed it would lead to high number of false negatives (FN: spam email that is classified/tagged as legitimate email) which is a source of annoyance for the user. Hence, another level of filtering at client’s end should be applied. In summary, server side mail filtering is not enough to classify incoming emails correctly, and client side filtering is essential. Many tools, both open source and commercial exist as an add-on to give another level of filtering at the client side. The major focus of these tools is to filter spam email that escapes the mail server filter. For this research, we are focusing on client level filtering that is at an optimum level of accuracy to reduce FP and FN. Earlier experiments have found that training this tool with user preferred training data reduces the FN in user inbox by 86% [28].

Since Naïve Bayesian Content Filtering has been very popular and achieved some level of success [22, 29–32], we explored Naïve Bayesian implementation. Many open source tools based on Naïve Bayesian classification techniques exist; one of the existing tools called SpamBayes was chosen to analyze the performance at client level filtering. SpamBayes classifies emails using text features into three categories: spam, ham and unsmes. Unsmes is an email that lies between the threshold values called the grey area and is not clearly classified as spam or ham. It contains features that belong to both spam and ham. From now on, we would refer to unsmes as grey in this paper. An example spam email that SpamBayes would successfully classify is given in Fig. 1. It would be able to identify the words such as ‘information’, ‘\$2 million’, ‘1–800’ as spam words and classify the email a spam.

```
Subject: re : information requested
hi , name is john ' m 27 years old . was able $ 2 million
working home , 'd share did . please few moments busy life
listen short message tell ! call listen , 1-800 - 764-6203
change life !
```

**Fig. 1.** Sample spam email with text features

Spammers keep innovating new ways to deceive the filters. The content of the spam has evolved to contain more than just words such as links, numeric digits, special characters etc. Most of these features are non-textual as shown in the sample spam (Fig. 2) and would not be identified by SpamBayes or any textual based filtering

```

Subject: free promotional offer
' ' own 100 % free web site site : http : / / 000000138 .
0000127 . 000044 . 00000005 . cearth . . ca / users /
freewebsites / * * * charge * * * * * * commitment * * * *
* * problem * * * opportunity s33kers internet m@rketers
small lagre site is . site linked thousands web sites ?
amazing site . . . http : / / 000000000138. 000027 . 44 . 5
. cearth . . ca / users / freewebsites / * * * charge * * *
* * * commitment * * * * * * problem * * * is truly going
site century ! * * * * * * * * * * * * * * * * * * * * * *
please excuse intrusion . one fr33 offer mailing * * * * *
* * * * * * * *

```

**Fig. 2.** Sample spam with non-text features

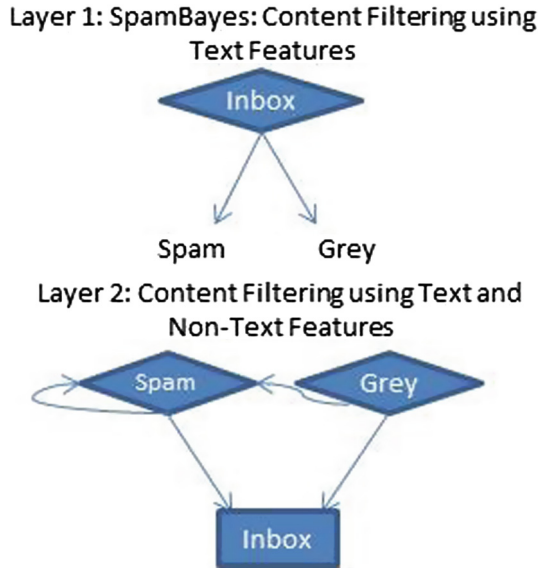
mechanism. Therefore, as suggested in [33], to improve the performance of SpamBayes we decided to introduce non-textual features.

The introduction of non-textual features is also testified by the Spam Reports published by Kaspersky labs for Quarter 2, 2015 [12] which highlighted the variation in features identified in spam emails that spammers are using to deceive the filtering solutions. These features listed are modified IP addresses, presence of upper case and lower case letters, special characters, use of number symbols, mis-spelt words, and number of links used to go to spam resources. In order to identify potential non-textual features, we analyzed the spam datasets and identified list of potential non-textual features. Subsequently to select the optimal non-textual features, we ran the program and observed the performance on the test set using F1 score and selected the features that gave high performance.

We explored the possibility to change the token type, size and thresholds in SpamBayes. To analyze the performance of the tool, we conducted further experiments to monitor its performance with various thresholds, token types (unigrams, bigrams and trigrams) and token sizes (15 to 25000 for different token types) which is elaborated in Sect. 2.2. We found that the overall performance would not substantially improve by increasing token type and size. Thresholds can be modified by the user as per need in SpamBayes.

In this paper we propose a multi-layer model that firstly builds an attribute set using many non-text features (Table 3) along with text features as a frequency matrix as shown in Fig. 3. Various non-text features such as number of link symbols, number of mis-spelt words, over use of numeric characters provide significant information about emails.

The aim of the model is to increase user productivity by not losing important emails as FP in greys and spam. This model eliminates unspams by classifying them as spam or ham. It also removes any FP by verifying the emails identified as spam in the junk folder. This model is based on CART, SVM,  $k$  nearest neighbor and Logistic regression machine learning techniques that have been used for spam categorization and classification [34, 35]. It has been tested with 10 datasets and results show that it has achieved 99+ % correct classification, with FP is as low as 0% to highest being 0.8%, at the max averaging at 0.3–0.4%.



**Fig. 3.** Flowchart showing 2 layer filtering process

The paper is structured as follows. Section 2 introduces Spam Bayes in Subsect. 2.1 followed by performance testing of SpamBayes using various token types, max-discriminators (token sizes) and thresholds in Subsect. 2.2. Results of SpamBayes experiments are reported in Sect. 2.3 along with the discussion justifying the need for a model to improve the performance of SpamBayes. Section 3 focuses on the multi-layer model and its integration with SpamBayes framework to eliminate greys from Spam-Bayes with the acceptable level of FP and FN. The experimental results to validate the model are reported in Sect. 4 followed by conclusion and future work in Sect. 5. In this paper, we would refer to spam email as spam and a legitimate email as ham.

## 2 Spambayes Performance Testing

### 2.1 SpamBayes

SpamBayes<sup>1</sup> is an open source content filtering tool that classifies emails on the basis of Naïve Bayesian techniques and can be installed as an add-on to the users email client such as Microsoft outlook. It builds the learning model from the training data and classifies the new incoming emails into three categories-spam, ham and grey (unsure). To classify an email, SpamBayes selects 150 significant unigram (single word) tokens called max\_discriminators from the header and body of an email and calculates the total spam score of the email. It uses the default threshold of 0.15 and 0.9 to classify

<sup>1</sup> <http://spambayes.sourceforge.net>.

emails. In SpamBayes manager a user can change the settings of thresholds such as ham and spam cut offs to suit their individual needs.

Readers are referred to [28] that elaborates background, training model and learning method of SpamBayes. Training data plays an important role in training the classifier. Our experiments on training the filter with user specific data [28] indicated that such training improves the performance of the tool. We noted similar trend during the experiments conducted while conducting this research as elaborated in Sect. 2.3.

## 2.2 Performance Testing Experiments

With an aim to find the optimal level of performance for SpamBayes, we conducted experiments using various datasets. As mentioned in Sect. 2.1, SpamBayes uses unigrams with 150 significant tokens based on the probability of occurrence of a particular token in the training data. To find the optimum performance parameters for SpamBayes we decided to conduct experiments using bigrams and trigrams with varying thresholds (Table 1) and tokens sizes (Table 2). Since getting access to live data is difficult, we decided to use publically available spam email datasets. The datasets chosen were Lingspam [36], PU1 [37], ENRON (divided into 6 preprocessed datasets since ENRON is very large dataset) [38], and CSDMC2010 SPAM corpus<sup>2</sup> available at csmining website. We also used dataset published at the Text REtrieval conference 2007 (TREC2007<sup>3</sup>), by University of Waterloo, Canada. In all, we used 10 datasets to conduct the experiments.

**Table 1.** Various thresholds

|              |     |      |     |     |     |
|--------------|-----|------|-----|-----|-----|
| Ham cut-off  | 0.5 | 0.15 | 0.2 | 0.3 | 0.8 |
| Spam cut-off | 0.5 | 0.9  | .9  | 0.8 | 0.6 |

**Table 2.** Various token sizes (max discriminators) used for 3 token types

| Unigram | Bigrams | Trigrams |
|---------|---------|----------|
| 15      | 150     | 150      |
| 50      | 500     | 500      |
| 75      | 5000    | 5000     |
| 150     | 10000   | 10000    |
| 200     | 20000   | 20000    |
|         | 25000   | 25000    |

Naïve Bayesian classification considers tokens as independent to each other; however correlations are possible between various tokens in an email. To identify these correlations we considered creating a correlation matrix. Since the size of matrix would

<sup>2</sup> <http://www.csmining.org/index.php/spam-email-datasets-.html>.

<sup>3</sup> <http://plg.uwaterloo.ca/~gvcormac/treccorpus07/>.

become very large, consume large amount of memory, and make the classifier performance very slow, we decided to consider the correlations between neighboring tokens by using bigrams and trigrams.

Bigrams are created only with neighbouring tokens. For example, if a, b and c are the three tokens then bigrams are ab, bc, a, b, and c. The frequencies of bigrams are calculated as the number of times they appeared in spam and ham emails. Trigrams are created in a similar manner. Subsequently, all tokens (normal, bigrams or trigrams) are sorted by their importance and only first max\_discriminators are taken for calculating the score of an email.

We divided each dataset randomly in a 70–30 ratio; 70% used for training and 30% used for testing. For each parameter set, we ran 20 iterations randomly selecting emails for training and testing. For example, for unigrams, token size 15 and thresholds as 0.5–0.5 for spam and ham cut off, 20 iterations for each dataset was conducted with emails selected randomly allocated into the pool of training and testing set for each iteration. Results were recorded and averaged for FP, FN and Grey rates along with the time taken to process for every parameter set. There were totals 85 parameter sets –25 for unigrams, 30 each for bigrams and trigrams.

### 2.3 Results and Discussion – Spam Bayes

Initial results indicated inconsistent behavior among various datasets. The value of FP, FN and grey varied for all 5 datasets. Different data sets have different optimum values for cutoffs and max discriminators. This may mean that each data set is a bit different since data sets belong to different times, probably have different styles of both spam and ham emails belonging to different authors and spam designs.

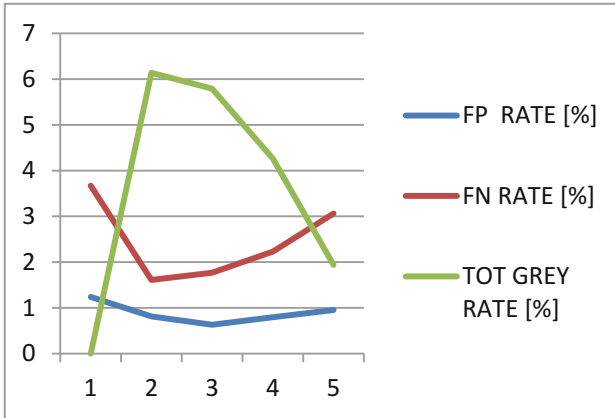
In this light, we can conclude that training data set has great impact on choice of parameters of a classification [28]. So we decided to run experiments with mix of data from all of the datasets and record the results. Mixed dataset namely, data from all the datasets that are messages belonging to different persons, involving different authors, styles, written in different times, showing different notions of what is spam, etc., thus preventing the classifier to “find the rule” for classification.

From the results for unigrams for mixed dataset shown in Fig. 4, it is evident that amount of greys are high, implying unacceptable loss of important emails.

The performance of the tool was monitored based on reduction in FP and FN verses the time it takes to achieve these values of FP and FN. Though we found that using bigrams with higher token sizes improved the performance of the tool with thresholds of .4 and .6, it was taking longer to process. Increasing the number of tokens (max\_discriminators) didn't contribute much to the performance of the classifier as shown in Fig. 5.

Hence, we conclude that though the values of unres was high for the default thresholds of .15 and .9 contributing to more FP and FN for default 150 token size, the scores were comparable for unigrams, bigrams and trigrams as shown in Fig. 4. Therefore, we decided to focus on improving the performance of SpamBayes for unigrams with token size of 150.

It is known that merely increasing the number of features does not necessarily (and usually does not) provide better models in machine learning. This is verified by the



**Fig. 4.** Rates for unigrams for mixed dataset

results obtained from our experiments. The trick is to find only a few most important features. Hence, there is a need to find the optimum text features and to consider the non-text features to be added to SpamBayes as stated in [33].

## 3 Multi-layer Model - A Hybrid Classifier

### 3.1 Definitions

#### 3.1.1 Text Features

Text Features help user make sense of what they are reading. It generally comprises of the actual text in the document that contributes towards user understanding the content and context of the document. They are the building block of the document that enhances the comprehension.

#### 3.1.2 Non-text Features

Non-text features contribute to the information about the document such as the size and structure of the document. It also includes the features such as illustrations, labels, subtitles, table of contents, glossary, maps, index, comparisons etc. as shown in Fig. 6. In case of emails non-text information would include date and time of an email, subject field, hyperlinks, numeric digits, word count, use of special characters, etc. (Fig. 6).

#### 3.1.3 Machine Learning Techniques

**CART.** Classification And Regression Trees (CART) algorithm is a supervised learning techniques for prediction, and classification. It constructs decision trees based upon attributes that belong to predefined classes from a collection of training data. For CART to construct these regression trees, we must define the list of attributes and the number of classes as an outcome. It then uses the training data with assigned classes to construct the rules via the decision trees. These decision trees are then used to classify new data into the said classes defined.

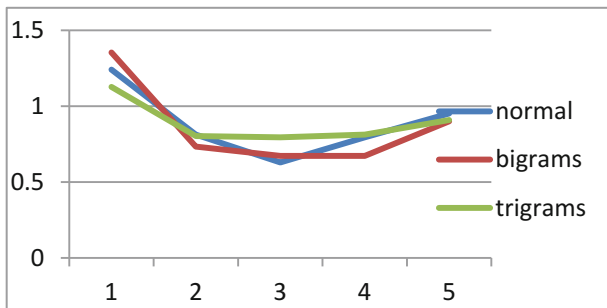


Fig. 5. Results for three token types for 150 max\_discriminators for mixed dataset



Fig. 6. Non Fiction Text Features for a document

For our case the attributes defined are text as well as non-text attributes are defined in Sect. 3.1.2 and the classes are spam and ham. The training data is the data used to train SpamBayes and for experiments purpose the datasets listed in Sect. 2.2.

**SVM.** Support vector machines [34] (SVMs) are a set of supervised learning methods used for classification, regression and detection. Unlike normal classification methods, SVM uses a subset of training points called support vectors and finds a boundary that has maximum margin by solving an optimization problem. In SVM, we tried to find a hyperplane in an n-dimensional space defined by the attributes of the emails in the training data. One side of this hyper plane is spam emails and other side is ham emails. Since support vector machine are effective not only in high dimensional spaces, but



also effective in cases where number of dimensions is greater than the number of samples.

*knn.* *k*-nearest neighbors is a supervised learning method that, as its name suggest, uses the labels of *k*-nearest training data points to figure out the label of the test data point under consideration. Once again we imagine all the vectors as points in a space and the distance being Euclidean distance.

KNN can be used for both classification and regression. In case of classification, we take the majority vote of *k*-nearest neighbors to find the label of a test data point. For regression, we take the average of the values for the *k*-nearest of a point to find the response for a new test point. In order make the regression more accurate, the response of the nearer neighbors is given more weight compared to the farther points. Usually, the weight given to points decrease inversely with their distance from the test point.

*Logistic Regression.* Logistic regression is a common classification technique used in situations where there is not much need to be very deterministic about the predictions made.

It works by maximizing likelihood, i.e. maximizing  $P[y|X]$  where *X* is the feature matrix each row of which is a feature vector and *y* is the vector of labels, one element for each row in *X*. We train the model such that it learns the probability distribution of the labels over the set of attributes.

Logistic regression is called regression even though we use it for classification because we try to approximate a real continuous function in this case. This function is the probability of getting a label; let's say the label 'spam', given a feature vector of an email. Since the probability function is continuous and real as opposed to discrete, it's called logistic regression.

Training a model for logistic regression involves defining an error measure. An error is a value whose magnitude tells us how far we are from the learned model that would predict correctly on the training data.

Once we have defined the error measure, the problem of learning the model is translated into an optimization problem wherein we have to reduce the error measure while changing the variables that it depends upon.

An example of error measure is that of likelihood, i.e. how likely are we to generate the training responses from the training features.

### 3.2 Multi-layer-Model

The proposed multi-layer model, a hybrid classifier is based on supervised alternative machine learning techniques applied to the features selected using text and non-text components of the email. This classifier carries out supervised learning, extracts text as well as non-text features from training data and applies that learning to detect and classify new email documents.

The emails that arrive at the email client are first filtered by the mail server. The filtering mechanism at the mail server marks an incoming email as ham, sends it to the Inbox or as spam, and sends to the Junk Mail. Some other work has gone into this area of adaptive, multi-stage learning systems to filter email spam using a variety of

machine learning techniques [18, 39] but none of them provide 2 layer filtering at the client side. SpamBayes provides a layer (Layer 1) of filtering that reclassifies the emails in Inbox and Junk mail as spam, ham and grey, landing ham in Inbox, greys in Junk Suspect and spam in Spam Folder. The proposed multi-layer model applies yet another layer (Layer 2) of filtering as shown in Fig. 3, to the greys as well as spam classified by SpamBayes to increase the precision and accuracy of classification since both would contain FP and FN and the aim is to reduce FP. It is very important that this model achieves a high level of performance (correct classification of spam and ham or least FP). To ensure this we analyzed the data sets to carefully identify the features that provide correct classification. The following sub-sections explain the selection of text and non-text features and how they were combined to build the multi-layer model.

### Text Features

To extract text features from the training data, bag-of-words approach has been applied to transform data into numerical features that can be used for machine learning techniques. Term frequency-inverse document frequency (tf-idf) has been applied to this bag of words. To do so tf-idf reflects the importance of each word related to a document in the training documents. The value of tf-idf increases proportionally to number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general in the training documents.

This gives us sparse Document-Term Matrix with a huge number of columns. To apply machine learning algorithm to such a matrix, we have to filter these columns. We tried different methods such as sparse LSA, mutual information and chi-square test for filtering and found that chi-square test works well. Using this method we filter 1000–2000 columns of the document-term matrix which is then used as feature for classification.

### Non-text Features

To extract non-text features, analysis of the selected datasets was conducted to identify the possible list of attributes. To select the correct features cross validation of manual analysis was done with classifier. The right feature is the feature that helps the classifier to improve its performance i.e. if we run the classifier without feature and the F1-score is  $F_{without}$  and then run it with feature and the score is  $F_{with}$ , then the feature is the “right” feature if  $F_{with} > F_{without}$  with statistical significance. This corresponds to suggestions made in [33]. Features set containing non-text features was developed as given in Table 3.

For combining features, we used a statistical non-parametric learning technique called Gradient Boosting Regression Trees classifier that gives the highest score. It’s one of the strongest methods in machine learning for classification. After extracting the features from training data and combining them using Gradient boosting, we tested the performance using F-1 score.

Once the features extraction is completed, the model then uses those features as attributes for classification. The model contains methods such as CART, Support vector machine (SVM), k-Nearest Neighbor and Logistic Regression. These methods utilize the decision boundaries that they identify from the training data and apply for classification. The detailed description of how these methods are applied has been provided

**Table 3.** List of non-text attributes

|                                    |
|------------------------------------|
| 0: email header and body lengths,  |
| 1: number of abnormal symbols,     |
| 2: number of numeric characters,   |
| 3: number of punctuation symbols,  |
| 4: number of links symbols,        |
| 5: ‘number of keywords’,           |
| 6: number of keyword ‘unsubscribe’ |
| 7: ‘send time in 8:00:00–18:59:59, |
| 8: length of subject field,        |
| 9: mis-spelt word count            |
| 10: similar to abnormal words      |
| 11: maximum run length of capitals |
| 12: average run length of capitals |

in Sect. 3.1. Each of these methods individually classifies the email spam and greys classified by Spam Bayes into clear spam and ham. The multi-layer model further uses rules on these classifications to predict an email as spam or ham. The rules are defined by the following voting system to classify an email document:

[all four methods agree]: outcome is the agreed classification decision

[Three methods agree]: outcome is the agreed classification decision

[Two methods agree]: outcome is Ham. This outcome has been chosen with an aim to reduce FP as some degree of FN is acceptable whereas FP is not acceptable.

### 3.3 SpamBayes Framework with Multi-layer Model

Since SpamBayes results in classification of emails into three categories, we apply the multi-layer model to the emails tagged as greys and spam. The rationale behind doing this has been explained in Sect. 2. The integration of the multi-layer model – a hybrid classifier to SpamBayes aims to eliminate greys category from the outcome classification Fig. 7.

## 4 Experiments and Results - Multi Layer Model

The experiments were conducted to test the multi-layer model at two levels. At the first level, we conducted experiments to test the performance of the multi-layer model using the datasets and at the second level after integrating the multi-layer model with SpamBayes in order to classify the greys and reclassify the ‘spams’ moved to the junk folder by the SpamBayes. The model was tested with same 10 datasets as mentioned in Sect. 2.2 to measure the performance in terms of FP and FN. Once a satisfactory level of performance was achieved with the multi-layer model on its own, the integrated SpamBayes Framework (Fig. 7) was tested with the same set of 10 datasets.

The performance of new SpamBayes framework was measured with respect to % of ham, greys and FP in SpamBayes.

The results showed that the multi-layer model improves the performance of SpamBayes by reducing the overall % of FP to less than 0.2. This means that model is performing at 99.8% which is very encouraging improvement (Table 4).

**Table 4.** Comparison of SpamBayes and multi-layer model showing % of FP, FN and greys

| Dataset name | Percentage of | SpamBayes % | Multi-layer model % |
|--------------|---------------|-------------|---------------------|
| CSDMC2010    | FP            | 0           | 0                   |
|              | FN            | 0           | 1.2                 |
|              | Ham greys     | 1.01        | 0                   |
| ENRON1       | FP            | 1.3         | 0.9                 |
|              | FN            | 0           | 2.3                 |
|              | Ham greys     | 5.1         | 0                   |
| ENRON2       | FP            | 0           | 0.2                 |
|              | FN            | 0.1         | 2                   |
|              | Ham greys     | 0.8         | 0                   |
| ENRON3       | FP            | 0.1         | 0.06                |
|              | FN            | 0.1         | 4.6                 |
|              | Ham greys     | 1.2         | 0                   |
| ENRON4       | FP            | 0.1         | 0.2                 |
|              | FN            | 0.3         | 0.9                 |
|              | Ham greys     | 0.2         | 0                   |
| ENRON5       | FP            | 0.1         | 0.4                 |
|              | FN            | 0.2         | 0.3                 |
|              | Ham greys     | 0.8         | 0                   |
| ENRON6       | FP            | 0.2         | 0.6                 |
|              | FN            | 0.3         | 0.7                 |
|              | Ham greys     | 1.3         | 0                   |
| Lingspam     | FP            | 0           | 0.1                 |
|              | FN            | 0           | 1.4                 |
|              | Ham greys     | 1.7         | 0                   |
| PU1          | FP            | 0.9         | 0.3                 |
|              | FN            | 0           | 5.4                 |
|              | Ham greys     | 4.8         |                     |
| TREC07       | FP            | 0           | 0                   |
|              | FN            | 2.9         | 2.9                 |
|              | Ham greys     | 0           |                     |

We also calculated the % improvement for each of the datasets for overall FP which includes ham greys in SpamBayes that contributes towards FP. Figure 8 below shows the percentage improvement multi-layer model bring to SpamBayes for each dataset.

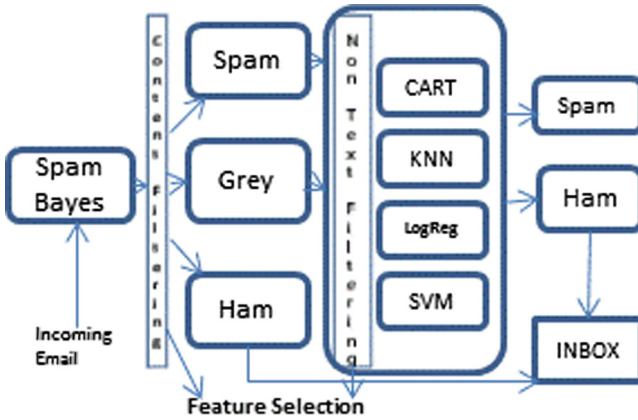


Fig. 7. Multi-layer model integrated with Spam Bayes

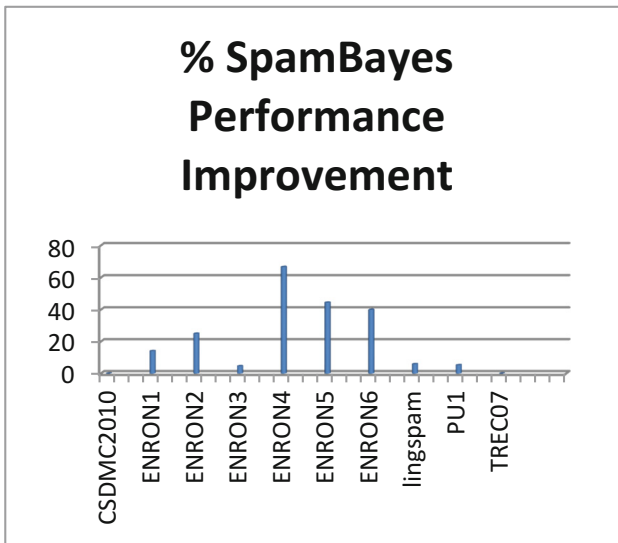


Fig. 8. % performance improvement the multi-layer model brings to SpamBayes

## 5 Conclusion and Future Work

Spam is an annoying and causes financial damage to organizations and individual users. This paper focused on supervised machine learning techniques based ensemble and its implementation via an open source tool called SpamBayes. As a base point we tested the performance of SpamBayes with various parameters such as different settings for thresholds and token size as well as the characteristics of feature sets such as unigrams, bigrams and trigrams of different sizes ranging from 75 to 20000 and noted that there is room for improvement. SpamBayes classifies a new email as spam, ham or

grey. Greys leave an area that user has to manually classify to eliminate the false positives and false negatives. Results reported showed that for optimum parameters that give the least amount of FP and FN, the size of greys needs to be reduced. A multi-layer model was proposed to eliminate the greys from SpamBayes. This model applied to SpamBayes framework was tested and results are reported. We would like to further compare these results of multi-layer model to some other models against the same datasets.

We believe the application of this multi-layer can also be applied for social network analysis. In our future work, we would like to extend the application of this multi-layer model to social networks such as Twitter, Facebook and alike.

## References

1. Wang, D., Irani, D., Pu, C.: A social-spam detection framework. In: Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference 2011, pp. 46–54. ACM, Perth (2011)
2. Bo, L., et al.: Filtering spam in social tagging system with dynamic behavior analysis. In: International Conference on Advances in Social Network Analysis and Mining, ASONAM 2009 (2009)
3. Wang, C., et al.: A behavior-based SMS antispam system. *IBM J. Res. Develop.* **54**(6) (2010)
4. Europe, M.: AdaptiveMobile launches anti-spam app for operators. *News Analysis* (2011)
5. Heymann, P., Koutrika, G., Garcia-Molina, H.: Fighting spam on social web sites: a survey of approaches and future challenges. *IEEE Internet Comput.* **11**(6), 36–45 (2007)
6. Clifford, M., et al.: Miracle cures and toner cartridges: finding solutions to the spam problem. In: 19th Annual Computer Security Applications Conference (ACSAC 2003) (2003)
7. Jung, J., Emil, S.: An empirical study of spam traffic and the use of DNS black lists. In: Internet Measurement Conference, Taormina, Italy (2004)
8. Jaeyeon, J., Emil, S.: An empirical study of spam traffic and the use of DNS black lists. In: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement 2004. ACM, Taormina (2004)
9. Leavitt, N.: Vendors fight spam's sudden rise. *Computer* **40**(3), 16–19 (2007)
10. Mikko, S., Carl, S.: Effective anti spam strategies in companies: an international study. In: 39th Hawaii International Conference on System Sciences. IEEE (2006)
11. Gina Spam statistics of third-quarter 2010. Panda Security Report (2010)
12. Shcherbakova, T., Vergelis, M., Demidova, N.: Spam and phishing in Q2 2015. *Quarterly Spam Reports* (2015)
13. Symantec, Symantec Intelligence Report: July 2015, p. 19, July 2015
14. Levenstein, J.: Email statistics report, 2013–2017. In: Radicati, S. (ed.) Reports. Radicate Group Inc.: 1900 Embarcadero Road, Suite 206, Palo Alto, CA 94303 (2013)
15. Meyer, T.A., Whateley, B.: SpamBayes: effective open-source, Bayesian based, email classification system. In: First Conference on Email and Anti-Spam (CEAS), Mountain View, CA (2004)
16. Pelletier, L., Almhana, J., Choulakian, V.: Adaptive filtering of spam. In: Proceedings of the Second Annual Conference on Communication Networks and Services Research (2004)
17. Saito, T.: Anti-spam system: another way of preventing spam. In: Proceedings of the Sixteenth International Workshop on Database and Expert Systems Applications (2005)

18. Yan, Z., Mulekar, M.S., Nerellapalli, P.: Adaptive spam filtering using dynamic feature space. In: 17th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2005 (2005)
19. Lai, C.-C.: An empirical study of three machine learning methods for spam filtering. *Knowl. Based Syst.* **20**(3), 249–254 (2007)
20. Youn, S., McLeod, D.: A comparative study for email classification. In: Elleithy, K. (ed.) *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, pp. 387–391. Springer, Netherlands (2007)
21. Karthika Renuka, D., et al.: Spam classification based on supervised learning using machine learning techniques. In: *International Conference on Process Automation, Control and Computing (PACC 2011)* (2011)
22. Zhang, Y., Yang, X., Liu, Y.: Improvement and optimization of spam text filtering system. In: *2nd International Conference on Computer Science and Network Technology (ICCSNT 2012)* (2012)
23. Sarafijanovic, S., Le Boudec, J.-Y.: Artificial immune system for collaborative spam filtering
24. Krasnogor, N., et al. (eds.): *Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)*, pp. 39–51. Springer, Berlin (2008)
25. Balakumar, M., Vaidehi, V.: Ontology based classification and categorization of email. In: *International Conference on Signal Processing, Communications and Networking, ICSCN 2008* (2008)
26. Chih-Chin, L., Ming-Chi, T.: An empirical performance comparison of machine learning methods for spam e-mail categorization. In: *Fourth International Conference on Hybrid Intelligent Systems, HIS 2004* (2004)
27. Caruana, G., Li, M.: A survey of emerging approaches to spam filtering. *ACM Comput. Surv.* **44**(2), 1–27 (2012)
28. Blanzieri, E., Bryl, A.: A survey of learning-based techniques of email spam filtering. *Artif. Intell. Rev.* **29**(1), 63–92 (2008)
29. Bajaj, K., Pieprzyk, J.: A case study of user-level spam filtering. In: *Proceedings of the Twelfth Australasian Information Security Conference*, vol. 1492014, pp. 67–75. Australian Computer Society, Inc., Auckland
30. Wu, J., Deng, T.: Research in anti-spam method based on bayesian filtering. In: *Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACIIA 2008)*, pp. 887–891 (2008)
31. Wei, Z., et al.: Active learning based spam filtering method. In: *2010 8th World Congress on Intelligent Control and Automation (WCICA)* (2010)
32. Hu, Y., Zhang, C.: An improved bayesian algorithm for filtering spam e-mail. In: *2011 2nd International Symposium on Intelligence Information Processing and Trusted Computing (IPTC)* (2011)
33. Seibel, P.: Practical: a spam filter. In: *Practical Common Lisp*, pp. 291–309 (2005)
34. Bajaj, S.K., Pieprzyk, J.: Can we CAN the email spam. In: *2013 Fourth Cybercrime and Trustworthy Computing Workshop (CTC)* (2013)
35. Drucker, H., Donghui, W., Vapnik, V.N.: Support vector machines for spam categorization. *IEEE Trans. Neural Netw.* **10**(5), 1048–1054 (1999)
36. Chao, X., Yiming, Z.: Transductive support vector machine for personal inboxes spam categorization. In: *International Conference on Computational Intelligence and Security Workshops, CISW 2007* (2007)
37. Androusoyopoulos, J.K., Chandrinou, K.V., Paliouras, G., Spyropoulos, C.D.: An evaluation of naive bayesian anti-spam filtering. In: *11th European Conference on Machine Learning, Barcelona, Spain* (2000)

38. Androutsopoulos, J.K., Chandrinos, K.V., Spyropoulos, C.D.: An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In: 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000), Athens, Greece (2000)
39. Metsis, V., Androutsopoulos, I., Paliouras, G.: Spam filtering with naive bayes—which naive bayes? In: 3rd Conference on Email and Anti-Spam (CEAS 2006), Mountain View, CA, USA (2006)
40. Islam, M.R., Wanlei, Z., Chowdhury, M.U.: Email categorization using (2 + 1)-tier classification algorithms. In: Seventh IEEE/ACIS International Conference on Computer and Information Science, 2008. ICIS 2008 (2008)