Optimized Incremental SVM based Classifier for Spam Filtering using Internet Acronyms

Indrajeet Singh Jhala
M. Tech Scholar
Department of Computer Science and Engineering
Srinathaji Institute of Technology and Engineering,
Nathdwara, Rajsamand-313301 (Rajasthan)

Dr. Pankaj Dalal
Professor
Department of Computer Science and Engineering
Srinathaji Institute of Technology and Engineering,
Nathdwara, Rajsamand-313301 (Rajasthan)

Abstract

The word Spam as applied to email means Unsolicited Bulk Email. Unsolicited means that the Recipient has not granted verifiable permission for the message to be sent. Bulk means that the message is sent as part of a larger collection of messages, all having substantively identical content. A message is Spam only if it is both Unsolicited and Bulk. Spam is a key problem in electronic communication, including large-scale email systems and the growing number of blogs. The anti-spam community has been divided on the choice of the best machine learning method for content-based spam detection. In many traditional machine learning applications, SVMs are applied in batch mode. That is, an SVM is trained on an entire set of training data, and is then tested on a separate set of testing data. Spam filtering is typically tested and deployed in an online setting, which proceeds incrementally. In this paper, an online incremental SVM is proposed which classifies emails as spam or ham (normal mail). The traditional textual features are augmented with character level Features, thus, extending the feature set used to train the SVM. This approach exploits the fact that most of the user written text on internet in the form of mails, tweets etc. consists of shorthand notations. The proposed model maps the popular internet shorthand notations to the corresponding words, thereby making a much more accurate classification as compared to traditional approaches. The proposed model is tested on real benchmark data set and performance is evaluated. The empirical results reveal that the proposed SVM, although is computationally heavier, nevertheless provides an improvement in classification accuracy as compared to those based only on textual features.

Keywords: Machine Learning, Spam Filtering, Support Vector Machines, Character Level Features

I. INTRODUCTION

A. Spam Filtering

Spam is a prime problem in electronic communication, together with large-scale email systems and the rising number of blogs. Content-based filtering is one consistent method of combating this risk in its various forms, but some academic researchers and industrial practitioners conflict on how best to filter spam. The former have advocated the use of Support Vector Machines (SVMs)[1] for content-based filtering, as this machine learning method gives high-tech performance for text classification. Though, similar performance gains have yet to be established for online spam filtering. Furthermore, practitioners cite the high cost of SVMs as reason to favor faster Bayesian methods.

The spam filtering problem has usually been presented as an example of a text categorization problem with the categories being ham and spam. Ham refers to normal mails sent as a one-to-one message, in contrast to spam which refers to unsolicited mails sent in bulk. Actually, the structure of email is better than that of flat text, with meta-level features such as the fields present in MIME compliant messages [2]. Researchers have presently acknowledged this, setting the problem in a semi-structured document classification framework. Numerous solutions have been proposed to conquer the spam problem. Along with the anticipated methods, much interest has paid attention on the machine learning techniques in spam filtering. They comprise rule learning, decision trees, Naive Bayes [3], Support Vector Machines or combinations of different learners. The essential and common concept of these inductive approaches is that using a classifier to filter out spam and the classifier is learned from training data instead of constructed by hand. Generally spam filtering task is a constant work with email sequence increasing in size, there is a requirement to scale up the learning algorithms to handle more training data. And as it is time consuming to retrain the classifier whenever a new example is added to the training set, it is more proficient from a computational point of view to reduce the number of labeled examples to learn a function at a definite accuracy level. A realistic classification model for spam filtering must not only consider the fact that spam evolves over time, but also that labeling a large number of examples for initial training can be costly in terms of both time and money.

B. Support Vector Machine (SVM)

The main idea of Support Vector Machine is to build a nonlinear kernel function to map the data from the input space into a perhaps high-dimensional feature space and then simplify the optimal hyper-plane with utmost margin between the two classes. Given a T-element training set$\{(x^i, y^i): x^i \in R^D, y^i \in \{-1,1\}, i = 1 \dots \dots T\}$ a linear SVM classifier can be written as:

$$F(x) = \sum_i a_i y^i \, x \cdot x^i + b = x \cdot \sum_i a_i y^i \, x^i + b = x \cdot w + b \quad eq.1$$

Where $a_i \geq 0$, b is a bias term and D is the dimension of the input space; $\cdot$ is a dot-product operator, and w is the normal vector of the classification hyperplane [4]. Normally, the multipliers a_i have non-zero values only for a small subset of the training set, which is known as the support set and its elements the support vectors. The best hyperplane is found such as to make best use of the classification margin, given by $2/\|w^2\|$, where w denotes the normal vector of the hyper plane. The soft-margin optimization task is formulated as:

$$minimize: \sum_i C_i \, \varepsilon_i^p + \frac{1}{2} \|w^2\| \qquad equation.2$$

$$\text{subject to: } y(w \cdot x + b) \geq 1 - \varepsilon_i$$

where $C_i \geq 0$, $p \geq 0$ and $\varepsilon_i = \big(1 - y(w \cdot x + b)\big)$; $z_+ = z$ for $z \geq 0$ and is equal to 0 otherwise; The slack variables ε_i take non-zero values only for bound support vectors, i.e., points that are misclassified or lie inside the classifier's margin. In equation 2, the accurateness over the training set is balanced by the "smoothness" of the solution. If the case is considered where p=1, a number of highly proficient computational methods have been developed.

So far learning strategies are considered in which data is obtained passively. However, SVMs build a hypothesis using a subset of the data consisting of the most informative patterns and therefore they are high-quality candidates for active or discriminating sampling techniques which look for out these patterns. Suppose the data is primarily unlabeled, a good heuristic algorithm would mainly request the labels for those patterns which will become support vectors. Active selection would be chiefly important for practical situations in which the process of labeling data is costly or the dataset is big and unlabeled. Tong et al. [5] proposed the Simple algorithm, which iteratively chooses *h* unlabeled instances closest to the separating hyperplane to solicit user feedback. Based on this algorithm, a first spam filter f can be trained on the labeled email pool L. Then the f is applied to the unlabeled e-mail pool *U* to compute each unlabeled e-mail's distance to the separating hyperplane. The h unlabeled e-mails closest to the hyperplane and comparatively apart are selected as the next batch of samples for conducting pool-queries. Though, the Simple algorithm may select too many similar queries, which harm the learning performance. Diversity measure presented by Brinker is adapted to build batches of new training examples and enforces chosen examples to be diverse regarding their angles. The main proposal is to choose a collection of emails close to the classification hyperplane, while at the same time preserving their diversity. The diversity of email is calculated by the angles. For instance, suppose x_i has a normal vector equal to(x_i). The angle between two hyperplanes h_i and h_j, corresponding to the email instances x_i and x_j , can be written in terms of the kernel operator K: The angle-diversity algorithm starts with an initial hyperplane h_i trained by the given labeled email set L. Then, for each unlabeled email x_i, it computes the distance to the classification hyperplane h_i. The angle between the unlabeled x_i and the current set S is defined as the maximal angle from x_i to any other x_j in set S. This angle measures how diverse the resulting S would be, if xi were to be chosen as a sample. Algorithm angle-diversity introduces a parameter to balance two components: the distance from emails to the classification hyperplane and the diversity of angles among different emails. After that, the algorithm chooses as the sample the unlabeled email that has the smallest score in U. The algorithm repeats the above steps *h* times to select *h* emails. In practice, the complete method uses a weighted sum of diversity and hyperplane distance, controlled by a parameter λ, where λ= 0 is corresponding of focusing exclusively on diversity and λ= 1 is the same as Simple.

C. Online Learning for Spam Filtering

Online learning is an important domain in machine learning with interesting theoretical properties and practical applications. Online learning is performed in a sequence of trials. At trial t the algorithm first receives an instance $x_t \in R_n$ and is required to predict the label associated with that instance. After the online learning algorithm has predicted the label, the true label is revealed and the algorithm pays a unit cost if its prediction is wrong. The ultimate goal of the algorithm is to minimize the total number of prediction mistakes it makes along its run. To achieve this goal, the algorithm may update its prediction mechanism after each trial so as to be more accurate in later trials. In this paper, it is assumed that the prediction of the algorithm at each trial is determined by a SVMs classifier. Usually in SVMs only a small portion of samples have non-zero α_i coefficients, whose corresponding x_i (support vectors) and y_i fully define the decision function. Therefore, the SV set can fully describe the classification characteristics of the entire training set. Because the training process of SVM involves solving a quadratic programming problem, the computational complexity of training process is much higher than a linear complexity. Hence, if the SVM is trained on the SV set instead of the whole training set, the training time can be reduced greatly without much loss of the classification precision. This is the main idea of online learning algorithm. Given that only a small fraction of training emails end up as support vectors, the support vector algorithm is able to summarize the data space in a very concise manner

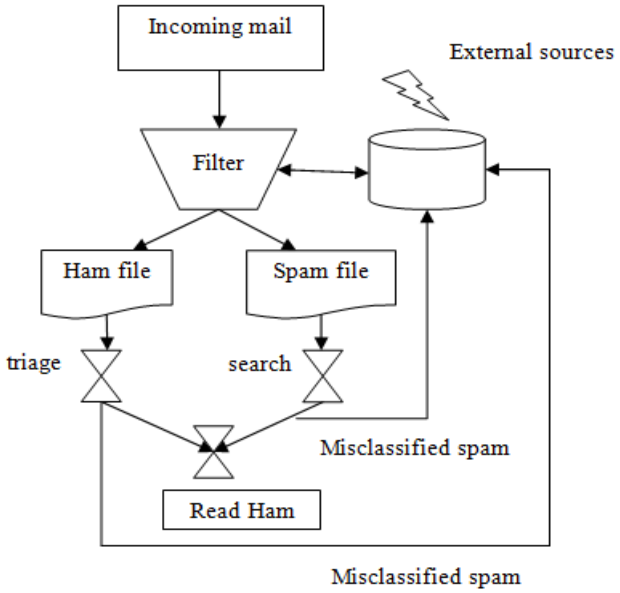

Fig. 1.1: Spam Filter Usage

The training would use only the historical SV samples and the incremental training samples in re-training. All non-SV samples are discarded after previous training.

D. Email Spam and Online SVMs

Unwanted email is not convenient, annoying and inefficient. Its quantity threatens to overcome the ability to distinguish desirable messages. An automatic spam filter can alleviate these problems, given that it acts in a consistent and expected manner, eradicates a large proportion of unwanted email, and create negligible risk of eliminating desirable email.

Figure 1.1 models spam filter deployment and use as it relates to an individual email recipient. Messages from an incoming email stream are presented to the spam filter, which classifies each as good email (ham) [6] or as indiscriminately sent unwanted email (spam). Messages classified as ham are placed in the recipient's mailbox (ham file) or quarantined or discarded (placed in the spam file). The recipient, in the normal course of reading the ham file, may notice spam messages (which have been misclassified by the filter) and may provide feedback to the filter noting these errors. From time to time the recipient may search the spam file for ham messages (which have also been misclassified) and may provide feedback on these errors as well. The filter may avail itself of external resources such as global statistics, collaborative judgments, network information, black lists, and so on. A perfect spam filter would avoid ham misclassification – incorrectly placing a ham message in the spam file – and spam misclassification – incorrectly placing a spam message in the ham file. Ham misclassification poses a serious risk; were the recipient to fail to retrieve the misclassified message from the spam file, it would be lost. The expected cost to the user is some combination of the probability of misclassifying a particular ham message, the likelihood of retrieving it from quarantine, and the value of the message. Spam misclassification, on the other hand, exposes to the recipient a degree of the original inconvenience, annoyance and risk associated with spam. An effective filter should mitigate the effects of spam while maintaining an acceptable risk of loss.

This research paper is organized as follows: Section 1 presents an overview of the subject matter. Section 2 provides the gives the problem statement and the approach for the research. Section 3 presents the proposed SVM based classifier for spam filtering. Section 4 gives the simulation results and the plots for various features for similarity measurement. Section 5 concludes the paper.

II. SCOPE AND MOTIVATION, PROBLEM STATEMENT AND RESEARCH APPROACH

In this study, the rapidly growing problem of spam e-mail is investigated. A classifier makes the binary decision whether an incoming e-mail is handled as spam or as desired e-mail. Text categorization is one of the major application areas of support vector machines (SVM). In an often cited experiment, Joachims (1998) achieved better results with an SVM algorithm than with standard methods in the classification of labeled Reuter's news stories that were to sort into categories such as money market, earning etc.

The problem of text categorization arises in many different areas such as preprocessing news agency stories and data mining. Nowadays, a prominent task is to classify incoming e-mails. An e-mail user wants his or her e-mail to be sorted to different folders according to their contents, filtering out undesired spam e-mail. In this paper, the classification of spam is focused using Support Vector Machine based classifier. SVM are proved much more accurate as compared to Bayesian Classifier for textual data analysis and classification. The goal is to develop a powerful online spam-filter that learns fully autonomously from

individual examples provided by the user, i.e. it learns to imitate its user's classification strategy. The feature set extracted for SVM training is extended using character level N Gram Features. Simulation is performed over real data set and the results indicate a considerable improvement in classification accuracy as compared to classifier trained only with textual features.
It is investigated that far support vector machines are suitable for filtering spam e-mail. Support vector machines have several advantages in comparison to other machine learning approaches and have been proven to be an efficient tool for classification of linguistically preprocessed text. A software program using R statistical package has been developed that learns the classification criteria of a benchmark data set and filters e-mail accordingly. The complete process is fully autonomous, based solely on sample e-mails. The contribution of this paper is the inculcation of shorthand internet notations widely used in the internet jargon. For example, f2f → Face-to-face, hru → How Are you, etc. Thus, proposed classifier classifies the spam using a much more extended feature set as compared to traditional content analysis techniques.

III. PROPOSED WORK

A. Online Learning Techniques

Many real-life machine learning problems can be more naturally viewed as online rather than batch learning problems. Indeed, the data is often collected continuously in time, and, more importantly, the concepts to be learned may also evolve in time. Significant effort has been spent in the recent years on development of online SVM learning algorithms. The elegant solution to online SVM learning is the incremental SVM which provides a framework for exact online learning. One should note, however, a significant restriction on the applicability of the above-mentioned supervised online learning algorithms: the labels may not be available online, as it would require manual intervention at every update step. A more realistic scenario is the update of the existing classifier when a new batch of data becomes available. The true potential of online learning can only be realized in the context of unsupervised learning. In this paper, an incremental model of learning is proposed to implement online SVM based on the manual classification of a batch of data as and when required.

An OnLine Support Vector Machine (OLSVM) based on the incremental SVM is proposed in this paper to classify the spam and ham. The proposed machine is trained on a small data set and then incrementally trained for new batches of manually classified data so as to train it further on a large dataset.

Online learning can be used to overcome memory limitations typical for kernel methods on large-scale problems. It has been long known that storage of the full kernel matrix, or even the part of it corresponding to support vectors, can well exceed the available memory. To overcome this problem, several sub-sampling techniques have been proposed. Online learning can provide a simple solution to the sub-sampling problem: make a sweep through the data with a limited working set, each time adding a new example and removing the least relevant one. This procedure results in an approximate solution. An experiment on the benchmark data presented in this paper shows that significant reduction of memory requirements can be achieved without major decrease in classification accuracy.

B. Incremental SVM

Following equation describes the abstract formulation of SVM optimization problem:

$$\max_{\mu} \min_{\substack{0<x<C \\ a^{T}x+b=0}} : W = -C^{T}x + \frac{1}{2}x^{T}kx + \mu(a^{T}x + b) \qquad \text{equation 1}$$

where C and a are $n \times 1$ vectors, K is a $n \times n$ matrix and b is a scalar. By defining the meaning of the abstract parameters a, b and c for the particular SVM problem at hand, one can use the same algorithmic structure for different SVM algorithms. In particular, for the standard support vector classifiers, take $c = 1$; $a = y, b = 0$ and the given regularization constant C; the same definition applies to the $v - SVC$ [] except that $C = \frac{1}{Nv}$ for the SVDD, the parameters are defined as: $c = \text{diag}(K)$; $a = y$ and $b = -1$. Incremental (decremental) SVM provides a procedure for adding (removing) one example to (from) an existing optimal solution. When a new point k is added, its weight x_k is initially assigned to 0. Then the weights of other points and μ should be updated, in order to obtain the optimal solution for the enlarged dataset. Likewise, when a point k is to be removed from the dataset, its weight is forced to 0, while updating the weights of the remaining points so that the solution obtained with $x_k = 0$ is optimal for the reduced dataset. The online learning follows naturally from the incremental/decremental learning: new example is added while some old example is removed from the working set.

C. Incremental SVM

The basic principle of the incremental SVM is that updates to the state of the example k should keep the remaining examples in their optimal state. In other words, the Kuhn-Tucker (KT) conditions:

$$g_i = -c_i + K_i; x + \mu a_i \geq 0, \text{if } x_i = 0 \qquad \text{eq. 2}$$
$$= 0, \text{if } 0 < x_i < C$$
$$\leq 0, \text{if } x_i = C$$

$$\frac{\partial W}{\partial \mu} = a^{T}x + b = 0 \qquad eq.3$$

must be maintained for all the examples, except possibly for the current one. To maintain optimality in practice, one can write out conditions (2)-(3) for the states before and after the update of x_k. By subtracting one from the other the following condition on increments of Δx and Δg is obtained:

$$\begin{bmatrix}\Delta g_k\\ \Delta g_s\\ \Delta g_r\\ 0\end{bmatrix} = \begin{bmatrix}a_k & K_{ks}\\ a_s & K_{ss}\\ a_r & K_{rs}\\ 0 & a_s^T\end{bmatrix}\begin{bmatrix}\Delta\mu\\ \Delta x_s\end{bmatrix} + \begin{bmatrix}K_{kk}^T\\ K_{KS}^T\\ K_{kr}^T\\ a_k\end{bmatrix}\Delta T_k \qquad \text{eq. 4}$$

The subscript s refer to the examples in the set S of unbounded support vectors, and the subscript r refers to the set R of bounded support vectors (E) and other examples (O). It follows from (2) that gs = 0. Then lines 2 and 4 of the system (4) can be re-written as:

$$\begin{bmatrix}0\\ 0\end{bmatrix} = \begin{bmatrix}0 & a_s^T\\ a_s & K_{ss}\end{bmatrix}\Delta s + \begin{bmatrix}a_k\\ K_{KS}^T\end{bmatrix}\Delta x_k \qquad \text{eq 5}$$

This linear system is easily solved for Δs:

$$\Delta s = \beta \Delta x_k \qquad eq\ 6$$

Where

$$\beta = -\begin{bmatrix}0 & a_s^T\\ a_s & K_{ss}\end{bmatrix}^{-1}\begin{bmatrix}a_K\\ K_{KS}^T\end{bmatrix} \qquad \text{eq 7}$$

is the gradient of the linear manifold of optimal solutions parameterized by x_k

One can further substitute (6) into the lines 1 and 3 of the system (4) and obtain the following relation:

$$\begin{bmatrix}\Delta g_k\\ \Delta g_r\end{bmatrix} = \gamma \Delta x_k \qquad \text{eq 8}$$

Where

$$\gamma = \begin{bmatrix}a_c & K_{ks}\\ a_r & K_{rs}\end{bmatrix}\beta + \begin{bmatrix}K_{KK}\\ K_{kr}^T\end{bmatrix} \qquad \text{eq. 9}$$

is the gradient of the linear manifold of the gradients of the examples in set R at the optimal solution parameterized by x_k.

D. Online SVM for Spam Classification

The model of online SVM for spam classification is shown in figure 3.1

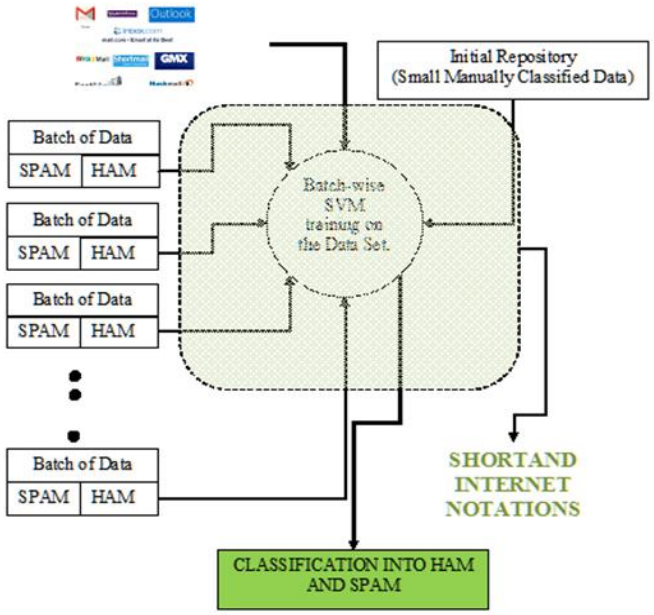

Fig. 3.1: Online SVM Batch Processing Model

As depicted in the figure given above, the SVM is trained on an initial data set comprising of manually classified SPAM and HAM. After this initial training, the SVM is subjected to test the incoming mails as HAM or SPAM. Accuracy of the classification is predicted based on user input and the SVM is incrementally trained on the batches of manually classified data, thereby, training the SVM on a larger set of data.

E. Shorthand Internet Notations

Most of the internet content written now-a-days comprises of a lot of shorthand notations. These notations are common in chats and text messages and forms a significant portion of the overall text body of most of the user contributed articles on the web. A

mapping of these shorthand notation to the actual text strings these corresponds to is necessary for accurate classification of the text under consideration. Table 3.1 list the common internet shorthand notations and the corresponding text strings.

Table 3.1:
Shortand Internet Notations Used On Web

Notation	*Text String*
B	*Be*
2	*To*
@	*At*
B4	*Before*
2day	*Today*
Zzz	*Sleeping*

The above list tabulates only 6 shorthand notations, nevertheless the number is large and is constantly increasing as specified in Netlingo[31]. In the proposed algorithm, these shorthand notations in the mails are replaced by the corresponding text string for a more accurate classification of the incoming emails. The list of these shorthand notations is imported from www.netlingo.com which comprises of about 100 most commonly used shorthand notations in general practice in facebook and twitter.

F. Algorithm for Online Classification

The proposed classification algorithm for spam filtering using Online SVM and shorthand notation is given below:

1) Initialized an SVM classifier
2) For a given data set of emails, replace the shorthand notations with the corresponding full text and classify manually. Label this set as 'Initial Dataset'.
3) Train the SVM on the classified dataset.
4) Test the incoming mails on the SVM and predict the accuracy.
5) If the accuracy falls below the threshold range, then

– Manually classify a batch of incoming mails and label it as Batch-1
– Train the SVM on the combined data set of Initial Dataset and Batch-1.

6) Repeat steps 2 to 5 unless desired prediction accuracy is achieved.

Dataset from UCI Machine Learning Repository [33] is classified using the proposed SVM. Section 4 tabulates the results and compare with those of the baseline, followed by Section 5 which concludes the paper.

IV. ANALYSIS OF PROPOSED WORK

A. SVM Based Classifier for E-mail Data

The samples of text are taken from a database consisting of about 5500 messages comprising of the messages which are sent from individual-to-individual and those that are sent as spam to a large number of recipients. As stated previously, unsolicited bulk messages are called SPAM, whereas those which are normally send from one-to-one in routine are known as HAM. Table 4.1 enlists a few of such samples. The complete database is provided in the CD ROM enclosed.

Table 4.1:
Normal (Ham) And Spam Text Messages

NORAMAL (HAM)	*SPAM*
Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...	*Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's*
Ok lar... Joking wif u oni...	*FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, £1.50 to rcv*
U dun say so early hor... U c already then say...	*WINNER!! As a valued network customer you have been selected to receivea £900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.*
Nah I don't think he goes to usf, he lives around here though	*Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030*
Even my brother is not like to speak with me. They treat me like aids patent.	*SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 info*
*As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends Callertune*	*URGENT! You have won a 1 week FREE membership in our £100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18*
I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today.	*XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here>> http://wap. xxxmobilemovieclub.com?n=QJKGIGHJJGCBL*
I've been searching for the right words to thank you for this	*England v Macedonia - dont miss the goals/team news. Txt ur national team to*

breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times.	*87077 eg ENGLAND to 87077 Try:WALES, SCOTLAND 4txt/ú1.20 POBOXox36504W45WQ 16+*
I HAVE A DATE ON SUNDAY WITH WILL!!	*Thanks for your subscription to Ringtone UK your mobile will be charged £5/month Please confirm by replying YES or NO. If you reply NO you will not be charged*
Oh k...i'm watching here:)	*07732584351 - Rodger Burns - MSG = We tried to call you re your reply to our sms for a free nokia mobile + free camcorder. Please call now 08000930705 for delivery tomorrow*

The illustration of the frequency of words through the text corpus can be done as shown in figure 4.1 and 4.2 for ham and spam respectively.

The data records in both spam and ham files consists of a large number of shorthand notation which are replaced with the corresponding full text words. Thus, making the SVM much more precise towards textual classification.

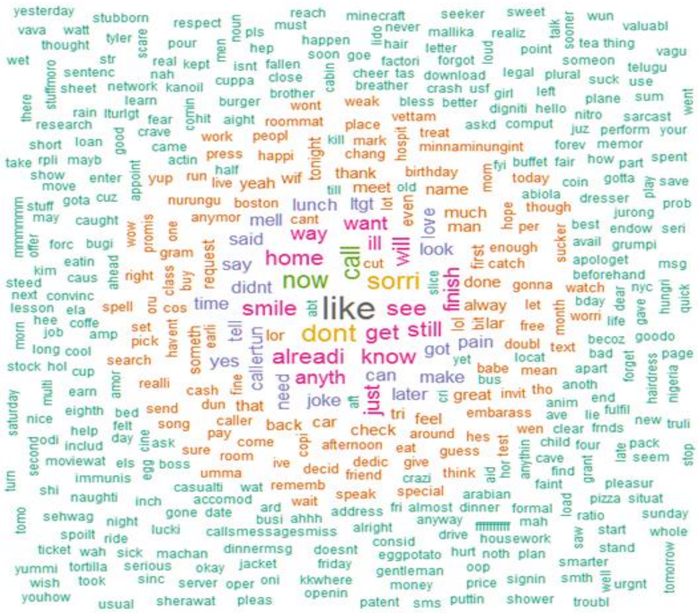

Fig. 4.1: Word Corpus from HAM

A word cloud is a pictorial description in which the size of the word in the word-cloud gives a qualitative measure of the frequency of the word in the document corpus. It is evident that most of the words have almost equal frequencies which are appearing in routine text messages with the frequency of words like "like", "sorry", "home" etc. being more than the other words.

Fig. 4.2: Word Corpus from SPAM

It is evident from figure 4.2 that the words like "free", "call", "claim" etc. have higher frequencies as compared to other words in the corpus all of which have approximately equal frequencies.

A set of two csv files are prepared from the data file ham_spam.csv (included in the CD ROM) downloaded from UCI Machine Learning Repository. These files are named as ham.csv and spam.csv consisting of 4825 and 747 samples respectively. First 100 records, each from file spam.csv and ham.csv are used to create a training dataset which is used to train the SVM model. In the training dataset, the data is manually labeled as +1 for spam and -1 for ham so as to enable the binary classification for spam and ham. The SVM model is initialized with the following specifications:

Parameters:
SVM-Type: C-classification
SVM-Kernel: linear
cost: 1
gamma: 0.001315789
Number of Support Vectors: 90

The model is tested for next 100 records of both the files, viz spam.csv and ham.csv. The tabulated record of 100 records from the file spam.csv, starting from record label 101 to 200 is tabulated as follows:

Table 4.2:
Prediction Of 100 Records From The File Spam.CSV (Hilighted Rows Consists Of Misclassified Mails)

Record-ID	*Prediction LABEL*	*Probability*
1	*1*	*0.9995702*
2	*1*	*0.9519550*
3	*1*	*0.9983121*
4	*1*	*0.5869087*
5	*1*	*0.9154620*
6	*1*	*0.9957437*
7	*-1*	*0.9102946*
8	*1*	*0.9918264*
9	*-1*	*0.8154085*
10	*1*	*0.9998329*
11	*-1*	*0.6061998*
12	*1*	*0.6085534*
13	*1*	*0.9915683*
14	*1*	*0.9485093*
15	*-1*	*0.6981974*
16	*1*	*0.9983136*
17	*1*	*0.9484896*
18	*1*	*0.9898180*
19	*1*	*0.9965161*
20	*1*	*0.9996945*
21	*1*	*0.9988662*
22	*1*	*0.9966512*
23	*1*	*0.9942330*
24	*1*	*0.8866217*
25	*1*	*0.9996556*
26	*1*	*0.9574848*
27	*1*	*0.8200228*
28	*1*	*0.9993530*
29	*1*	*0.7211017*
30	*1*	*0.9973896*
31	*1*	*0.9950533*
32	*1*	*0.9785676*
33	*-1*	*0.7951639*
34	*-1*	*0.7194932*
35	*1*	*0.5338641*
36	*1*	*0.8895303*
37	*1*	*0.8589145*
38	*1*	*0.9146740*
39	*1*	*0.9991473*
40	*1*	*0.7939379*
41	*1*	*0.9913378*
42	*1*	*0.9464291*
43	*1*	*0.9989176*
44	*1*	*0.9930358*

45	*1*	*0.9856058*
46	*1*	*0.9397194*
47	*1*	*0.9560812*
48	*1*	*0.9508536*
49	*1*	*0.9595862*
50	*1*	*0.9733204*
Record-ID	*Prediction LABEL*	*Probability*
51	*1*	*0.9988505*
52	*1*	*0.7256490*
53	*1*	*0.7582294*
54	*1*	*0.9868776*
55	*1*	*0.9619047*
56	*1*	*0.8901901*
57	*1*	*0.9991257*
58	*1*	*0.9646776*
59	*1*	*0.9787618*
60	*1*	*0.9998131*
61	*1*	*0.9967260*
62	*1*	*0.9981087*
63	*-1*	*0.8677420*
64	*1*	*0.9995708*
65	*1*	*0.9973896*
66	*1*	*0.9154620*
67	*1*	*0.9764284*
68	*1*	*0.6101991*
69	*1*	*0.9899836*
70	*1*	*0.9997420*
71	*1*	*0.9990154*
72	*1*	*0.9979755*
73	*1*	*0.9918264*
74	*1*	*0.8942190*
75	*1*	*0.7696849*
76	*1*	*0.7650680*
77	*1*	*0.9301759*
78	*1*	*0.9977292*
79	*1*	*0.9957065*
80	*1*	*0.9913783*
81	*1*	*0.9975009*
82	*1*	*0.9986739*
83	*1*	*0.9910953*
84	*1*	*0.9909675*
85	*1*	*0.9946137*
86	*1*	*0.9296562*
87	*1*	*0.9927860*
88	*-1*	*0.7951639*
89	*1*	*0.9905197*
90	*1*	*0.9711032*
91	*-1*	*0.8341424*
92	*1*	*0.9997781*
93	*1*	*0.9897955*
94	*1*	*0.9995704*
95	*1*	*0.9617120*
96	*-1*	*0.6501814*
97	*1*	*0.9992788*
98	*1*	*0.9945216*
99	*1*	*0.9909675*
100	*1*	*0.9956694*

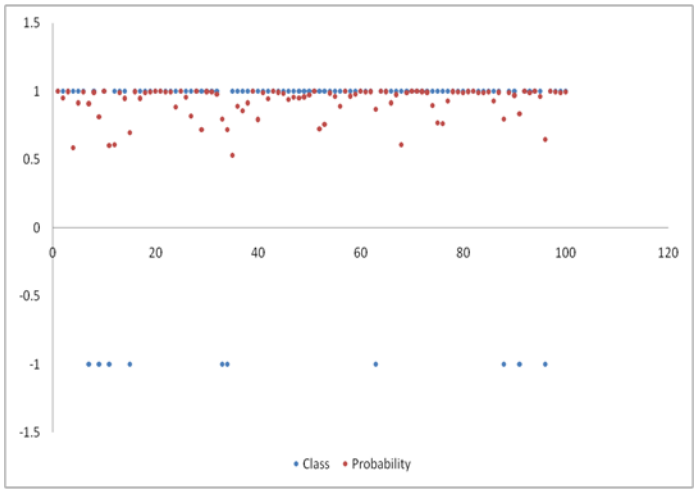

Fig. 4.3: Classification of 100 records of spam data from the test data. Horizontal axis shows the id of the record and vertical axis shows the class label and the corresponding probability.

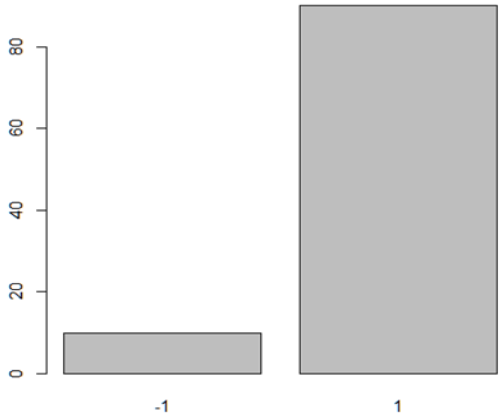

Fig. 4.4: Classification of 100 records of spam from test data. Vertical axis shows the count of records corresponding to class label shown on horizontal axis.

Figure 4.3 clearly indicates that a large proportion of the records are marked with class label +1 with high probability. Also, figure 4.4 indicates that out of the total of 100 records which belongs to spam category, 10 records are misclassified as ham using the proposed SVM, thus giving the accuracy of 90% in the classification.

The classification results of 100 records from the file ham.csv (starting from id 201 to 300), is depicted in table 4.3 given below.

Table 4.3:

Prediction Of 100 Records From The File Ham.CSV (Hilighted Rows Consists Of Misclassified Mails)

Record-ID	*Prediction LABEL*	*Probability*
1	*-1*	*0.827345*
2	*-1*	*0.8630545*
3	*-1*	*0.9334695*
4	*-1*	*0.8896606*
5	*-1*	*0.8536855*
6	*-1*	*0.9340552*
7	*-1*	*0.9673628*
8	*-1*	*0.5409831*
9	*-1*	*0.8032236*
10	*-1*	*0.9498397*
11	*-1*	*0.8423534*
12	*-1*	*0.8143575*
13	*-1*	*0.8834019*
14	*-1*	*0.933154*
15	*-1*	*0.9558561*
16	*-1*	*0.954938*
17	*-1*	*0.9390039*
18	*-1*	*0.8834019*

19	*-1*	*0.8834019*
20	*-1*	*0.8834019*
21	*-1*	*0.9074498*
22	*-1*	*0.8834019*
23	*-1*	*0.7112835*
24	*-1*	*0.8851515*
25	*-1*	*0.8834019*
26	*-1*	*0.8821307*
27	*-1*	*0.830576*
28	*-1*	*0.9139222*
29	*-1*	*0.9288399*
30	*1*	*0.6717017*
31	*-1*	*0.9045786*
32	*-1*	*0.9830544*
33	*-1*	*0.8834019*
34	*-1*	*0.8830488*
35	*-1*	*0.8834019*
36	*1*	*0.5230768*
37	*-1*	*0.8964304*
38	*-1*	*0.6140284*
39	*-1*	*0.890717*
40	*-1*	*0.890717*
41	*-1*	*0.8118697*
42	*-1*	*0.893867*
43	*-1*	*0.909989*
44	*-1*	*0.9172421*
45	*-1*	*0.8834019*
46	*-1*	*0.8243547*
47	*-1*	*0.8002979*
48	*-1*	*0.824264*
49	*-1*	*0.9102946*
50	*-1*	*0.8136093*
Record-ID	*Prediction LABEL*	*Probability*
51	*-1*	*0.9471174*
52	*-1*	*0.8834019*
53	*-1*	*0.8834019*
54	*-1*	*0.9422225*
55	*-1*	*0.9637474*
56	*-1*	*0.8693565*
57	*-1*	*0.8834019*
58	*-1*	*0.9018797*
59	*1*	*0.7850267*
60	*-1*	*0.8834019*
61	*-1*	*0.8834019*
62	*-1*	*0.9331984*
63	*-1*	*0.9396523*
64	*-1*	*0.8834019*
65	*-1*	*0.5508033*
66	*-1*	*0.8205345*
67	*-1*	*0.7148472*
68	*-1*	*0.851821*
69	*-1*	*0.9033542*
70	*-1*	*0.9137834*
71	*-1*	*0.8834019*
72	*-1*	*0.5647084*
73	*-1*	*0.9102946*
74	*-1*	*0.5308058*
75	*1*	*0.7955458*
76	*-1*	*0.8674316*
77	*-1*	*0.9679079*
78	*-1*	*0.888415*
79	*-1*	*0.897509*
80	*-1*	*0.8834019*

81	*-1*	*0.8895473*
82	*-1*	*0.880648*
83	*-1*	*0.8411894*
84	*-1*	*0.8834019*
85	*-1*	*0.9160945*
86	*1*	*0.6698279*
87	*-1*	*0.7051877*
88	*-1*	*0.8701915*
89	*-1*	*0.8834019*
90	*-1*	*0.918756*
91	*-1*	*0.8917017*
92	*-1*	*0.8834019*
93	*-1*	*0.9053883*
94	*-1*	*0.8151072*
95	*-1*	*0.932615*
96	*-1*	*0.7974066*
97	*-1*	*0.8834019*
98	*-1*	*0.9702297*
99	*-1*	*0.8902848*
100	*-1*	*0.9722694*

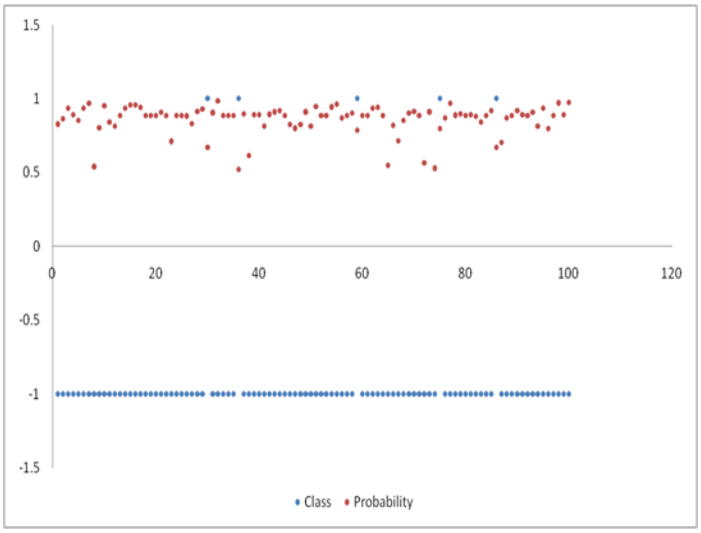

Fig. 4.5: Classification of 100 records of ham data from the test data. Horizontal axis shows the id of the record and vertical axis shows the class label and the corresponding probability.

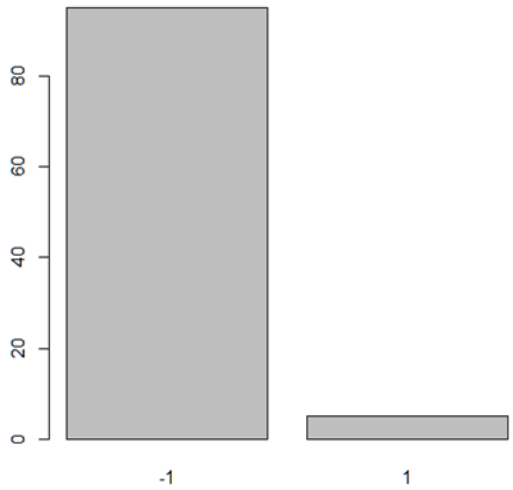

Fig. 4.6: Classification of 100 records of ham from test data. Vertical axis shows the count of records corresponding to class label shown on horizontal axis.

Figure 4.5 clearly indicates that a large proportion of the records are marked with class label -1 with high probability. Also, figure 4.6 indicates that out of the total of 100 records which belongs to ham category, 5 records are misclassified as spam using the proposed SVM, thus giving the accuracy of 95% in the classification.

B. SVM using Incremental Dataset

Classification of test data samples of 200 records after a training data of 150 records each of spam and ham yields the following results:

SPAM MAILS (LABEL = +1)

SVM_LABEL	SVM_PROB
-1: 14	Min. :0.5077
1 :186	1st Qu.:0.9210
	Median :0.9907
	Mean :0.9302
	3rd Qu.:0.9990
	Max. :1.0000

HAM MAILS (LABEL = -1)

SVM_LABEL	SVM_PROB
-1:191	Min. :0.5179
1 : 9	1st Qu.:0.8249
	Median :0.8566
	Mean :0.8517
	3rd Qu.:0.9184
	Max. :1.0000

Thus, in case of spam, there are a total of 200 test records out which 166 were classified correctly, thus providing an accuracy of 93%, thus providing a percentage improvement of 3% over what was achieved with 100 records training samples.

Also, in case of ham, the correctly classified records are 191, thus providing 95.5 % accuracy, thereby improving the efficiency of classification by 0.5% over what was achieved with a training set of 100 records.

The percentage improvement in the accuracy of the classification as a function of size of the training data set using the incremental SVM model can be depicted as shown in figure 4.7.

The normalized graph of percentage improvement in the accuracy of the classification as a function of percentage increment in the training data set is shown in figure 4.8.

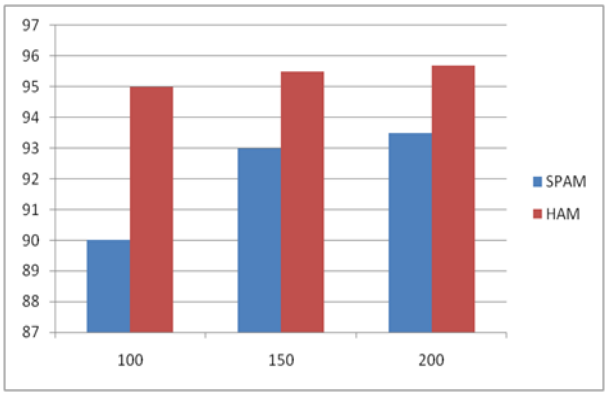

Fig. 4.7: Classification accuracy (vertical) as function of Training Set size (Horizontal Axis)

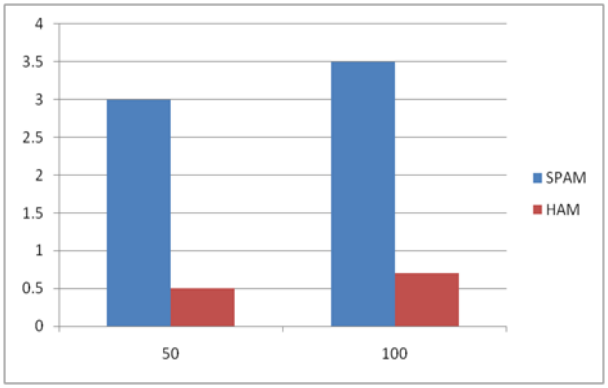

Fig. 4.8: Percentage Improvement in Classification accuracy (vertical) as function of Percentage Increment in Training Set size (Horizontal Axis)

It is evident that classification accuracy improves as the number of data records in the training set increases. Section 5 discusses the results obtained using normal and incremental mode of SVM and draws the conclusion and future scope.

V. CONCLUSION AND FUTURE SCOPE

SVMs produce state-of-the-art performance on text classification task with proper selection of the kernal function, but the main constraint is the cost that grows quadratically with the size of the data set. The high values of C required for best performance with SVMs show that the margin maximization of SVMs is overkill for classification task in low computational environment. Thus, a less expensive alternative is proposed in which the SVM is trained on an incremental dataset until the desired accuracy in the classification is obtained. Moreover, it is evident from the results produced in Section 4 that a moderate size training set of data records is sufficient to obtain desired accuracy, which cannot improve significantly even if the batch size of training data increases twofold. Thus, an optimum size training set can be figured out to achieve desired accuracy under given accuracy constraints. It is evident from the results obtained in Section 4 that training data set of size of 100 produces nearly equivalent results in comparison to training data set of 150 records. Moreover, the percentage improvement in the accuracy of classification for training batch size of 150 and 200 is merely 0.25% on average, at a quadratic cost of training of SVM. Moreover, associated with the cost of quadratic time is the cost of manually labeling of the training data needed to train the SVM. Thus is can be concluded that an optimal number of manually labeled records can be used so as to achieve desired accuracy under given time complexity, thereby providing optimal performance under all conditions.

A. Future Scope

The work as proposed and implemented in this paper can be extended by using the header and other meta contents of the mails as well the attachments provided therein. This will assist in extending the feature set used to train the SVM model and to more accurately classify the incoming mails as ham or spam. Extending the feature set would certainly improve the accuracy of the classification process, nevertheless increasing the computational complexity of the SVM, which already works in quadratic time proportional to the training data set. Increasing the accuracy of the classification process, while at the same time, minimizing the computational complexity will certainly be the main objective of study in this domain, which will be helpful for more appropriately managing the incoming mails, even on low computational complexity techniques.

REFERENCES

[1] Jian Zhong; YiLu Zhou; Wei Deng, "Filtering image-based spam using multifractal analysis and active learning feedback-driven semi-supervised support vector machine," Conference Anthology, IEEE , vol., no., pp.1,5, 1-8 Jan. 2013 doi: 10.1109/ANTHOLOGY.2013.6784950
[2] Wenxuan Shi; Maoqiang Xie; Yalou Huang, "Collaborative spam filtering technique based On MIME fingerprints," Intelligent Control and Automation (WCICA), 2011 9th World Congress on , vol., no., pp.225,230, 21-25 June 2011 doi: 10.1109/WCICA.2011.5970733
[3] Spam filtering: how the dimensionality reduction affects the accuracy of Naive Bayes classifiers, February 2011, Volume 1, Issue 3, pp 183-200, DOI- 10.1007/s13174-010-0014-7
[4] Lin-Fang Hu; Wei Gong; Li-Xiao Qi; Ping Wang, "A method for feature selection based on the optimal hyperplane of SVM and independent analysis," Machine Learning and Cybernetics (ICMLC), 2013 International Conference on , vol.01, no., pp.114,117, 14-17 July 2013 doi: 10.1109/ICMLC.2013.6890454
[5] A study of spam filtering using support vector machines, June 2010, Volume 34, Issue 1, pp 73-108, Springer Netherlands, DOI- 10.1007/s10462-010-9166-x
[6] A suffix tree approach to anti-spam email filtering, Kluwer Academic Publishers, DOI- 10.1007/s10994-006-9505-y
[7] Araujo, S.; Tran, D.T.; de Vries, A.P.; Schwabe, D., "SERIMI: Class-Based Matching for Instance Matching Across Heterogeneous Datasets," Knowledge and Data Engineering, IEEE Transactions on , vol.27, no.5, pp.1397,1440, May 1 2015 doi: 10.1109/TKDE.2014.2365779
[8] An Efficient Method for Filtering Image-Based Spam E-mail, 12th International Conference, CAIP 2007, Vienna, Austria, August 27-29, 2007. Proceedings, DOI- 10.1007/978-3-540-74272-2_117
[9] Chi-Yao Tseng; Ming-Syan Chen, "Incremental SVM Model for Spam Detection on Dynamic Email Social Networks," Computational Science and Engineering, 2009. CSE '09. International Conference on , vol.4, no., pp.128,135, 29-31 Aug. 2009 doi: 10.1109/CSE.2009.260
[10] Generating Balanced Classifier-Independent Training Samples from Unlabeled Data, 16th Pacific-Asia Conference, PAKDD 2012, Kuala Lumpur, Malaysia, May 29-June 1, 2012, Proceedings, Part I, DOI- 10.1007/978-3-642-30217-6_23
[11] Yinggang Zhao; Qinming He, "An Incremental Learning Algorithm Based on Support Vector Domain Classifier," Cognitive Informatics, 2006. ICCI 2006. 5th IEEE International Conference on , vol.2, no., pp.805,809, 17-19 July 2006 doi: 10.1109/COGINF.2006.365593
[12] Advances in Spam Filtering Techniques, Volume 394, 2012, pp 199-214, Springer Berlin Heidelberg, DOI- 10.1007/978-3-642-25237-2_12
[13] Davids, A., "Fraudulent Online Identity Sanctions Act: empowering law enforcement or limiting privacy?," Distributed Systems Online, IEEE , vol.5, no.4, pp.,, April 2004 doi: 10.1109/MDSO.2004.1301255
[14] Pfleeger, S.L.; Bloom, G., "Canning SPAM: Proposed solutions to unwanted email," Security & Privacy, IEEE , vol.3, no.2, pp.40,47, March-April 2005 doi: 10.1109/MSP.2005.38
[15] Schryen, G., "A Formal Approach towards Assessing the Effectiveness of Anti-Spam Procedures," System Sciences, 2006. HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on , vol.6, no., pp.129a,129a, 04-07 Jan. 2006 doi: 10.1109/HICSS.2006.9