International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 5 – No. 10, August 2013 – www.ijais.org

A Novel Technique of Email Classification for Spam Detection

Vinod Patidar
Student (M. Tech.),
CSE Department, BUIT
Bhopal, India

Divakar singh
HOD,
CSE Department, BUIT
Bhopal, India

Anju Singh
Assistant Professor,
IT Department, BUIT
Bhopal, India

ABSTRACT
Email spam is one of the unsolved problems of the today's Internet, annoying individual users and bringing financial damage to companies. Among the approaches developed to stop spam emails, filtering is a popular and important one. Common uses for email filters include organizing incoming email and computer viruses and removal of spam. As spammers periodically find new ways to bypass spam filters and distribute spam messages, researchers need to stay on the forefront of this problem to help reduce the amount of spam messages. Currently spam emails occupy more than 70% of all email traffic. The negative effect spam has on companies is greatly related to financial aspects and the productivity of employees in the workplace. In this paper, we propose the new approach to classify spam emails using support vector machine. It found that the SVM outperformed than other classifiers.

Keywords
Support vector, spam, non spam, email, ann

1. INTRODUCTION
Recently, e-mails have become an easy and important medium of communication for all the Internet users. However, spam, also known as bulk or unsolicited commercial e-mail, is a curse of e-mail communication. Spam is commonly compared to paper junk mail. However the difference between paper junk is that mailers pay a fee to distribute their junk materials, whereas with spam the ISP or recipient pays in the form of server resources, disk space, additional bandwidth and lost productivity. If spam continues to increases at the current rate, that will become unmanageable in the near future.

Through a study it is found that over 70% of today's business emails are spam [1]; therefore, there are many grave problems associated with increasing volumes of spam such as wasting storage space and communication bandwidth, engulfing important personal mail, filling users' mailboxes, , and consuming users' time to delete all spam mails. Spam emails vary significantly in content and they just about belong to the following categories: service provider advertisement, money making scams, improve business, fat loss, sexually explicit, make friends, etc. [2], one example of a spam email is shown as Fig. 1. E-mail users spend lot of time in reading email message and deciding whether it is spam or not and categorizing them into folders. E-mail service providers would like to take over the user's burden by installing server-based spam detector and filters that can classify e-mails as spam automatically [3]. Spam filtering classification can occur by the following reasons:

- Continually changing – Spam is continuously changing, as spam new topics are emerges. Also, spam mailers try to make their email messages as indistinguishable from canonical email as possible and change the spam patterns to debacle the filters [4].
- OCR computational cost – the OCR computational cost is consist text embedded in images analog with the immense amount of e-mails handled by server-side filter day by day [4].
- The used content obscuring techniques-the Spam mailers are applying content obscuring techniques to images, to able OCR systems ineffective without compromising human readability [5].

U.B.E. (Unsolicited Bulk Email) is another acronym for spam that effectively encapsulates this definition. To create training sets of spam emails and non-spam emails, each email is carefully examined according to this restrictive definition of spam. Although the most of the user oft considers all unwanted emails as "spam", emails that border on "solicited" should be rejected straight-out. Examples of these might include email sent from easily placeable domains, such as Youmint.Com Yahoo.com or Amazon.com. A good shibboleth to follow here is: "when in doubt, throw it out". Similarly, emails not are spam should be limited to personal email communications between individuals or groups, and keep off any forms of bulk mailings, regardless of whether they were solicited or not [14]. Once these data sets have been gathered and approved, the support vector machine is ready for training. The support vector machine uses a set of sophisticated mathematical equations to perform this type of computation.

2. LITERATURE REVIEW
This section gives a brief literature review of the work have done on spam classification. The Naive Bayes classifier is a simple statistical algorithm with a long chronicle of providing amazingly better results. It has been used in many spam classification studies [6, 7, 8, 9], and has become somewhat of a benchmark. It is based on Bayes' rule of conditional probability that's by its name become naive bayes classifier, combined with the "naive" assumption that all conditional probabilities are independent [10]. Another method for spam classification is artificial neural network (ANN) [12], normally called neural network (NN), and is a computational model or mathematical model that is inspired by the functional and structure aspects of biological neural networks. A neural network comprises of an interconnected group of artificial neurons, and it processes information using a connectionist approach for computation. In many cases an ANN is an adaptive system that modifies its structure based on internal or external information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modeling tools. To distinguish

spam patterns, the neural network must first be "trained". This training involves a computational analysis of email content using more number of samples for both spam and non-spam email messages. Essentially the network will "learn" to descry what the humans mean by "spam" and "non-spam". To assist in this process, authors first need to have a clear and concise definition of "spam".

The KNN (k-Nearest Neighbour) classifier is an example-based classifier; it means that the classifier needs training files for comparison rather than an precise category representation, like the category profiles used by existing other classifiers. As such, there is no real training phase. When a new file needs to be categorized, the k most similar files (neighbours) are found and if a large adequate proportion of them have been assigned to a certain category, the new file is also assigned to this category, otherwise not .Additionally, finding the nearest neighbours can be accelerated using traditional indexing methods. To decide whether a email is legitimate or not, we look at the class of the emails that are closest to it. The comparison between the vectors is a real time process. This is the basic idea of the k nearest neighbour algorithm.

A GANN (Generalized Additive Neural Network) is less susceptible being perceived as a black box, because partial residual plots are generated providing graphical results that aid the spam researcher in obtaining insight about the constructed models [13]. Pattern recognition is another benefit of a GANN and can help in the classification of spam messages.

3. SUPPORT VECTOR MACHINE

Support vector machines (SVMs) are relatively new approach that has rapidly gained popularity because of the very fast and accurate results they have achieved in a wide variety of machine learning problems. Support vector machine algorithms divide the n dimensional space represented data into two regions using a hyper plane. This hyper plane always maximizes the margin between the two regions (classes). The margin is defined by the longest distance between the examples of the two classes and is computed based on the distance between the closest instances of both classes to the margin, which are called supporting vectors. Instead of using linear hyper planes, many implementations of these algorithms use so-called kernel functions. These kernel functions lead to non-linear classification surfaces, such as polynomial, radial or sigmoid surfaces [11].

Support Vector Machines [2] is also a supervised learning method for automatic pattern recognition. SVM learns from a training data set to classifier which separates a set of positive examples from a set of negative examples of introducing the maximum margin between the two data sets. The training data set can be described by points in the dimensional space. $xi = \mathbb{Q}^n$ with two different labels $yi \in \{-1, +1\}$ depending on the class which is assigned to the point xi for all $i = 1,2, \ldots . l$. The points xi, which lie on the hyper plane, satisfy $w = xi + b = 0$, where $w \varepsilon Rn$ is the normal vector of the hyper plane. The Euclidean norm of w is $||w||$. Mathematically, the points xi can be expressed by two inequalities as follows:

$$w.xi + b \geq +1 \text{ for } yi = +1 \qquad (1)$$

$$w.xi + b \leq -1 \text{ for } yi = -1 \qquad (2)$$

We decided to define the "margin" of a separating hyper plane to be $d(+), d(-)$. Margin $= 2/||w||$ so, in order to find largest margin we have to find the smallest $||w||$. In order to minimize $||w||$, we need to allocate Lagrangian multipliers where $\alpha i \geq 0$.

We then calculate w, $w = \sum_{i=1}^{l} \alpha_i y_i x_i w$ which determines the set of support vectors by finding the indices such that $\alpha i > 0$.

We then calculate $b = \frac{1}{N_i} \sum_{s \epsilon S} (y_s - \sum_{m \epsilon S} \alpha_m y_m x_m . x_s)$ .Testing each new point x' can be done by the following: y' = sgn (w. x' + b).

4. PROPOSED APPROACH

In this section we describe process of Spam Classification in details. Many email services today provide spam filters that are able to classify emails into spam and non-spam email with high accuracy. In this we use SVMs to build our own spam filter. We will be train a classifier to classify whether a given email,x, is spam $(y = 1)$ or non-spam $(y = 0)$. In particular, we need to convert each email into a feature vector $x \epsilon \mathbb{Q}^n$.

Our proposed approach is dividing into four modules as shown in fig 2.

4.1 Preprocessing Emails

To preprocess the emails for classification here we have an example of email.

Delivered-To: vinod.patidar@gmail.com

Date: Wed, 17 Jul 2013 13:08:38 +0100

From: indian supplier needed urgent <supply@info.org>

Reply-to: peterowen78@outlook.com

I am greeting you, are you from India? I got your contact from India international business and investment daily then decided to contact you direct. Our Company use to purchase industrial chemicals and natural Pharmaceutical Products, from South Africa and India but it is very scarce now in South Africa but the products are cheaper in India. I can't come to India because of my new promoted post and I will want you to act as the direct dealer. our company's general director has asked me to contact the local dealer in India, for them to send the new purchasing manager to India to purchase the product directly from the local dealer in India.

After purchase from original local seller you would sell to our new purchasing manager, then we share the profit. Your role must be played perfectly and the least I expect from you is betrayal. I don't want my organization to know the real cost of the product for obvious reasons. Please take out a moment of your very busy schedule to respond back to me for more details if you are interested to source and supply to us via this email peterowen78@outlook.com

regards

DR PETER OWEN
BUSINESS DIRECTOR
NOVAS PHARMACEUTICAL LTD UK
LONDON.
peterowen78@outlook.com

Fig 1: Sample Email

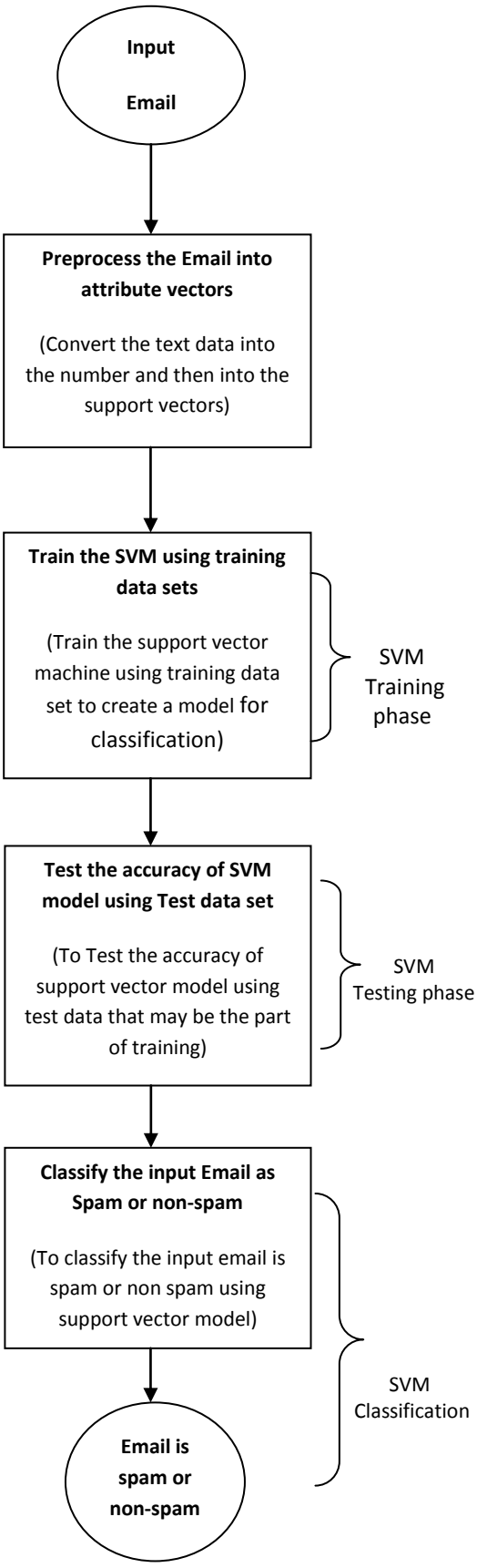

Fig 2: Flow chart of spam classification process

Before starting on a machine learning task, it is usually insightful to take a look at examples from the dataset. Fig 1 shows a sample email that contains a URL, an email address (at the end), numbers, and dollar amounts. While many emails would contain similar types of entities (e.g., numbers, other URLs, or other email addresses), the specific entities (e.g., the specific URL or specific dollar amount) will be different in almost every email. Therefore, one method often employed in processing emails is to \normalize" these values, so that all URLs are treated the same, all numbers are treated the same, etc. For example, we could replace each URL in the email with the unique string \httpaddr" to indicate that a URL was present.

This has the effect of letting the spam classifier make a classification decision based on whether any URL was present, rather than whether a specific URL was present. This typically improves the performance of a spam classifier, since spammers often randomize the URLs, and thus the odds of seeing any particular URL again in a new piece of spam is very small.

We have implemented the following email preprocessing and normalization steps:

- Lower-casing: The entire email is converted into lower case, so that capitalization is ignored (e.g., IndIcaTE is treated the same as Indicate).
- Stripping HTML: All HTML tags are removed from the emails. Many emails often come with HTML formatting; we remove all the HTML tags, so that only the content remains.
- Normalizing URLs: All URLs are replaced with the text \httpaddr".
- Normalizing Email Addresses: All email addresses are replaced with the text \emailaddr".
- Normalizing Numbers: All numbers are replaced with the text \number".
- Normalizing Dollars: All dollar signs ($) are replaced with the text \dollar".
- Word Stemming: Words are reduced to their stemmed form. For example, \discount", \discounts", \discounted" and \discounting" are all replaced with \discount". Sometimes, the Stemmer actually strips o_ additional characters from the end, so \include", \includes", \included", and \including" are all replaced with \includ".
- Removal of non-words: Non-words and punctuation have been removed. All white spaces (tabs, newlines, and spaces) have all been trimmed to a single space character.

The result of these preprocessing steps is shown in Fig 3. While preprocessing has left word fragments and non-words, this form turns out to be much easier to work with for performing feature extraction.

> emailaddr congratul beneficiari thi is mr walli marvin microsoft claim agent from the microsoft lotteri board i have deposit your win chequ of number number number pound for deliveri sinc the bank and ha refus to contact you for deliveri i have left for greec for a seminar so i have deposit a confirm bank draft number number number pound with the fed ex courier plc for deliveri i have also paid for deliveri i have been wait for you sinc to contact me for your confirm bank draft of number number number pound but i did not hear from you sinc that time

Fig 3: Preprocessed Sample Email

4.2 Vocabulary List

After preprocessing the emails, we have a list of words (e.g., Fig 3) for each email. The next step is to choose which words we would like to use in our classifier and which we would want to leave out. For this exercise, we have chosen only the most frequently occurring words as our set of words considered (the vocabulary list). Since words that occur rarely in the training set are only in a few emails, they might cause the model to overfit our training set. The complete vocabulary list is in the file vocab.txt and also shown in Fig 4. Our vocabulary list was selected by choosing all words which occur at least a 100 times in the spam corpus, resulting in a list of 1899 words. In practice, a vocabulary list with about 10,000 to 50,000 words is often used.

Given the vocabulary list, we can now map each word in the preprocessed emails (e.g., Fig 3) into a list of word indices that contains the index of the word in the vocabulary list. Fig. 5 shows the mapping for the sample email. Specifically, in the sample email, the word \anyone" was first normalized to \anyon" and then mapped onto the index 86 in the vocabulary list.

```
1 aa
2 ab
3 abil
...
86 anyon
...
916 know
...
1898 zero
1899 zip
```

Fig 4: Vocabulary List

```
531  1676  877   1074   1036   292  44   688
1666 1036  194   756    432    1895 1854
1162 1120  1120  1120   1282 666  426
1519 1666  153   74     743  1699 352
1893 666   426   756  941    666  666
1476 1538  756   432    341    153  1120
1120 1120 1282   1860   1666   666  426
756  60    1205  666    426    756  167
1805 666   1893  1519   1699   352  1018
666  1895  341   153    1162 1120   1120
1120 1282  225   449  1113   763
688  1893  1519  1665 1694
```

Fig 5: Word Indices for Sample Email

4.3 Extracting Features from Emails

In this we extract feature that converts each email into a vector in $\mathbb{Q}^n$. For this we have used $n = \#$ words in vocabulary list. Specifically, the feature $x_i \epsilon \{0,1\}$; 1g for an email corresponds to whether the $i-th$ word in the dictionary occurs in the email. That is, $x_i = 1$ if the $i-th$ word is in the email and $x_i = 0$ if the $i-th$ word is not present in the email. Thus, for a typical email, this feature would look like:

$$x = \begin{bmatrix} 0 \\ . \\ . \\ . \\ 1 \\ 0 \\ . \\ . \\ . \\ 0 \\ 1 \\ 1 \\ . \\ . \\ . \\ 1 \end{bmatrix} \epsilon \mathbb{Q}^n$$

4.4 Training SVM for Spam Classification

After you have completed the feature extraction functions, we will load a preprocessed training dataset that will be used to train a SVM classifier. Dataset contains 4000 training examples of spam and non-spam email, while test dataset contains 1000 test examples. Each original email was processed and converted into a vector $x^{(i)} \epsilon \mathbb{Q}^{1899}$. After loading the dataset, we will proceed to train a SVM to classify between spam ($y = 1$) and non-spam ($y = 0$) emails.

5. RESULTS

The Accuracy, which refers to the proportion of emails classified as accurate type in the total emails. Accurate circumstances are True Positive (TP) and True Negative (TN), while false detected situations are False Positive (FP) and False Negative (FN). Accuracy of the system is calculated by the following equation:

$$\text{Accuracy} = ((TP + TN)/(TP + TN + FP + FN)) \text{ X } 100$$

Once the training completes, we observed that the classifier gets a training accuracy of about 99.8% and a test accuracy of about 99.354%.

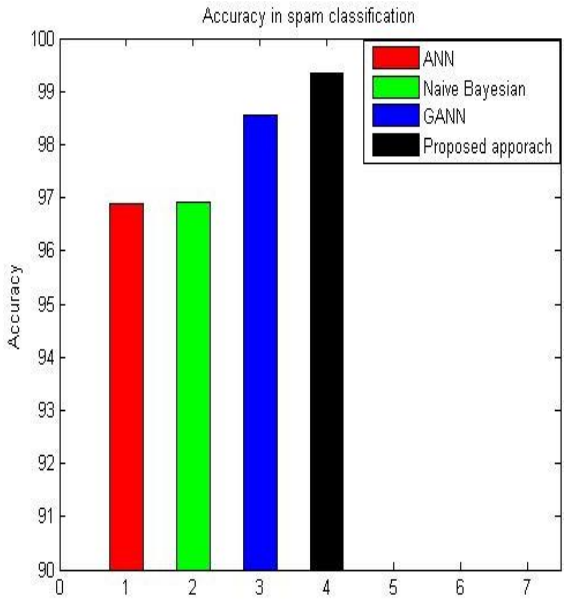

Fig 6: Comparison among classification algorithm for accuracy

Fig. 6 shows that the proposed approach support vector machine perform better in term of spam classification.

6. CONCLUSION

The results achieved by the support vector machine compared exceptionally well to the other three techniques. It was found that the SVM outperformed the other two classifiers in the case of spam messages. It can be concluded that SVMs are capable of spam classification more accurately than other techniques.

7. REFERENCES

[1] Aladdin Knowledge Systems, Anti-spam white paper, Retrieved December 28, 2011.

[2] F. Smadja, H. Tumblin, "Automatic spam detection as atext classification task", in: Proc. of Workshop on Operational Text Classification Systems, 2002.

[3] A. Hassanien, H. Al-Qaheri, "Machine Learning in Spam Management", IEEE TRANS., VOL. X, NO. X, FEB.2009

[4] P. Cunningham, N. Nowlan, "A Case-Based Approach to Spam Filtering that Can Track Concept Drift", [Online]

[5] F . Roli, G. Fumera, "The emerging role of visual pattern recognition in spam filtering: challenge and opportunity for IAPR researchers".

[6] I. Androutsopoulos, J. Koutsias, "An evaluation of naïve bayesian anti-spam filtering". In Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML 2000), pages 9–17, Barcelona, Spain, 2000.

[7] I. Androutsopoulos, G. Paliouras, "Learning to filter spam E-mail: A comparison of a naïve bayesian and a memory based approach". In Proceedings of the Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2000), pages 1– 13, Lyon, France, 2000.

[8] J. Hidalgo, "Evaluating cost-sensitive unsolicited bulk email categorization". In Proceedings of SAC-02, 17th ACM Symposium on Applied Computing, pages 615–620, Madrid, ES, 2002.

[9] K. Schneider, "A comparison of event models for naïve bayes anti-spam e-mail filtering". In Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics, 2003.

[10] I. Witten, E. Frank, "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations". Morgan Kaufmann, 2000.

[11] S. Amari, S. Wu, "Improving support vector machine classifiers by modifying kernel functions". Neural Networks, 12, 783– 789. (1999).

[12] C. Miller, "Neural Network-based Antispam Heuristics" , Symantec Enterprise Security (2011), www.symantec.com Retrieved December 28, 2011

[13] Du Toit, J. V., de Waal, D. A., 2010. Spam Detection using Generalized Additive Neural Networks. SATNAC 2010 Conference Proceedings.

[14] [Miller, Chris. "Neural Network-based Antispam Heuristics." Symantec Enterprise Security. White paper (2006).