NLP Based Customized Method For Spam Detection

Nikita Shukla[1], Shiv Tiwari[2]
[1]Dept of CSE
[2]Assistant Professor, Dept of CSE
[1, 2] Shri Ram Institute of Science & Technology, Jabalpur, Madhya Pradesh, India.

Abstract- *E-mail service is one of the most popular Internet communication services. Thousands of companies, organizations and individuals use e-mail every day and get benefit from it. The continuous growth of email users has resulted in the increasing of unsolicited emails also known as Spam. In current, server side and client side anti spam filters are introduced for detecting different features of spam emails. However, recently spammers introduced some effective tricks consisting of embedding spam contents into digital image, pdf and doc as attachment which can make ineffective to current techniques that is based on analysis digital text in the body and subject fields of email. Many of proposed working strategy provides an anti spam filtering approach that is based on data mining techniques which classify the spam and ham emails. The effectiveness of these approaches is evaluated on large corpus of simple text dataset as well as text embedded image dataset. But most of the filtering techniques are unable to handle frequent changing scenario of spam mails adopted by the spammers over the time.*

Therefore improved spam control algorithms or enhancing the efficiency of various existing data mining algorithms to its fullest extent are the utmost requirement. However, an amount of spam emails always hang around us and bring down our productivity. We urgently need a spam filtering to clean up our network environment. Through this thesis we design a baseline system that will train the machine and classify the emails as ham or spam based on naive bayes training. The emails identified as spam would be kept aside as correct spams, whereas, the ham emails would be sent to the NLP engine that we will design to classify them further. We propose to check some lexical and semantic features to help the bayes engine classify them correctly. Our results indicate that the NLP engine we had introduced proved to be of high significance in identifying spam or ham from email servers.

Keywords- E-mail service, Spam, Ham, Naive Bayes, NLP, Lexical, Semantic.

I. INTRODUCTION

Email is undoubtedly one of the Internet's killer applications. It satisfies the basic human need for communication and has become mission critical in every organization. Billions of emails are delivered each day connecting people around the globe. Unfortunately, not all emails are sent for serious purposes. More precisely, the majority of all emails circulating on the Internet are unsolicited bulk emails, in short: spam. To prevent spam from becoming email's killer application, a plethora of countermeasures have been proposed, for instance legal regulations, economic burdens, DNS-based attempts, and a variety of solutions exploiting different spam filtering techniques. However, the fight against spam has only been modestly successful so far: Recent studies report that currently more than 70 percent of all emails are spam, and that no improvement has been detected over the past years.

Email is one of the most prevalent forms of communication, both for business and individuals. Due to its efficiency, convenience, and low cost, it offers an ideal environment for users to connect with each other, and also provides a platform for personal information management [1]. However, like other media, email is used for illegitimate purposes as well. Criminals are employing email to facilitate their schemes. For instance, multiple items of evidence came from the events of 9/11 investigation demonstrate the use of email in terror plots [2]. In light of this, email proves to be a very important source of evidence in the digital investigation [3].

The Simple Mail Transfer Protocol (SMTP) is the root of all evil. Its authors did not foresee the danger of organized misuse, thus failing to devise a mechanism to prevent the flooding of millions of inboxes. Particularly, the lack of reasonable authentication schemes enables spammers to operate incognito. Proprietary mechanisms, such as the Sender ID Framework and the Domain Keys system, promise to help alleviating this deficiency, but it might take years before they are widely deployed and adopted.

1.1 Spam Overview: There is no exact definition of spam. Most of the spam can be termed as unwanted e-mail but not all of the unwanted e-mails are spam. Another term would be unsolicited commercial e-mail, but unfortunately spam is not only advertising material. Spam can be also defined as junk mail but it implicates the question: what is a junk mail? Although most of the e-mail users know what spam is, but it is

not obvious how to define spam and spamming. Spam originates from a Monty Python sketch [4] and is commonly used when referring to junk or unsolicited email. This can include email containing virus, chain letters, advertisement, political advocacy or fraud attempts. Ham is used when referring genuine email, the opposite of spam. Wikipedia, the biggest encyclopaedia on the Internet gives the following definitions:

E-mail spam: involves sending nearly identical messages to thousands (or millions) of recipients." [5].

Spamming: "Spamming is the abuse of any electronic communications medium to send unsolicited messages in bulk." [6].

As a summary one could agree that spam is something unsolicited, unwanted email what is mostly also an advertisement material. However not all unwanted e-mail letters are spam and not all spam is an advertisement. It is not an exact definition of spam these are only properties in order to explain, what is the relationship between spam and other e-mail sets.

1.2 Different Approaches to Spam Filtering

There are many ways to eliminate or reduce spam [7]. One technical way to stop spam is to deny outgoing email sessions from the receiver network and make every host within that network use a controlled, administered and secure email server when sending email. This will also help stop abuse from outside when a third party uses a network's resources to relay email. Spam can also be detected and filtered at the recipient's end. It can be stopped before it enters the recipient's system or it can be filtered after it has been accepted and entered the recipient's system. Filtering before it has entered the recipient's system is done by issuing error codes in the email delivery session.

Depending on whether it is a permanent or temporary error message, a genuine and standard compliant sender system will try to deliver the message after a delay if it is a temporary error message. Failure to deliver an email to the recipient will often result in a warning to the sender. When email is accepted by the recipient's system, it is not common to notify the sender if the email is caught by a spam filter. Therefore, the risk of losing information, when using a filter after the email is accepted, is higher. Technical anti spam methods like grey listing is also impossible at this stage.

1.2.1 Anti Spam Methods

There are a variety of anti spam methods. Some are based on email content and others are based on protocol and other technicalities. Centralized and decentralized anti spam methods are also covered. Every method has their strengths and weaknesses. Spam Filtering is applicable for both individual users as well as for enterprises. For individual it's enough to download a spam filtering application and run it on personal computers. This application directly communicates with Mail Transfer Agent (MTA) which helps to compose and receive mails. In the case of enterprise network the situation is more complex because there is a need to filter out the spam mails when it enters the network. Here the filter will be installed within the internal server which directly communicate with and thus allowing enterprises to manage their mailboxes [8]. In an enterprise once a mail is identified as a spam, and then it will be tagged as a spam for all users in that network. The mode of operations of a spam filter is depicted in Figure 1. [8]

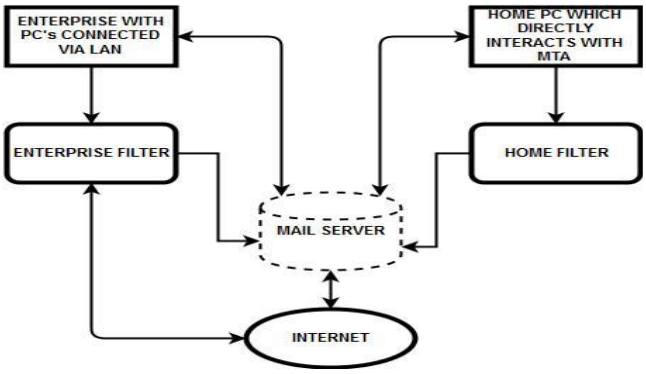

Figure 1.1: Modes of Operations in Spam Filtering.

The Spam Filtering methods can be classified mainly on two bases as mentioned in [9].

II. LITERATURE SURVEY

2.3 Email Spam Characteristics

Email spam can be characterized by sending behavior and content [10]. We can extract features based on sending behavior and content. In our framework, features are properties that our system can use to generate filter to find non-spam messages.

(1) Spam emails are often sent in bulk in order to spread out. Moreover, spammers use forged addresses for hiding and there are several IP hops between sender and receiver.
(2) Textual data in Spam emails are often more informal in style and do not follow established syntax or grammar rules.

Spam messages are mainly commercial advertisement. In addition, spam emails also contain URL links, HTML web pages and images. With the advance on URL camouflage techniques, the percentage of spam message embedding URL links increases significantly. According to Wang et al [11] statistics, the percentage of spam message containing image is less than 5%.

A type of unwanted email, commonly referred to as email spam, is becoming more sophisticated and capricious with the advance of authors [11]. Email spam has not only occupied the most percentage of all email traffic but increased to as much as 90% of entire mail volume nowadays [12]. Obviously, Email spam is a persistent problem. Yet, researchers always tend to downplay the importance of this kind of email in many investigations. Numerous techniques exist to detect email spam.

Researchers are more concerned about the accuracy of filtering methods. Few of them focus on relevant investigation method in digital forensics. Cormack and Lynam [13] defined email spam as "unsolicited, unwanted email that was sent indiscriminately, directly or indirectly, by a sender having no current relationship with the recipient." The main type of email spam message content is commercial advertisement. In order to attract users to browse spam messages, spam topic changes over time. Email spam is often in accompany with phishing, identity theft and malware distribution for negative purposes. Furthermore, it can contain incriminating information. Hiding potential evidence in the spam folder along with hundreds of real spam emails is an effective, partly because of the popularity of email spam. Meanwhile, many digital forensics practitioners do not pay attention to the content of spam. In digital forensics, the goal of any investigation is to answer questions about digital events. In investigations involving email forensics, investigators need to find the key piece of communication evidence between suspects from the unstructured textual data. In current practice, investigators typically employ modern computer forensics tools to execute keyword searches first, and then read those flagged emails for evidence one by one. This manual process requires comprehensive and detailed analysis by investigators with experience and expertise. This kind of analysis is tedious and still overlooks crucial information frequently. There have been several published works attempting to improve the effectiveness of text analysis. For example, Al-Zaidy et al.[14] present a method to extract information from email data to discover criminal networks by using a modified Apriori algorithm, Schmid et al. [15] demonstrate an application of customized associative classification techniques to address the email authorship attribution problem. These methods mentioned above are applied to analyze email rather than email spam. In contrast to emails in the inbox folder, most spam messages are sent in bulk and their content are irrelevant to each other. It is hard to discover direct or indirect associate information not only from the sending behavior but also from their contents. Furthermore, researchers in forensics have never realized the importance of analyzing spam emails until recently [16].

2.4.1 DATA ANALYSIS

In this section, we start with content analysis of Spam Archive dataset, followed by topic modelling and network analysis.

A. Content Analysis

The two main types of email message content are "Text" and "Multipart". Messages in type "Text" are simple text messages while messages in type "Multipart" have parts arranged in a tree structure where the leaf nodes are any non multipart content type and the non-leaf nodes are any of a variety of multipart types. To have a better sense of the distribution of main types in email spam,

In the first group, spam emails are detected based on senders' identifications [17]. In a white list based email system, a user needs to actively mark regular emails and add the senders' addresses into the white list, i.e., future emails from these senders will be considered regular ones. For the emails sent from others that are not in the white list, they will be treated as spam or junk emails. On the other hand, a user can also put the sender's email address into the blacklist list, if a spam is identified. In practice, both black and white lists can be applied. One limitation of this type of approach is that the sender may change its identity by using dynamic IP, IP proxy, and IP spoofing techniques.

In the second category, spam filtering is realized via rule-based approaches. A typical rule-based method is the decision tree based technique. The earliest decision tree based learning system was developed by Hunt, dating back to 1966. He created a concept learning system that uses, for the first time, a decision tree to learn concepts. It built the foundation for other decision tree based learning algorithms. For example, R.Quinlan proposed an iterative decision tree based classification algorithm ID3. To address the limitations of the ID3 algorithm, i.e., it cannot handle both continuous and discrete attributes; he later proposed an improved algorithm, the C4.5 algorithm [18]. One significant improvement in C4.5 lies in the feature selection method that is based on information gain theory. In 2002, S. Ruggieri proposed the EC4.5 algorithm [19] that uses binary search, instead of linear

search, to identify the threshold in the whole training set. To generate the same decision tree, the efficiency of EC4.5 is about five times better than the original C4.5 algorithm. The memory space requirement of EC4.5, however, is much larger than that of C4.5. The principle of decision tree based methods is to classify emails based on pre-defined rules. These rules are set by regular users and thus cannot be changed frequently. One limitation of these methods is the pattern about spam emails can hardly be identified by a regular user. In addition, the configuration and maintenance of these rules could a cumbersome task.

In the last category, spam emails are detected by machine learning based algorithms. One popular solution is based upon the support vector machine (SVM) technique. A SVM is a supervised learning technique for classification that is formally defined as a separating hyper plane in the space composed of training samples. The hyperplane essentially divides the training samples into different categories. Because it is able to handle small training set, non-linear and high-dimension classification problem, it is widely applied in text classification and spam email classification. The classification speed of SVM, however, depends highly on the number of support vectors extracted from training samples. In other words, larger the number of support vectors, slower the classification speed. When the number of samples is huge, implying large number of support vectors, SVM's classification speed becomes very slow. To address this issue, Scholkopf *et. al.* proposed a method to construct new vectors, and thus reduce the computational complexity of support vector decision functions .

B. Naive Bayes Classifier

Before introducing the proposed SVM-NB algorithm, we first briefly discuss the Naive Bayes and SVM methods. Let $x = (x1, x2, ..., xn)$ denote a feature vector and

$C = (c1, c2..... cm)$ denote all possible categories that the vector may belong to. The principle of a Naive Bayes classifier is to compute the probabilities $p1, p2..... pm$ for x where pj is the probability that $\_x$ belongs to category cj.

Determining the value of max $(p1, p2..... pj)$, we will know which category the feature vector x belongs to. Therefore, the classification problem can be considered as finding the maximum value of the following equation:

$$P(cj \mid x1, x2, \ldots, xn) = \frac{P(x1, x2, ..., xn|cj)P(cj)}{P(c1, c2..... cn)}$$

where $P(cj)$ denotes the probability that a random sample belongs to category cj . $P(x1, x2, ... , xn|cj)$ is the probability that category cj contains the feature vector $x = (x1, x2, ... , xn)$, if we already know the training sample is in cj . $P(c1, c2..... cn)$ is the joint probability of all possible categories. For all given categories, the denominator $P(c1, c2, ... , cn)$ is a constant, so equation 1 can be simplified as $cNB = \text{argmax}\ cj/C\ P(x1, x2, ... , xn|cj)P(cj)$ (2)

According to the assumption of Naive Bayes theorem, the terms in the feature vector are identically distributed. The assumption of applying Naive Bayes method is the independence between the feature vectors. In reality, however, there are always many dependent vectors presented in the training set. It can be seen from equation 3 that all probabilities P $(xi|cj)$ must be independent to each other. If not, NB will yield incorrect classification results. It is critical to have a mechanism to eliminate the dependency among feature vectors as much as possible. Fortunately, SVM is such a tool that can efficiently classify non-linear or dependent training samples into different categories. Therefore, we combined the NB and SVM methods to propose an innovative classification approach, called SVM-NB.

III. PROPOSED SYSTEM

3.1 Proposed System

Our main idea is to design a baseline system that will train the machine and classify the emails as ham or spam based on naïve bayes training. The emails identified as spam would be kept aside as correct spams, whereas, the ham emails would be sent to the NLP engine that we will design to classify them further. We propose to check some lexical and semantic features to help the bayes engine classify them correctly.

3.2 Detailed Working

We attempted to improve the naïve technique by applying lexical and semantic features by looking at the content of the text like emails. Figure below will represent complete process for training and testing of classifier.

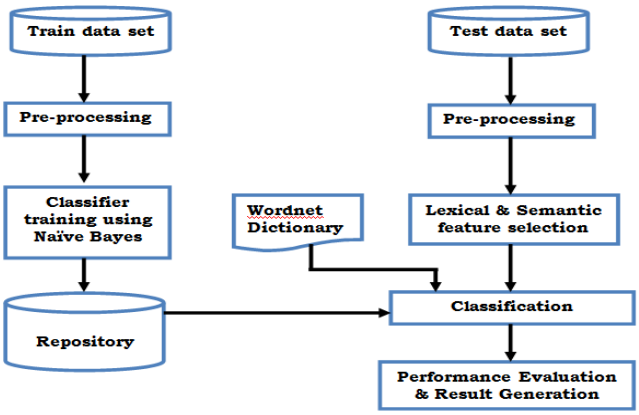

Figure 3.2: Training & testing process.

3.2.1. Lexical Features

At the very beginning, we tokenize the text into words and initialize their ham and spam word count based on their placement, whether in ham or spam. In the next step, we train our system by using some lexical features. For that purpose, we maintain the following 3 lists:

1. Stop words
2. Swear words
3. Common spam phrases.

3.2.2 Stop words removal

Stop words are words which do not add into much of a meaning to the topic and yet appear most frequently in the documents like articles or prepositions. We maintain a stop words list. During training phase, we skip the word if it is a stop word, and do not consider it in the Bayes probability calculation. Below are steps to find spam or ham using this method:

Step 1:- Let S be an email message. Take two variables ham and spam for counting, initialized to 0. Convert all words to lower case.
Step 2:- perform preprocessing on S.
Step 3:- For each word w_i in S
If w_i found in stopword list Then
w_i is removed from S.
End If.
End For
Step 4:-For each word w_j in S
If w_j found in spam word dataset Then
Increment count of spam
End if
Adjust probability of email S.
End For
Step 5:-For each word w_j in S
If w_j found in ham word dataset Then
Increment count of ham
End if
Adjust probability of email S.
End For

Step 6:- If Spam percent >Ham percent Then
S will be identified as Spam Email.
Else
S will be Ham Email.
End if

3.2.3 Swear words handling

Mostly, the spam emails may contain swear words, which make them categorized as spam emails. But this may not always be the case. So, while training, we make a note of the swear words occurring in the ham emails, and increase their ham count. This will increase the weight of that swear word being in ham.

3.2.4 Common spam phrases handling

We maintain a list of common spam phrases. We, scan the emails line by line, and check whether it contains the commonly occurring spam phrases. The occurrence of the spam phrase in the mail, increases the probability of the email being a spam. So, we accordingly increase the spam probability in the Bayes calculation.

3.2.5 Semantic Features

We observed that spammers use synonyms or hypernyms of spam words in text like emails.

A) Synonyms

If the data dictionary from the training phase does not contain the any word from test email, we look if we can find the synonyms of the test word in data dictionary, using WordNet. Then using the found synonyms in data dictionary, we average out their ham and spam probabilities and assign them to the test word.

B) Hypernyms

We use similar concept as that of synonyms. Likewise, we compute the hypernyms of the words of test email. And, calculate the average ham and spam count of those occurring in the training dictionary.

IV. IMPLEMENTATION AND EVALUATION

4.2 Third Party Tools

Third party tools we are using are:

1. *Stanford-core-nlp-3.5.2*
2. *WordNet Dictionary*
3. *MIT JWI API*

4.2.1 Stanford-core-nlp tool

Stanford CoreNLP provides a set of human language technology tools. It can give the base forms of words, their parts of speech, whether they are names of companies, people, etc., normalize dates, times, and numeric quantities, mark up the structure of sentences in terms of phrases and syntactic dependencies, indicate which noun phrases refer to the same entities, indicate sentiment, extract particular or open-class relations between entity mentions, get the quotes people said, etc.

Choose Stanford CoreNLP if you need:

- An integrated NLP toolkit with a broad range of grammatical analysis tools
- A fast, robust annotator for arbitrary texts, widely used in production
- A modern, regularly updated package, with the overall highest quality text analytics
- Support for a number of major (human) languages
- Available APIs for most major modern programming languages
- Ability to run as a simple web service

Stanford CoreNLP's goal is to make it very easy to apply a bunch of linguistic analysis tools to a piece of text. A tool pipeline can be run on a piece of plain text with just two lines of code. CoreNLP is designed to be highly flexible and extensible. With a single option you can change which tools should be enabled and disabled. Stanford CoreNLP integrates many of Stanford's NLP tools, including the part-of-speech (POS) tagger, the named entity recognizer (NER), the parser, the co reference resolution system, sentiment analysis, bootstrapped pattern learning, and the open information extraction tools.

4.2.2 WordNet Dictionary

WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and concepts can be navigated with the browser. WordNet is also freely and publicly available for download. WordNet's structure makes it a useful tool for computational linguistics and natural language processing.

WordNet superficially resembles a thesaurus, in that it groups words together based on their meanings. However, there are some important distinctions. First, WordNet interlinks not just word forms—strings of letters—but specific senses of words. As a result, words that are found in close proximity to one another in the network are semantically disambiguated. Second, WordNet labels the semantic relations among words, whereas the groupings of words in a thesaurus do not follow any explicit pattern other than meaning similarity.[2]

4.2.3 MIT JWI API

JWI (the MIT Java Wordnet Interface) is a Java library for interfacing with Wordnet. JWI supports access to Wordnet versions 1.6 through 3.0, among other related Wordnet extensions. Wordnet is a freely and publicly available semantic dictionary of English, developed at Princeton University. JWI is written for Java 1.5.0 and has the package namespace *edu.mit.jwi*. The distribution does not include the Wordnet dictionary files; these can be downloaded from the Wordnet download site.[3]

Results & Evaluation

Baseline method (Naive Base) and proposed methods are tested on two sets of directories: ham and spam. Firstly they are trained and then tested. For evaluation, parameter used is accuracy of the method.

Evaluation on the basis of accuracy

For Directory: SPAM

Method	No of Spam	No of Ham	Accuracy
Baseline	125	5	96.1538
Stop Words	128	2	98.4615
Swear Words	128	2	98.4615
Spam Phrases	126	4	96.9230
Synonymy	127	3	97.6923
Hypernymy	127	3	97.6923
All Combined	129	1	99.2307

Table 5.1: Result of All Methods for SPAM Directory.

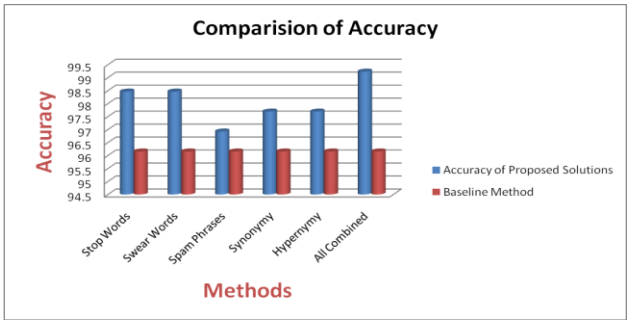

Figure 5.9: Chart of Resultant Accuracy for SPAM Directory.

V. CONCLUSION

As spammers had improve their spamming technique by obfuscate the spam email keywords to evade spam filter, normal rule-based spam filtering system will be very hard to detect this kind of spam. But this problem can be solve by this proposed engine due to it have the ability to detect obfuscated spam. This proposed engine is implemented in Netbeans and using MicroBlaze soft-core processor, thus it comes with the advantages of better speed performance. This spam filtering engine is able to provide better filtering speed compare with software-based spam filtering system.

However, users can always adjust the threshold to suit their needs and email environment. As a conclusion, this spam filtering engine is a recommended solution to improve email system for every email user by minimizes spam risks.

Future work

There are a few future work suggested here to improve the proposed spam filtering engine.

a. Graphical User Interface (GUI) tool for user control Currently this engine only display the filtering result in Hyper Terminal and there is no has any GUI tools for user to control this system, so this will give some trouble and inconvenient to users. In order to allow users to handle the filtering task more conveniently, it is suggested to create a GUI tool. The GUI tool should have a few modules such as:

- Filtering Module: Allow user to start, pause or stop spam filtering process.
- Reporting Module: Allow user to print the report of filtering result and alert user when spam detected. It also should have log file of the filtering process.
- Pattern Database Update Module: Provide an easy way for user to update the pattern database regularly.

REFERENCE

[1] S. Whittaker, V. Bellotti, and J. Gwizdka, "Email in personal information management," Communications of the ACM, vol. 49, no. 1, pp. 68–73, 2006.
[2] N. C. on Terrorist Attacks Upon the United States and U. S. of America, The 9/11 commission report," 2004.
[3] E. Casey, A. Blitz, and C. Steuart, "Digital evidence and computer crime," 2014.
[4] Monty Python's Flying Circus. Just the Words, volume 2, chapter 25, pages 27–28. Methuen Publishing Ltd, 1989.
[5] E-mail spam, http://en.wikipedia.org/wiki/E-mail_spam.
[6] Spam (electronic), http://en.wikipedia.org/wiki/Spam_%28electronic%29.
[7] S.L. Pfleeger and G. Bloom. Canning spam: Proposed solutions to unwanted email. IEEE Security & Privacy, 3(2):40–47, Mar-Apr 2005.
[8] Omar Saad, Ashraf Darwish, Ramadan Faraj, A survey of machine learning techniques for spam filtering, International Journal of Computer Science and Network Security, VOL.12 No.2, February 2012.
[9] Nazirova.S, Survey on spam filtering techniques Communications and Network, 2011.
[10] Paswan, M.K., Bala, P.S., and Aghila, G.: 'Spam filtering: Comparative analysis of filtering techniques', in Editor (Ed.)^(Eds.): 'Book Spam filtering: Comparative analysis of filtering techniques' (2012, edn.), pp. 170-176
[11] Zhang, Y., Yang, X., and Liu, Y.: 'Improvement and optimization of spam text filtering system', in Editor (Ed.)^(Eds.): 'Book Improvement and optimization of spam text filtering system' (2012, edn.), pp. 448-451.
[12] Caruana, G., Maozhen, L., and Man, Q.: 'A MapReduce based parallel SVM for large scale spam filtering', in Editor (Ed.)^(Eds.): 'Book A MapReduce based parallel SVM for large scale spam filtering' (2011, edn.), pp. 2659-2662.
[13] du Toit, T., and Kruger, H.: 'Filtering spam e-mail with Generalized Additive Neural Networks', in Editor (Ed.)^(Eds.): 'Book Filtering spam e-mail with Generalized Additive Neural Networks' (2012, edn.), pp. 1-8.
[14] Zheleva, E., Kolcz, A., and Getoor, L.: 'Trusting spam reporters: A reporter-based reputation system for email filtering', ACM Trans. Inf. Syst., 2008, 27, (1), pp. 1-27.
[15] Xiao-wei, W., and Zhong-feng, W.: 'Good word attack spam filtering model based on artificial immune system', in Editor (Ed.)^(Eds.): 'Book Good word attack spam filtering model based on artificial immune system' (2012, edn.), pp. 1106-1109.
[16] Pham, X., Lee, N.-H., Jung, J., and Sadeghi-Niaraki, A.: 'Collaborative spam filtering based on incremental ontology learning', Telecommun Syst, 2011, pp. 1-8.

[17] G. Caruana, M. Li, and Y. Liu, "An ontology enhanced parallel SVM for scalable spam filter training", Neurocomputing, 108, 45-57, 2013.

[18] K. L. Goh, A. K. Singh and K. H. Lim, "Multilayer perceptrons neural network based web spam detection application", Signal and Information Processing (ChinaSIP), 2013 IEEE China Summit & International Conference on (pp. 636-640), 2013.

[19] A. Arram, H. Mousa and A. Zainal, "Spam detection using hybrid Artificial Neural Network and Genetic algorithm", Intelligent Systems Design and Applications (ISDA), 13th International Conference on (pp. 336-340), IEEE, 2013.