PERFORMANCE OF MACHINE LEARNING TECHNIQUES FOR EMAIL SPAM FILTERING

M. Deepika¹ Shilpa Rani²
¹²Assistant Professor,
Department of Computer Science & Engineering,
Sreyas Institute of Engineering & Technology, Hyderabad

Abstract— Email spam, an Unsolicited Commercial e-mail (UCE), is one of the major problems for today’s Internet users. E-mail is an effective tool for communication as it saves time and cost, considerably. However, the increase in email users has resulted in a dramatic increase in spam emails during the past few years. There are several machine learning techniques such as Bayesian classification, Naive bayes, k-NN, SVMs, Rough sets etc., which help us with a technique to filter spam. In this paper, an email-filtering approach that is based on supervised classifier has been proposed. The model mentioned here has the advantage of combining supervised machine learning with Support Vector Machines (SVM). To implement SVM, we are using LIBSVM, a library from MATLAB tool and compared it with existing classifiers. The proposed solution for the undesired spam could achieve better results in the system function and is faster when compared to the other existing machine learning techniques.

Keywords— E-mail spam, Classification, Support vector machines.

I. INTRODUCTION (HEADING 1)

Spam is the result of the use of electronic messaging systems (including most broadcast media, digital delivery systems) to send unsolicited bulk messages indiscriminately [1]. E-mail spam, also known as junk e-mail or unsolicited bulk e-mail (UBE), is a subset of spam that involves nearly identical messages sent to numerous recipients by e-mail [2]. By the Symantec Message Lab forecast, spam becomes more culturally and linguistically diverse, in future. The amount of spam sent from the European countries will increase to 40% - 45% of all spam [16]. Peter J. Denning’s article [17] is one of the first papers to consider this serious problem. The first mathematical apparatus applied to spam filtering systems is the Bayes’ algorithm, which was used first by Sahami et.al in 1996 and then by the other researchers [5-8].

Electronic-mail (abbreviated as e-mail) is a fast, effective and inexpensive process of switching messages over the Internet. Whether it’s a personal message from a family member, a company-wide message from the boss, researchers across the continents sharing recent findings, or astronauts staying in touch with their family (via e-mail uplinks or IP phones), e-mail is a preferred means for communication. Used worldwide by 2.3 billion users, at the time of writing the article, email usage is projected to have increased up to 4.3 billion accounts by the end of 2016. But the increasing dependence on e-mail has induced the emergence of many problems caused by “illegitimate” e-mails, i.e. spam. According to the Text Retrieval Conference (TREC) the term “spam” is an unsolicited, unwanted e-mail that has been sent indiscriminately.

Spam e-mails are unsolicited, un-ratified and usually mass mailed. Being a carrier of malware, Spam causes the proliferation of unsolicited advertisements, fraud schemes, phishing messages, explicit content, promotions of cause, etc. On an organizational front, spam effects include: i) annoyance to individual users; ii) less reliable e-mails; iii) loss of work productivity; iv) misuse of network bandwidth; v) wastage of file server storage space and computational power; vi) spread of viruses, worms and Trojan horses; and, vii) financial losses through phishing, Denial of Service (DoS), directory harvesting attacks, etc.

The goal of the initial work is: to introduce the fact and the phenomenon of junk e-mail also called spam; and to understand what spam is; and, also to give a brief overview about the history of spam and spam filtering. This proposed work is followed by the explanation why spam exists at all and what it does. Furthermore, different kinds of spam are introduced and finally the spam filtering methods are reviewed. The proposed work contains the filtering methods using machine-learning techniques; and finally, the statistical results of different spam filter software’s are analyzed.
II. BACKGROUND STUDY AND RELATED WORK

A. Bayes Theoram

Bayesian algorithms were used by 1996 to sort and filter email. Although naive Bayesian filters did not become popular until later times, multiple programs were released in 1998 to address the growing problem of unwanted email.[1] The first scholarly publication on Bayesian spam filtering was done by Sahami et al. in 1998.[2] Particular words have particular probabilities of occurring in spam email and in legitimate email. The filter does not know these probabilities in advance; and, it must first be trained so that it can build them up. To train the filter, the user must manually indicate whether a new email is spam or not. For all the words in each training email, the filter adjusts the probabilities that each word will appear in the spam or legitimate email in its database. After training, the word probabilities (also known as likelihood functions) are used to compute the probability that an email with a particular set of words in it belongs to either category [4]. One of the main advantages of Bayesian spam filtering is that it can be trained on a per-user basis.

B. Naive Bayes

Naive Bayes is a simple technique for constructing classifiers. It is based on the basic rule of probability called Bayes Rule; and, it assumes that every word occurs independently of every other word. A Naive Bayes classifier is efficient in terms of time needed to train and the time needed to classify an unseen document and is easy to update as more training data are accumulated.

In the Evaluation criteria of Naive Bayes, it uses two metrics: Recall and Precision. In our case we have two categories as Spam or Legitimate. We want to classify an unseen document as one of these two categories based on the characteristics of the two classes during the training phase. Of the test document, an ideally trained classifier classifies each one as spam document; also an ideal classifier classifies as each real legitimate document as legitimate well. However, the real classifiers are not 100 percent accurate. Therefore, metrics are used to determine how well the classifiers are working.

Precision: Percentage of selected items which are correct.

Recall: Percentage of correct items those are selected.

C. Rough Set Theory

The rough set theory is a mathematical tool for modeling incomplete or imprecise information [5]. It has been used for both feature selection and knowledge discovery in a number of real world domains; and, the rough set theory uses feature selection for classification of mails. In training data set we first represent the top frequent words. Then, a feature selection method based on the rough sets is applied to remove redundant features from the training data. We use frequent words to represent the web pages in our training data. However, some frequent words may not be very relevant to our learning task. These words may have little power in differentiating the documents of different categories. Therefore, some further selection of relevant features is important. We apply a rough set-based feature-selection method for this purpose. In this method, we first introduce some concepts of rough sets and then describe an algorithm for removing the attributes.

D. K-Neighourest Neighbor

In k-Nearest Neighbors (kNN), the value of k designates the number of neighbors, used for classification. An important step of this method is the choice of similarity function between the messages. The method frequently used to calculate the similarity measure between the messages is the “cosine distance,” where cosine is defined as the angle between the vectors representing the compared messages. This distance function normalizes the length of the messages, and hence considered efficient[4].

E. Support Vector Machines

Support Vector Machines (SVMs) are relatively new techniques that have rapidly gained popularity because of the excellent results they have achieved in a wide variety of machine learning problems and because they have solid theoretical underpinnings in statistical learning theory [14]. Support Vector Machine (SVM) algorithms divide the n-dimensional space representation of the data into two regions using a hyper-plane. This hyper-plane always maximizes the margin between the two regions or 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 [15].
F. Neural networks
An Artificial Neural Network (ANN), usually called Neural Network (NN), is a mathematical model or computational model that is inspired by the structure and/or functional aspects of biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information to, using a connectionist approach. In most of the cases, an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modeling tools. They are usually used to model complex relationships between the inputs and outputs, or to find patterns in the data.

G. Decision Tree
Decision tree classifier is the structural data in the form a binary tree. A training set is a set of base tuples to determine the classes related to these tuples. A tuple X is represented by an adjective vector X = (x1, x2… xn). Assume that a tuple belongs to a predefined class that is determined by an adjective called class label [10]. The training set is randomly selected from the base; this step is called the learning step. This technique is very efficient and extensively uses classification. The structure of the tree can be implemented with the following factors:

- A node of the tree represents a test on an adjective;
- A branch exiting from a node represents possible outputs of a test; and,
- A leaf represents a class label.

III. METHODOLOGY
E-mail spam classification is a major issue in today’s electronic world. To solve this problem, different spam classification methods are used. Using this spam detection technique, we can identify the spam and non-spam mails in the mailbox. There are different steps that have been followed in order to identify the SPAM. To implement the classification algorithm, a linear SVM model is trained on each of the training sets and test is conducted to see the performance of each model. SVM shows better asymptotic performance as the amount of training data increases.

Installing LIBSVM
After downloading the LIBSVM Matlab Interface, following the instructions in the package’s README file, LIBSVM is built from its source code. Instructions are provided for both Matlab on Unix and Windows operating systems.

After successful installation of LIBSVM, 4 files with the suffix "mexglx" ("mexw32" on Windows) are shown. These are the binaries that will run from MATLAB, and need to make them visible to the working directory to implement classification. This can be done in any of the following 3 ways:

1. Creating links to the binaries from the working directory.
2. Adding the location of the binaries to the Matlab.
3. Copying the binaries to a particular working directory.

Text Classification
The classification is based on 4 training sets that are used for Naive Bayes and are now formatted for LIBSVM. They are named:

- a. email_train-50.txt (based on 50 email documents)
- b. email_train-100.txt (100 documents)
- c. email_train-400.txt (400 documents)
- d. email_train-all.txt (the complete 700 training documents)

The training is based on a linear SVM model on each of the four training sets with a default SVM value. After training, the performance of each model on a set is tested. This is done with the "svmpredict" command. During the test time, the accuracy on the test set is printed to the console. The classification accuracy for each training set is recorded and the answers with the solutions are checked.

Classification accuracy
The following are the expected results that are reported by LIBSVM.

- a. 50 documents: Accuracy = 75.3846% (196/260)
- b. 100 documents: Accuracy = 88.4615% (230/260)
- c. 400 documents: Accuracy = 98.0769% (255/260)
- d. the complete 700 training documents: Accuracy = 98.4615% (256/260)

Here are the comparisons with Naive Bayes:
The above results show that Naive Bayes performs better than SVM with less data, but SVM shows better asymptotic performance as the amount of the training data increases.

IV. CONCLUSION

There are many ways to filter a spam. At present, putting into practice of a reliable spam filter becomes more and more important for e-mail users since they have to face with the increasing amount of uninvited emails. Using different valid mails and spam mails, the present study developed a new model using the machine learning techniques. The proposed model demonstrated that the performance of the LIBSVM has a higher accuracy, compared to Naivesbayes classifier. The proposed method can be implemented using MATLAB in order to eliminate the problem of spam and to achieve higher classification accuracy in terms of the training and testing data.

REFERENCES

[10] Yun Wang, Zhiqiang Wu, Runxiu 2013 IEEE,” Spam Filtering System Based on Rough Set and Bayesian Classifier”