DOI 10.4010/2016.1440
ISSN 2321 3361 © 2016 IJESC

Research Article Volume 6 Issue No. 5

A Survey on Classification Algorithms for Email Spam Filtering

Simranjit Kaur Tuteja
Department of Computer Engineering
Savitribai Phule Pune University, Pisoli, Pune, India
simran_1410@yahoo.in

Abstract:
Millions of people use Electronic mail for communication across the globe and it is a critically important application for many businesses. Unsolicited bulk email has been a significant problem for email users over the past decade. A massive amount of spam is flowing into users mailboxes on a regular basis. Along with spam being frustrating for most email users, it also strains the IT infrastructure of organizations and costs businesses billions of dollars in lost productivity. Increasing need of effectively filtering spam has become very important. This paper reviews many classification algorithms and techniques for email spam filtering. It also studies the comparison of classification methods on Spambase dataset which helps to select the best method.

Keywords: Spam email detection, machine learning, feature selection, classification algorithm

I. INTRODUCTION

The internet has become an inherent part of daily day to day life and email has become a powerful tool for information exchange. As the importance and applications of the Internet and e-mail has grown, there has been a notable growth in spam in recent years. The origination of spam can be from any location across the world where Internet is readily available.

The count of spam messages persists to rapidly increase inspite of the development of antispam services and technologies. In accordance to counter the growing problem, determination of best possible techniques to counter spam with various tools available must be analyzed by organizations. Various tools, such as the contracted anti-spam services e-mail filtering gateways, corporate e-mail system and end user training, would be very helpful for any organization. However, attempting to counter huge amounts of spam on a daily basis is still unavoidable issue for users. Spam would still persist tomorrow after deluging network systems, hampering employee productivity, affecting bandwidth, if there is no anti spam activities performed.

Significant research has been carried out to generate algorithms competent of recognizing spam from legitimate emails as filtering spam has been one of the crucial applications of pattern recognition and encroachment of data mining. Mostly emails are filtered on the basis of their content inclusive of text, images or their header fields which supply details about the sender.

In this paper, we compare among some proposed content based filtering algorithms that rely on text classification to decide whether an email is spam or not. The rest of the paper continues as follows. Section II presents a survey summarizing the main classification algorithms. We provide a general background about the classification algorithms. In III we propose our system using BPNN(Back Propagation Neural Network) and Feed forward for Training and conclude with future work in Section IV.

II. LITERATURE REVIEW

Paper [1] presents a survey of some popular filtering algorithms that rely on text classification to decide whether an email is unsolicited or not. A comparison between the below methods is performed on the SpamBase dataset to identify the best classification algorithm in terms of accuracy, computational time, and precision/recall rates.

A. Support Vector Machines

SVM offer a principled approach to machine learning (ML) problems because of their mathematical foundation in statistical learning theory. SVM construct their solution as a weighted sum of SVs, which are only a subset of the training input. Beyond minimizing some error cost function based on the training data sets similarly to what other discriminant ML techniques do, SVM impose an additional constraint to the optimization problem; the hyper plane needs to be situated such that it is at a maximum distance from the different classes. Such a term forces the optimization step to find an optimal hyper plane that would eventually generalize better since it is situated at an equal and maximum distance between the classes. SVs and their corresponding weights are found after an exhaustive optimization step that uses Lagrange relaxation and solves Karush- Kuhn-Tucker (KKT) constraints to determine the parameters of this unique hyper plane. For linearly non-separable problems, SVM use the kernel trick, which consists of a kernel function satisfying Mercer's theorem to map the data to the feature space where the data would become at worst pseudo linearly separable.

B.Local Mixture Support Vector Machines (LMSVM)

LM-SVM is a variation of traditional SVM. It aims at preprocessing the dataset by reducing its size using local mixture measures before it's fed into the optimization stage of SVM. This is achieved by first clustering the data and then filtering the resulting clusters according to the aforementioned measures which act as fitness qualifiers.

C.Decision Trees (DT)

Decision trees are multistage decision systems that split feature space into regions associated with the various classes. Building the tree hierarchy is achieved through the use of candidate questions at the node level based on predefined splitting criteria. These criteria are usually strongly related to the notion of node impurity and its gradual decrease as the tree is traversed. Stopping criteria are used in supplement to control the growth of the tree while various pruning methods

ensure decent generalization is preserved. Upon the arrival of a new feature vector, sequential decisions are made by traversing the tree top to bottom and making the final decision at the leaf level where class assignment rules are utilized.

D.Artificial Neural Networks (ANN)

The basic structure of ANN consists of an input and output layers with hidden layers in between each of varying or constant number of neurons that operate as simple computing elements to mimic the biological signal propagation while minimizing the empirical risk during the training phase. Generally speaking, SVM distinguish themselves from ANN by the fact that they don't suffer from the classical multi-local minima, the curse of dimensionality, and over fitting.

DT and ANN were able to cut 50 features and still achieve ~90% accuracy and precision. LM-SVM was able to reduce the least yet retained and even increased precision.

The agenda of the [2] is to implement learning techniques on an embedded platform. This can be achieved in two ways:

1) The first method is to train the ANN off-board. The artificial neural network is implemented on the embedded platform only and the training of the artificial neural network (A.N.N) is implemented off board. This can be done using a PC based system. The weights obtained from this off board training are then programmed into the embedded system.

2) The second method is to provide the learning function on-board. In this method the A.N.N as well as the learning algorithm is implemented on the embedded system itself. No external system is required to train the A.N.N.

The focus of [2] is on the second approach. In this approach the A.N.N operation and learning is implemented on the same embedded platform. This provides us with the flexibility of training the network instantly. In the first approach, a database of possible inputs and their respective outputs is to be maintained. This database is then given to training algorithms on a PC based system which provides us with the weights of each node in the network. These weights are then transferred to the embedded platform. The approach described in it bypasses all these steps by directly providing learning on the same platform. Whenever a new input is to be trained, we can simply put the system in training mode and provide the possible inputs and required outputs. The system will automatically adjust the weights of each node as per requirement.

The accuracy of the output achieved is slightly less compared to what PC based applications can achieve. This is because of the rounding off which occurs in calculations due to the 8-bit architecture. However with more research on the subject an algorithm can be developed which can achieve accuracy similar to high-end processor systems.

In [3], email data was classified using four different classifiers (Neural Network, SVM classifier, Naïve Bayesian Classifier, and J48 classifier). The experiment was performed based on different data size and different feature size. The final classification result should be '1' if it is finally spam, otherwise, it should be '0'. This paper shows that simple J48 classifier which make a binary tree, could be efficient for the dataset which could be classified as binary tree.

For the email classification, various classification methods used to classify incoming messages as spam or legitimate were:

A. Neural Network (NN)

Generally, the classification procedure using the NN consists of three steps, data preprocessing, data training, and testing. The data preprocessing refers to the feature selection. Feature selection is the way of selecting a set of features which is more informative in the task while removing irrelevant or redundant features. For the text domain, feature selection process will be formulated into the problem of identifying the most relevant word features within a set of text documents for a given text learning task. For the data training, the selected features from the data preprocessing step were fed into the NN, and an email classifier was generated through the NN. For the testing, the email classifier was used to verify the efficiency of NN. In the experiment, an error BP (Back Propagation) algorithm was used.

B. Support Vector Machines (SVM) Classifier

SVMs are a relatively new learning process influenced highly by advances in statistical learning theory. SVM learn by example. Each example consists of a m number of data points($x_1 \ldots x_m$) followed by a label, which in the two class classification we will consider later, will be +1 or -1. -1 representing one state and 1 representing another. The two classes are then separated by an optimum hyperplane, illustrated in figure above, minimizing the distance between the closest +1 and -1 points, which are known as support vectors. The right hand side of the separating hyperplane represents the +1 class and the left hand side represents the -1 class.

This classification divides two separate classes, which are generated from training examples. The overall aim is to generalize well to test data. This is obtained by introducing a separating hyperplane, which must maximize the margin () between the two classes, this is known as the optimum separating hyperplane.

3. Naive Bayesian (NB) Classifier

Naïve Bayesian classifier is based on Bayes' theorem and the theorem of total probability. The probability that a document *d* with vector $x = < x_1 \ldots x_n >$ belongs to category *c* is:

$$P(C = c \mid \overline{X} = \bar{x}) = \frac{P(C = c) \cdot P(\overline{X} = x \mid C = c)}{\prod_{k \in \{spam, legit\}} P(C = k) \cdot P(\overline{X} = \bar{x} \mid C = k)}$$

P(Xi |C) and P(C) are easy to obtain from the frequencies of the training dataset. So far, a lot of researches showed that the Naïve Bayesian classifier is surprisingly effective.

D. J48 Classifier

J48 classifier is a simple C4.5 decision tree for classification. It creates a binary tree.

Naive Bayesian classifier also showed good result, but Neural Network and SVM didn't show good result compared with J48 or Naïve Bayesian classifier. Neural Network and SVM were not appropriate for the dataset to make a binary decision. From this experiment, we could find it that a simple J48 classifier can provide better classification result for spam mail filtering.

In [4] authors have applied neural network and spam model based on Negative selection algorithm for solving complex problems in spam detection. This is achieved by distinguishing spam from non-spam (self from non-self).An optimized technique for e-mail classification is proposed where the e-mail are classified as self and non-self whose redundancy was removed from the detector set in the previous research to generate a self and non-self detector memory. A vector with an array of two element self and non-self concentration vector are generated into a feature vector used as an input in neural network classifier to classify the self and non-self feature vector of self and non-self program. The hybridization of both neural network and our previous model will further enhance the spam detector by improving the false rate and also enable the two different detectors to have a uniform platform for effective performance rate.

1.	**DEFINITIONS:**
2.	x is a self data set (spam)
3.	y is a non-self data set (non-spam)
4.	N is the number of matching data
5.	SM(0)=0, NSM(0)=0;
6.	**INPUT:**
7.	∝ /* is a threshold
8.	b /* b is the detector of x;
9.	a /* a is the detector of y;
10.	**OUTPUT:**
11.	Finding matching detector of both self and
12.	non-self
13.	**BEGIN**
14.	Input N;
15.	Input SM(1), NSM(1) /*SM is self matching
16.	and NSM is non-self matching;
17.	For i=1 to N
18.	SM(i) = SM(i) + SM (1- i);
19.	Next;
20.	For i =1 to N
21.	NSM (i) = NSM (i) + NSM (i - 1);
22.	**If faffinity>=**
23.	f affinity (x) = max;
24.	f affinity (y) = max;
25.	end if
26.	**if fmatching = =.T.**
27.	(b,x) >= ;
28.	else
29.	(a,y) >= ;
30.	End if
31.	End

SELF AND NON-SELF DETECTOR LIBRARY FOR NEGATIVE SELECTION ALGORITHM.

[5] deals with the phishing detection problem and how to detect phishing emails. The proposed phishing detection model is based on the extracted email features to detect phishing emails, these features appeared in the header and HTML body of email using feed forward neural network to classify the tested email into phish or ham email. A multilayer feed forward artificial neural network with back propagation, as a training algorithm, has been used for detecting phishing emails.

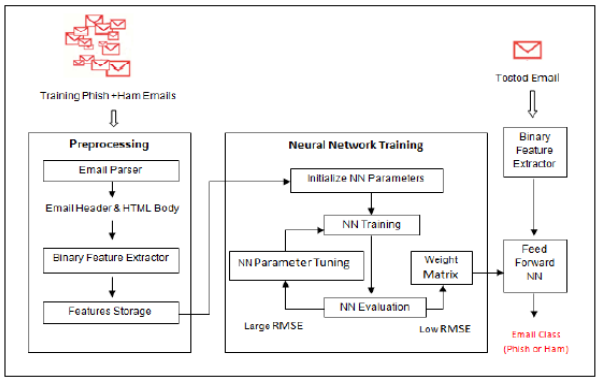

FIG. 1:MULTILAYER FEED FORWARD ANN ARCHITECTURE

To detect phishing emails using neural network the two phases (training and testing) need to be done. The steps used for detecting phishing emails using feed forward neural network is shown in Figure 1. The model consists of three stages, namely, pre-processing, neural network training and application of phish detection using feed forward neural network. In this approach, 18 features are implemented as a binary value (0 or 1); with a value 1 indicating this feature appeared in the tested email and 0 for non-appearance case.

This method is based on the features extracted from the header and the HTML body of the email. Eighteen common features have been extracted from each email in the training data set. The neural network consists of two phases:
1) Training Phase: In this phase the neural network was trained using 6000 phish and ham emails. The input to the neural network was 18 features extracted from each email with 1 hidden layer and 1 neuron in the output layer which is either (1 or 0). The output 1 indicates that the email is phish and 0 indicates that the email is ham.
2) Testing phase. Once the neural network was properly trained, it was tested over the test email data set (which is not used in the training phase), and also the neural network was tested using training data set.

The results of the conducted tests indicated good identification rate (98.72%) with short required processing time (0.00067 msec.).

The method in [5] employs attributes comprised from descriptive characteristics of the evasive patterns that spammers employ rather than the context or frequency of keywords in the messages to find out which ANN configuration will have the best performance and least error to desired output. As the mathematical modeling platform we were using NeuroDimensios graphical neural network development tool NeuroSolutions. According to Neural Network theory, for static pattern classification the best performance shows the layered feedforward networks, called Multilayer Perceptrons (MLPs), typically trained with static backpropagation. Their main advantage is that they are easy to use, and that they can approximate any input/output map. The key disadvantages are that they train slowly, and require lots of training data.

III. PROPOSED WORK

In our system, for detection of spam and phishing emails, back propagation along with multilayer feed forward artificial neural network has been used as a training algorithm. We

would also be implementing a k-mean clustering algorithm on the vector set to increase efficiency. To detect spam emails using neural network the two phases (training and testing) need to be done. The process of detecting spam and phishing emails using feed forward neural network is shown in Figure 2. There are mainly three stages in the proposed model : pre-processing, neural network training and application of phish detection using feed forward neural network. In this approach, 11 features have been implemented as a binary value (0 or 1); with a value 1 indicating this feature appeared in the tested email and 0 for non-appearance case.

3.1 Features used in Email Classification

Spam detection techniques are based on identification of a group of features consisting in the e-mail header and body. In this work a list of 11 features are extracted; they are binary features.

3.2 Training the Neural Network

This stage works on the binary files that are created in the pre-processing stage which consists of the features vectors of the emails. The input is the set of the extracted features from the header and HTML body of the email. Multiple hidden layers were used where the count of nodes in the hidden layer varies to get the best number of nodes which leads to minimum root mean square error (RMSE) for detecting spam emails. The number of output nodes are two, first to indicate email as Spam and second to indicate email as non-spam. The taken Activation sigmoid function is:

$output = 1/(1\text{-}e^{-input})$

3.3 Testing the Neural Network

The test emails are represented in form of the binary feature vector in the test phase of neural network. The binary feature vector is entered to the feed forward neural network that has the best found artificial neural network weight coefficients set, which is computed in the training phase, to classify the email into spam or ham email.

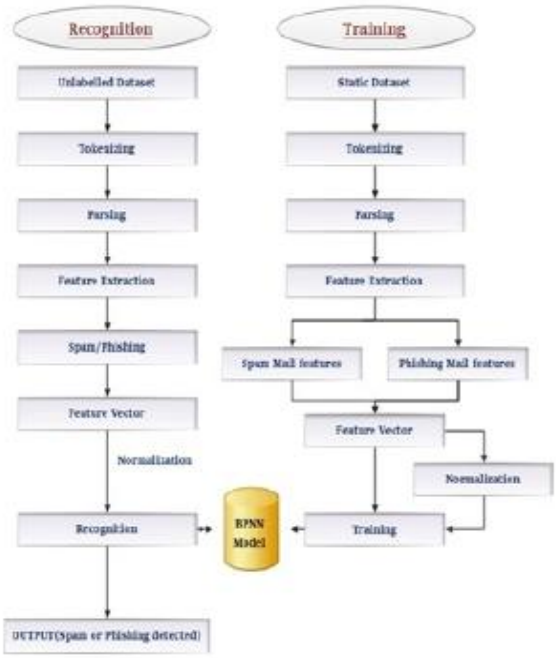

FIG. 2: PROPOSED SYSTEM ARCHITECTURE

IV. CONCLUSION

This paper focuses on survey of various context based classification algorithms for efficient email spam filtering. The paper also gives the comparison of result between different classification algorithms and also studies the various implementations methods to effectively classify emails into spam or non-spam. So that, this paper gives the brief idea about pit fall of different approaches and a comparison is performed on the SpamBase dataset to identify the best classification algorithm in terms of accuracy, computational time, and precision/recall rates.

ACKNOWLEDGEMENT

We thank the mysterious referees for their valuable suggestions to improvise the content and quality of this paper. The author is grateful to our principal for availing necessary facilities which helped for successful completion of work. We acknowledge the diligent efforts of our Head of the Department to guide us towards implementation of this review paper.

REFERENCES

[1] S. A. Saab, N. Mitri and M. Awad, "Ham or Spam? A comparative study for some Content-based Classification Algorithms for Email Filtering", Faculty of Electrical and Computer Engineering American University of Beirut, 2014.

[2] S. Shahane, S. Shendye and A. Shaikh, "Implementation of Artificial Neural Network Learning Methods on Embedded Platform", ISSN, 2014.

[3] S. Youn and D. McLeod, "A comparative study for email classification", In Advances and Innovations in Systems, Computing Sciences and Software Engineering, 2007.

[4] Ismaila Idris, "E-mail Spam Classification With Artificial Neural Network and Negative Selection Algorithm", Federal University of Technology, Minna, Nigeria.

[5] N. Ghazi, M. Jameel and L. E. George, "Detection of Phishing Emails using Feed Forward Neural Network", International Journal of Computer Applications, 2013.

[6] D. Punuskis ,R. Laurutis and R. Dirmeikis, "An artificial neural nets for spam email recognition", Electronics and Electrical Engineering, 2006.