https://doi.org/10.36375/prepare_u.a66

Spam E-mail classification using Machine Learning techniques

Sharika Anjum Mondal, Koustav Pal, Kalyan Chatterjee, Sayanti Banerjee
Department of Electronics and Communication Engineering
Amity University Kolkata
Major Arterial Road, Action Area II, New Town, Kolkata 700135
sharikaanjumm@gmail.com

Abstract – In today's world, bulk of emails is received by every individual out of which many fraudulent or spam emails are also present. The task of a good email service provider is to create an algorithm so that such fraudulent or spam messages are automatically detected and then they are sent to the spam folder. In this paper, the authors proposed a novel technique by which this sorting of email can be done automatically. Using machine learning method, the authors implemented a method in which spam mail and fraudulent messages have been successfully detected and those mails have been sent to the spam folder of the mailbox. The authors, in this paper, presented the description of the algorithm along with the test results.

Keywords – E-mail classification; **Machine learning algorithms**; **classifier**; **Naïve-byes.**

INTRODUCTION

Recently unsolicited commercial / big email, also known as spam, is becoming a big internet epidemic. Spam is waste of - time, memory space and bandwidth for communication. The spam email epidemic has been on the rise for years. In recent statistics, 40 percent of all emails are spam, which cost about 15.4 billion emails a day and about $355 million a year to internet users. At the moment, automated e-mail filtering seems to be the most efficient way to counter spam and there is a close rivalry between spammers and spam-filtering methods. Just a few years ago, most of the spam could be managed efficiently by blocking emails from certain addresses or filtering out messages with certain subject lines. Spammers began using several tricky strategies to circumvent filtering methods such as using random sender addresses and/or inserting random characters to the start or end of the subject line of the document. Nevertheless, we addressed the problem with the approach to machine learning, rather than using the approach to software engineering. Machine learning approach is more powerful than approach to software engineering; it does not allow any rules to be laid down. Alternatively, such samples are a set of pre-classified email addresses, a collection of training samples. Afterwards, a complex algorithm is used to learn from these email messages the classification rules. In unsupervised learning of ML, one tries to discover secret regularities (clusters) or detect irregularities in the data such as spam messages or intrusion into the network. Several features in the e-mail filtering process could be the word bag or the subject line review. The input to the function of e-mail classification can thus be interpreted as a two-dimensional matrix, whose axes are the messages and features. Tasks for classifying e-mails are often divided into several subtasks. First, Data collection and representation are mostly problem specific (i.e. e-mail messages), second, e-mail feature selection and feature reduction attempt to reduce the dimensionality (i.e. the number of features) for the remaining steps of the task. Finally, the process's e-mail classification step will find the actual mapping between the training set and the test set.

https://doi.org/10.36375/prepare_u.a66

Naïve Bayes classifier method

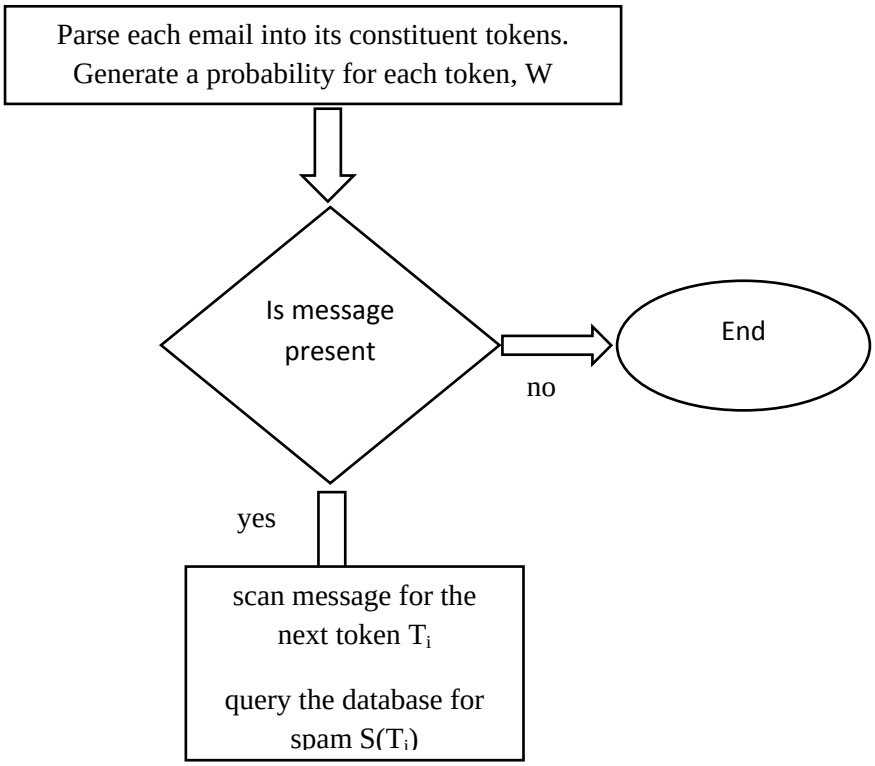
In 1998 the Naïve Bayes classifier was proposed for spam recognition. Bayesian classifier operates on the dependent events and the probability that an incident will occur in the future which can be predicted from the previous occurrence of the same event. You can use this technique to classify spam e-mails; word probabilities play the main rule here. If some words frequently occur in spam but not in ham, then this incoming e-mail is likely spam. Naïve Bayes classification technique has become a very popular method in software for mail filtering. Bayesian filter should be qualified for successful operation. Every word in its database has a certain probability of occurring in spam or ham email. If the total probabilities of words exceed a certain limit, then the filter marks the e-mail to either category. Here, it only takes two categories: spam or ham. Nearly all statistical-based spam filters use the Bayesian likelihood method to add the statistics of individual tokens to an overall score and make performance-based filtering decisions. The statistic we are mostly interested in a token T is its spam, calculated as follows:

$$S[T] = \frac{C[spam](T)}{C[spam](T) + C[ham](T)}$$

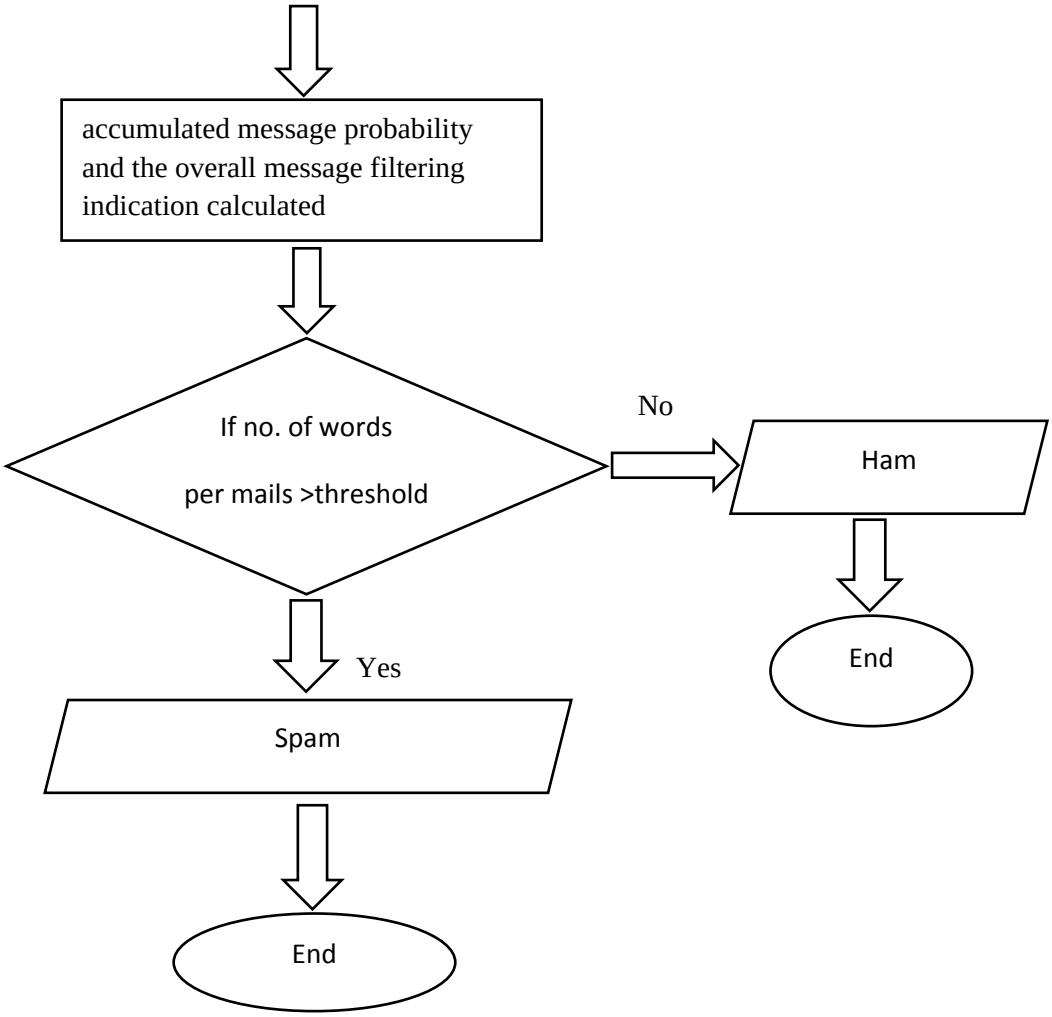
Where C[spam](T) and C[ham](T) are the number of spam or ham messages containing token T, respectively. To calculate the possibility for a message M with tokens {T_1, ..., Tn}, in order to determine the total spam message, one has to combine the individual spam token. A simple way to make classifications is to calculate the spam product of each token and to compare it with the ham product of the individual token.

$$H[M] = \sum(1 - S[T])$$

The message is considered spam if the overall spam product S[M] is larger than the ham product H[M]. The above description is used in the following algorithm:



https://doi.org/10.36375/prepare_u.a66



MACHINE LEARNING METHODS PERFORMANCE

Experiment Implementation

Some corpora of spam and legitimate emails had to be collected to check the efficiency of the above-mentioned method; other sets of emails are publicly accessible for researchers to use. Kaggle will be used in this experiment for having the dataset.

Total no of mails	4210
No. of Spam mails	2700
No. of Ham mails	1510
Total no. of words dealt with	1083821

Apart from the body letter of an email, an email has a different element called the header. The header's purpose is to store information about messages, and it includes several fields such as field (From) and (Subject), we decided to split the email into 3 different parts. The first item is the (subject) that can be considered to be the most relevant part of the report, it found that most of the new incoming emails have succinct subjects that can be used to make a clear distinction between spam and ham. The second part is (From) who is the person responsible for the message, this field we store it in a database and use it after the decision of the classifier has been made, that is the way to compare the field (From) stored in the database to the field (From) in the incoming new email if they are the same so that the decision of the incoming email is spam. The (Body) is the third part which represents the bulk of the message. We have also implemented two pre-processing procedures. Stop is used for the deletion of common words. Case-change is used to use small letters to represent the (Body). The experiment is carried out in spam email with the most commonly used words. In the processing stage, we pick 3000 words.

Detailed algorithm steps

Step 1: Email pre-processing

The email content is received through our software, the information is then extracted as mentioned above, then the extracted information (Feature) is saved into a corresponding database. For all the algorithms this function extraction scheme has been used.

Step 2: Description of the feature extracted

Feature extraction module extracts spam text and ham text, then produces feature dictionary and features vectors as the selected algorithm's input, the feature extraction function is to train the classifier and check it out. For the train portion, this module account word frequency in the email text, we take words that appear as the feature word of this class is more than three times the time of appearance. And in class, denote every email as a function vector.

Training data	80%
Testing data	20%

Step 3: Spam classification

Through the above steps, we take standard classification of email documents as a training document, pre-treatment of email, extracting useful information, saving in text documents according to fix format, splitting the entire document into words, extracting the feature vector of the spam document and converting it into the form of a fix format vector. Using the chosen algorithm, which is constructed using the feature vector of spam papers, we search for the optimal classification.

Step 4: Performance evaluation

We used the most common assessment methods used by the spam filtering researchers to evaluate the efficiency of the six above listed methods. Spam Precision (SP), Spam Recall (SR), Accuracy (A). Spam Precision (SP) is the number of relevant documents identified as

https://doi.org/10.36375/prepare_u.a66

$$SP = \frac{\#\text{ of Spam Correctly Classified}}{\text{Total \# of messages classifies as spam}} - \frac{M(spam \rightarrow spam)}{\text{M(spam} \rightarrow \text{spam)} + \text{M(ham} \rightarrow \text{spam)}}$$

percentage of all documents identified; this shows the noise that filter presents to the user (i.e. how many of the messages classified as spam will actually be spam)

Spam Recall (SR) is the percentage of all spam emails that are correctly classified as spam.

$$SR = \frac{\#\text{ of Spam Correctly Classified}}{\text{Total \# of messages}} = \frac{\text{M(spam} \rightarrow \text{spam)}}{\text{Mspam} \rightarrow \text{spam} + \text{M(spam} \rightarrow \text{ham)}}$$

Accuracy (A) is the percentage of all emails that are correctly categorized

$$A = \frac{\#\text{ of E} - \text{mails correctly categorized}}{\text{Total \# of E} - \text{mails}} = \frac{\text{M(ham} \rightarrow \text{ham)} + \text{M(spam} \rightarrow \text{spam)}}{\text{M(ham)} + \text{M(spam)}}$$

Where M(ham→ham) and M(spam→spam) are the number of messages that have been correctly classified to the legitimate email and Spam email respectively; M(ham→spam) and M(spam→ham) are the number of legitimate and spam messages that have been misclassified; M(ham) and M(spam) are the total number of legitimate and spam messages to be classified.

Output 1:

```
Enter the mail : Hi! you have won a lottery of 1 crore. To claim your reward call on 9674803230
Out[48]: "Alert!It's a SPAM mail!!"
```

Output 2:

```
Enter the mail : Hi Sharika!! How are you?
Out[49]: 'New Mail!!'
```

CONCLUSION

Spam and fraudulent email detection are a very important task and the automated process will reduce the overall complexity on the process. This model is correctly working with all kinds of email and is having an accuracy of 94.78%. Thus, this can be deployed for industrial purpose.

References:

[1] M. N. Marsono, M. W. El-Kharashi, and F. Gebali, “Binary LNS-based naïve Bayes inference engine for spam control: Noise analysis and FPGA synthesis”, IET Computers & Digital Techniques, 2008
[2] Muhammad N. Marsono, M. Watheq El-Kharashi, Fayez Gebali “Targeting spam control on middleboxes: Spam detection based on layer-3 e-mail content classification” Elsevier Computer Networks, 2009
[3] Yuchun Tang, Sven Krasser, Yuanchen He, Weilai Yang, Dmitri Alperovitch ”Support Vector Machines and Random Forests Modeling for Spam Senders Behavior Analysis” IEEE GLOBECOM, 2008

International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011
184
[4] Guzella, T. S. and Caminhas, W. M. ”A review of machine learning approaches to Spam filtering.” Expert Syst. Appl., 2009
[5] Wu, C. ”Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks” Expert Syst., 2009
[6] Khorsi. “An overview of content-based spam filtering techniques”, Informatica, 2007
[7] Hao Zhang, Alexander C. Berg, Michael Maire, and Jitendra Malic. "SVM-KNN: Discriminative nearest neighbour classification for visual category recognition", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006
[8] Carpinteiro, O. A. S., Lima, I., Assis, J. M. C., de Souza, A. C. Z., Moreira, E. M., & Pinheiro, C. A. M. "A neural model in anti-spam systems.", Lecture notes in computer science.Berlin, Springer, 2006
[9] El-Sayed M. El-Alfy, Radwan E. Abdel-Aal "Using GMDH-based networks for improved spam detection and email feature analysis"Applied Soft Computing, Volume 11, Issue 1, January 2011
[10] Li, K. and Zhong, Z., “Fast statistical spam filter by approximate classifications”, In Proceedings of the Joint international Conference on Measurement and Modeling of Computer Systems. Saint Malo, France, 2006
[11] Cormack, Gordon. Smucker, Mark. Clarke, Charles " Efficient and effective spam filtering and re-ranking for large web datasets" Information Retrieval, Springer Netherlands. January 2011
[12] Almeida,tiago. Almeida, Jurandy.Yamakami, Akebo " Spam filtering: how the dimensionality reduction affects the accuracy of Naive Bayes classifiers" Journal of Internet Services and Applications, Springer London , February 2011