Email based Spam Detection

Thashina Sultana, K A Sapnaz, Fathima Sana, Mrs. Jamedar Najath
Dept. of Computer Science and Engineering
Yenepoya Institute of Technology
Moodbidri, India

***Abstract*— Nowadays, a big part of people rely on available email or messages sent by the stranger. The possibility that anybody can leave an email or a message provides a golden opportunity for spammers to write spam message about our different interests .Spam fills inbox with number of ridiculous emails . Degrades our internet speed to a great extent .Steals useful information like our details on our contact list. Identifying these spammers and also the spam content can be a hot topic of research and laborious tasks. Email spam is an operation to send messages in bulk by mail .Since the expense of the spam is borne mostly by the recipient ,it is effectively postage due advertising. Spam email is a kind of commercial advertising which is economically viable because email could be a very cost effective medium for sender .With this proposed model the specified message can be stated as spam or not using Bayes' theorem and Naive Bayes' Classifier and Also IP addresses of the sender are often detected .**

***Keywords*— Term Frequency, Inverse Document Frequency, language tool kit.**

I. INTRODUCTION

In recent years, internet has become an integral part of life. With increased use of internet, numbers of email users are increasing day by day. This increasing use of email has created problems caused by unsolicited bulk email messages commonly referred to as Spam. Email has now become one of the best ways for advertisements due to which spam emails are

generated. Spam emails are the emails that the receiver does not wish to receive. a large number of identical messages are sent to several recipients of email. Spam usually arises as a result of giving out our email address on an unauthorized or unscrupulous website .There are many of the effects of Spam .Fills our Inbox with number of ridiculous emails .Degrades our Internet speed to a great extent .Steals useful information like our details on you Contact list .Alters your search results on any computer program .Spam is a huge waste of everybody's time and can quickly become very frustrating if you receive large amounts of it .Identifying these spammers and the spam content is a laborious task . even though extensive number of studies have been done, yet so far the methods set forth still scarcely distinguish spam surveys, and none of them demonstrate the benefits of each removed element compose .In spite of increasing network communication and wasting lot of memory space ,spam messages are also used for some attack . Spam emails, also known as non-self, are unsolicited commercial or malicious emails, sent to affect either a single individual or a corporation or a bunch of people. Besides advertising, these may contain links to phishing or malware hosting websites found out to steal confidential information. to solve this problem the different spam filtering techniques are used. The spam filtering techniques are accustomed protect our mailbox for spam mails.

II. LITERATURE SURVEY

In the paper[1], authors have highlighted several features contained in the email header which will be used to identify and classify spam messages efficiently .Those features are selected based on their performance in detecting spam messages. This paper also communalize each features contains in Yahoo mail,Gmail and Hotmail so a generic spam messages

detection mechanism could be proposed for all major email providers.

In the paper[2], a new approach based on the strategy that how frequently words are repeated was used. The key sentences, those with the keywords, of the incoming emails have to be tagged and thereafter the grammatical roles of the entire words in the sentence need to be determined, finally they will be put together in a vector in order to take the similarity between received emails. K-Mean algorithm is used to classify the received e-mail. Vector determination is the method used to determine to which category the e-mail belongs to.

In the paper[3],authors described about cyber attacks .Phishers and malicious attackers are frequently using email services to send false kinds of messages by which target user can lose their money and social reputations. These results into gaining personal credentials such as credit card number, passwords and some confidential data .In This paper ,authors have used Bayesian Classifiers .Consider every single word in the mail. Constantly adapts to new forms of spam.

In the paper[4],proposed system attempts to use machine learning techniques to detect a pattern of repetitive keywords which are classified as spam. The system also proposes the classification of emails based on other various parameters contained in their structure such as Cc/Bcc, domain and header. Each parameter would be considered as a feature when

applying it to the machine learning algorithm. The machine learning model will be a pre-trained model with a feedback mechanism to distinguish between a proper output and an ambiguous output. This method provides an alternative architecture by which a spam filter can be implemented. This paper also takes into consideration the email body with commonly used keywords and punctuations.

In the paper[5],authors investigated the use of string matching algorithms for spam email detection. Particularly this work examines and compares the efficiency of six well-known string matching algorithms, namely Longest Common Subsequence (LCS), Levenshtein Distance (LD), Jaro , Jaro - Winkler, Bi-gram, and TFIDF on two various datasets which are Enron corpus and CSDMC2010 spam dataset. They observed that Bi-gram algorithm performs best in spam detection in both datasets.

III. PROPOSED SYSTEM

In this system, to solve the problem of spam, the spam classification system is created to identify spam and non-spam. Since spammers may send spam messages many times, it is difficult to identify it every time manually .So we will be using some of the strategies in our proposed system to detect the spam. The proposed solution not only identifies the spam word but also identifies the IP address of the system through which the spam message is sent so that next time when the spam message is sent from the same system our proposed system directly identifies it as blacklisted based on the IP address.

In the proposed model ,the web application is done using dot net and spam detection is done using machine learning .The web application consists of following modules:

1. User Management :

The user who is using this for the very first time must register, by using the website the user or the individual should get registered into it, by registering this will help to maintain separate account for each user. Registration of the user is must before they log in. The user will login to the main page with his registered name and password. Once the user successfully login the authorized page will be displayed otherwise that shows the error messages. Login is compulsory.

Login: The user will login to the main page with his registered name and password. Once the user successfully login the authorized page will be displayed otherwise that shows the error messages. Login is compulsory.

Registration: First time while using the website the user or the individual should get registered into it, by registering this will help to maintain separate account for each user. Registration of the user is must before they log in.

2. Compose

Input: the sender will compose the new email; the
sender should add the address of the recipient, the subject and the message.

Output: the email will be sent based to the address mentioned by the recipient.

3. Inbox

This page will store all of the mails received by user. All the received Mails will be listed sorted in order of date.

Input: the inbox page will accept all the incoming emails sent to an individual.

Output: the receiver can open and read the email received to their address.

4. Sent

This folder stores all the mails sent from the user.

Input: here the sender will compose an email and send to the recipient.

Output: Sent email can be be read out .

5. Trash

This folder will store all of mails deleted by the user.

Input: select and Delete all the unwanted emails.

Output: all the deleted emails are added in the trash bin. Trash bin stores all the deleted emails.

6. Voice Message

Input: The Email has been sent in the form of the text message by the sender

Output: The email has been read through the use of voice note by the receiver.

7. Offline notification

Input: The sender sends an email

Output: the receivers receive a notification offline in the text format as SMS.

8. Delete For everyone

Input: here the sender deletes the email which he has sent

Output: the email has been erased or deleted for both the sender as well as the receiver.

9. Read Message

Input: The receiver will read the email.

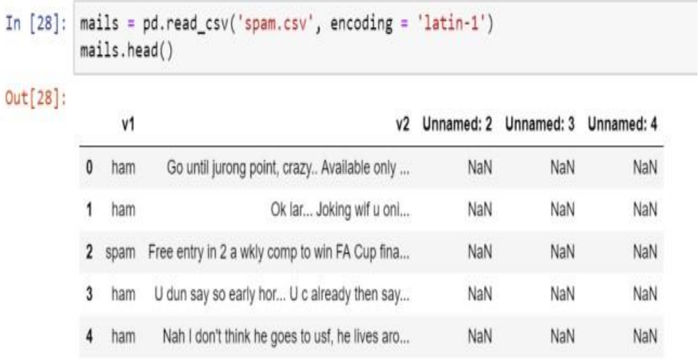
Output: the sender will get a notification stating the sender as read the message.

When we receive message in the inbox ,that message will be exported to dataset. This message will be detected as spam or not using Naïve Bayes Classifier.

Before detecting whether received message is spam or not ,the model has to be trained which is explained in the below section.

IV. SPAMDETECTION USING MACHINE LEARNING|

1. For training the algorithm dataset from Kaggle is used which is shown below



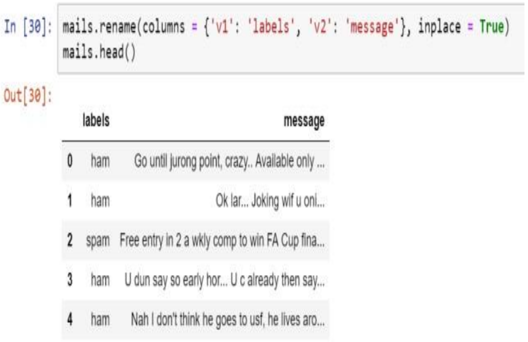
```
In [28]: mails = pd.read_csv('spam.csv', encoding = 'latin-1')
         mails.head()
```

Out[28]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

Fig.1. Dataset

2. It has many fields, some of these columns of the dataset are not required. So remove some columns which are not required. We need to change the names of the columns.



```
In [30]: mails.rename(columns = {'v1': 'labels', 'v2': 'message'}, inplace = True)
         mails.head()
```

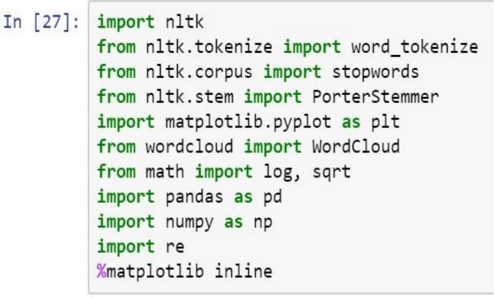
Out[30]:

	labels	message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

Fig.2. Classification dataset

With the help of NLTK (Natural Language Tool Kit) for the text processing, Using **Matplotlib** you can plot graphs , histogram and bar plot and all those things ,Word Cloud is used to present text data and pandas for data manipulation and analysis, NumPy is to do the mathematical and scientific operation.

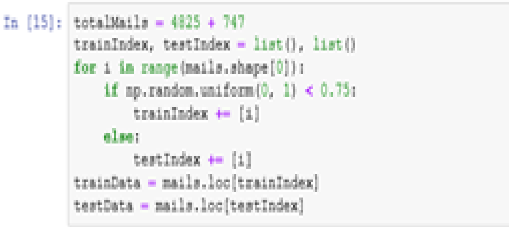
The packages used in the proposed model are shown below.



```
In [27]: import nltk
         from nltk.tokenize import word_tokenize
         from nltk.corpus import stopwords
         from nltk.stem import PorterStemmer
         import matplotlib.pyplot as plt
         from wordcloud import WordCloud
         from math import log, sqrt
         import pandas as pd
         import numpy as np
         import re
         %matplotlib inline
```

Fig.3.Packages

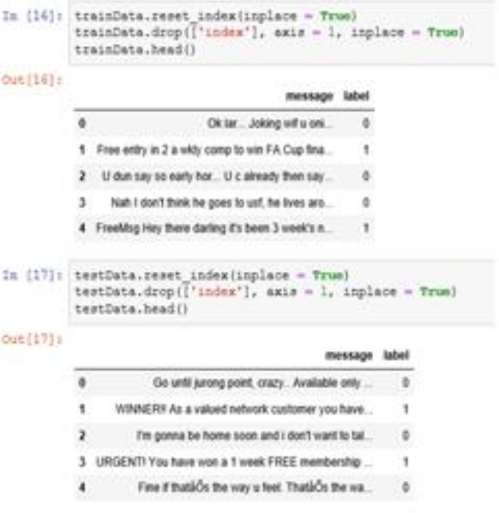
3. Split the data into training and testing sets as shown below. Some percentage f the data set is used as train dataset and the rest as a test dataset.



```
In [15]: totalMails = 4825 + 747
         trainIndex, testIndex = list(), list()
         for i in range(mails.shape[0]):
             if np.random.uniform(0, 1) < 0.75:
                 trainIndex += [i]
             else:
                 testIndex += [i]
         trainData = mails.loc[trainIndex]
         testData = mails.loc[testIndex]
```

Fig.4.Train dataset

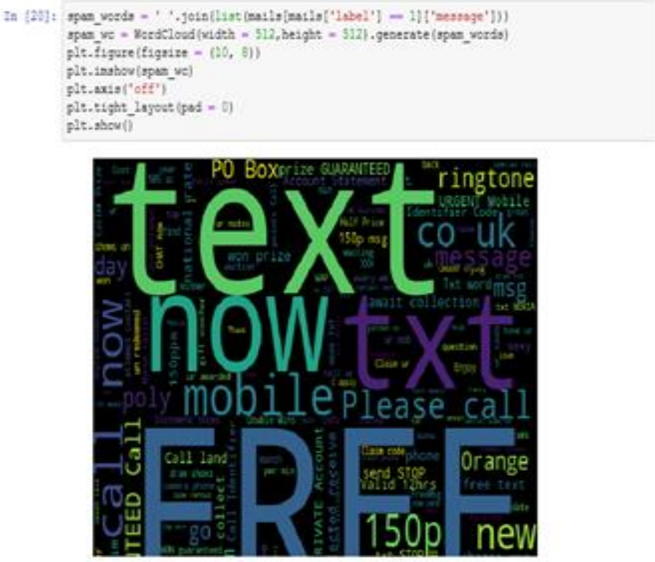
4. Reset train and test index as shown in the next column:



```
In [16]: trainData.reset_index(inplace = True)
         trainData.drop(['index'], axis = 1, inplace = True)
         trainData.head()

Out[16]:
                                          message  label
         0                   Ok lar... Joking wif u oni...      0
         1   Free entry in 2 a wkly comp to win FA Cup fina...      1
         2   U dun say so early hor... U c already then say...      0
         3   Nah I don't think he goes to usf, he lives aro...      0
         4   FreeMsg Hey there darling it's been 3 week's n...      1

In [17]: testData.reset_index(inplace = True)
         testData.drop(['index'], axis = 1, inplace = True)
         testData.head()

Out[17]:
                                            message  label
         0     Go until jurong point, crazy.. Available only ...      0
         1       WINNER!! As a valued network customer you have...      1
         2       I'm gonna be home soon and i don't want to tal...      0
         3   URGENT! You have won a 1 week FREE membership ...      1
         4        Fine if thatåÕs the way u feel. ThatåÕs the wa...      0
```

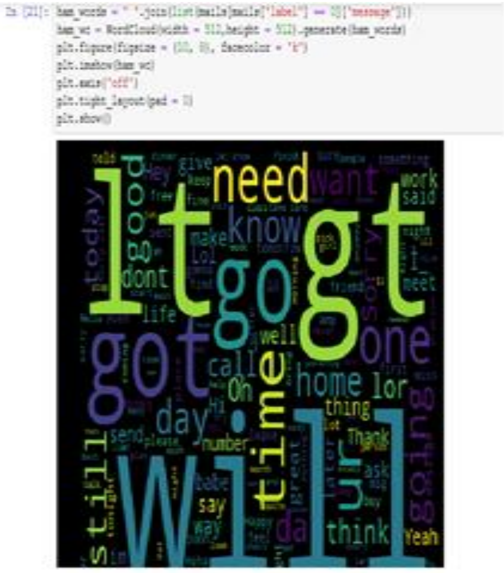
Fig.5. Reset train and test index

5. We need to find out the most repeated words in the spam and ham messages.So Word Cloud library is used.



```
In [20]: spam_words = ' '.join(list(mails[mails['label'] == 1]['message']))
         spam_wc = WordCloud(width = 512,height = 512).generate(spam_words)
         plt.figure(figsize = (10, 8))
         plt.imshow(spam_wc)
         plt.axis('off')
         plt.tight_layout(pad = 0)
         plt.show()
```

Fig.6.Spam word cloud



```
In [21]: ham_words = ' '.join(list(mails[mails['label'] == 0]['message']))
         ham_wc = WordCloud(width = 512,height = 512).generate(ham_words)
         plt.figure(figsize = (10, 8), facecolor = 'k')
         plt.imshow(ham_wc)
         plt.axis('off')
         plt.tight_layout(pad = 0)
         plt.show()
```

Fig.7. Ham word cloud

5. Whenever there is any message, we must first preprocess the input messages. We need to convert all the input characters to lowercase.

6. Then split up the text into small pieces and also removing the punctuations. So the **Tokenization** process is used to remove punctuations and splitting messages.

7. The **Porter Stemming Algorithm is used for** stemming**. Stemming** is the **process** of reducing words to their root word.

8. We need to find the probability of the word in spam and ham messages.

$$P(word) = \frac{Total\ occurrences\ of\ word}{Total\ number\ of\ words}$$

Eqn.1. Frequency of word

Then spam word frequency is calculated as follows:

$$P(word|spam) = \frac{Total\ occurrences\ of\ the\ word\ in\ spam\ message}{Total\ number\ of\ words\ in\ the\ spam\ message}$$

Eqn.2.Spam word frequency

9. **Tf –idf(**term frequency-inverse document frequency) has to be calculated.

TF: Term Frequency, which measures how many times a term occurs in a document.

TF(t) = (Number of times t appeared in a document) / (Total terms in the document).

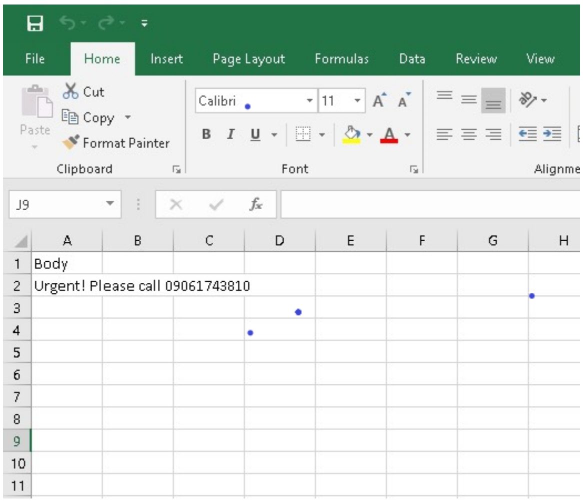
IDF: Inverse Document Frequency, which measures the significance of the term.

IDF(t) = loge(Total documents / documents with term t in it).

10. See how well the model performed by evaluating **Naïve Bayes Classifier** and showing the accuracy score.

V. RESULTS AND DISCUSSIONS

When we receive message in the inbox ,that message will be exported to dataset as shown below. This message will be detected as spam or not.



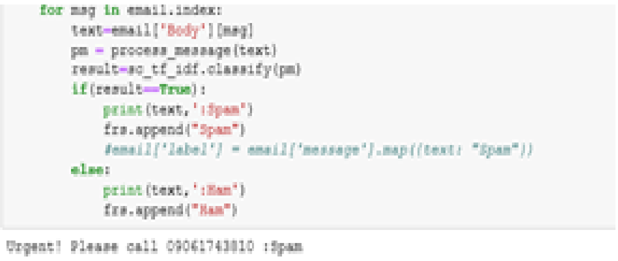
	A
1	Body
2	Urgent! Please call 09061743810

Fig.8. Exported Dataset

The exported message will be detected as spam or not using **Bayes' theorem and Naive Bayes' Classifier** following all the steps discussed above along with finding probability of words in spam and ham messages to detect it as spam or not.

The below figures shows message which got detected as spam and ham.

If "Urgent! Please call 09062703810" is an exported message from the inbox to the dataset then based on trained dataset and using **Bayes' theorem and Naive Bayes' Classifier,** the above message is detected as **Spam** as shown below.

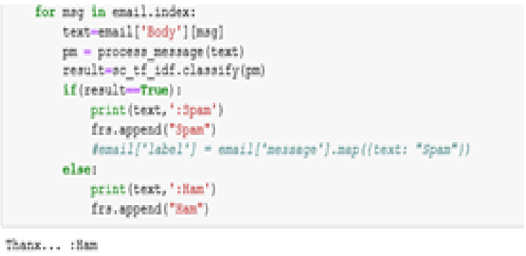


```
for msg in email.index:
    text=email['Body'][msg]
    pm = process_message(text)
    result=sc_tf_idf.classify(pm)
    if(result==True):
        print(text,':Spam')
        frs.append("Spam")
        #email['label'] = email['message'].map((text: "Spam"))
    else:
        print(text,':Ham')
        frs.append("Ham")
```

```
Urgent! Please call 09061743810 :Spam
```

Fig.9.Spam Message

If "Thanx" is an exported message from the inbox to the dataset then using **Bayes' theorem and Naive Bayes' Classifier,** the above message is detected as **Ham** as shown below.

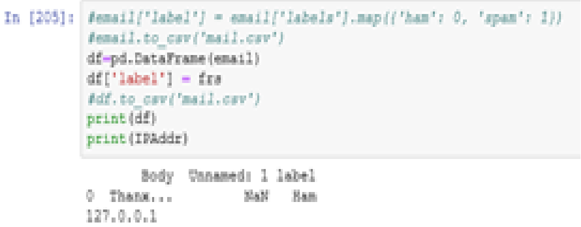


```
for msg in email.index:
    text=email['Body'][msg]
    pm = process_message(text)
    result=sc_tf_idf.classify(pm)
    if(result==True):
        print(text,':Spam')
        frs.append("Spam")
        #email['label'] = email['message'].map((text: "Spam"))
    else:
        print(text,':Ham')
        frs.append("Ham")
```

```
Thanx... :Ham
```

Fig.10.Ham message

The IP address of the sender can also be detected.



```
In [205]: #email['label'] = email['labels'].map({'ham': 0, 'spam': 1})
          #email.to_csv('mail.csv')
          df=pd.DataFrame(email)
          df['label'] = frs
          #df.to_csv('mail.csv')
          print(df)
          print(IPAddr)
```

```
       Body  Unnamed: 1 label
0  Thanx...         NaN   Ham
127.0.0.1
```

Fig.11.IP address of the sender

VI. CONCLUSION

Email has been the most important medium of communication nowadays, through internet connectivity any message can be delivered to all aver the world. More than 270 billion emails are exchanged daily, about 57% of these are just spam emails. Spam emails, also known as non-self, are undesired commercial or malicious emails, which affects or hacks personal information like bank ,related to money or anything that causes destruction to single individual or a corporation or a group of people. Besides advertising, these may contain links to phishing or malware hosting websites set up to steal confidential information. Spam is a serious issue that is not just annoying to the end-users but also financially damaging and a security risk. Hence this system is designed in such a way that it detects unsolicited and unwanted emails and prevents them hence helping in reducing the spam message which would be of great benefit to individuals as well as to the company .In the future this system can be implemented by using different algorithms and also more features can be added to the existing system.

REFERENCES

[1] Shukor Bin Abd Razak, Ahmad Fahrulrazie Bin Mohamad "Identification of Spam Email Based on Information from Email Header" 13th International Conference on Intelligent Systems Design and Applications (ISDA), 2013.

[2] Mohammed Reza Parsei, Mohammed Salehi "E-Mail Spam Detection Based on Part of Speech Tagging" 2nd International Conference on Knowledge Based Engineering and Innovation (KBEI), 2015.

[3] Sunil B. Rathod, Tareek M. Pattewar "Content Based Spam Detection in Email using Bayesian Classifier", presented at the IEEE ICCSP 2015 conference.

[4] Aakash Atul Alurkar, Sourabh Bharat Ranade, Shreeya Vijay Joshi, Siddhesh Sanjay Ranade, Piyush A. Sonewa, Parikshit N. Mahalle, Arvind V. Deshpande "A Proposed Data Science Approach for Email Spam Classification using Machine Learning Techniques", 2017.

[5] Kriti Agarwal, Tarun Kumar "Email Spam Detection using integrated approach of Naïve Bayes and Particle Swarm Optimization", Proceedings of the Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018.

[6] Cihan Varol, Hezha M.Tareq Abdulhadi "Comparison of String Matching Algorithms on Spam Email Detection", International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism Dec, 2018.

[7] Duan, Lixin, Dong Xu, and Ivor Wai-Hung Tsang. "Domain adaptation from multiple sources: A domaindependent regularization approach." IEEE Transactions on Neural Networks and Learning Systems 23.3 (2012).

[8] Mujtaba, Ghulam, et al. "Email classification research trends: Review and open issues." IEEE Access 5 (2017).

[9] Trivedi, Shrawan Kumar. "A study of machine learning classifiers for spam detection." Computational and Business Intelligence (ISCBI), 2016 4th International Symposium on. IEEE, 2016. [10] You, Wanqing, et al. "Web Service-Enabled Spam Filtering with Naïve Bayes Classification." 2015 IEEE First International Conference on Big Data Computing Service and Applications (BigDataService). IEEE, 2015.

[10] Rathod, Sunil B., and Tareek M. Pattewar. "Content based spam detection in email using Bayesian classifier." International Conference on. IEEE, 2015.

[11] Sahın, Esra, Murat Aydos, and Fatih Orhan. "Spam/ham e-mail classification using machine learning methods based on bag of words technique." 2018 26th Signal Processing and Communications Applications Conference (SIU). IEEE, 2018.