IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

SPAM MAIL PREDICTION USING MACHINE LEARNING

[1]B. Uday Reddy, [2]S. Nagasai Tej , [3]Md. Shoheb, [4]Dr. Krishna Samalla, [5]Y. Sreenivasulu

[1]Student, [2]Student, [3]Student, [4]Professor, [5]Associate Professor

[1,2,3,4,5]Department of Electronics and Communication

[1,2,3,4,5]Sreenidhi Institute of Science and Technology, Ghatkesar, Hyderabad, India

Abstract: Online reviews are often the primary factor in a customer's decision to purchase a product or service, and are a valuable source of information that can be used to determine public opinion on these products or services. Because of their impact, manufacturers and retailers are highly concerned with customer feedback and reviews. This practice is known as Opinion (Review) Spam, where spammers manipulate and poison reviews (i.e., making fake, untruthful, or deceptive reviews) for profit or gain. Since not all online reviews are truthful and trustworthy, it is important to develop techniques for detecting review spam. By extracting meaningful features from the text using Natural Language Processing (NLP), it is possible to conduct review spam detection using various machine learning techniques. The majority of current research has focused on supervised learning methods, which require labelled data, a scarcity when it comes to online review spam. Research on methods for Big Data are of interest, since there are millions of online reviews, with many more being generated daily.

Keywords : Spam, Natural Processing Language (NLP), Ham, Big Data

I. INTRODUCTION :

Email is the worldwide use of communication applications. It is because of the ease of use and faster than other communication applications. However, its inability to detect whether the mail content is either spam or ham degrades its performance. Nowadays, a lot of cases have been reported regarding stealing of personal information or phishing activities via email from the user. This project will discuss how machine learning helps in spam detection. Machine learning is an artificial intelligence application that provides the ability to automatically learn and improve data without being explicitly programmed. Binary classifier will be used to classify the text into two different categories; spam and ham.

II. LITERATURE SURVEY

Today, spam has become a big internet issue. In 2017, the statistics showed spam accounted for 55% of all e-mail messages, same as during the previous year. Spam which is also known as unsolicited bulk email has led to the increasing use of email as email provides the perfect ways to send the unwanted advertisement or junk newsgroup posting at no cost for the sender. This chance has been extensively exploited by irresponsible organizations and resulting to clutter the mail boxes of millions of people all around the world. Evolving from a minor to major concern, given the high offensive content of messages, spam is a waste of time. It also consumed a lot of storage space and communication bandwidth. End user is at risk of deleting legitimate mail by mistake. Moreover, spam also impacted the economy which led some countries to adopt legislation. Text classification is used to determine the path of incoming mail/message either into the inbox or straight to the spam folder. It is the process of assigning categories to text according to its content. It is used to organize, structure and categorize text. It can be done either manually or automatically. Machine learning automatically classifies the text in a much faster way than manual technique. Machine learning uses pre-labelled text to learn the different associations between pieces of text and its output. It used feature extraction to transform each text to numerical representation in the form of a vector which represents the frequency of word in a predefined dictionary

PROBLEM STATEMENT:

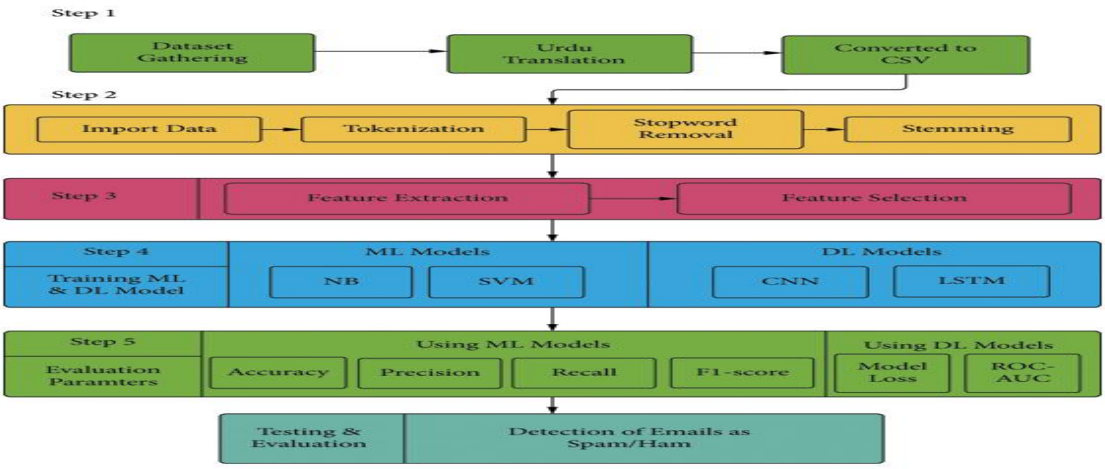
"To study on how to use machine learning techniques for spam detection, modify machine learning algorithms in computer system settings, leverage modified machine learning algorithms in knowledge analysis software and to test the machine learning algorithm real data from a machine learning data repository"

III. PROPOSED SYSTEM

Data Processing and Methodology

a. As for data model, it refers to the documenting a complex system and data flow between different data elements and design as an easily understood diagram using text and symbols. The data flow below shows how the data flow of these projects in order to detect the spam messages and classify them into two separate types which are spam and ham message. The data model flow is essential to this project to show the structure of the project on how it should be built and how the process is related to each other. It helps to organize the process of the project smoothly and clearly. Based on the framework, Azure ML Studio is used as the platform to develop the project. First, study and discover all the functionality on Azure ML to make sure the project can achieve its objectives. After that, make sure to download and use the real existing dataset from the machine learning data repository as training and testing data. Preprocessing data starts with reformatting the dataset into 2 separate files which are training.csv and testing.csv format. Then, upload the formatted dataset into Azure ML under dataset function/menu and drag them onto the workspace to visualize the data. Choose any desired filters to clean the raw data such as “remove numbers" filter. In feature extraction, we will transform the data so that it can be used to train the classifier by using the Vowpal Wabbit algorithm. First, use the feature hashing step to change the message hashing bit size. This step is important to extract all pairs of bigrams, compute an 8 bit hash for each bigram and create a new column for each hast in the output dataset. 22 Model training step include 2 steps which is picking a classifier and scoring the classifier. Two-Class Logistic Regression is used to predict the probability of spam detection either it is spam or ham. After the data have been trained, the model needs to be tested to evaluate its accuracy and overstrain the model so that it memorizes the data. Web service is set up so that the model can be used. First, select only the message column by using Select Columns in the Dataset step so that the data can be tested in the web service.

b. The data model flow is essential to this project to show the structure of the project on how it should be built and how the process is related to each other. It helps to organize the process of a project smoothly and clearly. Based on the framework, Azure ML Studio is used as the platform to develop the project. First, study and discover all the functionality on Azure ML to make sure the project can achieve its objectives.



The first stage is to import the dataset, which is downloaded from ‘Kaggle’ and then converted to CSV format in Urdu. The dataset containing 5000 emails were already classified as spam and ham. The data was obtained while being written in the English Language. As explained before, unlike the usual approach, we have translated the data set into Urdu, to achieve our goal of Urdu spam e-mail detection. We created our own dataset, because Roman Urdu is written using English alphabets, and Urdu script is based on Arabic alphabets. Urdu scripts are distinct from Roman Urdu scripts.

Being a critical phase of preprocessing, in this step, all the words from emails are gathered, and the number of times each word appears and location of appearance are counted. With the aid of Count Vectorizer, we were able to find the repetition of words in our dataset. Each word is given a unique number, and hence, they are called tokens, also depicting their occurrences and quantity of occurrences. The token includes one of a kind feature values that will later help in the creation of feature vectors. In a tokenization phase, every word is assigned a unique token.

IV . METHODOLOGY

Process model is a series of steps, concise description and decisions involved in order to complete the project implementation. In order to finish the project within the time given, the flows of the project need to be followed. The framework below shows the overall flow of this project in order to separate between a spam and ham message.

Methodology is an important role as a guide for this project to make sure it is on the right path and working as well as plan. There are different types of methodology used in order to do spam detection and filtering. So, it is important to choose the right and suitable methodology thus it is necessary to understand the application functionality itself.

IMPLEMENTATION AND CODING PHASE

This project is developed by using Python Language and combining with the Vowpal Wabbit algorithm. Azure machine learning studio is the platform to develop the project. It contains an important function for preprocessing the dataset. Then, the dataset is going to be used to train and test whether the model of machine learning will achieve the objectives.

3.3 PROJECT REQUIREMENT AND SPECIFICATION

System requirement is needed in order to accomplish the project goals and objectives and to assist in development of the project that involves the usage of hardware and software. Each of these requirements is related to each other to make sure that system can be done smoothly.

Step 1. Pick a random mail from the collection for testing purposes.
Step 2. The e-mail in question is in its unprocessed state. E-mail must be preprocessed before the feature extraction and classification procedure can begin. Tokenization, stemming, and stop word elimination are all steps in the preprocessing process:
(1)To begin, split down the e-mail into distinct words and tokenize it. Tokenization separates each word into its own token.
(2)Eliminate all punctuation marks from the characters you obtained through tokenization.
(3)Stemming is done with the tokens earned in the previous stage. The stemming process decreases the size of a word to its base word. For stemming, a predetermined range of available words is examined, as well as the irrespective stem words.
(4)For stemming, a list of suffixes keywords is maintained in an array with their base words.
(5)Check to see whether there are any tokens available in the base input text.
(6)Stem the phrase to the proper base word from the array list if the test token's suffixes are true.
(7)Otherwise, stemming is unnecessary. Word has already been converted to its root word format. Therefore, proceed to the next token.
Step3. To use the feature extraction technique, select suitable attribute words from the validation set. Just the set of features that is most nearly connected to the category is selected.
Step4. Use extracted features and created tokens to train ML and DL models. That model can easily distinguish between spam and ham emails.
Step5. Tokens are classified as spam or ham based on their feature similarity as ML models determine.
Step 6. Finally, the likelihood of spam or ham tokens in a sentence is evaluated for final classification:
(1)The mail is regarded spam if the significance level of spam tokens is higher than zero
(2)Otherwise, e-mail is regarded as ham e-mail
Step7. Mark the e-mail as spam or ham and proceed with the rest of the emails.

ALGORITHM EMPLOYED :

To get started, first, run the code below:

```
spam = pd.read_csv('spam.csv')
```

In the code above, we created a spam.csv file, which we'll turn into a data frame and save to our folder spam. A data frame is a structure that aligns data in a tabular fashion in rows and columns, like the one seen in the following image.

ham	Hi there, here's the newest game of our studio
spam	Hi from Mrs. Alice
ham	Click on the llink to verify your email address
ham	Here's what you requested, tada!
spam	Hey buddy! How you doin'?

Run the command below:

```
z = spam['EmailText']
y = spam["Label"]
z_train, z_test,y_train, y_test = train_test_split(z,y,test_size = 0.2)
```

z = spam['EmailText'] assigns the column EmailText from spam to z. It contains the data that we'll run through the model. y = spam["Label"] assigns the column Label from spam to y, telling the model to correct the answer. You can see a screenshot of the raw dataset below.

The function z_train, z_test,y_train, y_test = train_test_split(z,y,test_size = 0.2) divides columns z and y into z_train for training inputs, y_train for training labels, z_test for testing inputs, and y_test for testing labels.

test_size=0.2 sets the testing set to 20 percent of z and y. You can see an example of this in the screenshot below, where the ham label indicates non-spam emails, and spam represents known spam emails:

For example, in the image above, the word corresponding to 1841 is used twice in email number 0.

Now, our machine learning model will be able to predict spam emails based on the number of occurrences of certain words that are common in spam emails.

Building the model

SVM, the support vector machine algorithm, is a linear model for classification and regression. The idea of SVM is simple, the algorithm creates a line, or a hyperplane, which separates the data into classes. SVM can solve both linear and non-linear problems

Let's create an SVM model with the code below:

```
model = svm.SVC()
model.fit(features,y_train)
```

model = svm.SVC() assigns svm.SVC()

In the model.fit(features,y_train) function, model.fit trains the model with features and y_train. Then, it checks the prediction against the y_train label and adjusts its parameters until it reaches the highest possible accuracy.

Testing our email spam detector

Now, to ensure accuracy, let's test our application. Run the code below:

```
features_test = cv.transform(z_test)
print("Accuracy:{}".format(model.score(features_test,y_test)))
```

The features_test = cv.transform(z_test) function makes predictions from z_test that will go through count vectorization. It saves the results to the features_test file.

print(model.score(features_test,y_test)) function, mode.score() scores the prediction of features_test against the actual labels in y_test.

The full script for this project is below:

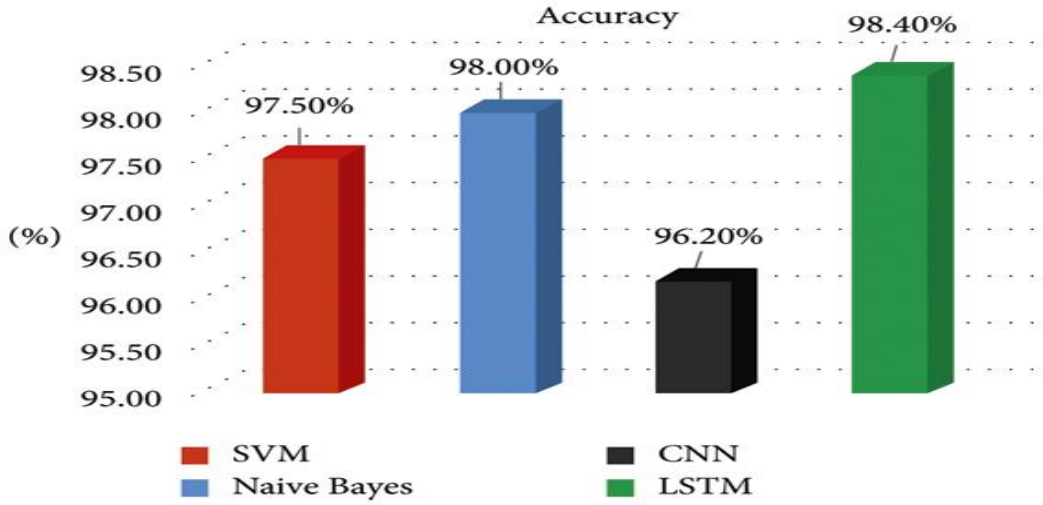
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn import svm

spam=pd.read_csv('C:\\Users\\nethm\\Downloads\\spam.csv')
z = spam['EmailText']
y = spam["Label"]
z_train, z_test,y_train, y_test = train_test_split(z,y,test_size = 0.2)
cv = CountVectorizer()
features = cv.fit_transform(z_train)

model = svm.SVC()
model.fit(features,y_train)

features_test = cv.transform(z_test)
print(model.score(features_test,y_test))
```

Implementation and Results:

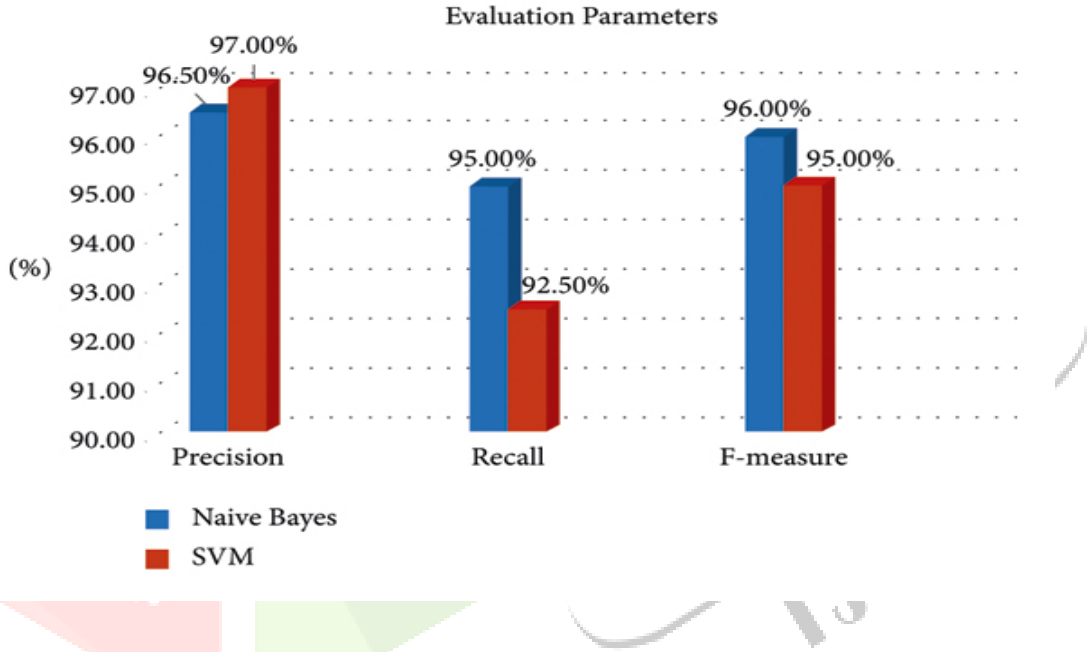
In this section, the programming language, libraries, implementation platform along with the data modeling and the observations and results obtained from it are discussed.

Implementation Platform and Language :

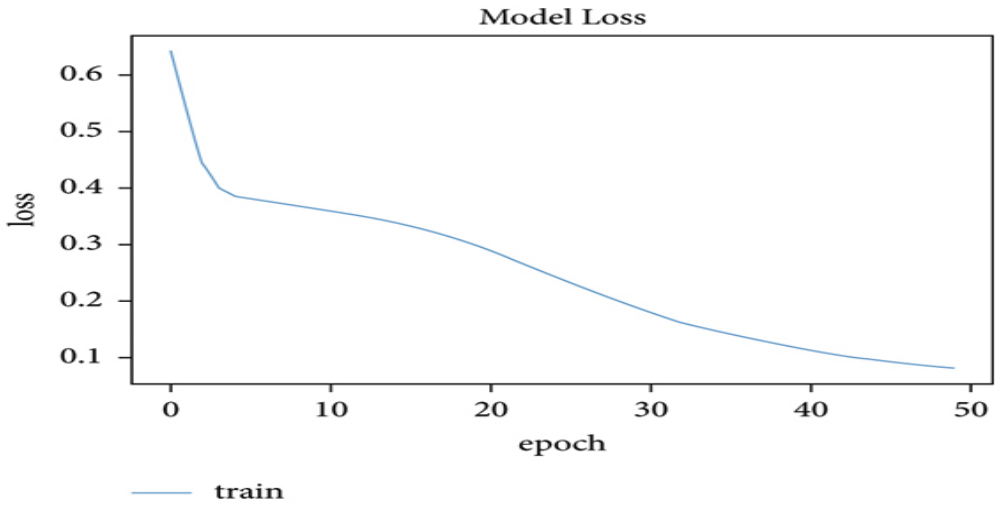
ML models (i.e., SVM and Naive Bayes) are used to calculate evaluation parameters such as precision, recall, and f-measures, which are described in Table 5. In terms of recall and f-measures, we found that Naive Bayes is more successful and produces better results; however, SVM produces the highest precision percentage when compared to Naive Bayes. The results of the comparative analysis provided in Table 5 show that Naive Bayes achieves better results in terms of recall and f-measure, while SVM achieves better results with respect to precision.

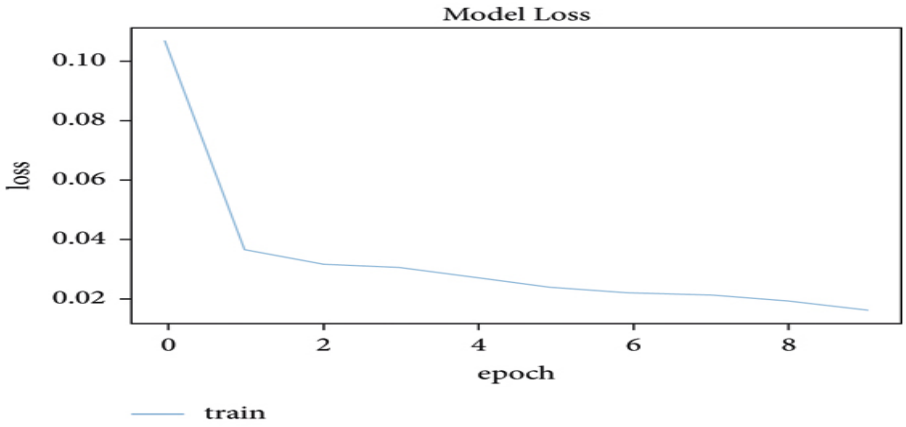


In the mentioned table we have compared the accuracy of four different ML and DL models. We can see that the DL model (LSTM) is the most accurate among all the models, but it takes a long time to train. ML models like SVM and Naive Bayes are around the same accuracy percentage lower than LSTM/CNN, which is also a DL model and has the lowest accuracy percentage. Below figure shows accuracy comparison of ML and DL models.



The following graphs (Figures 15 and 16) demonstrate the model loss for CNN and LSTM for each epoch. The graph line is obviously decreasing as the epochs increase, as can be seen. When the number of epochs is increased, the model loss rate decreases.





This depicts that DL models are particularly good at detecting and classifying Urdu spam emails, as they produce more accurate and precise detection.
In this study, we used existing models for detection of Urdu spam emails, and more training and better detection were also explained for SVM, Naive Bayes, CNN, and LSTM. Furthermore, the accuracy of each model was calculated, and evaluation measures such as precision, recall, and f-measure for SVM and Naive Bayes and for CNN and LSTM as well as the measures ROC-AUC and model loss were used for comparative evaluation. According to the findings, the LSTM model obtained higher accuracy than the other models with a score of 98.4%.

Conclusion and Future Scope:

From this project, it can be concluded that Microsoft Azure Machine Learning Studio is a cloud collaborative tool which has capabilities to predict analytics solutions on particular data. This research has leveraged the Azure Machine learning by modifying Vowpal Wabbit algorithm in order to detect spam. The classification model and score weights based on words used will determine the spam.

The performance of a classification technique is affected by the quality of data source. Irrelevant and redundant features of data not only increase the elapsed time, but also may reduce the accuracy of detection. Each algorithm has its own advantages and disadvantages. As stated before, supervised ML is able to separate messages and classify the correct categories efficiently. It is also able to score the model and weigh them successfully. For instance, Gmail's interface is using the algorithm based on a machine learning program to keep their users' inbox free of spam messages. During the implementation, only text (messages) can be classified and scored instead of domain name and email address. This project only focuses on filtering, analysing and classifying messages and does not block them. Hence, the proposed methodology may be adopted to overcome the flaws of the existing spam detection.

With the increased usage of emails, this study focuses on using automated ways to detect spam emails written in Urdu. The study uses various machine learning and deep learning algorithms to detect them. In the study, a translated emails dataset including spam and ham emails is generated from Kaggle, which is preprocessed for various approaches. Accuracy, precision, recall, F-measure, ROC-AUC, and model loss are used as comparative measures to examine performance. The study concludes that deep learning models are more successful in classifying Urdu spam emails. Comparatively, the LSTM algorithm has a high accuracy rate of around 98% with a low model loss rate of 5%. Even though LSTM takes a little longer to train than CNN, SVM, or Naive Bayes, its efficiency and accuracy rate are far better than those of the other approaches. The creation of an actual dataset of Urdu emails can be considered as a viable future task. In addition, more recent artificial intelligent approaches may also be considered to detect spams.

References:

1.N. Kumar, S. Sonowal, and Nishant, "Email spam detection using machine learning algorithms," in *Proceedings of the 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 108–113, IEEE, Coimbatore, India, July 2020.

2. G. Jain, M. Sharma, and B. Agarwal, "Optimizing semantic lstm for spam detection," *International Journal of Information Technology*, vol. 11, no. 2, pp. 239–250, 2019.

3. F. Masood, G. Ammad, A. Almogren et al., "Spammer detection and fake user identification on social networks," *IEEE Access*, vol. 7, pp. 68140–68152, 2019.

4. A. Akhtar, G. R. Tahir, and K. Shakeel, "A mechanism to detect Urdu spam emails," in *Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp. 168–172, IEEE, New York, NY, USA, Oct 2017.

5. H. Drucker, D. Donghui Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1048–1054, 1999.

6. H. Afzal and K. Mehmood, "Spam filtering of bi-lingual tweets using machine learning," in *Proceedings of the 2016 18th International Conference on Advanced Communication Technology (ICACT)*, pp. 710–714, IEEE, PyeongChang, Korea (South), Feb 2016.

7. S. K. Tuteja and N. Bogiri, "Email spam filtering using bpnn classification algorithm," in *Proceedings of the 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, pp. 915–919, IEEE, Pune, India, Sep 2016.

8. M. Mohamad and A. Selamat, "An evaluation on the efficiency of hybrid feature selection in spam email classification," in *Proceedings of the 2015 International Conference on Computer, Communications, and Control Technology (I4CT)*, pp. 227–231, IEEE, Kuching, Malaysia, Apr 2015.