MACHINE LEARNING BASED SPAM DETECTION SYSTEM

Saurabh Masurkar[1], Arjunsingh Rajput[2], Anish Angane[3], Simran Madaan[4], Shaveta Malik[5]

[1-4]Student, Department of Computer Engineering, TEC, University of Mumbai, Mumbai, India
[5]Faculty, Department of Computer Engineering, TEC, University of Mumbai, Mumbai, India

Abstract - *Due to its convenient, economical, fast, and easy to use nature Electronic mail is a vital revolution taking place over traditional communication systems. A main obstruction in electronic communications is the vast publicizing of unwanted, harmful emails known as spam emails. Lots of time of client is being wasted for sorting approaching mail and erasing undesirable correspondence, so there is a need for spam detection so that its outcomes can be reduced. The main aim is to development of suitable filters that can appropriately detect those emails and results in a high-performance rate.*

In this project, Spam Detection aims to differentiate between spam and authorized mail messages. Here, the evaluation of it is done by using a Machine Learning algorithm named SVM. Machine learning algorithms, especially Support Vector Machine (SVM), can play a major role in spam detection. In this project, the classification is done by defining feature vectors calculated by TF-IDF values.

Keywords — Electronic emails, Email spam, Spam Detection, Machine Learning, TF-IDF(Term Frequency – Inverse Document Frequency), SVM(Support Vector Machine), Classification.

1. INTRODUCTION

To communicate with each other the Internet has become an indispensable method, because of its popularization, low cost, and fast delivery of messages. In recent years, there has been a dramatic growth in spam along with the growth of the Internet and email. Spam can arise from any location across the globe where internet access is available. Spamming is the misuse of electronic messaging systems to send voluntary bulk messages or to promote products or services, that are almost universally undesired. The Spam problem is currently of serious and surge concern, and it is challenging to develop spam filters that can effectively eliminate the increasing volumes of unwanted emails automatically before they enter a user's mailbox. If people have to spend time and effort on identification E-mail every day it is evident that their work efficiency and their emotions will be influenced.

Automatic distinguishing of spam has important meaning and applying value. Automated email filtering using Machine Learning (ML) is one popular solution. One of the most used techniques as the base classifier to overcome the spam problem is the Support Vector Machine. SVM Classify spam, that is to differentiate between spam and authorized mail message. Hence application of this SVM classifier with SVC model is studied in this paper.

First Dataset containing 5725 email samples containing both spam and ham(non-spam) type emails is used. Where 70% mails are used for training and remaining 30% are used for testing. Later vocabulary is built, which contains a set of most frequently words chosen from the training/testing set. Vocabulary is then used to calculate the TF-IDF values, where each email will be represented on the basis of the importance of word in the entire dataset which is a n-dimensional vector. This vector is feature vector. A Machine learning algorithm, Support Vector Machine(SVM) is trained to classify the given mails. Each of these mails belongs to only one of two classes. The idea of SVM classification is find a linear separation boundary that correctly classifies training samples. And later this model is used to predict new mails given, which is the main aim or the system.

The paper is organized as follows. In section II, we give a brief summary of various existing work done. In section III, a detailed description of the SVM model is given along with all the major terminologies. In section IV, a system has been proposed which will classify the new mails into spam and non-spam using the SVM classifier. Each and every step involved in the system has been briefly explained in the paper.

2. LITERATURE REVIEW

- Ni Zhang et al. [1] developed a method for filtering spam emails from the Internet service providers in its heavy traffic. They applied their method to email traffic data captured at one of the largest commercial Internet service providers in China. They achieved a reduction of junk mail traffic by 70.4%.
- Youn Seongwook et al. [2] proposed a comparative study for email classification. Neural Network, SVM, Naive Bayesian and J48 classifiers are used to filter spam from the datasets of emails. A neural network consists of data preprocessing, data training and testing.
- Y. Zhang et al. [3] presented a statistical framework which generalizes the bag-of-words representation and aim to supply a theoretical understanding for vector quantization and its effect on object categorization from the point of view of statistical consistency.

- Laorden et al. [4] presented an in depth revision of the usefulness of anomaly discovery used for Email spam filtering that decreases the need of classifying email spam messages and only works with the representation of single class of emails.
- Sanz, Hidalgo, and Pérez [5] detailed the research issues associated with email spams, in what way it affects users, and by what means users and providers can reduce it effects. The paper also enumerates the legal, economic, and technical measures count to mediate the e-mail spams.
- Graham [6] on the other hand, gives a comprehensive analysis of the similarities and differences between traditional techniques of message filtering that were used at the time and the machine learning technique at the turn of the millennium.
- Friedman et. al. [7] proposed the TAN (tree augmented naive Bayes) classifier that relaxes the independence assumption in the Naive Bayesian algorithm. In the model, a node in the tree can rely on no more than one non-class node. TAN achieve a great trade-off between the classification speed and classification accuracy.
- Scholkopf et. al. [8] proposed a method to construct new vectors, and thus reduce the computational complexity of support vector decision functions. Although the simplification greatly accelerates an SVM's classification speed, the SVM's precision and recall rate are significantly reduced, due to the difference between the new vectors and the original vectors.
- M. Basavaraju et al [9] proposed the text based clustering method for spam detection. Preprocessing of data, methodology of classification, vector space model, and data reduction are the methodologies used for spam filtering. The Porter stemming and lemmatization algorithm are used for preprocessing of data. Hierarchical and partition clustering algorithms are used for partitioning and clustering.
- Ali Cıltık et al [10] proposed a way of spam e-mail filtering methods with high accuracies and low time complexities. They took Turkish mails for his or her research. They used PC-KIMMO system, a morphological analyzer to extract root sorts of words as input and produce parse of words as output. This method is predicated on the n-gram approach and heuristics.

3. SVM (SUPPORT VECTOR MACHINE)

A discriminative classifier formally defined by a separating hyperplane is a Support Vector Machine (SVM). In other words, if labeled data is given for training i.e. supervised learning, the output of the algorithm is an optimal hyperplane that categorizes new examples. In two-dimensional space this hyperplane is a line dividing a plane into two parts wherein each class lay on either side.

To find a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points is the objective of the support vector machine algorithm. The below figure 3.1 gives an idea about the support vector machine (SVM) theory.

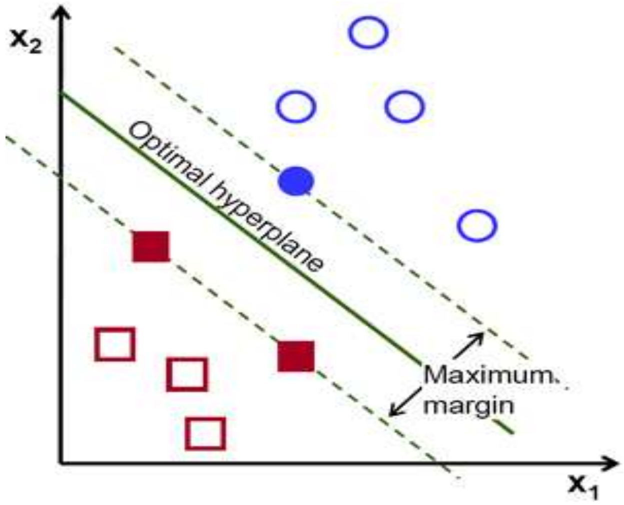


Fig - 3.1: Support Vector Machine [11]

There are many possible hyperplanes that could be chosen, to separate the two classes of data points. To find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes is our objective. Reinforcement can be provided by maximizing the margin distance so that future data points can be classified with more confidence.

3.1 Hyperplanes

Decision boundaries that help to classify the data points are called as Hyperplanes. Data points can be attributed to different classes if they are falling on either side of the hyperplane. Also, the hyperplane dimension depends upon the number of features.

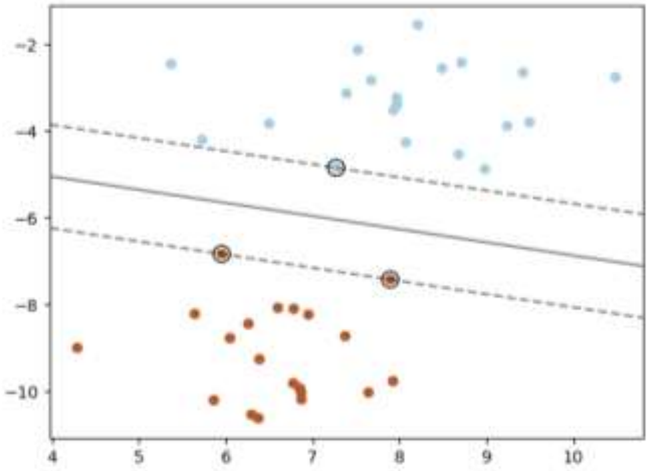


Fig -3.2: Hyperplanes [11]

The hyperplane is just a line if the number of input features is 2. The hyperplane becomes a two-dimensional plane if the number of input features is 3. The above figure 3.2 gives an idea about the theory of hyperplanes.

3.2 Support Vectors

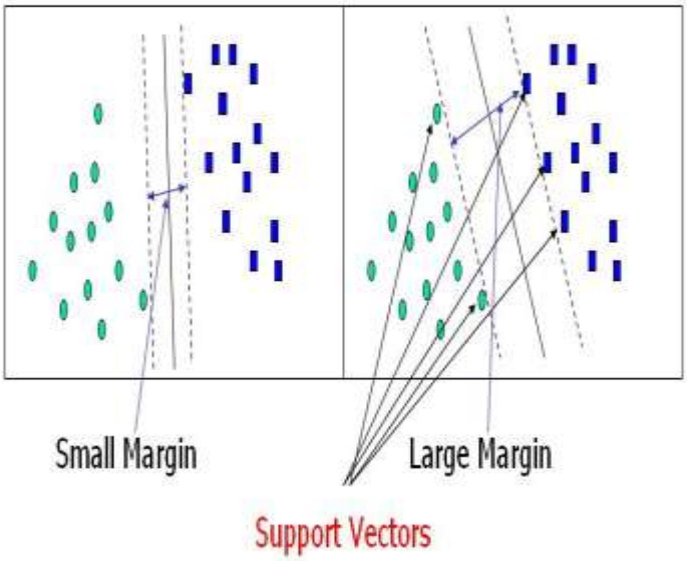


Fig -3.3: Support Vectors [11]

The Data points that are closer to the hyperplane and influence the position and orientation of the hyperplane are called Support vectors. These support vectors can be visualized in above figure 3.3. We maximize the margin of the classifier, using these support vectors. The position of the hyperplane will be changed by deleting the support vectors. These are the point that helps to build the SVM.

4. METHODOLOGY

The process followed for the implementation of the project is given below in the form of flowchart. Also, the details of document processing are mentioned below.

Flowchart

The following figure 4.1 gives an idea about the flowchart representation of this project. It details about all the steps followed while building the classification filter for the mails. Right from the preprocessing stage to actually classifying the emails, all the steps are explained in detail. There are multiple stages as shown in the figure.

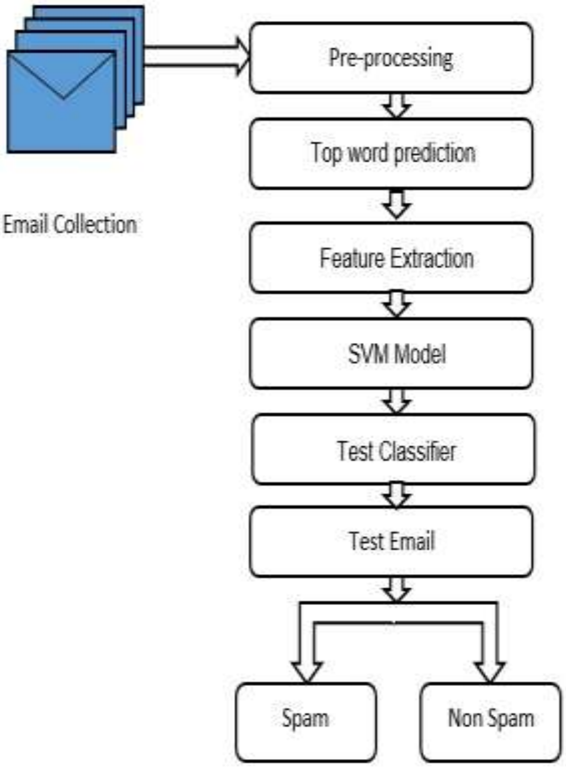


Fig -4.1: Block Diagram of Methodology

1. Pre-processing

The pre-processing step is used to remove the noises from the email which are irrelevant and need not be present. The pre-processing step includes:

- Removal of Numbers
- Removal of Special Symbol
- Removal of URLS
- Stripping HTML
- Word Stemming

2. Feature Extraction

To extract the important and relevant features from the email body Feature Extraction is used. The feature transforms the email into a 2D vector space having features number. These features are mapped from the vocabulary list. Feature vector of an email can be described as follows:

$$x = [0.123\ \ 0.523\ \ 0.428\ \ 0.902 \ldots\ldots\ldots 0.014\ \ 0.890]$$

Feature vector is defined by calculating TF-IDF values. TF-IDF stands for Term Frequency – Inverse Document Frequency. It is calculated by using the following formula.

$$\text{tf-idf}(t, d) = (n(t, d) / n(d)) * \log (N / n(t))$$

Where,

n(t, d) = Number of times the word (term) t appears in the email (document) d.

n(d) = Total Number of words in the email.

N = Total Number of emails (documents) in the training set.

n(t) = Total Number of emails those contain the word t.

3. SVM Model

Support Vector Machine is used for classification and also for regression problems where the datasets are used to train the SVM to classify any new data that it receives. It is a supervised machine learning algorithm that works by finding a hyperplane that classifies the dataset into different classes. The SVM maximizes the distance between different classes because of the existence of many linear hyperplanes which is called as margin maximization.

SVM has verified to be one in all the foremost economical kernel strategies. The success of SVM is principally owing to its high generalization ability. the employment of positive definite kernel within the SVM may be taken as associate degree embedding of the input area into a high dimensional feature area wherever the classification is meted out while not exploitation expressly this feature area The email spams are used for training purposes. The training dataset contains spam content and classifier are trained using it. After training, the classifier is ready to classify the spam emails.

4. Test Classifier

To test the accuracy of the classifier, the classifier is tested with numerous training data. Testing data is the set of samples from the dataset which are not used for training. Here 30% data is used for testing purpose. This data is selected randomly from the dataset. Up to 94% accuracy in classifying emails is achieved by the proposed solution.

5. Test Email

After the training phase is completed, a new sample email is given as input to the classifier to classify the email. The classifier produces output in the forms of 0 or 1, 1 means it is spam and 0 means it is not a spam.

4.1 Documentation Processing:

4.1.1 Tokenization:

Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. The list of tokens becomes an input for further processing like parsing or text mining.

Typically, tokenization occurs at the word level. However, it's sometimes difficult to define what's meant by a "word". A tokenizer relies on simple heuristics, for example:

- All contiguous strings of alphabetic characters are a part of one token; like-wise with numbers.
- Tokens are separated by whitespace characters, like an area or line break, or by punctuation characters.
- Punctuation and whitespace may or might not be included within the resulting list of tokens.

In languages like English (and most programming languages) where words are delimited by whitespace, this approach is simple. However, tokenization is harder for languages such as Chinese which haven't any word boundaries. Simple whitespace-delimited tokenization also presents difficulties when word collocations like NY should be treated together token. By developing more complex heuristics, querying a table of common collocations, or fitting the tokens to a language model that identifies collocations during a later processing step are some ways to address this problem.

For example:
Input: “Practice make man perfect”

- Output: Tokens
- Practice
- make
- man
- perfect

An instance of a sequence of characters is called as Token.

4.1.2 Stemming:

Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or the roots of words known as a lemma.

When a new word is found, it can present new research opportunities. Often, the best results can be attained by using the basic morphological form of the word: the lemma. To find the lemma, stemming is performed by an individual or an algorithm, which may be used by an AI system. Stemming uses several approaches to reduce a word to its base from whatever inflected form is encountered.

To develop a stemming algorithm, lemma can be used. Some simple algorithms will simply strip recognized prefixes and suffixes. However, these simple algorithms are prone to error. For example, an error can reduce words like laziness instead of lazy. Such algorithms may also have difficulty with terms whose inflectional forms don't perfectly mirror the lemma such as with saw and see.

4.1.3 Lemmatization:

Lemmatization in linguistics is that this process of grouping together the various inflected sorts of a word so they are often analyzed as one item.

The algorithmic process of determining the lemma for a given word is Lemmatization, in Linguistics. Since the method may involve complex tasks like understanding context and determining the part of speech of a word during a sentence (requiring, for instance, knowledge of the grammar of a language) it is often a tough task to implement a lemmatizer for a replacement language.

Lemmatization is also used in natural language processing and many other fields that deal with linguistics in general. It also provides a productive thanks to generate generic keywords for search engines or labels for concept maps. Lemmatization is closely associated with stemming. The difference is that a stemmer operates on one word without knowledge of the context, and thus cannot discriminate between words which have different meanings counting on a part of speech. However, stemmers are typically easier to implement and run faster, and therefore the reduced accuracy might not matter for a few applications.

For instance:
1. The word "worse" has "bad" as its lemma. This link is missed by stemming, because it requires a dictionary look-up.
2. The word "walk" is that the base form for word "walking", and hence this is often matched in both stemming and lemmatization.
3.The word "meeting" are often either the bottom sort of a noun or a sort of a verb ("to meet") counting on the context, e.g., "Looking forward to see you tomorrow" or "with reference to our last meeting". Unlike stemming, lemmatization can in theory select the acceptable lemma counting on the context.

4.1.4 Removal of Stop Word:

Sometimes, the extremely common word which might appear to be of little or no value in helping select documents matching user need are deleted from the dictionary. These words are called stop words and therefore the technique is named stop removal.

The general strategy for determining a stop list is to sort the terms by collection frequency then to make the foremost frequently terms, as a stop list, the members of which are discarded during indexing.

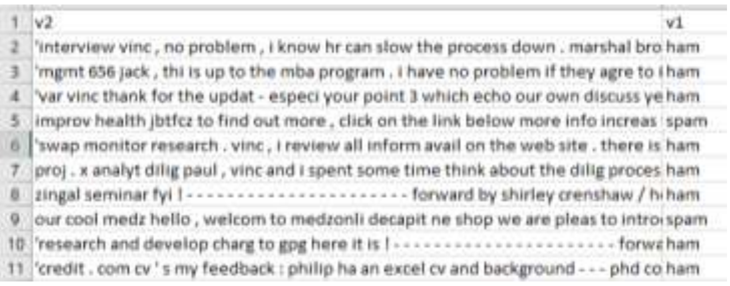
Some of the samples of stop-word are: a, an, the, and, are, as, at, be, for, from, has, he, in, is, it, its, of, that, the, to, on, were, was, will, with, etc.

5. DATA ANALYSIS

Dataset of 5725 emails is taken for this project.
It is a csv file with 2 columns viz ‘emails’ and ‘label’.
Labels are the class of the email i.e. class is either ‘spam’ or ‘ham’. ‘Spam’ label is given to emails which are spam and ‘ham’ is given to the mails which are not spam.
The emails column consists of the content of the email.

Email description-
- Only content of the email is written.
- The subject of the email is not included.
- Image or any other media files are not included.
- Sender and recipients email address is also excluded.

Dataset Used:



1	v2	v1
2	'interview vinc , no problem , i know hr can slow the process down . marshal bro	ham
3	'mgmt 656 jack , thi is up to the mba program . i have no problem if they agre to i	ham
4	'var vinc thank for the updat - especi your point 3 which echo our own discuss ye	ham
5	improv health jbtfcz to find out more , click on the link below more info increas	spam
6	'swap monitor research . vinc , i review all inform avail on the web site . there is	ham
7	proj . x analyt dilig paul , vinc and i spent some time think about the dilig proces	ham
8	zingal seminar fyi ! - forward by shirley crenshaw / hi	ham
9	our cool medz hello , welcom to medzonli decapit ne shop we are pleas to intro	spam
10	'research and develop charg to gpg here it is ! - forwa	ham
11	'credit . com cv ' s my feedback : philip ha an excel cv and background - - - phd co	ham

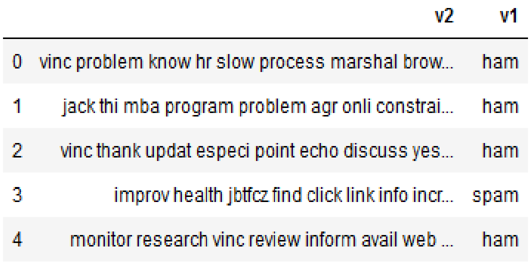
Fig -5.1: Dataset used

The above figure 5.1 represents the dataset used for the data analysis of the project and it represents the label of ‘spam’ or ‘ham’ for a particular email.

6. RESULTS

1. Dataset after Preprocessing:

Dataset after having performed preprocessing steps such as:
1) Removal of numbers
2) Removal of html tags
3) Removal of punctuation marks
4) Removal of stop words
5) Stemming and Lemmatization
6) Converting all data to lower case
is shown in figure 6.1 below.



	v2	v1
0	vinc problem know hr slow process marshal brow...	ham
1	jack thi mba program problem agr onli constrai...	ham
2	vinc thank updat especi point echo discuss yes...	ham
3	improv health jbtfcz find click link info incr...	spam
4	monitor research vinc review inform avail web ...	ham

Fig -6.1: Preprocessed Dataset

The above figure 6.1 represents the dataframe of the dataset, which contains 2 columns. First column v2 contain email texts after being preprocessed whereas the second column represents the respective label.

2. Visualizing the Data:

Studying the dataset and visualizing it as per the requirement.

a. Spam vs Ham Value Count

Total number of spam emails and total number of non-spam or ham emails in the entire dataset are shown in the bar graph Figure 6.2(a) below.

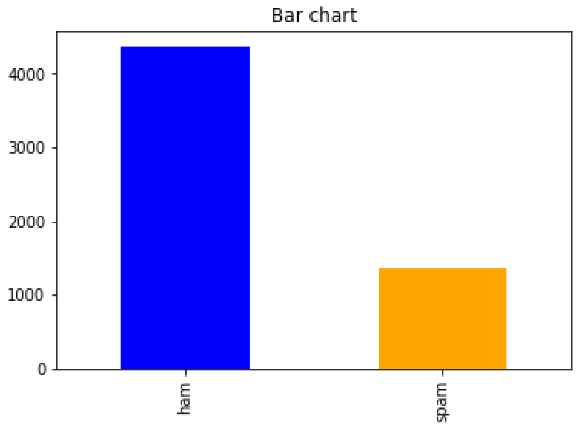


Figure 6.2(a): Spam vs Ham Value Count

According to the above bar chart shown in figure 6.2(a) conveys that around 4250 emails are ham(non-spam) emails and around 1500 emails are spam emails in the entire dataset.

b. Most frequent words in spam mails:

Counting frequency of each word and displaying 25 most frequent words appeared in spam emails in bar graph format in below shown figure 6.2(b).

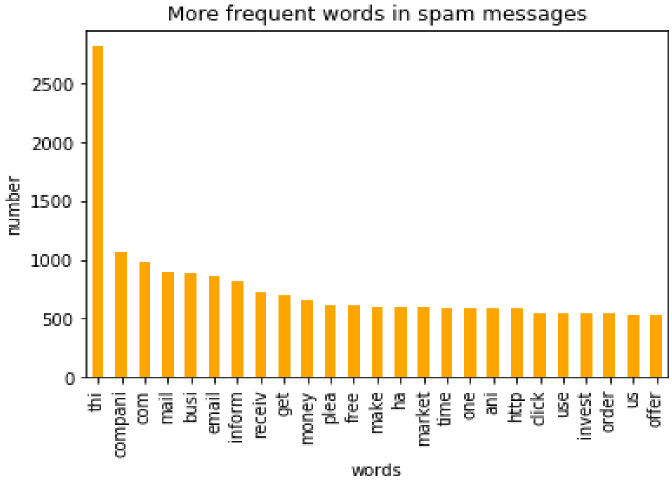


Fig -6.2(b): Most frequent words in spam mails

From the above bar graph, the count of the words which are mostly appeared in the spam emails is shown.

c. Most frequent words in non-spam (ham) mails

Counting frequency of each word and displaying 25 most frequent words appeared in non-spam emails in bar graph format in below figure 6.2(c).

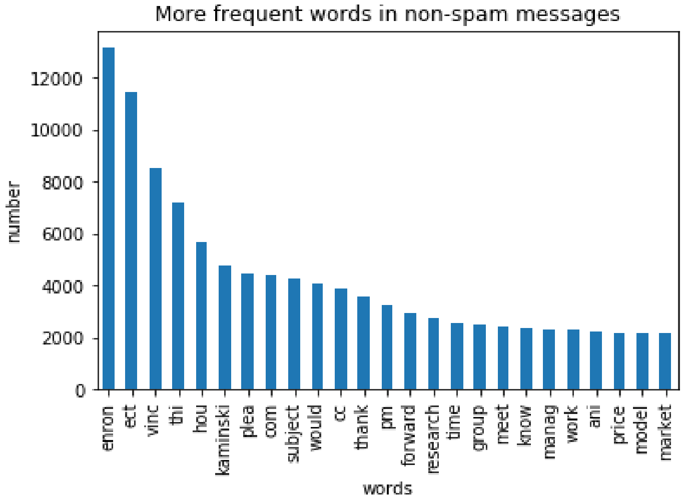
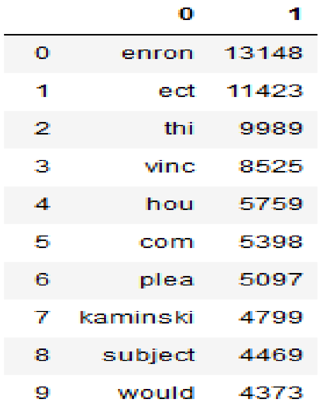


Fig -6.2(c): Most frequent words in ham mails

From the above bar graph, the count of the words which are mostly appeared in the non-spam emails is shown.

3. Built Vocabulary

Vocabulary built on the frequency of most appeared words in the dataset is shown in figure 6.3



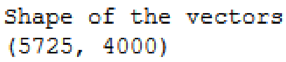
	0	1
0	enron	13148
1	ect	11423
2	thi	9989
3	vinc	8525
4	hou	5759
5	com	5398
6	plea	5097
7	kaminski	4799
8	subject	4469
9	would	4373

Fig -6.3: Vocabulary

The above table shown in the figure 6.3 conveys that the words mentioned in that table are arranged in the descending order. These are the important words by which feature vectors are built.

4. Shape of the feature vector:

After calculating the feature vector for each email, the total shape of all the feature vectors is shown below in figure 6.4.



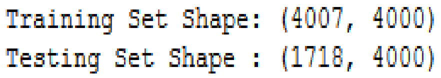
```
Shape of the vectors
(5725, 4000)
```

Fig -6.4: Shape of Vector

The above matrix represents that each email is being converted into an array which is of length 4000 and there are such 5725 emails in this project.

5. Splitting data in train and test set:

After cross validating the vectors in 7:3 ratio. The resultant shape of each training and testing set is shown below.



```
Training Set Shape: (4007, 4000)
Testing Set Shape : (1718, 4000)
```

Fig -6.5: Shape of Train Test Split

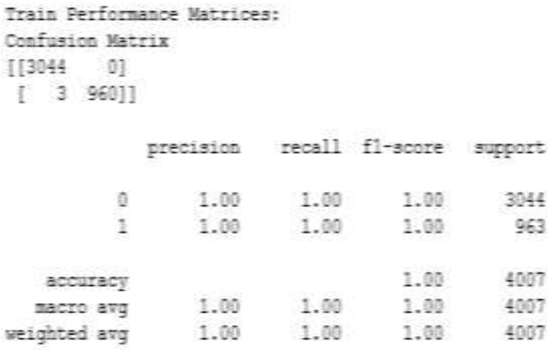
The above figure 6.5 gives the information that the dataset is been split into two sets, training and testing. Where training set consists of 4007 samples of data and testing set consists of 1718 samples of data.

6. Performance Evaluation:

After training the model on 70% data, testing is performed and accuracy and other evaluation matrices of the trained model is calculated.

a. Training set

Trained model is tested on the same training data and its accuracy is calculated. Performance matrices for training set is as shown below.



```
Train Performance Matrices:
Confusion Matrix
[[3044    0]
 [   3  960]]

              precision    recall  f1-score   support

           0       1.00      1.00      1.00      3044
           1       1.00      1.00      1.00       963

    accuracy                           1.00      4007
   macro avg       1.00      1.00      1.00      4007
weighted avg       1.00      1.00      1.00      4007
```

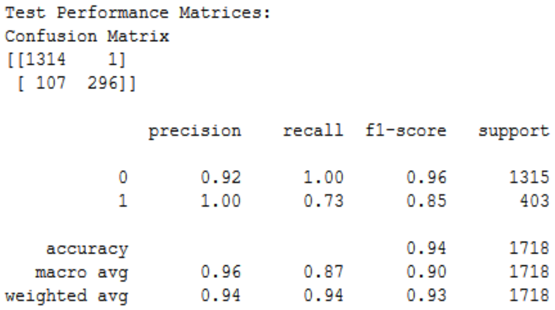
Fig -6.6(a): Performance Evaluation of Train Set

The confusion matrix shown in above figure 6.6(a) says that 3044 emails are True Positive, 960 emails are True Negative, whereas no emails are False Positive and 3 emails are False Negative out of total 4007 samples in training set.

Accuracy is observed to be 100% and other performance evaluation scores are displayed.

b. Testing set

Trained model is tested on the test dataset i.e. the subset of dataset which is not used for training. Its performance matrices are as shown in below figure 6.6(b)



```
Test Performance Matrices:
Confusion Matrix
[[1314    1]
 [ 107  296]]

              precision    recall  f1-score   support

           0       0.92      1.00      0.96      1315
           1       1.00      0.73      0.85       403

    accuracy                           0.94      1718
   macro avg       0.96      0.87      0.90      1718
weighted avg       0.94      0.94      0.93      1718
```

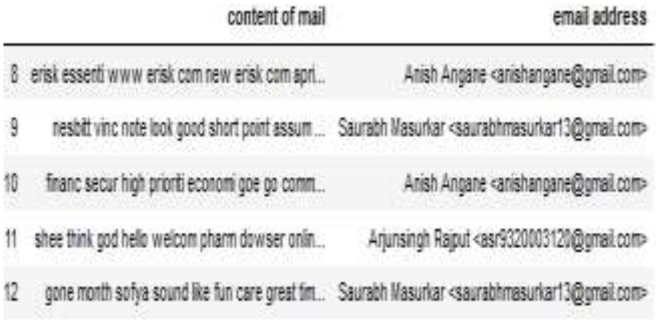
Fig -6.6(b): Performance Evaluation of Test Set

The confusion matrix shown in above figure 6.6(b) says that 1314 emails are True Positive, 296 emails are True Negative, whereas 1 email is False Positive and 107 emails are False Negative out of total 1718 samples in testing set.

Accuracy is observed to be 94% and other performance evaluation scores are displayed.

7. New Mails for Prediction:

Emails extracted from the g-mail account which are to be given to the classifier for prediction is as shown in below figure 6.7



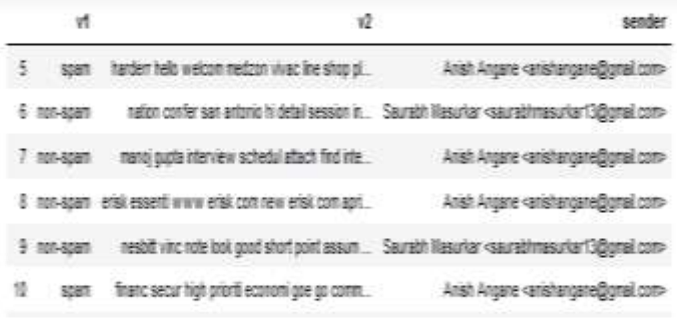
	content of mail	email address
8	erisk essenti www erisk com new erisk com apri...	Anish Angane <arishangane@gmail.com>
9	nesbitt vinc note look good short point assum...	Saurabh Masurkar <saurabhmasurkar13@gmail.com>
10	financ secur high prioriti economi goe go comm...	Anish Angane <arishangane@gmail.com>
11	shee think god hello welcom pharm dowser onlin...	Arjunsingh Rajput <asr9320003120@gmail.com>
12	gone month sofya sound like fun care great tim...	Saurabh Masurkar <saurabhmasurkar13@gmail.com>

Fig -6.7: Input mails for prediction

Above figure 6.7 shows the dataframe with 2 columns. First column contains content of the mail which is being extracted from the g-mail account and second column contains the email address of the respective sender.

8. Final Output of Prediction:

The final output of the prediction done on the mails given for prediction is as shown below



	v1	v2	sender
5	spam	harden hello welcom medzon vivac line shop pl...	Anish Angane <anishangane@gmail.com>
6	non-spam	nation confer san antonio hi detail session in...	Saurabh Masurkar <saurabhmasurkar13@gmail.com>
7	non-spam	manoj gupta interview schedul attach find inte...	Anish Angane <anishangane@gmail.com>
8	non-spam	erisk essenti www erisk com new erisk com apri...	Anish Angane <anishangane@gmail.com>
9	non-spam	nesbitt vinc note look good short point assum...	Saurabh Masurkar <saurabhmasurkar13@gmail.com>
10	spam	financ secur high prioriti economi gse go comm...	Anish Angane <anishangane@gmail.com>

Fig -6.8: Output of prediction

The above figure 6.8 displays 3 columns. The second column displays the content of the emails and third column consists of email ids of respective sender. And the first column displays the predicted output of the email.

7. CONCLUSIONS

There are many ways to filter spam emails. Considering the daily growth of spam and spammers, it is essential to provide effective mechanisms and to develop efficient software packages to manage spam. Using legitimate emails and spam emails the present study extracted data from emails using machine learning algorithms to develop a new model. In this research one such model named SVM(Support Vector Machine) is studied and the feature vectors are calculated by TF-IDF(Term Frequency – Inverse Document Frequency) values for each mail.

This method has extracted only the features from the content of an email instead of extracting all the features from the mail. The vectors are built by building the vocabulary and mapping the new given word with the built vocabulary. later comparing the new words with the words in the vocabulary, a spam email can be identified. The Feature Vectors formed by using. TF-IDF values provide a better result than a bag of words or frequency count method.

The system achieves an accuracy of 94% for the test dataset. AS compared with other classifiers such as Naïve Bayes and Logistic Regression by using TF-IDF as a feature vector, the SVM classifier used has better accuracy.

Currently, the SVM algorithm is only applicable to text-based spams detection, but modifications can be done to the algorithm and make it suitable for filtering spams with different formats of data e.g., pictures and other types of multimedia files.

REFERENCES

[1] Ni Zhang, Yu Jiang, Binxing Fang, Xueqi Cheng and Li Guo, "Traffic Classification-Based Spam Filter," IEEE International Conference on Communications, vol. 5, p. 2130 –2135, 2006.

[2] Youn, Seongwook, and Dennis McLeod, "A Comparative Study for Email Classification," Editor. Khaled Elleithy, Advances and Innovations in Systems, Computing Sciences and Software Engineering, pp. 387-391, 2007.

[3] Y. Zhang, R. Jin, and Z. Zhou, "Understanding bag-of-words model:A statistical framework," Int. J. Mach. Learn. Cybern., vol. 1, p. 43–52, 2010.

[4] C. Laorden, X. Ugarte-Pedrero, I. Santos, B. Sanz, J. Nieves, and P.G. Bringas, "Study on the effectiveness of anomaly detection for spam filtering," Inf. Sci., vol. 277, pp. 421-444, 2014.

[5] E.P. Sanz, J.M.G. Hidalgo, J.C.C. Pérez, "Email spam filtering Adv. Comput," no. 74, pp. 11-45, 2008.

[6] P. Graham, "A Plan for Spam," 2012. [Online]. Available: http://paulgraham.com/spam.html. [Accessed Sept 2019].

[7] N. Friedman, D. Geiger and M. Goldszmidt, "Bayesian network classifiers," Machine learning, vol. 29, no. 2-3, pp. 131-163, 2017.

[8] B. Scholkopf, S. Mika, C. J. Burges, P. Knirsch, K.-R. Muller, G. Ratsch, et al., "Input space versus feature space in kernel-based methods," IEEE Transactions on neural networks, vol. 10, no. 5, pp. 1000-1017, 2009.

[9] M. Basavaraju and Dr. R. Prabhakar, "A Novel Method of Spam Mail Detection using Text Based Clustering Approach," IJCA, pp. 15-25, 2010.

[10] Çıltık, Ali, and Tunga Güngör, "Time-Efficient Spam E-mail Filtering Using n-gram Models," Pattern Recognition Letters, 2008.

[11] Rohith Gandhi, "Support Vector Machine — Introduction to Machine Learning Algorithms," towards data science, june 2018. [Online]. Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47. [Accessed jan 2020].

[12] M. Singh, R. Pamula and S. k. shekhar, "Email Spam Classification by Support Vector Machine," International Conference on Computing, Power and Communication Technologies, pp. 878-882, 2018.

[13] Trivedi, S. K., "A study of machine learning classifiers for spam detection," 4th ISCBI, pp. 176-180, 2016.

[14] S. Nandhini and D. J. Marseline.K.S, "Performance Evaluation of Machine Learning Algorithms for Email Spam Detection," International Conference on Emerging Trends in Information Technology and Engineering, pp. 1-4, 2020.

[15] W. Feng, J. Sun, L. Zhang, C. Cao and Q. Yang, "A support vector machine based naive Bayes algorithm for spam filtering," IEEE 35th International Performance Computing and Communications Conference, pp. 1-8, 2016.

[16] A. Alzahrani and D. B. Rawat, "Comparative Study of Machine Learning Algorithms for SMS Spam Detection," SoutheastCon, Huntsville, pp. 1-6, 2019.

[17] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. A. Abetunmbi, O. E. Ajibuwa, "Machine learning for email spam filtering: review, approaches and open research problems," Heliyon, vol. 5, no. 6.

[18] A. A. Aski, N. K. Sourati, "Proposed efficient algorithm to filter spam using machine learning techniques," Pacific Science Review A: Natural Science and Engineering, vol. 18, no. 2, pp. 145-149, 2016.

[19] Shradhanjali, Verma Toran, "E-Mail Spam Detection and ClassificationUsing SVM and Feature Extraction," International Journal of Advance Research Ideas and Innovations in Technology, vol. 3, no. 3, pp. 1491-1495, 2017.

[20] K sai Prasanthi, T Deepika, S Anudeep, M Sai Koushik, "An Efficient Email Spam Detection using Support Vector Machine," International Journal of Innovative Technology and Exploring Engineering, vol. 9, no. 2, pp. 5258-5262, 2019.