Email Spam Detection

[1]S. Sivapriya, [2]Roshni Musaddi, [3]Anny Jaiswal
[1]Assistant Professor, [2]Student, [3]Student
[1]Department of Computer Science and Engineering,
[1]SRM Institute of Science and Technology, Chennai, India

Abstract : Emails have become an essential mode of technical communication which is widely used across the globe. Along with transportation and communication facilities, they come with various malicious threats like viruses and spams. Cyberpunks can do anything to access the data in these mails and use them for illegal activities. Various legal measures have already been taken to suppress this issue. Many filters are used to distinguish legitimate mails from the spam mails on the basis of text analysis and their capacity to achieve the goal. However, these designs have not been as effective. This paper uses Naïve Bayes Classifier together with Support Vector Machine to detect Spam and ham (non - spam) messages. It shows the comparative study between these two algorithms to conclude which one gives more accurate results.

IndexTerms - **Machine Learning, Spam Email, Text Analysis, Feature Engineering, Naïve Bayes Classification, Support Vector Machine (SVM).**

I. INTRODUCTION

The E-mail or Electronic mail is an important source of communication. It was of limited use in the 1960s but their importance and convenience starting ruling the medium of communication. According to a statistical research (Figure 1), there were approximately 4 billion active email accounts some 4 years back . This amount is expected to go up to around 5.6 billion by 2020. This shows the relevance of emails in today's world and how much people depend on them. These accounts may get some spam messages which may not be of any use to the user.

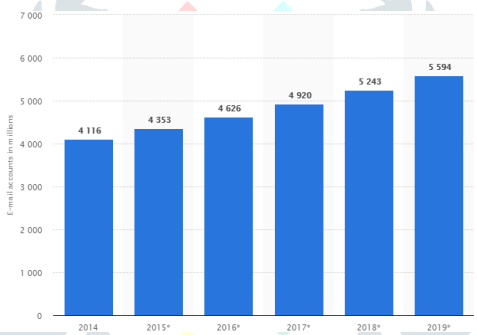

Fig.1: Statistical report of active email accounts

In fact, they can be treacherous and unsafe at times. Spam messages are the unsolicited mail which contribute major amount to the email traffic today. They can be any phishing spams, advanced free frauds, money transfer job offers, etc. The person behind these spam can either be an advertising authority or a traitor who can steal the respective name and id and transfer amount from the users bank accounts to their own. There are various machine learning algorithms that are available to detect and classify spam emails from the legitimate mails such as Random Forest (RF), Support Vector Machine, Naive Bayes, K-Nearest Neighbour, etc. In our paper, two of these algorithms are compared to conclude which one is more efficient on the basis of their accuracies. These two algorithmic design models are Support Vector Machine model and Multinomial Naïve Bayes Classification. Naïve Bayes approach is a probabilistic classification method while SVM is a non-probabilistic machine learning algorithm used for regression analysis. After applying these designs, it was seen that SVM outperforms MNB (Multinomial Naïve Bayes). The main variation between these models is that SVM is a geometric method which considers the interactions between the features whereas MNB takes independent features and works well for snippets than for large documents.

II. LITERATURE SURVEY

Paras Sethi [1] compared various machine learning algorithms like Naïve Bayes, Random Forest Algorithm, Logistic Regression to classify SMS Spam Detection. They took dataset of around 5574 short messages from Kaggle under SMS Spam collection Database. They took two important features like the message length and the count vectorizer matrix under SMS classification and concluded that Naïve Bayes performs better than Logistic Regression (LR) and Random Forest Model (RFM). Naïve Bayes achieved a high accuracy of 98.445% and classified text as either spam or ham.

Akash Iyengar [2] emphasised on integrated approach over traditional approach and have observed that accuracy has increased while taking real world data set. Their model checks for the URLs given in the mail and displays failure message when the filter rejects the mail. Their datasets contained spam and ham mails from Gmail and Yahoo. The pre-processing is done by

removing tag, some predefined words, frequency of words, tokenization. They used Bayesian Classifier for classifying text in data mining.

Linda Huang [3] proposed an algorithm for enhancing the accuracy of Naïve Bayes Spam Filtration. They implemented this algorithm in real – time environments. Their results showed that with a slight addition of coding to the current servers, the accuracy of the algorithm improved from 23.9% to 62% which was an increase of almost 259%. Their algorithm proved to be a valuable addition to the current systems and could save millions. This impacted the customers of E- Mail servers and prevented malware damages.

Shubhi Shrivastava [4] took three benchmark datasets: first dataset from Indiana State University repository containing around 5200 spam and ham emails, second and third dataset from Ling - Spam corpus of around 5000 files. They used Bernoulli's and continuous probability distribution to spam classification and showed that their models works better with Bernoulli's probability distribution. They also concluded that there is no such superior classifier model between Naïve Bayes and Decision Tree classifier, rather the performance of classifier depends on various other factors such as probability distribution, dataset used and the involved problems. By reducing the overfittings, the efficiency can be increased significantly.

Dat Tran [5] proposed possibility theory - based method to detect spam Mails and blacklisted list of keywords to detect similar spam messages. They took 100 Emails with 33 spam Mails having image attachments. Their approach used misspelling of keywords as a feature which they used to calculate possibility score for spam detection. Using the possibility score and Trigram method, they showed around 91% of accuracy.

Naghmed Moradpoor [6] proposed a model based on Neural Network (NN) classification of Phishing Emails. They had a dataset of around 14370 and used 70% of data for training, 15 % for validation and 15 % for testing. They used datasets from "SpamAssassin" and "Phishcorpus". Their model had 10 hidden layers with 5 input features,1 output layer and 1 output feature. They presented an accurate and satisfactory performance with 100% sensitivity and 100 % specificity for detection of spam Emails.

Sunil B. Rathod [7] used Bayesian Classifier to detect Content based spam Mails. They took datasets from Gmail and applied various features to detect the spam texts and the URLs available in the body of mails. They achieved an accuracy of about 96.46%.

Shukor Bin Abd Raza [8] commonalized several features contained in Email header in Hotmail, Gmail and Yahoo Mail so that a generalised spam Email detection mechanism can be formed for all the Email providers.

III. METHODOLOGY

To implement Naïve Bayes classification, Python language has been used which is the most powerful language in use today. Since, Python has its predefined packages, it makes the work a lot easier.

3.1 DATASET

The Dataset (Figure 2) is an assortment of tagged messages that has been assembled for SMS Spam analysis. It contains 5,574 messages along with their tags to define whether they are legitimate (non-spam) or not. The files accommodates one message per line. Each of these line consists of two columns: v1 containing the tag (spam or ham) and v2 containing the message text.

A total of 425 spam texts was manually withdrawn from Grumbletext website. This is actually a UK forum where mobile phone customers make public claims about spam messages. Though most of these messages are never even reported. These messages are identified after scanning thousands of web pages which becomes a very long and time-consuming work. A group of 3,375 ham messages were randomly picked from the NSC (NUS SMS Corpus). This forms a dataset of around 10,000 legitimate texts collected for research and analysis at the Computer Science Department in NUS. The messages mostly emerge from Singaporeans and largely from students of the University. The volunteers were made aware that their message contributions were to be made accessible by the general public. A group of 450 ham messages were taken from Caroline Tag's PhD. It has 1,002 messages as ham and 322 messages as spam.

	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
5	spam	FreeMsg Hey there darling it's been 3 week's n...	NaN	NaN	NaN
6	ham	Even my brother is not like to speak with me. ...	NaN	NaN	NaN
7	ham	As per your request 'Melle Melle (Oru Minnamin...	NaN	NaN	NaN
8	spam	WINNER!! As a valued network customer you have...	NaN	NaN	NaN
9	spam	Had your mobile 11 months or more? U R entitle...	NaN	NaN	NaN

Fig.2: Dataset

3.2 TEXT ANALYZING AND FEATURE ENGINEERING

The dataset was explored to make a distribution of the statistic of spam messages. A pie chart was plotted on the whole dataset and the results were as follows in Figure 3. Only 13% of the messages were spam and all others were considered important. Also to get a comparison analysis, a bar chart was plotted. These 2D figures were plotted with the Python Matplotlib plotting library. Pyplot is a module which is most commonly used as it provides a variety of features to control formatting, axis lines, fonts and

other properties. It is used with NumPy which then becomes an open source substitute for MATLAB. The figure obtained from bar chart plotting is in Figure 4.

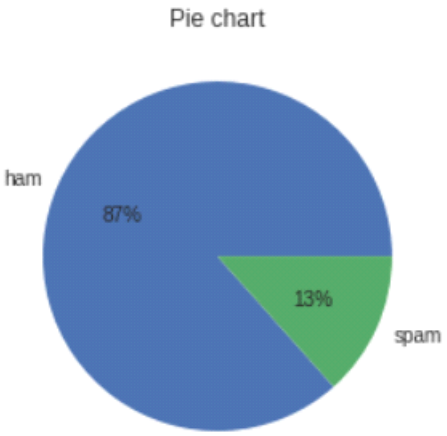

Fig.3: Pie chart for the dataset

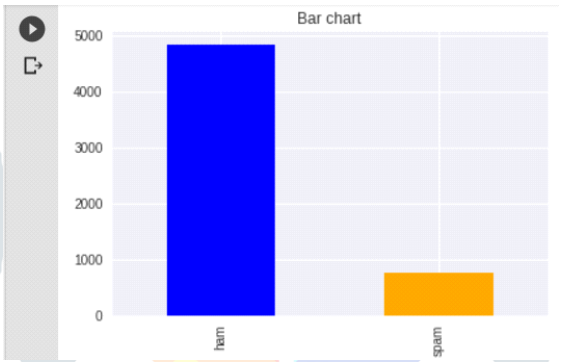

Fig.4: Bar chart for the dataset

Text Analysis is an approach to extract meaningful structured data from the text which can further be mined as trends or patterns. This process of computationally analysing the data was done by parsing the text messages. We have used text analysis to find the recurrence of certain words in both the category of messages. These words will then be model features. To perform this task Counter function was used and the most common words were plotted along with their respective frequencies for spam and non-spam messages individually.

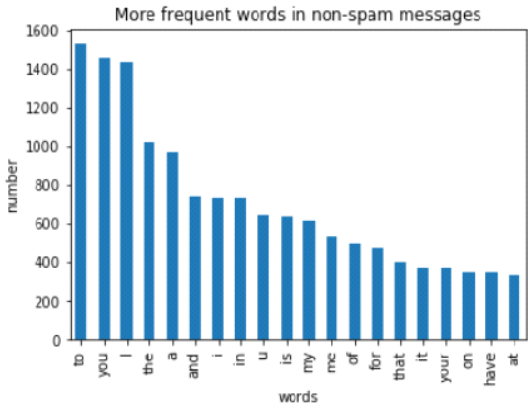

Fig.5: Most frequent words in non-spam messages

The majority of words in both the plots are words such as 'to', 'your', 'a', 'the', etc. These words are known as stop words which has to be ignored because these are present everywhere in a sentence and do not contribute in classifying our messages (Figure 5, Figure 6).

Text processing and feature engineering processes were implemented to extract features in order to execute the algorithm which is our goal. Around 8400 features were created. The j new feature in the i row is equivalent to one if the word w_j occurs in that text sample. If not, it is set to 0.

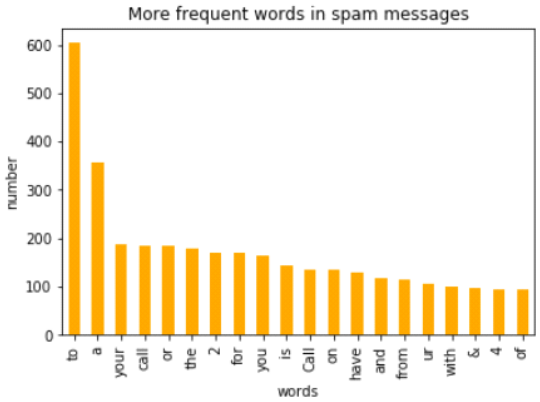

Fig.6: Most frequent words in non-spam messages

3.3 NAIVE BAYES AND SUPPORT VECTOR MACHINE

We will anticipate whether a given message was spam or not. The algorithm proves to be worse if it misclassifies non-spam as spam than when it misclassifies a spam message because it shouldn't have any false positives. The reason being that not everyone usually checks their spam messages and hence, the important messages shouldn't be tagged as being spam.
The two potential situations are:

1. False negatives or new spam messages in our inbox.
   CONCLUSION: We delete it.
2. False positives or new non-spam message in our spam folder
   CONCLUSION: We presumably don't even read it.

The first option is preferable.

The variable spam/non-spam is transformed into binary variables and then separated as training and test set. Spam is considered as 1 and ham as 0. Now, we train different Bayes models by α which is also called the regularization parameter. With the test data, we calculate the accuracy and precision of the first ten models and their metrics. After taking the model with the maximum precision, we can conclude that it does not produce any false positives.

	Predicted 0	Predicted 1
Actual 0	1587	0
Actual 1	56	196

Fig.7: Confusion matrix for Naïve Bayes

The confusion matrix (Figure 7) clearly shows that we did not misclassify any non-spam messages as spam, which was our main aim. However, we did misclassify 56 spam texts as non-spam and they may show in our inbox. We have also implemented Support Vector Machine. It is a supervised design in machine learning chiefly used for classification. Co-ordinates are obtained by plotting each data in a feature dimensional space. It segregates our binary classes (ham and spam) with a hyper-plane. This model is applied with the gaussian kernel. Here also, we train our model with different regularization parameters and then evaluate the accuracy and precision with the test dataset. The confusion matrix (Figure 8) was obtained for our support vector machine model. Though we misclassified 31 spam messages as non-spam, we did not produce any false positives. No non-spam or ham messages is predicted as spam.

	Predicted 0	Predicted 1
Actual 0	1587	0
Actual 1	31	221

Fig.8: Confusion matrix for Naïve Bayes

Support Vector Machine's best model produces an accuracy of 98.3%. It classifies every non-spam message correctly. It classifies 87.7% of spam messages correctly.

SYSTEM ARCHITECTURE

The training dataset is distributed into bar charts and also pie charts which shows the statistical percentage of the tagged messages. Text analysis is executed to take out certain words which are used the most and repeated in almost all the messages. These words are called stop words and are removed as they do not prove to be a differentiating factor for our model. Feature engineering is performed to convert the most used words into features. 8400 features were obtained from these words. The system architecture can be figured out as Figure 9.

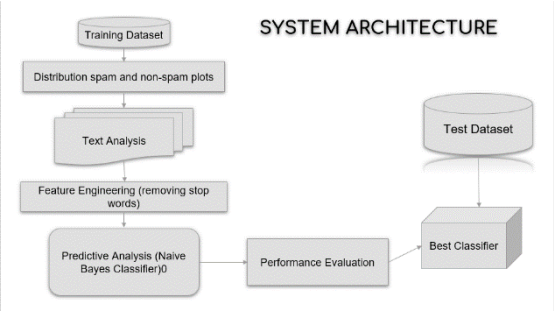

Fig.9: System Architecture Model

Finally, Multinomial Naïve Bayes Classification is implemented on the manipulated data with different regularization parameters to evaluate the best performed model. Support Vector Machine was found to be most efficient in classifying our binary variables as spam or ham.

CONCLUSION

Though there are already many methods in use which classify spam from non-spam texts, they are not very effective. Hence, we compared Naïve Bayes algorithmic model and our Support Vector Machine model to find out which gives the most accurate predictions. The main difference between Support Vector Machine and Multinomial Naïve Bayes is that SVM has dependent features which interact with each other. Contrarily, Naïve Bayes operates best with independent features. Naïve Bayes produced an accuracy of 96% whereas Support Vector Machine gave 98.3% accuracy. Thus, SVM surpasses Multinomial Naïve Bayes algorithm as it can handle non-linearities in the data.

	C	Train Accuracy	Test Accuracy	Test Recall	Test Precision
0	500.0	0.994910	0.982599	0.873016	1.0
1	600.0	0.995982	0.982599	0.873016	1.0
2	700.0	0.996785	0.982599	0.873016	1.0
3	800.0	0.997053	0.983143	0.876984	1.0
4	900.0	0.997589	0.983143	0.876984	1.0

Fig.10: Test accuracy for SVM

Hence, the test accuracy in Figure 10 for Support Vector Machine was found out to be 98.3% and a test precision of 1.0 was achieved which is pretty good.

REFERENCES

[1] Paras Sethi, et al. SMS spam detection and comparison of various machine learning algorithms. (978-1-5386-0627-8/17/$31.00 c 2017 IEEE)

[2] Akash Iyengar, et al. INTEGRATED SPAM DETECTION FOR MULTILINGUAL EMAILS. INTERNATIONAL CONFERENCE ON INFORMATION, COMMUNICATION & EMBEDDED SYSTEMS (ICICES 2017)

[3] Linda Huang, et al. Enhancing the Naive Bayes Spam Filter through Intelligent Text Modification Detection. (2018 17th IEEE International Conference on Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering).

[4] Shubhi Shrivastava, et al. Spam Mail Detection through Data Mining Techniques. (2017 International Conference on Intelligent Communication and Computational Techniques (ICCT) Manipal University Jaipur, Dec 22-23, 2017).

[5] Vishagini V, et al. An Improved Spam Detection Method with Weighted Support Vector Machine. (978-1-5386-4855-1/18/$31.00 ©2018 IEEE)

[6] Naghmeh Moradpoor, et al. Employing Machine Learning Techniques for Detection and Classification of Phishing Emails. (Computing Conference 2017 18-20 July 2017 | London, UK)

[7] Sunil B. Rathod, et al. Content Based Spam Detection in Email using Bayesian Classifier. (978-1-4799-3516-1113/$31.00 ©2013 IEEE)

[8] Shukor Bin Abd Raza, et al. Identification of Spam Email Based on Information from Email Header. (978-1-4799-8081-9/15/$31.00 © 2015 IEEE)
[9] Moradpoor, et al. (2015, September). SQL-IDS: evaluation of SQLi attack detection and classification based on machine learning techniques. (In Proceedings of the 8th International Conference on Security of Information and Networks (pp. 258-266). ACM).
[10] Zhan, et al. "Phishing detection using stochastic learningbased weak estimators". (In Computational Intelligence in Cyber Security (CICS), 2011 IEEE Symposium on, pp. 55-59, 2011).
[11] Khonji, et al. "Enhancing Phishing E-Mail Classifiers: A Lexical URL Analysis Approach,". (International Journal for Information Security Research (IJISR), vol. 2, no.1/2, 2012).
[12] Khonji, M., et al. "Lexical url analysis for discriminating phishing and legitimate websites". (In Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference, CEAS '11, pages 109–115, New York, NY, USA, 2011. ACM).
[13] M. Khonji, et al. "A study of feature subset evaluators and feature subset searching methods for phishing classification". (In Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference, ser. CEAS '11. New York, NY, USA: ACM, 2011, pp. 135–144).
[14] Ammar Almomani, Tat-Chee Wan, et al. "Evolving Fuzzy Neural Network for Phishing Emails Detection". (Journal of Computer Science, vol. 8, no.7, pp. 1099-1107, 2012).
[15] Barraclough, et al. Intelligent phishing detection and protection scheme for online transactions. (Expert Systems with Applications, 40(11), pp.4697-4706. Vancouver).
[16] Martin, A., et al. "A framework for predicting phishing websites using neural networks". (International Journal of Computer Science Issues (IJCSI), 2(8)).
[17] Smadi, S., et al. Detection of phishing emails using data mining algorithms. (In 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA) (pp. 1-8). IEEE. Vancouve).
[18] Barraclough, et al. "Online phishing detection toolbar for transactions." (Science and Information Conference (SAI), 2015. IEEE, 2015).