Email Spam Detection using Machine Learning and Neural Networks

Manoj Sethi[1], Sumesha Chandra[2], Vinayak Chaudhary[3], Yash[4]

[1]Assosiate Professor/Programmer, Department of Computer Science and Engineering, Delhi Technological University, New Delhi, Delhi, India

[2,3,4]Student, Department of Computer Science and Engineering, Delhi Technological University, New Delhi, Delhi, India

---***---

ABSTRACT- *Spam emails are known as unrequested commercialized emails or deceptive emails sent to a specific person or a company [5]. Spams can be detected through natural language processing and machine learning methodologies. Machine learning methods are commonly used in spam filtering. These methods are used to render spam classifying emails to either ham (valid messages) or spam (unwanted messages) with the use of Machine Learning classifiers. The proposed work showcases differentiating features of the content of documents [4]. There has been a lot of work that has been performed in the area of spam filtering which is limited to some domains. Research on spam email detection either focuses on natural language processing methodologies [25] on single machine learning algorithms or one natural language processing technique [22] on multiple machine learning algorithms [2]. In this Project, a modeling pipeline is developed to review the machine learning methodologies.*

Keywords: *Email Spam Detection, Spam Detection, Machine Learning, Neural Networks, Naive Bayes, Support Vector Classifier, Logistic Regression, Spam, Social Media, Email.*

1. INTRODUCTION

Technology has become a vital part of life in today's time. With each passing day, the use of the internet increases exponentially, and with it, the use of email for the purpose of exchanging information and communicating has also increased, it has become second nature to most people. While e-mails are necessary for everyone, they also come with unnecessary, undesirable bulk mails, which are also called Spam Mails [29]. Anyone with access to the internet can receive spam on their devices. Most spam emails divert people's attention away from genuine and important emails and direct them towards detrimental situations. Spam emails are capable of filling up inboxes or storage

capacities, deteriorating the speed of the internet to a great extent. These emails have the capability of corrupting one's system by smuggling viruses into it, or steal useful information and scam gullible people. The identification of spam emails is a very tedious task and can get frustrating sometimes.

While spam detection can be done manually, filtering out a large number of spam emails can take very long and waste a lot of time. Hence, the need for spam detection softwares has become the need of the hour. To solve this problem, various spam detection techniques are used now. The most common technique for spam detection is the utilization of Naive Bayesian [5] method and feature sets that assess the presence of spam keywords. The main purpose is to demonstrate an alternative scheme, with the use of Neural Network (NN) [4] classification system that utilises a collection of emails sent by several users, is one of the objectives of this research. One other purpose is the development of spam detection with the help of Artificial Neural Networks, resulting in almost 98.8% accuracy.

2. LITERATURE SURVEY

Email :

Electronic mail (email) is a messaging system that electronically transmits messages across computer networks. Anyone is free to use email services through Gmail, Yahoo or people can even register with an Internet Service Provider (ISPs) and be provided with an email account. Only an internet connection is required, otherwise being a free service.

Spam :

Bulk mails that are unnecessary and undesirable can be classified as Spam Mails. These spam emails hold the power to corrupt one's system by filling up inboxes, degrading the speed of their internet connection.

Spam Detection :

Many spam detection techniques are being used now-a-days. The methods use filters which can prevent emails from causing any harm to the user. The contributions and their weakness have been identified.

There are several methods that are accessible to spam, for example location of sender, it's contents, checking IP address or space names. [26]. Spammers use refined variations to avoid spam identification. Few measures connected with spam identification are; Blacklist and white-list, Machine learning approaches, Naïve Bayes,

Support Vector Machine, Neural Network Classification. [27]
A mobile system was proposed by Mahmoud et al. [28] with the motive of blocking and identifying spam SMS. In their work, they attempted to protect smartphones by filtering SMS spam that contains abbreviations and idioms. The system was based on the Artificial Immune System (AIS) and Naïve Bayesian (NB) algorithm. By the use of the Naive Bayes algorithm, the messages are classified based on their features. It used an SMS dataset with 1324 messages. Results from this system gave detection rate 82%, 6% positive rate and 91% accuracy.

Table 1 : Spam Categories

Categories	Descriptions
Health	The spam of fake medications
Promotional products	The spam of fake fashion items like clothes bags and watches
Adult content	The spam of adult content of pornography and prostitution
Finance & marketing	The spam of stock kiting, tax solutions, and loan packages
Phishing	The spam of phishing or fraud

An approach using random forest algorithm approach is proposed by Akinyelu and Adewumi [1] in order to identify the phishing or spam emails. It used 200 emails. The main motto of research was to reduce features and increase efficiency/accuracy. Accuracy of up to 99.7% with a minimal amount of 0.06% false positives is achieved by the proposed algorithm.
The research only covered the classification aspect without considering vital information which can affect the results, especially, in case of limited text in the email.

Yüksel et al. [3] aimed to resolve the problem of spam by inhibiting the spam emails from being spread within the email systems. To achieve this, they propose a cloud base system, which involves the identification of spam emails using analytics and machine learning algorithms like support vector machines and decision trees. The results of the tests show that the SVM leads to a higher accuracy of up to 97.6% and a false-positive rate of 2.33%. The decision tree attains a lower accuracy of 82.6% and a false-positive rate of 17.3%. Results reveal that the increase in spam emails is affected by the no. of received emails. Lee et al. [28] proposed an optimal technique for spam detection.

2.1. EXISTING SYSTEMS

Due to the increase in the number of email users, the amount of spam emails have also risen in number in the past years. It has now become even more challenging to handle a wide range of emails for data mining and machine learning. Therefore, many researchers have executed comparative studies to see various classification algorithms performances and their results in classifying emails accurately with the help of a number of performance metrics. Hence, it is important to find an algorithm that gives the best possible outcome for any particular metric for correct classification of emails and spam or ham.
The present systems of spam detection are reliant on three major methods:-

A. Linguistic Based Methods

Unlike humans, who can grasp linguistic constructs along with their exposition, machines cannot and hence it is necessary to teach machines some languages to help them understand these constructs. This is the technique that is used in places like search engines in order to ascertain the next terms for suggestions to the user while they are typing their search. Sentences are divided into two Unigrams (words taken are one by one) and two Bigrams (words that are taken two at a time). Since this technique requires that every expression be remembered, this method is not feasible and also time-intensive. [29]

B. Behavior-Based Methods

This technique is Metadata-based. This approach requires that users generate a set of rules, and the users must have a thorough understanding of these rules. Since the attributes of spam change over time so the rules also need to be reformed from time to time. As a result, it still requires a human to scrutinise the details and is majorly user-dependent. [29]

C. Graph-Based Methods

This technique uses a single graphical representation by incorporating numerous, heterogeneous particulars. Graph-based anomaly recognition algorithms are executed which detect abnormal forms in the data showing behaviours of spammers. This method is not dependable, so it is taxing to recognise faulty opinions. [29] Feature

Engineering mostly depends on the commercial appeal of terms and is absolutely content-oriented, and does not depend on statistics. All these attributes lead to a noteworthy decline of this structure.

3. PROPOSED METHOD

The dataset is taken from SpamAssassin [7], 2500 non-spam messages belong to easy_ham and they should be easily differentiated from spam. Instead of using sophisticated and hybrid models, this study relies on relatively simple classification algorithms to solve this problem like Logistic Regression, Naive Bayes, and Support Vector Machine. The concept of Neural Networks is also used to select the best activation function for spam detection. The dataset is in the form of HTML files which are converted into plaintext during text preprocessing. This paper has used two feature sets to find the most optimal feature set and respective models.

In order to perform efficient operations, Compressed Sparse Row (CSR) is used to feed data to models. Hence, the data is converted into a compressed sparse row matrix format for modeling.

A perfect (or best) model should be the one that reduces underfitting or overfitting. There are three practices for identification. They are datasets splitting, cross-validation, and bootstrap. In proposed work to prevent underfitting and overfitting, the modeling results will be evaluated first through a 10-fold cross-validation score, and then evaluated by evaluation metrics of classification.

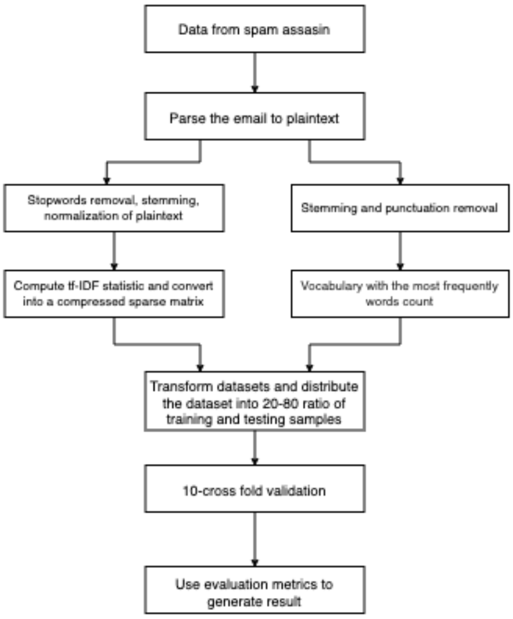
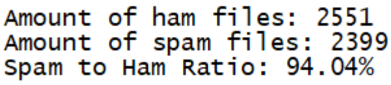

Data from spam assasin
Parse the email to plaintext
Stopwords removal, stemming, normalization of plaintext
Stemming and punctuation removal
Compute tf-IDF statistic and convert into a compressed sparse matrix
Vocabulary with the most frequently words count
Transform datasets and distribute the dataset into 20-80 ratio of training and testing samples
10-cross fold validation
Use evaluation metrics to generate result

Fig: 1) Flow Chart of Method

A - Data Set Reading and Inspection

The Data set can be taken from Spam Assassin [6][7]. It consists of nearly 5000 email files. These emails taken from Spam Assassin are used so that models can be created that can distinguish between spam and ham (non - spam) emails. The email data consists of either spam or hams. Spams, aka junk emails, are unsolicited messages sent in bulk by email. Hams are non-spams expected by email recipients. Data is read and inspected according to the existing kernel method [6]. Each file in the data source represents an email message.



```
Amount of ham files: 2551
Amount of spam files: 2399
Spam to Ham Ratio: 94.04%
```

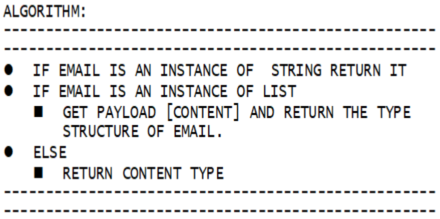
Fig: 2) Data set Files Record

All emails can be read by the python email package.

B - Text Preprocessing

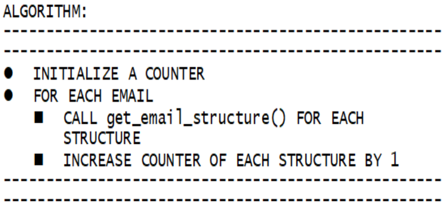
In this section, the email structure will be extracted and the content of the emails will be converted to plain text for the text analysis. This is executed through the following functions on the existing kernel [6]:

- get_email_structure()
This function is used to get the structure of email and its content.



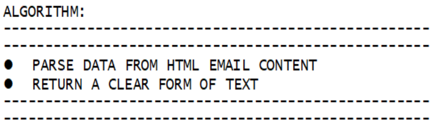
```
ALGORITHM:
-----------------------------------------------------
-----------------------------------------------------
●  IF EMAIL IS AN INSTANCE OF  STRING RETURN IT
●  IF EMAIL IS AN INSTANCE OF LIST
   ■  GET PAYLOAD [CONTENT] AND RETURN THE TYPE
      STRUCTURE OF EMAIL.
●  ELSE
   ■  RETURN CONTENT TYPE
-----------------------------------------------------
-----------------------------------------------------
```

- structures_counter()
This function is used to return the count of structures to find the most common parts.



```
ALGORITHM:
-----------------------------------------------------
-----------------------------------------------------
●  INITIALIZE A COUNTER
●  FOR EACH EMAIL
   ■  CALL get_email_structure() FOR EACH
      STRUCTURE
   ■  INCREASE COUNTER OF EACH STRUCTURE BY 1
-----------------------------------------------------
-----------------------------------------------------
```

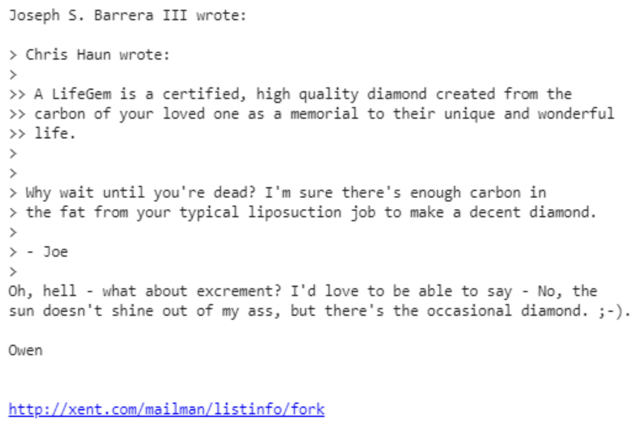
- html_to_plain()
This Function is used to get plain text emails as the email files in the dataset are read in HTML format with HTML tags present which needs to be removed.



```
ALGORITHM:
-----------------------------------------------------
-----------------------------------------------------
●  PARSE DATA FROM HTML EMAIL CONTENT
●  RETURN A CLEAR FORM OF TEXT
-----------------------------------------------------
-----------------------------------------------------
```

- email_to_plain()

This function is a driver of all the functions mentioned above. It is the final function which calls the above functions to return the emails in the dataset to plain text emails.



```
Joseph S. Barrera III wrote:

> Chris Haun wrote:
>
>> A LifeGem is a certified, high quality diamond created from the
>> carbon of your loved one as a memorial to their unique and wonderful
>> life.
>
>
> Why wait until you're dead? I'm sure there's enough carbon in
> the fat from your typical liposuction job to make a decent diamond.
>
> - Joe
>
Oh, hell - what about excrement? I'd love to be able to say - No, the
sun doesn't shine out of my ass, but there's the occasional diamond. ;-).

Owen


http://xent.com/mailman/listinfo/fork
```

Fig: 3) Email to Plain function sample output

C - Feature Sets & Vectorization

Two feature sets will be prepared for the modeling task :

1. The feature set 1 - Stopwords with N-gram and Term Frequency Inverse Document Frequency (tf-idf).

2. The feature set 2 - Most Frequent Word Count with Count Vectorization.
   Both Feature sets are developed using the existing kernel [6].

Feature set 1

It is created by exploring the text structure to exploit the contextual features.

```
ALGORITHM:
---------------------------------------------------
---------------------------------------------------
●  USE STOP-WORDS ALGORITHM USING NATURAL LANGUAGE
   TOOLKIT PACKAGE [NLTK]
●  USE THE STEMMER OF NLTK PACKAGE TO REDUCE
   DERIVED WORDS IN EMAILS
●  NORMALISATION IS THE NEXT STEP. IT IS DONE BY:
   ■  HAVING CERTAIN EXPRESSIONS LIKE web URL,
      EMAIL ADDRESSES, PHONE NUMBERS ETC BELONG TO
      THE SAME PATTERN BY GROUPING THEM.
   ■  IT SHOULD RETURN THE EMAIL IN PLAIN LOWER
      TEXT WITH GROUPED FEATURES INSTEAD OF
      ADDRESSES AND LINKS.
●  TOKENIZATION TO CHOOSE N-GRAMS MODE FOR
   VECTORIZATION
   ■  AFTER SELECTING N-GRAMS, WE CAN CALCULATE
      THE FREQUENCY OF EACH N-GRAM
   ■  USING TERM FREQUENCY AND INVERSE DOCUMENT
      FREQUENCY
NOTE: SCIKIT-LEARN LIBRARY PROVIDES COMPLETE
FRAMEWORK FOR TOKENIZATION PROCEDURE
---------------------------------------------------
---------------------------------------------------
```

Feature set 2

It is based on counting the most frequently occurring words from the email content.

```
ALGORITHM:
---------------------------------------------------
---------------------------------------------------
●  CONVERT EMAIL TO WORDS
   ■  LOWERCASE CONVERSION
   ■  PUNCTUATION REMOVAL
   ■  STEMMING
   ■  COUNT FREQUENCY OF EACH WORD
   ■  RETURN THE LIST OF WORDS WITH THEIR
      FREQUENCIES
●  CONVERT WORDS AND THEIR COUNTS TO VECTORS
   ■  CONSIDER ONLY MOST FREQUENTLY OCCURING WORDS
   ■  CONVERT THEM USING COUNT VECTORIZATION
      METHOD.
---------------------------------------------------
---------------------------------------------------
```

D - Pipeline

A pipeline is created so it's easy to compare different models and feed data to them with their feature set. The models being used and metrics to compare them are shown as below:

1. Naive Bayes

A standard Multinomial Naive Bayes.
This algorithm is a supervised learning algorithm [25] that relies on the Bayes theorem.

2. Logistic Regression

Dependent variables are incorporated in the Logistic Regression [5] techniques that denote binary values (0 or 1, true or false, yes or no), implying that the results can only be in two forms. Finding the probability of a favorable or failed event can be seen as an example of binary values.

3. Support Vector Machine

Support Vector Machine or SVM is one of the foremost in style Supervised Learning algorithms [23], that is employed for Classification similarly as Regression issues.

4. Neural Network

A neural network with 2 hidden layers is constructed and tanh activation function is used. Tuned by sklearn and TensorFlow packages. [4]

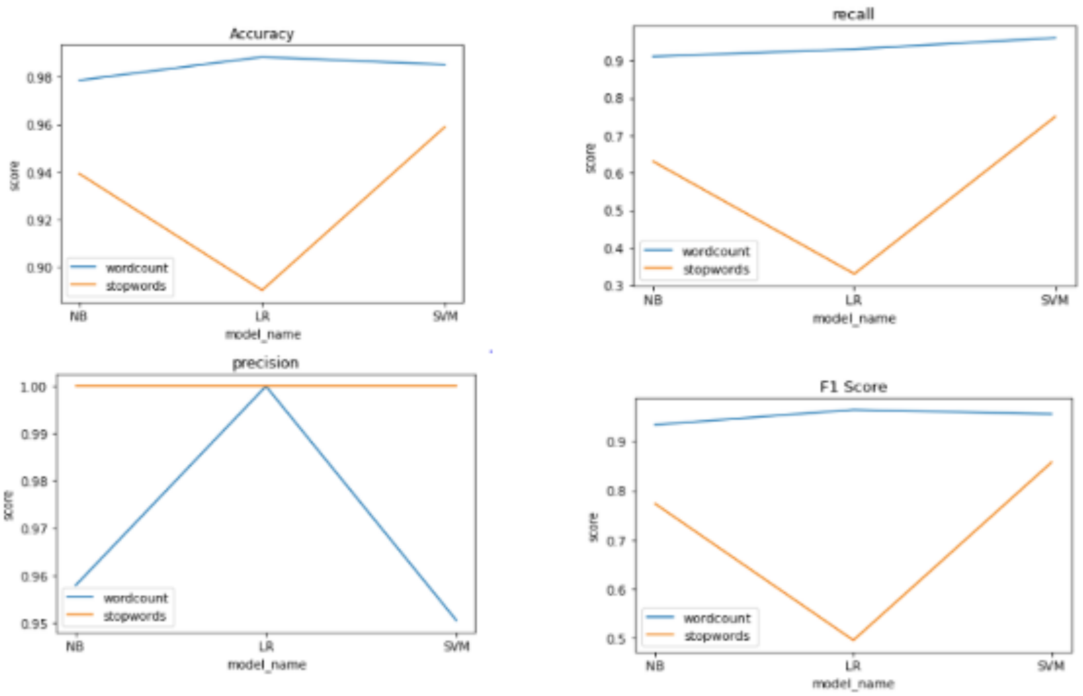
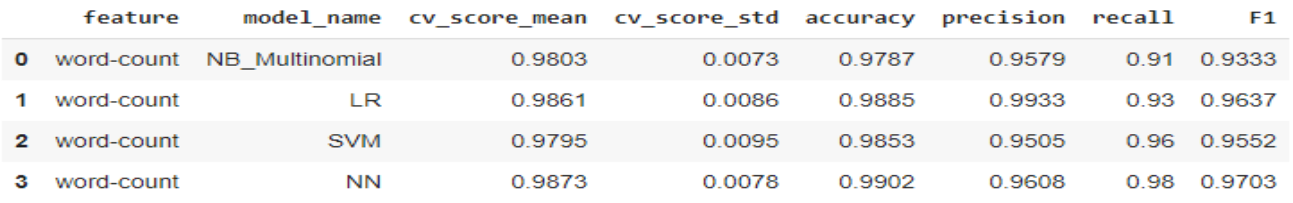


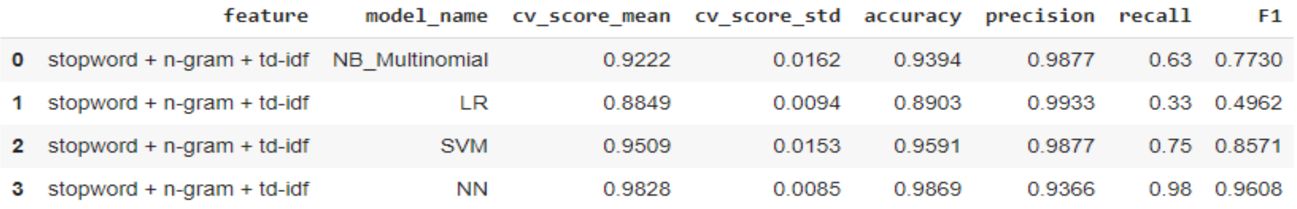
Fig. 4 Graphs of Evaluation metrics: - Accuracy; Precision; Recall; F-score

Table 2 Feature set 1 Outputs



	feature	model_name	cv_score_mean	cv_score_std	accuracy	precision	recall	F1
0	word-count	NB_Multinomial	0.9803	0.0073	0.9787	0.9579	0.91	0.9333
1	word-count	LR	0.9861	0.0086	0.9885	0.9933	0.93	0.9637
2	word-count	SVM	0.9795	0.0095	0.9853	0.9505	0.96	0.9552
3	word-count	NN	0.9873	0.0078	0.9902	0.9608	0.98	0.9703

Table 3 Feature set 2 Outputs



	feature	model_name	cv_score_mean	cv_score_std	accuracy	precision	recall	F1
0	stopword + n-gram + td-idf	NB_Multinomial	0.9222	0.0162	0.9394	0.9877	0.63	0.7730
1	stopword + n-gram + td-idf	LR	0.8849	0.0094	0.8903	0.9933	0.33	0.4962
2	stopword + n-gram + td-idf	SVM	0.9509	0.0153	0.9591	0.9877	0.75	0.8571
3	stopword + n-gram + td-idf	NN	0.9828	0.0085	0.9869	0.9366	0.98	0.9608

4. RESULTS

The evaluation criteria is simply based on the following evaluation metrics:

- Accuracy
- Precision
- Recall
- F1 score

These four factors comprehend the performance of a model with the feature set.
In the figure 4 above, it is shown how different models perform with these respective metrics.

As shown by the accuracy graphs it can be seen that the artificial neural network has the highest detection rate of whether a file is spam or ham. Also as shown by recall and F-Score it can be seen that the Neural Network out performs every other model. However results can also be seen in terms of precision logistic regression is the better, however it's not the best model as its poor performance compared to others.

Table 2 and Table 3 showcases the output of the results of feature set 1 and feature set 2 with the models respectively. cv_score_mean refers to cross validation score, and is used to verify accuracy results. cv_score_std refers to deviation in cross validation and also how much overfitting is there in the model.

Among all models using the feature 1 stopwords + n-gram + tf-idf as shown in Table 2, Neural Network using tanh activation function achieved maximum accuracy and viz. 98.69%. Logistic Regression got the highest precision 99.33% so false-positives are least there.

Among all models using the feature 2 word-count as shown in Table 3, Neural Network using tanh activation function achieved maximum accuracy and viz. 99.02%. Logistic Regression got the highest precision 99.33% so false-positives are least there but it's score and recall are less than Neural Network.

5. CONCLUSION

As shown in Figure 4, all the models based on the feature set 2 most-frequent-word-count have higher accuracy and F1 score than those based on the feature set 1 stop words + n-gram + tf-IDF.

If the use case is to introduce a beta version of an email spam detector like no-spam in the inbox. In this case, the model: Neural Network with tanh activation function and the feature set 1 stop words + n-gram + tf-IDF serves this purpose.

According to the graphs in Figure 4, if the use case is to introduce an email spam detector to reduce bad user experience in searching for important emails from junk mailboxes and filtering spam from the inbox. In this case, Neural Network with a feature set 2 - 'most frequent word count' gives a better user experience in general.

The future work includes testing the model with various standard datasets. This research proposes that the outcome that is obtained should be compared with additional spam datasets from various sources. Also, more classification and feature algorithms should be analyzed with email spam datasets.

6. REFERENCES

[1] AKINYELU, A. A., & ADEWUMI, A. O. (2014). "Classification of phishing email using random forest machine learning technique". Journal of Applied Mathematics.

[2] Vinodhini. M, Prithvi. D, Balaji. S "Spam Detection Framework using ML Algorithm" in IJRTE ISSN: 2277-3878, Vol.8 Issue.6, March 2020.

[3] YÜKSEL, A. S., CANKAYA, S. F., & ÜNCÜ, İ. S. (2017). "Design of a Machine Learning Based Predictive Analytics System for Spam Problem." Acta Physica Polonica, A., 132(3).[26] GOODMAN, J. (2004, July). "IP Addresses in Email Clients." In CEAS.

[4] Deepika Mallampati, Nagaratna P. Hegde "A Machine Learning Based Email Spam Classification Framework Model" in IJITEE, ISSN: 2278-3075, Vol.9 Issue.4, February 2020.

[5] Javatpoint, "Machine Learning Tutorial" 2017 https://www.javatpoint.com/machine- learning

[6] SpamAssassin, "Spam and Ham Dataset'', Kaggle, 2018. https://www.kaggle.com/veleon/ham-and-spam-dataset

[7] Apache, "open-source Apache SpamAssassin Dataset", 2019 https://spamassassin.apache.org/old/publiccorpus/

[8] SpamAssassin, "Spam Classification Kernel", 2018 https://www.kaggle.com/veleon/spam-classification

[9] SpamAssassin, "REVISION HISTORY OF THIS CORPUS", 2016 https://spamassassin.apache.org/old/publiccorpus/readme.html

[10] Jason Brownlee, "Naive Bayes for Machine Learning" The Machine Learning Mastery, April 11, 2015. https://machinelearningmastery.com/naive-bayes-for-machine- learning/

[11] Wikipedia, "History of email spam," Internet Free Encyclopedia, 2001. https://en.wikipedia.org/wiki/History_of_email_spam

[12] Rohith Gandhi, "Support Vector Machine" The Machine Learning Mastery, June 7, 2018. https://towardsdatascience.com/support-vector-machine-introduction-to-machine- learning-algorithms-934a444fca47

[13] Jason Brownlee, "Logistic Regression for Machine Learning" The Machine Learning Mastery, April 1, 2016. https://machinelearningmastery.com/logistic-regression-for-machine-learning/

[14] Jason Brownlee, "How to Encode Text Data for Machine Learning with scikit- learn" The Machine Learning Mastery, September 29, 2017. https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/

[15] I. Androutsopoulos, J. Koutsias, K. Chandrinos and C. D. Spyropoulos, "An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal email messages," Computation and Language, pp. 160-167, 2000.

[16] G. V. Cormack, "Email Spam Filtering: A Systematic Review," Foundations and Trends® in Information Retrieval, vol. 1, no. 4, pp. 335-455, 2006.

[17] M. Siponen and C. Stucke, "Effective Anti-Spam Strategies in Companies: An International Study," Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06), 2006.

[18] Guzella, T. S. and Caminhas, W. M."A review of machine learning approaches to Spam filtering." Expert Syst. Appl., 2009.

[19] Jianying Zhou, Wee-Yung Chin, Rodrigo Roman, and Javier Lopez, (2007) "An Effective MultiLayered Defense Framework against Spam", Information Security Technical Report 01/2007.

[20] Xiao Mang Li, Ung Mo Kim, (2012) "A hierarchical framework for content-based image spam filtering", 8th International Conference on Information Science and Digital Content Technology (ICIDT), Jeju, June, pp. 149-155.

[21] Linda Huang, Julia Jia, Emma Ingram, Wuxu Peng, "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.

[22] W.A. Awad, S.M. Elseuofi, Machine learning methods for spam E-mail classification, Int. J. Comput. Sci. Inf. Technol. 3 (1) (2011) 173–184.

[23] K.R. Dhanaraj, V. Palaniswami, Firefly and Bayes classifier for email spam classification in a distributed environment, Aust. J. Basic Appl. Sci. 8 (17) (2014) 118–130.

[24] M. Zavvar, M. Rezaei, S. Garavand, Email spam detection using combination of particle swarm optimization and artificial neural network and support vector machine Int. J Mod Educ. Comput.Sci. (2016) 68-74.

[25] Deepika Mallampati, "An Efficient Spam Filtering using Supervised Machine Learning Techniques" in IJSRCSE, Vol.6, Issue.2, pp.33-37, April (2018).

[26] [Deepika Mallampati, K.Chandra Shekar and K.Ravikanth "Supervised Machine Learning Classifier for Email Spam Filtering", © Springer Nature Singapore Pte Ltd. 2019 and Engineering, https://doi.org/10.1007/978-981-13-7082-341.

[27] GUPTA, H., JAMAL, M. S., MADISETTY, S., & DESARKAR, M. S. (2018, January). "A framework for real-time spam detection in Twitter." In Communication Systems & Networks (COMSNETS), 2018 10th International Conference on (pp. 380-383).

[28] MAHMOUD, T. M., & MAHFOUZ, A. M. (2012). "SMS spam filtering technique based on artificial immune system." International Journal of Computer Science Issues (IJCSI), 9(2), 589.

[29] AN ANTI-SPAM DETECTION MODEL FOR EMAILS OF MULTI-NATURAL LANGUAGE Mazin Abed Mohammed a,*, Salama A. Mostafa b,*, Omar Ibrahim Obaid