Study of Spam Email Filtering Methods using Supervised Machine Learning Techniques

Mallikarjuna Rao Ch
Computer Science and Engineering Department
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
cmrao@griet.ac.in

Sahithi Reddy S
Computer Science and Engineering Department
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
sahithisreddy@gmail.com

Sathvika Konlyada
Computer Science and Engineering Department
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
sathvikarao09@gmail.com

Vineela Konagala
Computer Science and Engineering Department
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
vineelakonagala145@gmail.com

Vanditha Das Chenda
Computer Science and Engineering Department
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
vanditha.das@gmail.com

Abstract – **The development of the Internet paved the way for the commercial exchange of data-extensive messages in the form of E-Mails. But, a big issue is spam E-Mails which are unwanted and incessant emails that the user may or may not have signed up for. Hence, users need to have a filter that distinguishes spam emails from ham to avoid unnecessary and potentially harmful messages. This paper includes the framework for automatically detecting spam emails using supervised machine learning techniques. The models are trained on an openly available dataset with additional methods that help in gaining insight into the data. The performance and the accuracy of different models for segregating incoming emails are then tested and compared.**

Keywords – Spam email, machine learning, Naïve Bayes, Support Vector Machine.

I. INTRODUCTION

E-Mails are the most commonly used forms of communication due to their accessibility and the relative speed at which messages are transferred. Due to its large and ever-growing user base, it is also an ideal platform to reach a huge number of people with minimum effort. It has helped build lucrative businesses by allowing effective communication with the customers directly about their products and services and also receiving feedback. At the one-to-one user level too, it facilitates an easy and quick transfer of messages and data. However, there has also been an increase in spam emails that reach people's inboxes. Spam or junk mails are unwarranted or unsolicited emails that are usually sent out in bulk persistently. They can also be fraudulent and often lead to a loss of productivity and resources. Some might also be counterfeit messages that trick the user into revealing sensitive information.

It is imperative that businesses install a spam filter to lower the chance that customers may click on something inappropriate and thereby protect their internal data from a cyber assault. Hence, spam mail detection increases the security of sensitive data within an organization or even of a single user and provides more control and privacy. It would basically also act as an anti-malware tool.

An email generally consists of a header and a body. The header consists of a brief explanation of the email content and information regarding the subject, sender, and receiver. It has fields such as the sender's address and the recipient's address and it may also contain a timestamp that indicates when the message was sent and delivered. The body consists of the main message in the email and has all the details the sender wants to convey. The data can be text, audio, video, images, files, or HTML markup. All the information available has to undergo some processing before the classifier is used for filtering. The data from an email message is processed in multiple stages before categorization by identifying and selecting the important attributes and features that would have maximum effect.

The learning techniques used in this outline are Gaussian Naive Bayes and SVM algorithms. They are used to analyze the dataset obtained on Kaggle and to predict the output values. The results are compared using certain metrics. There exists a general consensus that the simple Naive Bayes algorithm performs comparatively well in text classification. This project aimed to understand this significance by collating it with the more powerful SVM algorithm.

II. RELATED WORK

1. U Murugavel, R Santhi[1] elucidate an approach to classify spam mail using K-Nearest Neighbours Classifier which requires extracting particular attributes to increase efficiency. But, predominantly, K-Nearest Neighbours can be computationally expensive and also require a greater amount of memory as the algorithm needs to store the data for processing. Moreover, it can perform fairly well in circumstances when the new spam mails are extremely similar in information to the trained data as a distance metric is used for classification.

2. Another supervised learning technique used is the Decision Tree algorithm[2] which is a popular classifier but it can be extremely sensitive to outliers in the training set, thereby, affecting the overall accuracy.

3. Sankar K.V. et al. (2015) [3] proposed the detection of masked spam synonym relation completion and keyword concatenation. The entire content of an E-Mail is considered instead of some keywords.

4. The survey[4] also details some unsupervised learning techniques that were employed with an improved digest algorithm or on the basis of string equivalence. Though these approaches are unconventional and not widely used, they provided satisfactory results.

5. I. AbdulNabi and Q. Yaseen, [11] offered a fix for the problem of word embedding in email classification. BERT (Bidirectional Encoder Representations from Transformers), a pre-trained transformer model, has been improved to distinguish spam emails from non-spam emails (HAM). The outcomes are contrasted with a baseline DNN (deep neural network) model that included a Bi-LSTM (bidirectional Long Short Term Memory) layer and two stacked Dense layers. Additionally, the outcomes are contrasted with those of a group of industry-standard classifiers called k-NN (k-nearest neighbors) and NB (nearest neighbor) classifiers (Naive Bayes). The model is trained using one open-source data set, while the other is utilized to test the model's robustness and persistence in the presence of unlabeled input. The recommended approach achieved the highest F1 score (98.66%) and the highest accuracy (98.67%).

6. Rohit Giyanani and Mukti Desai [12] presented a model to detect spam emails using statistical Natural Language Processing (NLP) to distinguish between legitimate and spam messages. The threshold counter helps reduce congestion, but increases the required storage space. The system blocks incoming emails based on the sender and the content of the email message.

7. Weimiao Feng et al., [13] have designed a SVM-NB system that effectively classifies emails as either spam or ham (legitimate). This system uses the SVM and Naive Bayes algorithms to separate the training dataset by constructing an optimal

hyperplane. The experiment was conducted using the DATAMALL dataset, and the results showed that this approach provides improved accuracy and faster classification compared to other methods.

8. Savita Teli and SantoshkumarBiradar [14] have studied various methods for detecting spam and the related issues. The methods they discussed include list-based or rule-based filters (such as blacklists, whitelists, black holes, and greylists), content-based filters, and Bayesian filters. The authors have concluded that Bayesian classifiers are more effective than other methods because they take into account the entire message and continually adapt to changing conditions.

9. Authors in [15] developed a method for changing the email classification problem into a graph classification problem. The email text does not need to be translated into a vector representation for this project. In contrast, this approach uses a graph neural network to classify spam emails by converting the email's content into a graph (GNN). In [9], authors developed several techniques, including the B-TransE mode, to identify false news based on news content and knowledge graphs. The author provided various fresh approaches for identifying fake news based on incomplete and imperfect knowledge graphs, using the existing TransE model and the newly presented B-TransE model.
10. Authors in [16] concentrated on ways to efficiently hone SMS spam. The Naive Bayes, Gradient Boost Logistic Regression, SGD classifier, and Deep learning-based models like CNN and LSTM were among the machine learning-based classifiers that were tested. According to their findings, the CNN model, which had an accuracy of 99.44% on randomly generated tenfold cross validation data, performed best for screening real text messages. However, the effort was constrained by the fact that it was solely dependent on English-language messaging.

III. ARCHITECTURE

The project employs supervised machine learning algorithms on a sizable and openly available dataset of emails that include ham and spam mail. Fig.1. elucidates the architecture and workflow of the email spam filtering process. The dataset is obtained from the "Kaggle" website for training and testing and then is splitted. The whole dataset is located in a directory, in which it contains all the data in the form of text files. In the data set, there exists 3000 emails in which 2551 are spam and 500 are non-spam. For efficient processing of the data, it is initially cleaned by removing non-relevant content which is an important step for better performance further on. For extracting the features from the dataset, a few preprocessing techniques are used such as tokenization, lemmatization, removal of stop words etc. Then among the extracted attributes feature selection is done by removing the unimportant attributes. The rudimentary pre-processing steps needed to analyze text data in E-Mails are tokenization and feature selection and extraction. Tokenization helps in separating a block of text into smaller units called tokens. The tokens can either be words, characters, or sub-words. It helps in the generalization of the relationship between the texts or words. The entire email text is divided into tokens for better analysis and classification. Subsequently, the most commonly occurring terms across the documents are retrieved and analyzed. This helps in building a model that can identify particular terms attributed to spam mails and thereafter classify incoming mails into spam and ham.

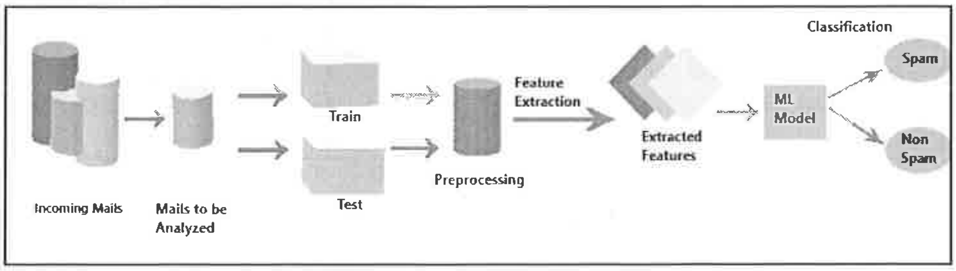


Fig. 1. System Architecture

IV. PROPOSED WORK

A. *Naïve Bayes Classifier*

The Naive Bayes method is a supervised machine learning algorithm. A naïve Bayes classifier is a straightforward probabilistic classifier with robust independence assumptions[11]. It makes the assumption that, given the value of the class variable, all characteristics are independent. As a result, a Naive Bayes classifier makes the assumption that a specific property of a class has no relationship to any other property. Naive Bayes is preferable to conditional models for email spam filtering due to their simplicity, ease of use, and speedy convergence[12]. The Naive Bayes classifier also has the benefit of just requiring a small training dataset.

i. Process

The first step is to collect a meaningful and legitimate dataset which is then entirely explored and analyzed. Then, preprocessing is done. In this step, first, the data is converted to lowercase. Then stop words are removed. Stop words are the most common words which do not add much importance to the text. So removing the stop words reduces the dataset's size and the computation time, which increases the classification accuracy. Since the stop words occur commonly in both spam and ham mail, these words don't hold much importance in classification. Hence, these words are removed. Next lemmatization is done. Lemmatization is the process of converting a word into its base form. This helps in categorizing and grouping similar words so they can be analyzed as a single word. This holds great importance in analyzing text messages over stemming. The reason is that when lemmatization is used it converts the word into its base form without changing its meaning, but when stemming is used it removes the last letter to convert it into its base form. For example, in lemmatization, caring is converted into care. Whereas stemming converts caring into a car, which changes the original meaning of the word. Hence, lemmatization is used for preprocessing.

Word cloud is a model used to gain insights into the dataset as it visually represents the repetition and frequencies of some common words. Before splitting the data, the data is vectorized using tf-idf vectorization. Tf-Idf stands for term frequency and inverse document frequency. Term frequency defines the total number of repetitions of a particular term in the document. Document frequency indicates how frequently the term is occurring in all documents. Inverse document frequency aims to reduce the weight of a term if the occurrences of the term are scattered throughout all the documents present. So splitting the data after vectorization increases the accuracy of the model which is better than the accuracy calculated on raw data. The data is split into training and testing datasets.

Then the training dataset is used for training the model. The model used for training is gaussian which is a probabilistic classification algorithm that has strong independence assumptions. This independence among features is generally considered a poor assumption but in practice, its working has proved to be tantamount to the ones by more advanced classifiers[13]. It is a straightforward yet effective technique for supervised learning algorithms' predictive modelling.. After training the model is tested on the testing dataset. Performance metrics are calculated:

ii. Algorithm

- Libraries such as NumPy, Pandas, and sklearn are imported into the proposed model
- Data is then read from the drive
- The whole dataset is labelled such that spam emails are labelled as 1 and ham emails are labelled as 0
- Data is explored and analyzed
- In preprocessing, stopwords are removed and lemmatization is done
- Word cloud is designed labe
- Vectorizing the data
- Splitting of dataset
- A gaussianNB model is used for training
- Performance measurement. The performance of the classification algorithm is usually examined by evaluating the accuracy of the classification

B. Support Vector Machine

Support Vector Machine is a supervised learning algorithm used for both classification and regression problems. In SVM, the best decision boundary line is created to separate the n-dimensional space into separate classes. This helps in categorizing the new incoming data into the correct class easily. The decision boundary is called the hyperplane. The extreme points chosen to create the hyperplane are called the support vectors. Therefore, this algorithm is called the Support Vector Machine. The hyperplane distinguishes the space into well-defined classes with a set of attributes. SVM has the advantage of increasing class separation and reducing expected prediction error. SVM is flexible for both linear and nonlinear-based analysis.

i. Process

Data collection is done and it undergoes pre-processing methods such as feature selection to remove unwanted and redundant data. Data is split into training and testing datasets. Pipeline allows the steps to be specified and evaluated in a sequential manner. It can be applied for testing and optimizing the model by using lemmatization and noise-reduction methods in email spam filtering [14]. This gives more accurate results and better predictions. The pipeline method involves continuous deployment of the code which will reduce the total training cost without compromising the quality. Whereas, state-of-the-art deployment costs more comparatively [15]. The first step in the pipeline method is scaling which is done using MinMaxScaler and the second step is model training which is done using the support vector classifier.

MinMaxscaler scales and translates every feature individually such that it is in the given range in the training dataset. MinMaxscaler scales every feature in the range of [0,1] generally and if there are any negative values present in the dataset then the values are scaled in the range of [-1,1]. In this project, the scaling is done in the range of [-1,1]. This is done due to the presence of negative values in the dataset. This is done to maintain consistency in the data values across the different attributes.

This project employs a Support Vector Classifier, which maps data to a high-dimensional feature space in order to classify data points. This is possible even when the data cannot be separated linearly. It transforms data from a lower dimension to a higher dimension so that it may be distinguished by clearly defined borders. The classifier uses Kernel as a function that takes input data and manipulates it into the required form. There are various values for kernel functions such as linear, RBF and sigmoid. Radial basis function kernel or RBF kernel is a kernel function more complex than other kernel functions. It can combine multiple polynomial kernels. Sigmoid kernel is an activation function for artificial neurons. The linear kernel is used for linearly separable data where there are more features[16]. The data is divided into different classes by a single line. It is easy to implement and more efficient as compared to other kernels and is used in problems related to text data. Hence a linear kernel is used for this model. Using the kernel function, non-linearly separable data can be converted into separable data. The default gamma function is taken which has the value “auto”. Gamma is a parameter of the non-linear hyperplane. The impact of a single training example is studied through the gamma parameter. If the value is low, support vectors are far and high values mean that they are close.

ii. Algorithm

- The data set is read and labelled as 0 for spam and 1 for ham mails.
- The model is trained using a pipeline which is used for executing the methods in a sequential manner.
- The methods in the pipeline include MinMaxScaler and support vector classifier.
- MinMaxScaler scales the dataset values in a given range and scaling is done.
- In support vector classifier, the linear kernel is used and the gamma value is set to default i.e, auto
- The dataset is split into training and testing datasets.
- The results of the classification model are given by the confusion matrix.

V. RESULT ANALYSIS

The performance of the models is evaluated based on a set of parameters. They are

- Accuracy
- Precision
- Recall
- F1-score
- Support

Fig 2 is the confusion matrix, the result of the Naive Bayes classifier. It is observed that 1683 images are correctly classified as ham mails which have a label as 0, and 407 images are correctly classified as spam mails which are labelled as 1. And a few others are incorrectly classified.

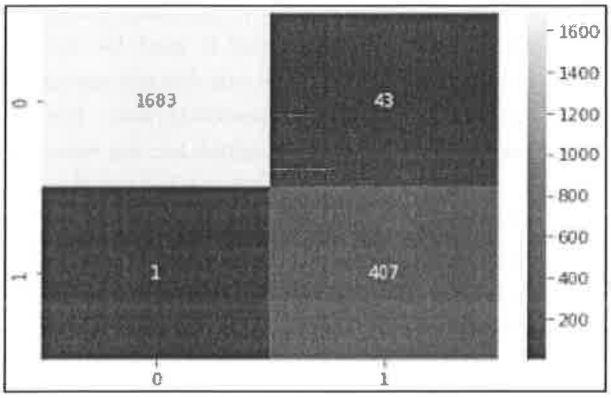
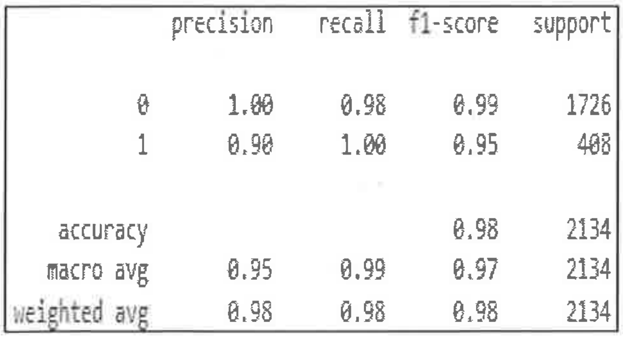


Fig. 2. Confusion Matrix of Naïve Bayes Model

Fig 3 presents the classification report of Naive Bayes with an accuracy of 0.97



```
              precision    recall  f1-score   support

           0       1.00      0.98      0.99      1726
           1       0.90      1.00      0.95       408

    accuracy                           0.98      2134
   macro avg       0.95      0.99      0.97      2134
weighted avg       0.98      0.98      0.98      2134
```

Fig. 3. Classification Report of Naïve Bayes Model

Fig 4 is the confusion matrix for the SVM classifier. It is observed that 959 images are correctly classified as ham mails which have a label as 0, and 111 images are correctly classified as spam mails which are labelled as 1. It can be observed that 455 mails have been classified wrongly which reduced the accuracy.

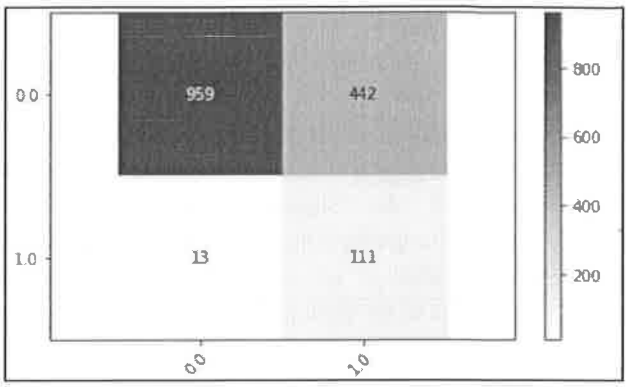
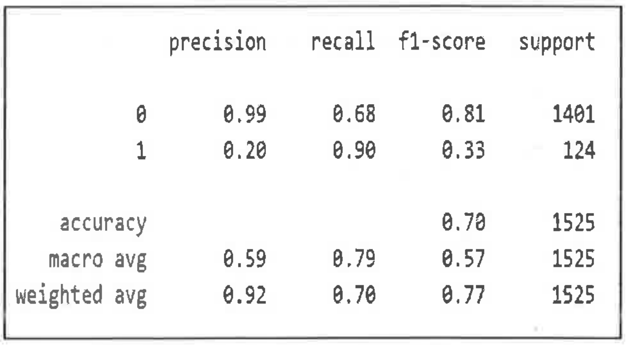


Fig. 4. Confusion Matrix of SVM Model

Fig 5 represents the classification report of SVM with an accuracy of 0.70



```
              precision    recall  f1-score   support

           0       0.99      0.68      0.81      1401
           1       0.20      0.90      0.33       124

    accuracy                           0.70      1525
   macro avg       0.59      0.79      0.57      1525
weighted avg       0.92      0.70      0.77      1525
```

Fig. 5. Classification Report of SVM Model

As per the accuracy values observed from the two matrices, we can conclude that Naive Bayes has better accuracy compared to SVM. The recall and F1-Score also show that Naive Bayes is more efficient comparatively.

VI. CONCLUSION AND FUTURE SCOPE

In this project, using the same training and testing datasets for both the Naive Bayes classifier and the SVM model allowed for a fair comparison of the performance of the two algorithms. The results show that the Naive Bayes classifier has a higher accuracy of 97% over SVM's 70% for email spam classification, making it a more efficient and effective option for this task. It has the ability to handle large datasets and high-dimensional feature spaces efficiently. It is relatively easy to implement and can be easily integrated into existing email systems. This can lead to improved user experience and better performance of the email platform.

Spam email detection is an ongoing challenge as spammers continuously find new ways to evade detection. Spam filters are an important tool in fighting against spam, but they are not foolproof and may generate false positives or negatives. So, in the future, it is important to continue to improve and update spam filtering techniques by the addition of new features like stop words in order to effectively combat spam and protect users from unwanted emails.

The research has shown that the Naive Bayes classifier is an excellent algorithm for email spam detection, and it can be used in various email platforms and applications. Since it is scalable, this system can be developed by integrating it with various other algorithms.

The models are currently trained on emails written in English but the scope of spam email detection can be increased to different languages too.

VII. BENEFITS

The huge volume of spam emails flowing through computer networks has destructive effects on the memory space of email servers, communication bandwidth, CPU power and user time[4] and not to mention the breach in security and privacy. Implementation of a spam filter reduces IT administration and network costs as expenditure need not be allocated to recuperate from any unwarranted exposure of internal data. It tremendously improves user experience as unnecessary and dangerous messages are kept out of their inboxes. Business emails also benefit from a higher quality of life because they function properly as they are only ever utilized for their intended purposes.

VIII. REFERENCES

[1] U Murugavel, R Santhi, “K-Nearest neighbor classification of E-Mail messages for spam detection”, ICTAT Journal on Soft Computing, vol. 11, Issue 1, 2020, pp. 2218-2221.

[2] A. Wijaya and A. Bisri, "Hybrid decision tree and logistic regression classifier for email spam detection", 2016 8th International Conference on Information Technology and Electrical Engineering, 2016, pp. 1-4, doi: 10.1109/ICITEED.2016.7863267.

[3] K.V. Sankar, S.Uma, P.S. Subin, T Abhimannan, “Mask spam detection using difficult keyword identification and relation completion”, International Journal of Computer Science and Mobile Computing, vol.4, Issue 4, April- 2015, pp. 451-457.

[4] N. Ahmed, R. Amin, H. Aldabbas, D. Koundal, B. Alouffi, T. Shah, "Machine learning techniques for spam detection in Email and IoT platforms: analysis and research challenges", Security and Communication Networks, vol. 2022, 2022, doi: 10.1155/2022/1862888.

[5] I. AbdulNabi, Q Yaseen, “Spam Email Detection Using Deep Learning Techniques”, Procedia Computer Science, Vol. 184, Pages 853-858, 2021, doi:10.1016/j.procs.2021.03.107

[6] R. Giyanani, M. Desai, “Spam Detection using Natural Language Processing”, IOSR Journal of Computer Engineering, ISSN: 2278-0661, Vol. 16, Issue 5, Sept-Oct 2014.

[7] W. Feng, J. Sun, L. Zhang, C. Cao, Q. Yang, “A Support Vector Machine based Naive Bayes Algorithm for Spam Filtering”, IEEE 2016.

[8] S. Teli, S. Biradar, “Effective Spam Detection Method for Email”, IOSR Journal of Computer Science, pp: 68-75.

[9] W. Pan, J. Li, L. Gao, L. Yue, Y. Yang, L. Deng, C. Deng, "Semantic Graph Neural Network: A Conversion from Spam Email Classification to Graph Classification", Scientific Programming, vol. 2022, 2022. doi: 10.1155/2022/6737080.

[10] J. Fattahi & M. Mejri. “SpaML: a Bimodal Ensemble Learning Spam Detector based on NLP Techniques”.2020.

[11] M. tope, “Email Spam Detection using Naive Bayes Classifier”, IJSDR, vol. 4, Issue 6,2019.

[12] N.F. Rusland, N. Wahid, S. Kasim, and H. Hafit, “Analysis of Naïve Bayes algorithm for Email spam filtering across multiple datasets”, IOP Conference Series: Materials Science and Engineering, vol. 226, International Research

and Innovation Summit (IRIS2017), May 2017, doi: 10.1088/1757-899X/226/1/012091.

[13] I. Rish, "An empirical study of the Naïve Bayes classifier", IJCAI, 2001 Work Empir Methods Artif Intell. 3.

[14] A. Occhipinti, L. Rogers, C. Angione, "A pipeline and comparative study of 12 machine learning models for text classification", Expert Systems with Applications, vol. 201, 2022, doi: 10.1016/j.eswa.2022.117193.

[15] B. Derakhshan & V. Markl, "Continuous deployment of machine learning pipelines", EDBT, 2019, doi: 10.5441/002/edbt.2019.35.

[16] A. Vasileios, "SVM classification with linear and RBF kernels", 2015, doi: 10.13140/RG.2.1.3351.4083.