Archives available at journals.mriindia.com

International Journal on Advanced Computer Engineering and Communication Technology

ISSN: 2278-5140
Volume 14 Issue 01, 2025

Spam Mail Detection using Machine Learning

Dhanashri Bachche[1], Kavita Hirugade[2], Kavita Oza[3]
[1,2] Research Student [3]AssosiateProfessor,
[1,2,3]Department of Computer Science, Shivaji University, Kolhapur, Maharashtra
dhanashribachche1@gmail.com, kavitahirugade2002@gmail.com , kso_csd@unishivaji.ac.in

Peer Review Information

Submission: 16 Jan 2025
Revision: 13 Feb 2025
Acceptance: 12 March 2025

Keywords

Machine Learning
NLP
Spam Email Detection
Supervised Learning
Text Classification

Abstract

The aim is to create a spam filter that is efficient enough to enhance email security and productivity. Then we will evaluate some ML algorithms like Naive Bayes, Support Vector Machines (SVM) or Random Forests to find out which is going to be the best model to classify spam. Spam emails have become the bane of the internet world, but they have also turned to be a huge problem in the world of criminal activities such as phishing scams and frauds; as spam emails became more common, they needed better anti spam filters. Nowadays, machine learning techniques are used to detect and filter spam emails successfully. This study reviews popular machine learning based spam filters and their strengths, weaknesses and future directions. It also talks about how Gmail and Yahoo use machine learning to filter spam.

Introduction

This project will discuss how machine learning can identify spam emails. Machine learning is a form of artificial intelligence that can improve and learn automatically without being programmed in an explicit way. We will employ a binary classifier to classify emails into two categories: spam and legitimate (ham).

This has been a huge problem on the web, wasting time, storage space, and bogging down mail services. Spamming is simple to get through filters, which makes it difficult to block it. Machine learning can be used to identify spam email. A method is to look at the content of an email, but this can generate false positives where good emails get blocked. Some other methods involve blacklisting (blocking certain mail addresses) and white listing (only letting certain approved mail in). Spam mail is unsolicited and can be advertisements, phishing links, or from unknown people. 'Ham' mail is good mail which is not spam

Literature Review

In this project researcher use machine learning (ML) and natural language processing (NLP) techniques to classify email messages as either spam or legitimate [1]. This research paper mainly focuses on the content of the message to detect received mail is spam or ham. This research aims to tackle the issue of spam emails by proposing a new approach that combines TFIDF and SVM algorithms, achieving high accuracy in detecting spam emails [2]. This project will discuss how machine learning helps in spam detection. Fake online reviews are a problem, and researchers are using machine learning and NLP to detect them and ensure that reviews are genuine and trustworthy [3].This research paper clarify how machine learning techniques are used by top Internet service providers (ISPs) including Gmail and Yahoo junk mail filters to filter email spam[4].This research paper specified message can be stated as spam or not also IP addresses of the sender are often detected. This research aims to tackle the problem of spam emails by using mathematical techniques to identify spam messages and detect the senders' IP addresses, making it easier to block unwanted emails and protect personal information 5]. This research project proposes to address the issue of spam emails through Natural Language Processing (NLP) methods to identify and filter out spam emails. The project has a dataset of spam emails and compares the performances of various deep learning models (Dense classifier, Sequential Neural Network, LSTM, and Bi-LSTM) to classify the emails as either spam or valid (ham) [6].This project aims to use machine learning to identify and block spam emails that can harm your computer or steal your personal information, and to find the most effective way to do so[7].This article explores a Python-based method to block spam emails by using machine learning to identify and filter important words, and then using those words to train algorithms to detect spam[8].This project aims to create a reliable email classification model using Bayesian methods in Python to reduce the impact of spam emails on communication[9].This project aimed to improve spam email detection by analyzing browser data and testing algorithms to find the most effective ones, which can help reduce online harassment and fraudulent activities[10].In This paper, we tested different algorithms to see which one is best for classifying emails. We found that the Naïve Bayes algorithm is the most accurate and precise in detecting spam emails, using a tool called WEKA and a library called php-ml. We also compared our results with another algorithm called SVM and previous systems, and found that our approach is more accurate[11].This study proposes a new machine learning approach that combines two methods (Random Forest and J48) to classify emails as spam or legitimate (ham). And this study proposes a new machine learning approach that combines two methods to effectively classify emails as spam or legitimate[12].

DATA PROCESSING

Data processing is an important step in machine learning-based spam mail detection. The process starts with importing and gathering a dataset of labeled emails (spam and non-spam). The data is preprocessed by tokenizing the text, eliminating stop words, stemming or lemmatizing words, and converting all text to lowercase. After that, feature extraction methods like Bag-of-Words (BoW) or Term Frequency-Inverse Document Frequency (TF-IDF) are used to convert the text data into numeric vectors. Data is divided into training and test sets, and machine learning algorithms like Naive Bayes, Support Vector Machines (SVM), or Random Forest are trained on the training dataset to identify spam emails. Lastly, the model's performance is tested on the test data based on accuracy, precision, recall, and F1-score.

The first step is to import the dataset, which is downloaded from 'Kaggle' and then imported into CSV format.

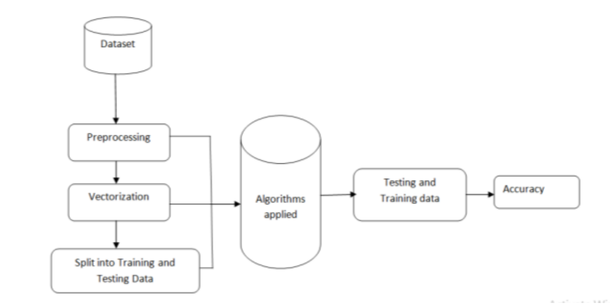

Dataset
Preprocessing
Vectorization
Split into Training and Testing Data
Algorithms applied
Testing and Training data
Accuracy

Fig.1: Data processing

METHODOLOGY

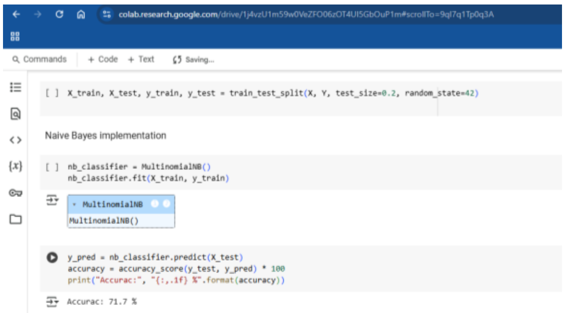
Spam mail detection employs a methodology which combines both machine learning methods along with NLP. One first accumulates and preprocesses a set of emails with respective labels as either spam or not spam, which in turn provide certain features of emails such as words, sentences, and addresses from where emails originated. Next, a machine learning algorithm like Naive Bayes, Support Vector Machine (SVM), or Random Forest is trained on the data to learn the patterns and features of spam emails. The trained model is then utilized to classify new, unseen emails as spam or non-spam. Besides that, methods like tokenization, stemming, and lemmatization are utilized to normalize the text data and feature selection algorithms are utilized in order to diminish the dimension of the feature space. Last but not least, the performance of the spam model is assessed with metrics like accuracy, precision, recall, and F1-score.

ALGORITHMS

1. Naïve Bayes:

Naive Bayes algorithm is employed in spam mail filtering by using Bayes' Theorem to determine whether an email is spam or not spam based on the words used in it. It operates by determining the probability of an email being spam or not spam when certain words are present. The model has the assumption that every word is independent of others (thus "naive") and has a training set of labeled emails on which it is trained to discover the probability of words being present in spam vs. non-spam emails. When an email comes in, the algorithm will

examine the frequency of known words and classify into the most likely class (spam or not spam), so it is an effective and easy-to-implement technique for spam detection.
P(A|B) = P(B|A) P(A) [7]
P(B)


colab.research.google.com/drive/1j4vzU1m59w0VeZFO06zOT4UI5GbOuP1m#scrollTo=9ql7q1Tp0q3A
Commands
+ Code
+ Text
Saving...

```
[ ] X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
```

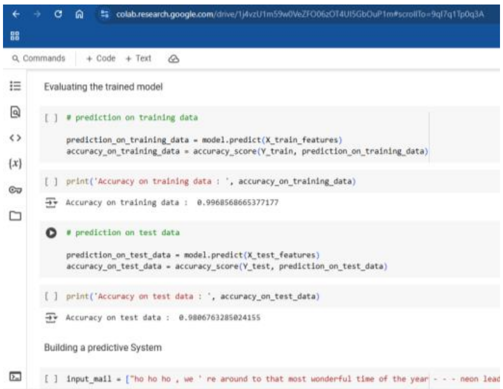
Naive Bayes implementation

```
[ ] nb_classifier = MultinomialNB()
    nb_classifier.fit(X_train, y_train)

    MultinomialNB
    MultinomialNB()

    y_pred = nb_classifier.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred) * 100
    print("Accurac:", "{:,.1f} %".format(accuracy))

    Accurac: 71.7 %
```

Fig. 2: sample image showing accuracy of the naïve bayes algorithm

2. Logistic Regression:-

Logistic Regression algorithm operates in spam filtering by learning to predict the probability of a message being spam or not spam given its features. The algorithm applies a logistic function to transform the input features into a probability score between 0 and 1, with scores closer to 1 indicating a greater chance of being spam. In training, the model learns the weights and the biases of individual features, like the occurrence of specific words or phrases, to optimize the accuracy of its prediction. Once a new message is received, the model feeds the features into the logistic function to produce a probability score, which is then compared with a threshold to determine the message as spam or non-spam.


colab.research.google.com/drive/1j4vzU1m59w0VeZFO06zOT4UI5GbOuP1m#scrollTo=9ql7q1Tp0q3A
Commands
+ Code
+ Text

Evaluating the trained model

```
[ ] # prediction on training data

    prediction_on_training_data = model.predict(X_train_features)
    accuracy_on_training_data = accuracy_score(Y_train, prediction_on_training_data)

[ ] print('Accuracy on training data : ', accuracy_on_training_data)

    Accuracy on training data :  0.9968568665377177

    # prediction on test data

    prediction_on_test_data = model.predict(X_test_features)
    accuracy_on_test_data = accuracy_score(Y_test, prediction_on_test_data)

[ ] print('Accuracy on test data : ', accuracy_on_test_data)

    Accuracy on test data :  0.9806763285024155
```

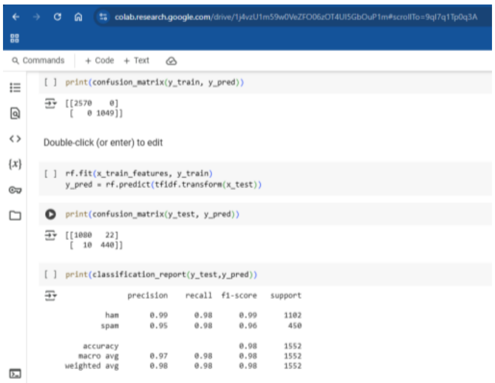
Building a predictive System

```
[ ] input_mail = ["ho ho ho , we ' re around to that most wonderful time of the year - - - neon lead
```

Fig. 3: sample image showing accuracy of the Logistic regression algorithm

3. Random Forest :-

Random Forest algorithm operates in spam detection by learning an ensemble of decision trees over a labeled data set, with every tree learning to predict messages as spam or not spam based on a random subset of features. The trees are constructed during training by selecting a random subset of features and splitting data between spam and not spam using these features. The trees are aggregated to create a "forest" of classifiers, which vote on the classification of new messages. When a new message is received, each tree in the forest predicts the probability that it is spam or non-spam, and the final classification is determined by combining the predictions.


colab.research.google.com/drive/1j4vzU1m59w0VeZFO06zOT4UI5GbOuP1m#scrollTo=9ql7q1Tp0q3A
Commands
+ Code
+ Text

```
[ ] print(confusion_matrix(y_train, y_pred))

    [[2570    0]
     [   0 1049]]
```

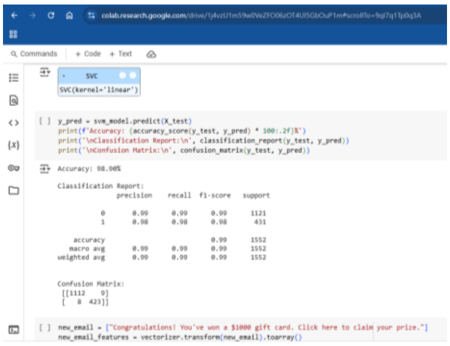
Double-click (or enter) to edit

```
[ ] rf.fit(x_train_features, y_train)
    y_pred = rf.predict(tfidf.transform(x_test))

    print(confusion_matrix(y_test, y_pred))

    [[1080   22]
     [  10  440]]

[ ] print(classification_report(y_test,y_pred))

                  precision    recall  f1-score   support

             ham       0.99      0.98      0.99      1102
            spam       0.95      0.98      0.96       450

        accuracy                           0.98      1552
       macro avg       0.97      0.98      0.98      1552
    weighted avg       0.98      0.98      0.98      1552
```

Fig. 4: Sample image showing accuracy of the Random Forest algorithm

4. SVM (Support Vector Machine):-

The SVM algorithm operates in spam filtering through identifying a hyper plane (decision boundary) that distinguishes between spam mails and non-spam mails. The algorithm starts with data preprocessing, in which a dataset of labeled mails (spam, non-spam) is gathered and preprocessed through tokenizing, stop words elimination, stemming, and transforming text data into numerical vectors. The SVM model is then trained on the labeled dataset with the features extracted to learn the best hyper plane that widens the margin between the nearest spam and non-spam emails. After the hyper plane is determined, unseen new emails can be labeled as spam or non-spam by checking which side of the hyper plane they are on. The SVM algorithm also uses kernel functions to map the data into a higher-dimensional space to enable non-linear classification. Through the determination of the best hyper plane, SVM is capable of high accuracy in spam detection and is hence a robust algorithm for this purpose.


colab.research.google.com/drive/1j4vzU1m59w0VeZFO06zOT4UI5GbOuP1m#scrollTo=9ql7q1Tp0q3A
Commands
+ Code
+ Text
SVC

```
SVC(kernel='linear')
```

```
y_pred = svm_model.predict(X_test)
print(f'Accuracy: {accuracy_score(y_test, y_pred) * 100:.2f}%')
print('\nClassification Report:\n', classification_report(y_test, y_pred))
print('\nConfusion Matrix:\n', confusion_matrix(y_test, y_pred))
```

```
Accuracy: 98.90%

Classification Report:
               precision    recall  f1-score   support

           0       0.99      0.99      0.99      1121
           1       0.98      0.98      0.98       431

    accuracy                           0.99      1552
   macro avg       0.99      0.99      0.99      1552
weighted avg       0.99      0.99      0.99      1552


Confusion Matrix:
 [[1112    9]
 [   8  423]]
```

```
new_email = ["Congratulations! You've won a $1000 gift card. Click here to claim your prize."]
new_email_features = vectorizer.transform(new_email).toarray()
```

Fig. 5: Sample image showing accuracy of the naïve bayes algorithm

RESULTS

Our model employs several classifiers to verify and compare outcomes for more accuracy. Each classifier provides its own output, and the user can compare them to determine whether an email is "spam" or "ham." The outputs are displayed in graphs and tables for easy comprehension. We trained our model using a dataset from Kaggle named "spam.csv" and tested it using a new dataset named "emails.csv" that the model had not previously seen.

Comparison of all algorithms:

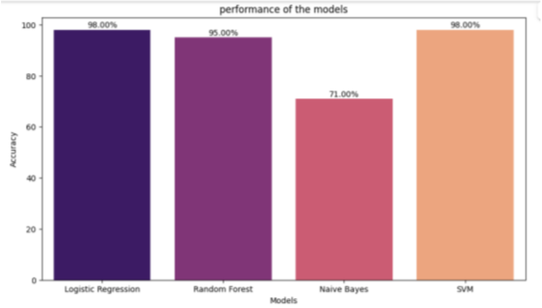


Fig.6: Comparison of all algorithms

Future Scope

The future horizon for this project is wide and exciting. One possibility is to work on the implementation of deep learning methods, e.g., CNNs and RNNs, to enhance spam detection accuracy. Another research possibility is to work on a system that can detect spam messages across languages, e.g., including non-Latin script languages. In addition, research into the application of transfer learning, image-based spam detection, and real-time spam detection can also bring about significant contributions. Also, research into the application of explainable AI, collaborative filtering, and domain adaptation can offer worthwhile insights and enhance the resilience of spam detection models. Last but not least, integrating human-in-the-loop approaches, like active learning and crowd sourcing, can assist further in enhancing the precision and efficiency of spam detection systems.

CONCLUSION

This research paper proposes a robust study on spam mail detection using machine learning techniques. The proposed system leverages natural language processing and machine learning algorithms in conjunction with each other to efficiently detect and categorize spam emails. Experimental results validate the effectiveness of the proposed system with high accuracy for spam email detection. In this paper we use four algorithms Logistic Regression, Random Forest, Naïve Bayes and SVM. Among these algorithms Logistic Regression produces the best results.

Reference

Mrs. Anitha Reddy, Kanthala Harivardhan Reddy, A. Abhishek, Myana Manish, G. ViswaSaiDattu, Noor Mohammad Ansari. (2023) “Email Spam Detection Using Machine Learning” Sreyas Institute of Engineering and Technology, 10(1) 2658-266

Darshana Chaudhari, Deveshri Kolambe, Rajashri Patil, Sachin Puranik. (2022) “Email Spam Detection Using Machine Learning And Python”, SSBT’s College of Engineering and Technology, Bambhori, Jalgaon, India.

B. Uday Reddy, S. Nagasai Tej, Md. Shoheb, Dr. Krishna Samalla, Y. Sreenivasulu. (2023) “Spam Mail Prediction Using Machine Learning” Sreenidhi Institute of Science and Technology, Ghatkesar, Hyderabad, India

Manu Garg, Parveen, Muskan Gupta, Ojasvi. (2022) “Email Spam Detection Using Logistic Regression” Meerut Institute of Engineering and Technology, Meerut

Thashina Sultana, K A Sapnaz, Fathima Sana, Mrs. JamedarNajath. (2022) “Email based Spam Detection” Yenepoya Institute of Technology Moodbidri, India

Pooja Malhotra, Sanjay Kumar Malik. (2021) “Spam Email Detection using Machine Learning and Deep Learning Techniques” USIC&T, GGSIPU

Nikhil Kumar, Sanket Sonowal, Nishant. (2020) “Email Spam Detection Using Machine Learning Algorithms” Delhi Technological University New Delhi, India

Sabah Mohammed, Osama Mohammed, Jinan Fiaidhi, Simon Fong and Tai hoon Kim. (2020) “Classifying Unsolicited Bulk Email (UBE) using

Python Machine Learning Techniques" Lakehead University, Ontario, Canada

Olubodunde Stephen Agboola. (2020) "Spam Detection Using Machine Learning" Louisiana State University Patrick F. Taylor Hall, Baton Rouge, LA 70808

Panem Charanarur, Harsh Jain, G. SrinivasaRao, Debabrata Samanta, Sandeep Singh Sengar, Chaminda Thushara Hewage. (2023) "Machine Learning Based Spam Mail Detector"

Asma Bibi, Rasia Latif, Samina Khalid, Waqas Ahmed, Raja Ahtsham Shabir, Tehmina Shahryar (2020) "Spam Mail Scanning Using Machine Learning Algorithm" Mirpur University of Science and Technology, Pakistan.

Alanazi Rayan (2022)" Analysis of e-Mail Spam Detection Using a Novel Machine Learning-Based Hybrid Bagging Technique" Jouf University, Sakaka, Saudi Arabia.