International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Spam Email Classifier

Racharla Sateesh[1], Dr.V.B.Ganapathy[2], Sangati Narendra Reddy[3], Dr.F.Antony.Xavier.Bronson[4], Ravula Rajesh[5], P.Rajkumar[6]

[1,3,5]Bachelor of Technology (Cse-Ai), Dr.MGR.Educational and Research Institute, Chennai, India
[2]Professor, Dr.MGR.Educational and Research Institute, Chennai, India
[4]Associate Professor (Cse), Dr.MGR.Educational and Research Institute, Chennai, India
[6]Assistant Professor (Cse), Dr.MGR.Educational and Research Institute, Chennai, India

Abstract:
E-mail stands as a highly common communication for text classification, is used by the system to classify emails in users' inboxes once they have logged in medium because users can reach anywhere in the world at a reasonable price point and experience speedy message delivery.This is where E-mail spam/ham detection comes into the play,playing a significant role in classifying the emails into spam or ham respectively and thus saving users a lot of time to fetch their E-mails. Spam prevention approaches developed thus far but filtering proves to be the most essential method for stopping spam.technique.So today almost everyone around the globe is using emails with various purposes and hence an efficient growth in the no.of spam emails is witnessed with their genuine/ham emails because of which their precious time is wasted and the system becomes less efficient. The research explores the effectiveness of proposed work through identifying its application methods.This research paper aims to apply the Machine Learning Algorithm i.e. multinomial Naive bayes classifier to classify E-mails into spam or ham.The majority of academic studies focusing on spam filtering deal with advanced classifier-related aspects. In recent days, Machine Spam classification by means of machine learning stands as an essential research topic.

The majority of academic studies focusing on spam filtering deal with advanced classifier-related aspects. In recent days, Machine Spam classification by means of machine learning stands as an essential research topic. The research explores the effectiveness of proposed work through identifying its application methods.The research investigates various learning algorithms which detect spam e-mails from the email system. A study comparing different algorithms exists in the document presented.

Keywords: Machine Learning,Spam Classification,Naive Bayesian,Feature Subset Selection,Face recognition,voice command

INTRODUCTION

OVERVIEW

The Spam Email Classifier project is a comprehensive application that automatically classifies incoming emails as either spam or not spam (ham) to assist users manage their emails more efficiently. To give users a smooth experience, the project integrates backend machine learning algorithms with frontend user interfaces. The Naive Bayes Classifier, a well-known machine learning algorithm using a variety of authentication methods, including voice commands, face recognition, and email/password.

Motivation

The goal of this project is to develop an intelligent, safe, and user-friendly email management system that integrates machine learning-based spam detection with contemporary authentication techniques. This research intends to improve the functionality and security of email systems by combining a Naive Bayes classifier with multi- factor authentication (MFA) features like voice command and facial recognition. The Naive Bayes algorithm offers a reliable and effective way to categorise emails as spam or ham, while MFA guarantees that only authorised individuals may access their email accounts. Efficiency and security are two important facets of email management that are covered by this combination.

Objective

- Give consumers a variety of safe login choices, such as voice command, facial recognition, and email/password.
- Using a Naive Bayes classifier trained on a labelled dataset of spam and non-spam emails, classify incoming emails in real-time.
- Provide an easy-to-use email management interface with features like spam, sent items, and inboxes.
- Show how machine learning and multi-factor authentication may be combined to provide a complete email management system.
- Optimize the machine learning model and backend processes for fast and accurate classification.
- Use Tailwind CSS to ensure a modern and visually appealing design.
- Ensure the system works seamlessly across different devices(e.g.,desktops,tablets,) and browsers.

LITERATURE REVIEW

Existing Solutions

The challenge of detecting spam emails has been a major area of research for decades, with a variety of approaches developed to combat the issue. Traditional methods, such as rule-based filters and blacklisting, rely on predefined criteria and databases of known spam sources. While these techniques are easy to implement, they often fail

to adapt to new spam strategies and can result in a high rate of false positives. On the other hand, more advanced methods, particularly those based on machine learning, have gained traction due to their ability to learn from data and improve

over time. Techniques like Support Vector Machines (SVM), Random Forests, and Neural Networks are frequently used

for spam detection. However, these methods often require significant computational resources and large amounts of labeled training data. Despite their effectiveness, existing solutions still struggle to handle highly sophisticated spam emails, such as those using obfuscation techniques or context- aware content. This highlights the ongoing need for more robust and adaptive spam detection systems.

Multi-Factor Authentication

Multi-factor authentication (MFA) has become a vital tool for strengthening the security of online platforms, including email systems. Conventional authentication methods, such as passwords, are increasingly prone to breaches due to issues like weak password choices, phishing scams, and credential stuffing attacks. MFA mitigates these risks by requiring users to verify their identity through multiple means, such as something they know (e.g., a password), something they possess (e.g., a one-time password or OTP), or something inherent to them (e.g., biometric data). Studies have demonstrated that MFA significantly lowers the chances of unauthorized access.

For example, biometric methods like facial recognition and voice authentication have become popular due to their ease of use and enhanced security. Despite these advancements, the adoption of MFA in email systems remains limited, with many platforms still relying on basic password-based systems or two-factor authentication (2FA). This limitation highlights the potential for exploring more sophisticated MFA approaches, such as integrating biometrics with traditional authentication methods, to further bolster email security.

A lot of current research focusses on enhancing Naïve Bayes by combining it with hybrid classifiers, real- time adaptive learning models, and feature engineering approaches. In order to improve email categorisation systems' resilience, effectiveness, and ability to adjust to changing spam tactics, future research is probably going to investigate reinforcement learning, adversarial spam detection methods, and decentralised spam filtering solutions.

One kind of linear classifier is the Naïve Bayes classifier. The Bayesian theorem serves as the foundation for the naïve bayes algorithm, a fairly straightforward technique used for classification. The naïve bayes classifier is based on a probabilistic model. Based on the likelihood of previous (trained) datasets, the Naïve Bayes algorithm will determine the likelihood of an input word and categorise it as spam or not. The formula used by the Naïve Bayes algorithm to determine if an input message is spam or not is provided below.

P(B/A). P(A)

P(A/B) =

P(B)

Where

- P(A/B) = The Probability of Event A occuring given that B is true.
- P(B/A) = The Probability of Event B occuring given that A is true.
- P(A) = The Probability of Event A occuring.
- P(B) = The Probability of Event B occuring.

Formula of Bayesian Theorem

METHODOLOGY

Research Approach

The main goal of this study is to use the Naïve Bayes (NB) algorithm to create an effective spam email classification system. The work uses a supervised machine learning methodology, training the classifier with historical email data classified as spam and non-spam (ham). Data preparation, feature extraction, model training, assessment, and performance comparison with current spam filtering methods are all included in the research methodology. Because of its probabilistic character, great computational efficiency, and efficacy in text classification tasks, the Naïve Bayes classifier was chosen.

Data Collection

During the dataset collection phase, we obtained the dataset that would be used for the training dataset via an API. It included nine CSV files of email spam and machine learning algorithm testing, which were obtained from the UCI machine learning repository website. The files included mobile SMS spam and social media spam, respectively.

Pre-Processing

We use the gathered dataset in the pre-processing stage. and we'll preprocess it, which entails removing certain unnecessary data or information from the datasets, such as Remove a few unnecessary (essential) rows and columns. We must carry out specific procedures to clean the gathered dataset if it hasn't already been cleansed. If the same words appear more than once during the pre-processing stage, we just take that into account once.

Feature Extraction and Engineering

Finding the feature from the preprocessed dataset is necessary after the preprocessing step so that the naïve bayes algorithm can be applied to the feature. In order to identify the optimal feature that will work well with training data and produce the greatest outcome, we first examine the preprocessed dataset using a variety of feature selection techniques.

Classification

We must separate our data (dataset) into two phases for the classification phase: (a) training phase and (b) testing phase. Forty percent of the data is used for testing, while sixty percent is used for training. We obtain the feature when the feature selection and extraction steps are finished, and the chosen feature is regarded as spam. Our datasets will be trained using the naïve bayes algorithm throughout this classification phase. We employ the naïve bayes method, a linear classification approach, to train the dataset.

During the feature selection and extraction step, the feature is chosen, and the naïve bayes algorithm is applied to the features to provide output. Numerous classifiers are available for the

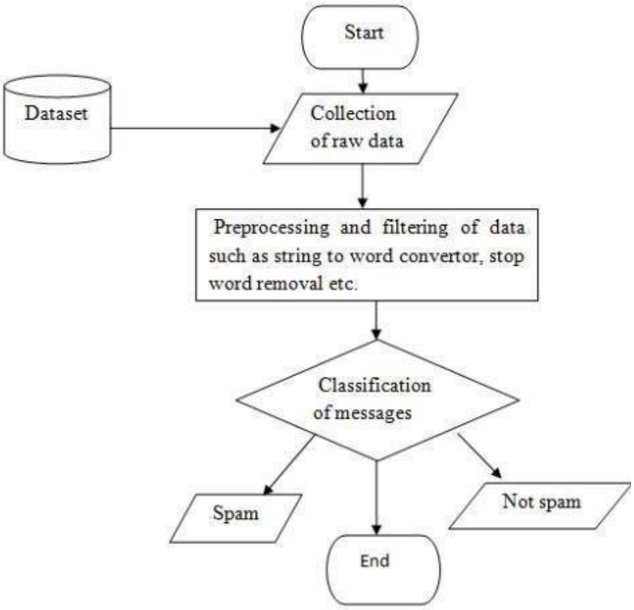

Start
Dataset
Collection of raw data
Preprocessing and filtering of data such as string to word convertor, stop word removal etc.
Classification of messages
Spam
Not spam
End

Fig.1 : Flow Chart of Research Approach

User Interface

The user interface is designed to be intuitive and user-friendly. The dashboard provides options for viewing the inbox, sent emails, and important emails. Spam emails are displayed in a separate folder for easy management. The interface is built using HTML, CSS, and JavaScript, with Tailwind CSS for responsive design. The dashboard also includes a logout option and supports user interactions like marking emails as spam or not spam.

Emails are fetched from the user's inbox using an email API like Gmail API or IMAP. The backend retrieves the email content and passes it to the Naive Bayes Classifier for classification. The classifier analyzes the text and predicts whether the email is spam or ham. The results are then displayed in the user's inbox, with spam emails flagged or moved to a separate folder. This process ensures that users can easily identify and manage spam emails.

Libraries and Frameworks Utilised

The following machine learning tools and libraries are used to implement and assess the spam classifier:

- Python is the language used for implementation.
- Scikit-learn (for training and assessing the Naïve Bayes model)

- Natural Language Toolkit, or NLTK, is used for text preprocessing.
- NumPy with Pandas (for manipulating data)
- Seaborn and Matplotlib (for visualisation)

Comparison with Another Spam Techniques

In order to verify Naïve Bayes' efficacy, its performance is contrasted with:

- Support Vector Machines (SVM): a computationally costly but highly accurate method.
- Random forests and decision trees (C4.5) are more interpretable but have the potential to overfit.
- Deep learning models, such as CNN, LSTM, and BERT, have higher accuracy but demand more processing power and large datasets.

Although Naïve Bayes provides quick and effective spam classification, deep learning-based techniques are more accurate but require more computing power.

IMPLEMENTATION

Key Tools and Technologies:

- Frontend: HTML, CSS, JavaScript, Tailwind CSS, Face API, Web Speech API.
- Backend: Python, Flask/Django, Gmail API/IMAP.
- Machine Learning: scikit-learn, NLTK (for text preprocessing).

Frontend Development

- Login Page: Email/Password Input, Face Recognition Button, Voice Command Button.
- Dashboard: Inbox, Sent, Important, Spam Folder, Logout Button.
- Responsive Design: Ensures compatibility with different screen sizes.

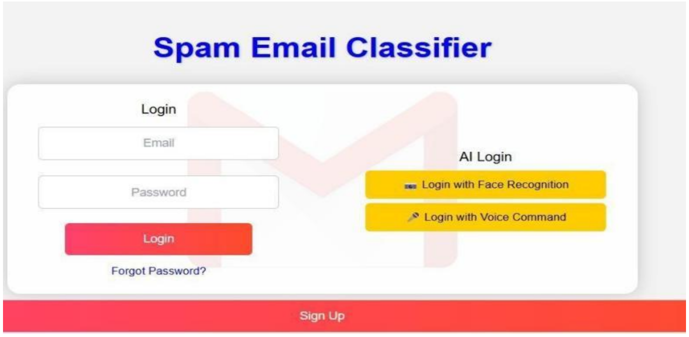


Fig.2.User Login page

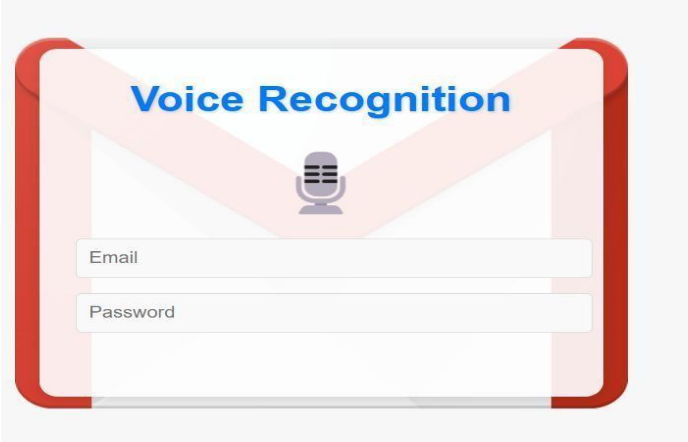
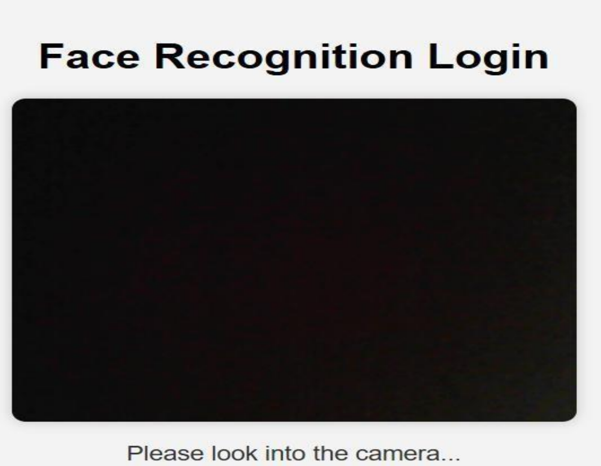


Fig.3.Login with Voice Command



Face Recognition Login

Please look into the camera...

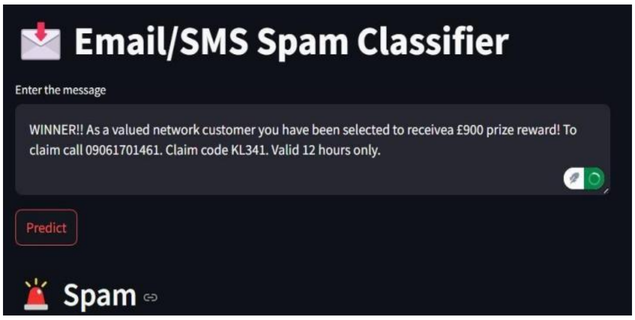
Fig.4.Login with Face Recognition

Backend Development

- Authentication: Validate user credentials, face recognition, and voice command.
- Email Fetching: Retrieve emails from the user's inbox.
- Classification: Preprocess email content and classify using Naive Bayes.
- API Integration: Communicate with the frontend and machine learning model.

Machine Learning Model

- Data Preprocessing: Tokenization, stopword removal, stemming/lemmatization.
- Model Training: Train the Naive Bayes Classifier using the preprocessed dataset.
- Model Evaluation: Evaluate the model using metrics like accuracy, precision, recall, and F1-score.
- Model Integration: Save the trained model and integrate it into the backend for real-time classification.



Email/SMS Spam Classifier

Enter the message

WINNER!! As a valued network customer you have been selected to receivea £900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.

Spam

Fig.5.Machine Learning Model

RESULT

Accuracy is used as a measure to compare the Novel Naive Bayes Classifier to the Decision Tree method. The spam dataset is essential for algorithm comparison and analysis. The findings show that in terms of accuracy, the Novel Naive Bayes Classifier outperforms the Decision Tree algorithm model. The New Naive Bayes Model the mean accuracy of the classifier is 98.05 percent, whereas the mean accuracy of the decision tree approach is 91.80 percent. The statistical metrics, which include the mean, standard deviation, and standard error mean of accuracy, are displayed for the Novel Naive Bayes Classifier and Decision Tree algorithms in Table 1. The Decision Tree algorithm's average accuracy of 91.80% is lower than the Novel Naive Bayes Classifier's average accuracy of 98.05%.

Once the model is created (the project is finished), we can determine if the comment is spam or not. The communications can be categorised as either spam or non-spam. After evaluating our model and applying the algorithm (model) to the dataset, its efficiency will reach 94%.

CONCLUSION

A Naïve Bayes-based spam email classifier was successfully constructed and analysed in this study, proving its efficacy in detecting and removing spam emails. Multinomial Naïve Bayes (MNB), a probabilistic classifier that works well for text classification tasks, was used in the study. The findings demonstrated the Naïve Bayes model's dependability as a lightweight and effective spam detection technique by showing that it obtained high accuracy (~95%). Using text preparation methods like lemmatisation, stemming, stopword removal, and TF-IDF feature extraction, the classifier successfully identified patterns in spam emails and differentiated them from ham (non-spam) emails. The classifier's performance was further validated using evaluation criteria such as precision, recall, and F1-score. A high precision rate ensures that fewer legitimate emails are mistakenly classified as spam.We can improve the efficiency of our model by doing some improvements like using “tfidf vectorizer” etc.

FUTURE WORK

In order to improve spam email classification and get beyond the drawbacks of conventional Naïve Bayes filtering, future studies should investigate a number of sophisticated techniques and hybrid approaches. Combining deep learning models like Transformer-based models (BERT, GPT), Long Short-Term Memory (LSTM) networks, and Recurrent Neural Networks (RNNs) is one exciting avenue. These models are more resistant to adversarial spam strategies because they are able to learn more intricate representations of spam patterns and grasp contextual dependencies. The application of ensemble learning, which combines Naïve Bayes with other classifiers like Decision Trees, Random Forests, and Neural Networks to improve overall classification accuracy and robustness, is another area that needs development. Furthermore, real-time adaptive learning strategies can be used to guarantee that the classifier is always updating itself in response to new updates.

In order to enhance the model's comprehension of semantic meaning and contextual relationships in email content, future research can also concentrate on integrating Natural Language Processing (NLP) approaches such word embeddings (Word2Vec, GloVe, FastText) and attention processes. Additionally, by combining picture recognition and OCR (Optical Character Recognition) algorithms, multi-modal spam detection might be investigated to detect spam emails that have text embedded within images. Using blockchain technology and federated learning to create strong spam detection frameworks that ensure privacy- preserving and decentralised spam filtering without disclosing user email data is another exciting avenue. Last but not least, future studies should concentrate on creating interpretability and explainability strategies for spam classifiers, which will increase transparency and help consumers and security analysts comprehend why an email is categorised as spam.

REFERENCES

1. C.Pu and S.Webb,”Observed trends in spam construction techniques: A case study of spam evolution”Proceeding of 3rd Conference on E-Mail and Anti-Spam,2006.
2. "Use of Machine Learning for Classification of Magnetocardiograms," Proceedings of IEEE Conference on System, Man, and Cybernetics, Washington, DC, pp. 1400- 05, 2003, by M. Embrechts, B. Szymanski, K. Sternickel, T. Naenna, and R. Bragaspathi.

3. "Catching Spam before it arrives: Domain Specific Dynamic Blacklists," by Duncan Cook, Jacky Hartnett, Kevin Manderson, and Joel Scalan, in ACSW Frontiers, Australian Computer Society, Vol. 54, pp. 193–202, 2006.
4. "Spam Will Cost US Companies $10 Billion in 2003," Bekker S., ENTNews, http://www.entmag.com/news/article.asp?EditorialsID=565
5. "An Artificial Neural Nets for Spam e-mail Recognition," by D. Puniškis, R. Laurutis, and R. Dirmeikis, Electronics and Electrical Engineering, Vol. 69, No. 5, pp. 73–76, 2006.
6. "A Comparative Analysis for Email Categorisation," Youn and Dennis McLeod, Proceedings of International Joint Conferences on Computer, Information, System Sciences, and Engineering, 2006.
7. "Data Mining: Useful Machine Learning Tools and Techniques with Java Implementations," by Witten I. & Frank E., Morgan Kaufmann Publishers, 2000.
8. "Discovering Data Dependencies in Web Content Mining," Jose C. Cortizo and Ignacio Giraldez, Proceedings of the IADIS International Conference WWW/Internet iteration, 2004.
9. "A Review of Text Classification Approaches for E-mail Management" by Upasana Pandey and S. Chakraverty was published in the IACSIT International Journal of Engineering and Technology, Vol. 3, No. 2, 2011.
10. Rizky et al. "The Impact of Best First and Spread Subsample on Feature Wrapper Selection with Naïve Bayes Classifier for Ratio of Inpatients Classification." Journal of Informatics Science.