Journal of Science, Computing and Engineering Research (JSCER)
Volume-9, Issue-4, April 2026.
DOI: https://doi.org/10.46379/jscer.2024.0709

EMAIL SPAM CLASSIFIER

[1]J.R.Arun Kumar, [2]Aman Chauhan, [3]Hitesh Sharma, [4]Lokesh Kumar

[1] Professor, Department of AI&DS, Modern Institute of Technology and Research Centre, Rajasthan, India.

[2,3,4,5] UG Student, Department of CSE, Modern Institute of Technology and Research Centre, Rajasthan, India

Article Information

Received: 02 April 2026

Revised: 03 April 2026

Accepted: 04 April 2026

Published : 05 April 2026

Corresponding Author:

Aman Chauhan

***Abstract*—** The **Email Spam Classifier** project aims to develop a machine learning-based system capable of classifying emails into spam and non-spam (ham) categories. The system uses Natural Language Processing (NLP) techniques to analyze the content of email messages and extract meaningful features from the text data. Various machine learning algorithms such as Naïve Bayes, Support Vector Machine (SVM), and Logistic Regression can be applied to train the model using a dataset of labeled email messages.

Keywords: application; LSTM; machine learning; natural language processing; prediction; preprocessing; react; tokenization

Citation: J.R.Arun Kumar, Aman Chauhan, Hitesh Sharma, Lokesh Kumar, "Email Spam Classifier", Journal of Science, Computing and Engineering Research, 9(4), April 2026

I. INTRODUCTION

Email has become one of the most widely used communication tools in the modern digital world. People use email services for personal communication, business transactions, online registrations, and information sharing. However, the increasing use of email has also led to a significant rise in unwanted and harmful messages known as spam emails. Spam emails often contain advertisements, phishing links, fraudulent offers, and malicious attachments that can pose serious security risks to users and organizations. As a result, managing and filtering spam emails has become an important challenge in maintaining secure and efficient communication systems.

Traditional spam filtering methods relied on simple rule-based techniques such as keyword matching, blacklisting suspicious email addresses, and manual filtering. Although these methods were effective in the early stages of email communication, they are no longer sufficient to handle the growing volume and complexity of spam messages. Modern spammers continuously change their strategies by using new words, symbols, and message formats to bypass traditional filters. Therefore, there is a need for more advanced and intelligent systems that can automatically learn from data and adapt to new spam patterns. The **Email Spam Classifier** project focuses on developing an automated system that uses machine learning and Natural Language Processing (NLP) techniques to classify emails as spam or non-spam (ham). The system processes email text data through several steps, including data preprocessing, tokenization, removal of stop words, and feature extraction. Machine learning algorithms such as Naïve Bayes, Support Vector Machine (SVM), or Logistic Regression are then used to train the classification model using a dataset

II. PROBLEM STATEMENT

Existing Traditional spam detection methods rely on simple rule-based systems such as keyword filtering, sender blacklisting, and manual sorting of emails. While these methods were effective in the past, they are no longer sufficient to handle the increasing volume and complexity of modern spam messages. Spammers continuously modify their techniques by using new words, symbols, and message structures to bypass existing filters. As a result, many spam emails still reach users' inboxes, leading to reduced productivity, security risks, and increased storage and network usage.

Another major challenge is the need for an automated system that can accurately identify spam emails without human intervention. Manual filtering of emails is time-consuming and inefficient, especially when dealing with large volumes of messages. Therefore, there is a need for a more advanced and

intelligent solution that can analyze email content, learn from past data, and continuously improve its performance over time.

The main problem addressed in this project is to design and develop a reliable **Email Spam Classifier** system using machine learning and Natural Language Processing (NLP) techniques. The system should be capable of automatically classifying incoming emails as spam or non-spam with high accuracy,

III. PROPOSED METHOD

The proposed method for the **Email Spam Classifier** project focuses on developing an intelligent system that can automatically detect and classify emails as spam or non-spam (ham) using Machine Learning and Natural Language Processing (NLP) techniques. The system follows a structured process that includes data collection, preprocessing, feature extraction, model training, and prediction

A. Data Collection

The first step in the proposed method is collecting a dataset containing labeled email messages. The dataset includes both spam and legitimate (ham) emails. These emails are used to train and test the machine learning model.

B. Data Preprocessing

Data preprocessing is an important step in preparing the email data for analysis. In this step, unnecessary information such as punctuation, special characters, and duplicate words is removed. The text is converted into lowercase letters to maintain consistency. Stop words (commonly used words such as "the", "is", and "and") are also removed .

C. Tokenization

Tokenization is the process of breaking down email text into smaller units called tokens or words. Each email message is divided into individual words so that the system can analyze the content more effectively. Tokenization is a key step in Natural Language Processin.

D. Feature Extraction

Feature extraction converts text data into numerical values that can be processed by machine learning algorithms. Techniques such as Bag of Words (BoW) or Term Frequency–Inverse Document Frequency (TF-IDF) are used to represent text data in a structured format.

E. Model Training

In this step, a machine learning algorithm such as Naïve Bayes, Support Vector Machine (SVM), or Logistic Regression is used to train the classification model. The model learns from the labeled dataset and identifies patterns that distinguish spam emails from legitimate emails. T

F. Model Testing and Evaluation

After training, the model is tested using a separate dataset to evaluate its performance. Evaluation metrics such as accuracy, precision, recall, and F1-score are used to measure the effectiveness of the spam classifier.

G. Prediction and Classification

Once the model is trained and tested, it is used to classify new incoming emails. The system analyzes the content of each email and predicts whether it is spam or non-spam.

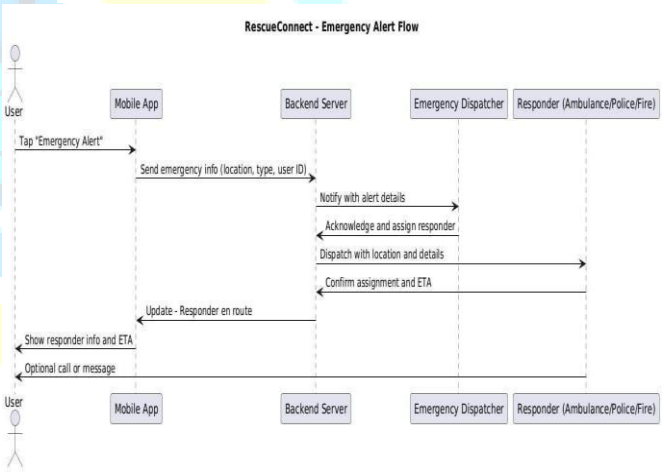

Fig. 1. Proposed Work Model

IV. TECH STACK

A. Frontend Technologies

Frontend technologies are used to create a user-friendly interface where users can input email text and view classification results. Technologies such as HTML are used to structure the web pages, CSS is used to design and style the user interface, and JavaScript is used to add interactivity to the web application. In some implementations, React may also be used to develop a modern, responsive, and interactive user interface that enhances the overall user experience

B. Backend Framework

The backend framework is responsible for handling the logic of the application and connecting the machine learning model to the user interface. Frameworks such as Flask or Django are commonly used to build the backend server and integrate the machine learning model with the web application. The backend receives email input from the user, processes the data using the trained model, and returns the prediction result indicating whether the email is spam or not spam.

C. Programming Language:

Python is used as the primary programming language for developing the Email Spam Classifier system. It provides powerful libraries and tools for machine learning, data analysis, and Natural Language Processing (NLP). Python is easy to use, flexible, and widely supported in the field of data science and artificial intelligence, making it suitable for building intelligent systems that can analyze email content and classify messages accurately.

D. Machine Learning Libraries:

Various machine learning libraries are used to build and train the spam classification model in this project. Libraries such as Scikit-learn are used for implementing machine learning algorithms like Naïve Bayes, Logistic Regression, and Support Vector Machine (SVM), which help in predicting whether an email is spam or not. Pandas is used for data manipulation, cleaning, and preprocessing of email datasets, while NumPy is used for performing numerical operations and handling arrays efficiently.

E. Natural Language Processing (NLP):

Natural Language Processing techniques are applied to analyze and process the content of email messages effectively. NLP enables the system to understand human language and identify patterns in textual data. In this project, NLP techniques such as tokenization.

F. Development Tools:

Various development tools are used to support the coding, testing, and deployment of the Email Spam Classifier system. Visual Studio Code (VS Code) is commonly used as the code editor for writing and managing project files due to its simplicity and wide range of extensions. Jupyter Notebook is used for experimenting with machine learning models, performing data analysis, and visualizing results

.

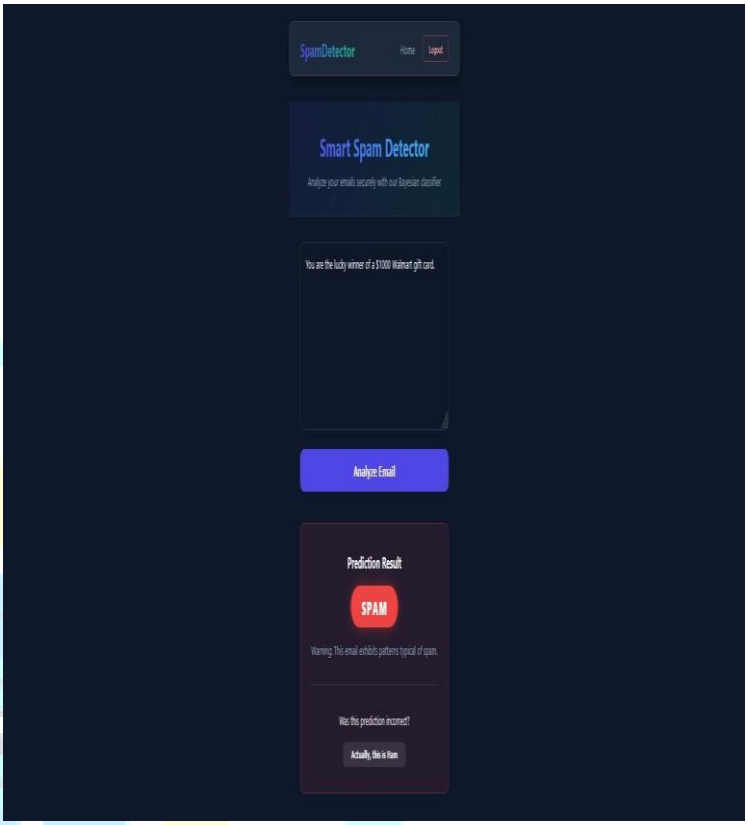

Fig. 2. Email Predictor

V. RESULTS

The results of the Email Spam Classifier project demonstrate the effectiveness of machine learning techniques in accurately identifying spam and non-spam emails. The system was trained using a dataset containing labeled email messages and tested using a separate set of data to evaluate its performance. After preprocessing the email text and applying feature extraction techniques, the machine learning model was able to learn patterns that distinguish spam emails from legitimate messages. The trained model successfully classified incoming emails with a high level of accuracy, reducing the number of unwanted messages reaching the user's inbox.

The performance of the spam classifier was evaluated using standard evaluation metrics such as accuracy, precision, recall, and F1-score. These metrics help measure how well the model performs in identifying spam emails correctly while minimizing false predictions. The results showed that the system achieved reliable classification performance, indicating that the proposed method is suitable for real-world email filtering applications. The classifier was able to detect most

spam emails efficiently while maintaining a low error rate for legitimate emails.

VI. CONCLUSION

In conclusion, the Email Spam Classifier project successfully demonstrates the application of machine learning and Natural Language Processing techniques in detecting and filtering unwanted email messages. The system was designed to automatically classify emails as spam or non-spam based on their content, helping users avoid harmful or unnecessary messages. By using data preprocessing, tokenization, feature extraction, and machine learning algorithms, the model was able to learn patterns from the dataset and make accurate predictions on new email messages. The developed system improves the efficiency of email communication by reducing the number of spam messages reaching the user's inbox and enhancing overall email security. Compared to traditional rule-based filtering methods, the machine learning-based approach provides better adaptability and accuracy, as it can continuously learn from new data and adjust to changing spam patterns. This makes the system mor

REFERENCES

[1] Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers.

[2] Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media

[3] Jurafsky, D., & Martin, J. H. (2020). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson Education.

[4] Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1), 1–47.

[5] Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., & Spyropoulos, C. D. (2000). An Experimental Comparison of Naïve Bayesian and Keyword-Based Anti-Spam Filtering. *Proceedings of the Workshop on Machine Learning in the New Information Age*.

[6] Almeida, T. A., Gómez Hidalgo, J. M., & Yamakami, A. (2011). Contributions to the Study of SMS Spam Filtering: New Collection and Results. *Proceedings of the 11th ACM Symposium on Document Engineering*.

[7] Mc Callum, A., & Nigam, K. (1998). A Comparison of Event Models for Naïve Bayes Text Classification. *AAAI Workshop on Learning for Text Categorization*.

[8] Ramos, J. (2003). Using TF-IDF to Determine Word Relevance in Document Queries. *Proceedings of the First Instructional Conference on Machine Learning*.

[9] HealthIT.gov, "Privacy and security considerations for digital mental health applications," U.S. Department of Health & Human Services, 2022.

[10] Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian Approach to Filtering Junk E-Mail. *Proceedings of the AAAI Workshop on Learning for Text Categorization*.