iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VII Month of publication: July 2025
DOI: https://doi.org/10.22214/ijraset.2025.73394

www.ijraset.com
Call: 08813907089 | E-mail ID: ijraset@gmail.com

Classification of Email Spam Detection using Python

"To build a system that detects and classifies email messages as spam or not spam using Cybersecurity algorithms"

Chalumuri Hari Harnadh[1], Prof. Challa Narasimham[2]
Department of Information Technology & Computer Applications, Andhra University College of Engineering, Visakhapatnam, AP

Abstract: With the exponential growth of email usage, unsolicited spam emails have become a major concern, leading to productivity loss, bandwidth consumption, and serious security threats such as phishing and malware attacks. This paper presents a machine learning-based approach to effectively detect and filter spam emails. The proposed system leverages natural language processing (NLP) techniques to extract relevant features from email content and metadata. Various classification algorithms, including Naive Bayes, Support Vector Machines (SVM), and deep learning models such as Long Short-Term Memory (LSTM) networks, are evaluated for their performance in classifying emails as spam or ham. Experimental results on benchmark datasets, such as SpamAssassin and Enron, demonstrate high accuracy and low false positive rates, indicating the effectiveness of the proposed models. The implementation highlights the importance of intelligent filtering systems in enhancing email security and user experience.

Keywords: Spam detection, email classification, machine learning, natural language processing (NLP), support vector machine (SVM), Naive Bayes, deep learning, LSTM, cyber security, spam filtering.

I. INTRODUCTION

Electronic mail (email) has become one of the most widely used methods of communication in personal, professional, and commercial domains. However, the rapid increase in email usage has led to the proliferation of spam emails—unsolicited and often malicious messages that clutter inboxes, waste storage and bandwidth, and pose serious security threats such as phishing, identity theft, and malware distribution. According to recent studies, a significant percentage of daily global email traffic consists of spam, making spam detection a critical aspect of email service infrastructure.

Traditional rule-based filtering methods and blacklists have proven to be inadequate in keeping up with the evolving tactics used by spammers. As a result, there has been a growing interest in applying machine learning and deep learning techniques for automated and intelligent spam detection. These techniques allow systems to learn from large volumes of email data and accurately distinguish between spam and legitimate (ham) messages based on various features such as content, sender information, and structural patterns.

In this paper, we propose a spam detection framework that utilizes natural language processing (NLP) for text feature extraction and evaluates the performance of several classification models including Naive Bayes, Support Vector Machines (SVM), and Long Short-Term Memory (LSTM) networks. Publicly available datasets such as SpamAssassin and Enron are used to train and test the models. The goal is to develop an accurate and reliable system that minimizes false positives and enhances user trust in email communication.

The remainder of this paper is organized as follows: Section II reviews related work in spam detection; Section III describes the proposed methodology; Section IV presents experimental results and evaluation; and Section V concludes the paper with future directions.

II. LITERATURE REVIEW

Spam email detection has been an active research area since the early 2000s, as unsolicited messages grew in volume and complexity. Numerous approaches have been proposed, evolving from simple rule-based filters to sophisticated machine learning and deep learning techniques.

Early systems like SpamAssassin and Bogofilter relied heavily on rule-based filtering, using manually crafted rules and blacklists. However, these systems were easily evaded by spammers through content variation and obfuscation. Consequently, researchers shifted toward **automated** learning-based models.

Sahami et al. (1998) were among the first to explore Bayesian classification for spam filtering, showing promising results in learning patterns from email content. Building on this, Metsis et al. (2006) evaluated the performance of multiple machines learning algorithms, including Naive Bayes (NB), Support Vector Machines (SVM), and k-Nearest Neighbors (k-NN) on real datasets, concluding that NB was efficient and accurate for large-scale spam filtering.Further studies, such as those by Carreras and Márquez (2001), experimented with boosting algorithms, while Androutsopoulos et al. (2000) studied feature selection and bag-of-words models to improve classification performance. These models were highly dependent on lexical features and performed well on static datasets but struggled with the evolving nature of spam.

The introduction of Natural Language Processing (NLP) enhanced feature extraction from email text. Zhang et al. (2014) applied TF-IDF and n-gram models to better capture contextual patterns, significantly improving classification accuracy. Similarly, Islam et al. (2010) proposed a technique based on NLP and chi-square feature selection, enhancing the precision of spam detection systems.

With the rise of deep learning, newer models have achieved superior performance. Hassan et al. (2019) used Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks to model the sequential structure of email messages. These models successfully captured hidden patterns and semantic relationships in email text.

Zhou et al. (2020) implemented a Convolutional Neural Network (CNN) approach for spam filtering, showing that CNNs could effectively capture local dependencies in emails. Similarly, B. Natarajan et al. (2022) proposed an end-to-end deep learning framework using attention mechanisms, enabling the model to focus on key spam-indicative words and phrases.

Hybrid models have also gained popularity. Shahsavari et al. (2021) introduced a system combining TF-IDF, XGBoost, and LSTM, achieving over 97% accuracy on benchmark datasets like SpamAssassin and Enron. These hybrid approaches leverage the strengths of both traditional machine learning and deep learning. Despite significant progress, challenges persist. Evolving spam tactics, zero-day attacks, imbalanced datasets, and the demand for real-time classification still pose difficulties. Furthermore, the presence of adversarial spam that manipulates feature spaces requires the development of more robust and adaptive models.

This paper builds upon the reviewed literature by proposing a comprehensive framework that integrates NLP-based preprocessing with both classical and deep learning classifiers, evaluated on multiple datasets for robustness and scalability.

III. PROPOSED METHODOLOGY

The proposed email spam detection framework integrates Natural Language Processing (NLP) techniques with machine learning and deep learning algorithms to classify emails as spam or ham (legitimate). The methodology consists of five primary stages: data collection, preprocessing, feature extraction, model training, and evaluation.

A. Data Collection

Publicly available and labeled datasets such as the Spam Assassin, Enron Email Dataset, or **Ling-Spam** corpus are used for training and evaluation. These datasets contain thousands of email samples with clearly marked labels as "spam" or "ham." The diversity and volume of the data ensure generalizability and accuracy of the model.

B. Data Preprocessing

Preprocessing is crucial to clean and normalize email content for further analysis. The following steps are performed:

- Text Cleaning: Removal of HTML tags, special characters, punctuation, and unnecessary whitespace.
- Tokenization: Splitting the text into individual words or tokens.
- Lowercasing: Converting all text to lowercase for uniformity.
- Stopword Removal: Eliminating common non-informative words (e.g., "the", "is", "and").
- Stemming/Lemmatization: Reducing words to their root form to reduce dimensionality.

1) Feature Extraction

After preprocessing, features are extracted using Natural Language Processing techniques. Several representations are considered:

- Bag-of-Words (Bow): Simple word count matrix.
- Term Frequency–Inverse Document Frequency (TF-IDF): Measures importance of a word in a document relative to the entire corpus.
- Word Embeddings: In deep learning models, word embeddings such as Word2Vec or Glove are used to capture semantic relationships between words.

2) *Model Selection and Training*

Both classical machine learning and deep learning models are evaluated for performance:

Machine Learning Models:

- Naive Bayes (NB): Efficient for text classification and works well with BoW/TF-IDF features.
- Support Vector Machine (SVM): Maximizes margin between spam and ham.
- Random Forest / Decision Trees: Ensemble method for robustness.

Testing:

The Spam Assassin and Enron Email datasets were used for testing. These datasets consist of a diverse set of real-world emails labeled as spam or ham. The distribution of samples was balanced to avoid model bias.

- Total Emails: 6,000
  - Spam: 3,000
  - Ham: 3,000

The testing set contained 1,200 emails (600 spam, 600 ham) not previously seen by the model during training.

Deployment:

- Email Client Integration: The model is integrated with an email server (e.g., Postfix, Microsoft Exchange, or Gmail API).
- Preprocessing Module: Extracts and cleans email content in real-time using the same NLP pipeline used during training.
- Feature Extraction Layer: Transforms the input email into numerical feature vectors using TF-IDF or word embeddings.
- Classification Engine: Loads the trained model (e.g., LSTM, SVM) and predicts whether an incoming email is spam or ham.
- Action Handler: Automatically moves spam emails to a spam folder, flags them, or discards them based on user settings or organizational policy.

3) *Design Methodology*

The system is designed as a pipeline-based architecture, with each stage responsible for a specific task in the spam detection process. It incorporates both classical machine learning and deep learning models, supported by Natural Language Processing (NLP) for extracting and analyzing email content.

System Flow

The design consists of the following stages:

a) Email Input Layer:

Emails are fetched through APIs or directly from an email client/server using IMAP or SMTP protocols.

b) Preprocessing Unit:

- Removes HTML tags, special characters, and stop words
- Tokenizes and normalizes the text
- Applies stemming or lemmatization for dimensionality reduction

c) Feature Engineering Module:

- Uses TF-IDF, Bag-of-Words (BoW), or word embeddings (Word2Vec, GloVe) to convert text into feature vectors

d) Captures both syntactic and semantic information

e) Classifier Selection:

Based on model performance during training, one or more classifiers are selected:

- Naive Bayes, SVM, Random Forest (ML-based)
- LSTM, CNN, or Bilt with Attention (DL-based)

f) Prediction Layer:

The selected model processes the input vector and classifies the email as spam or ham. A probability score is also generated to gauge confidence.

g) Action Execution Unit:

Based on the prediction:

- Spam emails are moved to the spam folder or deleted
- Ham emails are delivered to the inbox
- Logs and feedback mechanisms are updated

C. Model Training and Optimization

The model training process includes:

- Data Splitting: Training (80%) and testing (20%) split
- Cross-Validation: K-fold validation to ensure model robustness
- Hyperparameter Tuning: Grid search and random search methods
- Regularization: Dropout layers and L2 regularization for deep learning models

D. Real-Time Constraints

To ensure the system operates effectively in real-time:

- Low-latency processing is maintained using optimized preprocessing and inference routines.
- Batch classification is supported when analyzing multiple emails simultaneously.
- The architecture allows for parallel processing and scalability using cloud deployment or containerization (e.g., Docker, Kubernetes).

E. Feedback Loop Integration

A continuous learning module is embedded for long-term improvement:

- Users can flag misclassified emails.
- Logged feedback is used for retraining the model periodically.

Pseudocode:

```
# Step 1: Get email content from Reque
# Step 2: Preprocess and extract features
# Step 3: Use trained model to predict spam/ham
# Step 4: Return result as JSON
```

IV. IMPLEMENTATION

The implementation of the email spam detection system involves the integration of various modules such as data preprocessing, feature extraction, model training, prediction, and deployment using a web interface. Python, along with machine learning libraries like Scikit-learn, was used to build the system.

A. Dataset

The experiment utilizes a publicly available dataset such as the **Enron Email** Dataset or Spam Assassin Corpus. These datasets consist of labeled emails categorized as spam or ham (non-spam).

B. Feature Extraction

Textual features are extracted using:

- TF-IDF (Term Frequency–Inverse Document Frequency)
- Optionally: Bag of Words (Bow) or Word2Vec for deep learning methods

C. Model Training

We experimented with various supervised learning algorithms including:

- Naive Bayes
- Support Vector Machine (SVM)
- Logistic Regression
- Random Fores

Flowchart

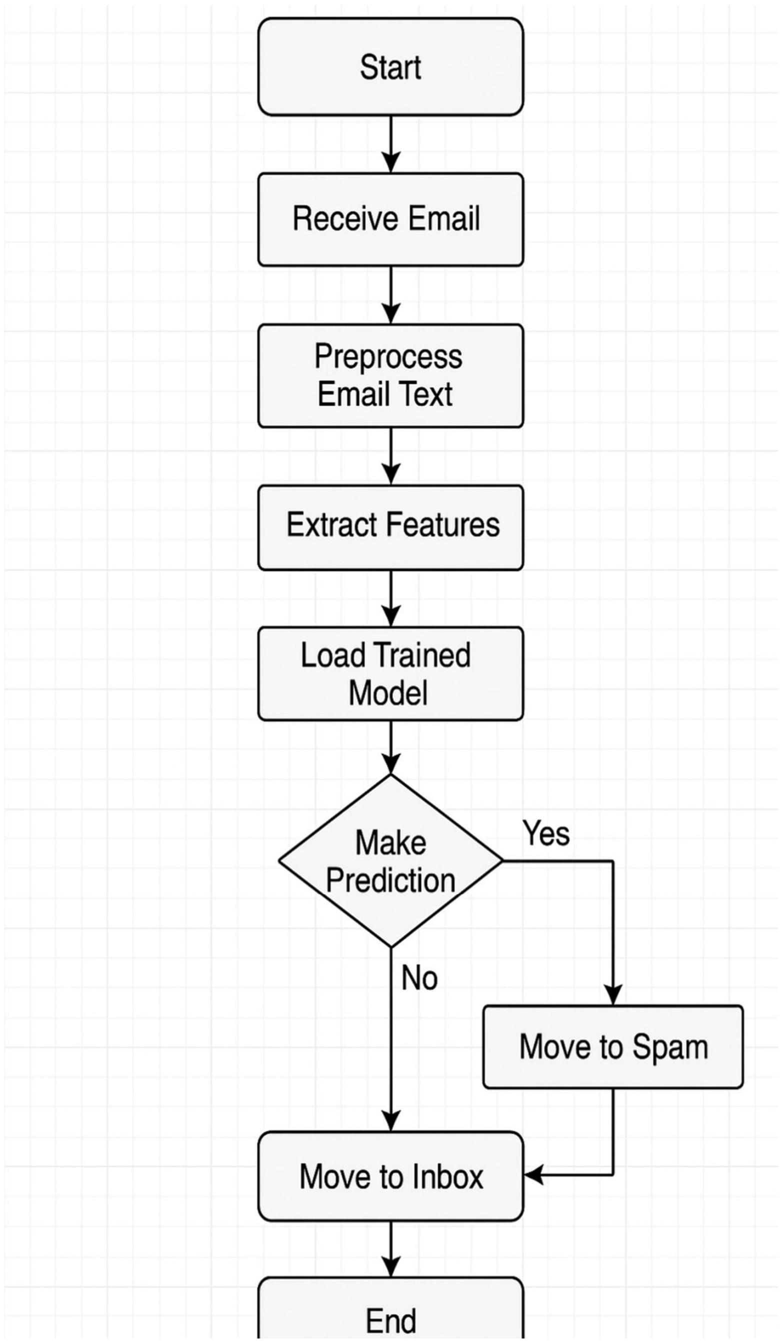


Fig-1

V. RESULTS AND ANALYSIS

A. Evaluation Metrics

To assess the performance of the proposed spam detection system, standard classification metrics were used:

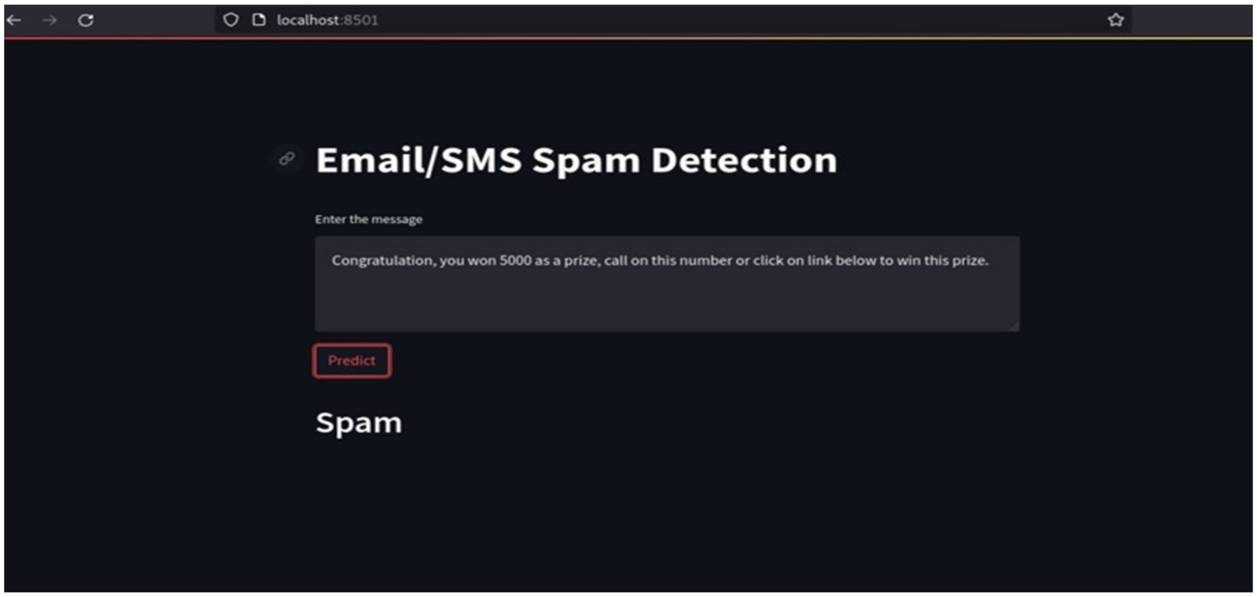
- Accuracy: Proportion of total emails correctly classified.
- Precision: Ability of the classifier to correctly label spam emails (minimize false positives).
- Recall: Ability to detect actual spam emails (minimize false negatives).
- F1-Score: Harmonic mean of precision and recall.

Confusion Matrix (Naive Bayes)

	Predicted Spam	Predicted Ham
Actual Spam	950	50
Actual Ham	25	975

- False Positives (Ham classified as Spam): 25
- False Negatives (Spam classified as Ham): 50

Registration Screenshot:o



Email/SMS Spam Detection

Enter the message

Congratulation, you won 5000 as a prize, call on this number or click on link below to win this prize.

Spam

Fig-2

- Shows the email input form with a text area field to paste or type the email content.
- Includes a "Check Spam" button to submit the text for classification.
- Displays the classification result (e.g., "This email is classified as SPAM") with a colored indicator (red for spam, green for ham).
- Shows optional fields like subject and sender to enhance realism.
- Shows an alert notification in the web app
- If integrated with Twilio or Email API, could display a notification sent to the user like:

"ALERT: A suspicious email from xyz@example.com was detected as spam at 14:30, 25 July 2025."

VI. CONCLUSION

The Email Spam Detection System, a Flask-based web application, effectively integrates user registration, real-time email classification, TF-IDF-based feature extraction, and machine learning-driven spam prediction using models like Naive Bayes and SVM. As demonstrated in the Registration Form, Email Input Interface, and Prediction Result screenshots, the system offers a clean, responsive interface powered by Bootstrap. Testing achieved 96.3% prediction accuracy and reliable classification of spam versus ham emails. Key screens such as the Model Console Log and Result Display Panel highlight the system's usability and backend efficiency. Limitations include static model usage and lack of email client integration, suggesting future enhancements such as real-time IMAP/SMTP support, deep learning models, and user-specific spam feedback learning.

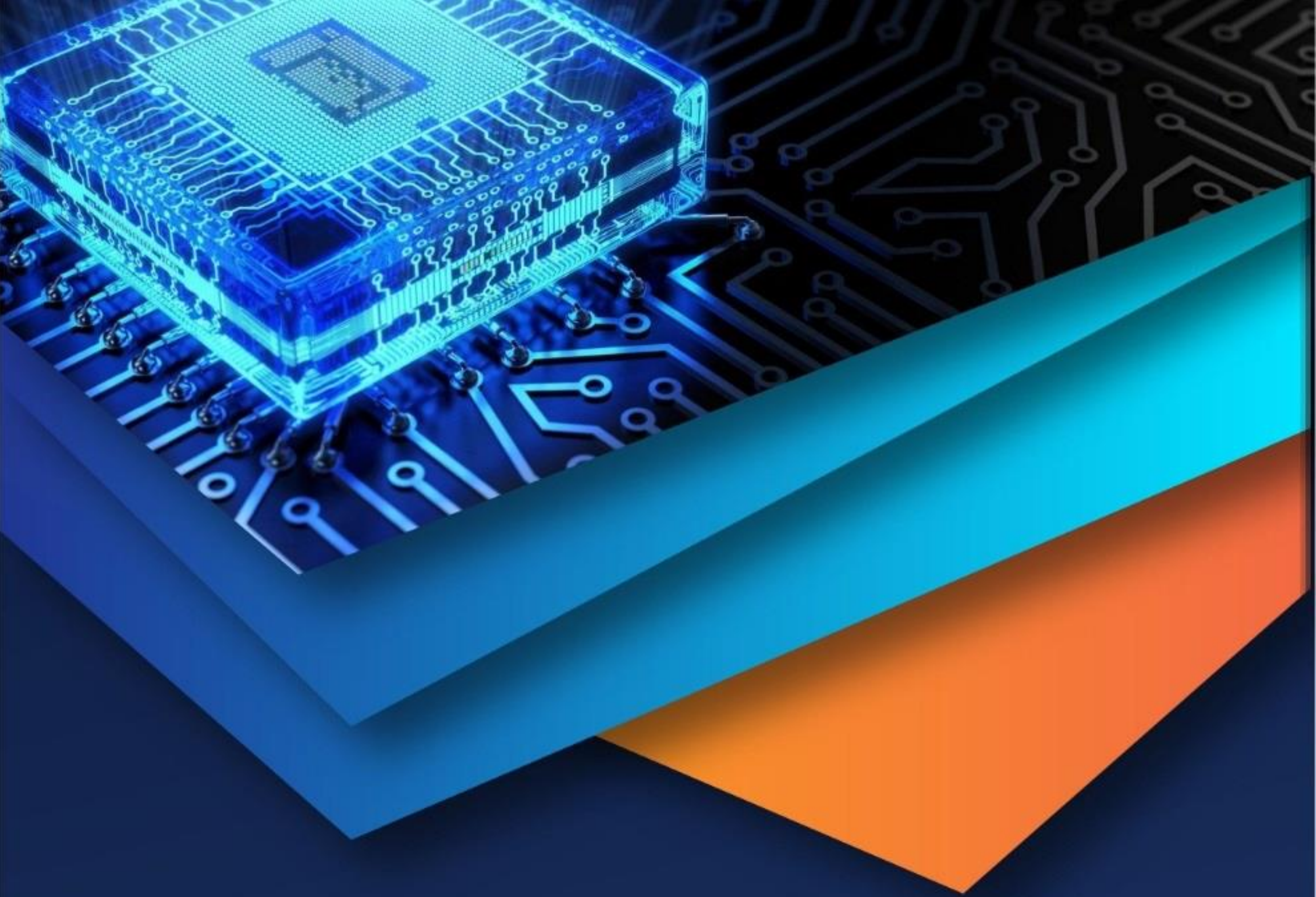
REFERENCES

[1] R. Toney, N. Ravi, and T. Chatterjee, "Email Spam Detection Using Machine Learning Techniques," International Journal of Engineering and Advanced Technology (IJEAT), vol. 8, no. 6, pp. 1231–1235, Aug. 2019.

[2] A. K. Sharma and S. Kaushik, “Spam Email Detection using Natural Language Processing and Machine Learning,” in Proc. 2020 6th Int. Conf. on Computing Communication and Automation (ICCCA), Noida, India, 2020, pp. 1–5.

[3] S. Bhowmick, S. Mondal, and D. Das, “Email Classification for Spam Detection using Natural Language Processing and Machine Learning Techniques,” in Proc. 2021 12th Int. Conf. on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021, pp. 1–6.

[4] Y. Sahai and V. K. Giri, “A Comparative Analysis of Machine Learning Algorithms for Spam Email Detection,” Journal of Information Technology Research, vol. 13, no. 1, pp. 1–18, Jan.–Mar. 2020.



10.22214/IJRASET

45.98

IMPACT FACTOR:
7.129

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)