A Machine Learning and NLP-Based Approach for Efficient Email Spam Detection Using TF-IDF and Logistic Regression

Amisha Kumari[1], Smriti Kumari[2], Pushpa Kumari[3], Sanjyoti Kumari[4], Kumar Amrendra[5]

[1,2,3,4]Bachelor of Computer Application, Jharkhand Rai University, Ranchi, Jharkhand, India

[5]Assistant Professor, Faculty of Computer Science Engineering and Information Technology, Jharkhand Rai University Ranchi, Jharkhand, India

Corresponding author: anshu.amrendra@gmail.com

Abstract

In the digital era, the rapid growth of spam emails poses significant risks, including fraud and phishing attacks. This study presents a machine learning-based spam email detection system that classifies emails as spam or non-spam. The system employs Natural Language Processing (NLP) techniques such as tokenization, stopword removal, and TF-IDF vectorization to preprocess and transform textual data. A Logistic Regression model is trained on labeled datasets to identify patterns associated with spam messages. Additionally, a user-friendly interface is developed using Streamlit for real-time classification. The proposed system achieves high accuracy and demonstrates an effective approach to enhancing email security, with potential for further improvement using advanced models and larger datasets.

Keywords - *Machine Learning algorithm, Random Forest (RF), Gradient Boosting (GBT), Hybrid approach*

I. Introduction

This project highlights the integration of advanced machine learning techniques with modern web technologies to develop an efficient spam detection system. It demonstrates how real-time email classification can be applied in practical applications to improve email security and user experience. The system has real-world significance for individuals and organizations aiming to protect their communication from spam, phishing attacks, and fraudulent messages.

Furthermore, the system is designed to be modular and scalable, making it adaptable for additional features such as advanced filtering mechanisms, real-time monitoring, analytics dashboards, and integration with email platforms. It can also be extended to detect more complex spam patterns using deep learning techniques and larger datasets. In essence, the Spam Email Detection System is a combination of machine learning, natural language processing, and software development, providing a smart and efficient solution for identifying and filtering unwanted emails in an automated and structured manner.

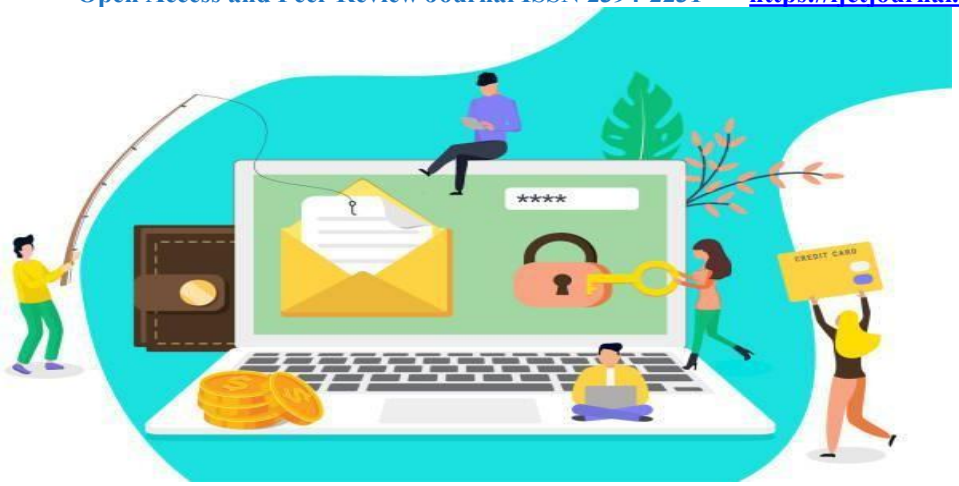

Fig.1 Email Spam

A. **Problem Statement:**

In today's digital environment, email communication plays a crucial role in both personal and business activities. However, the increasing number of spam emails has become a major concern for users and organizations. Spam emails often include unwanted advertisements, phishing attempts, and fraudulent messages that can lead to data theft, financial loss, and security risks.

Users receive a large volume of emails daily, making it difficult to manually identify and filter spam messages. Traditional filtering methods are not always effective, as they rely on basic rules and cannot detect advanced or evolving spam techniques. As a result, important emails may be missed, while harmful messages may go unnoticed.

Manual identification of spam emails is not only time-consuming but also prone to human error and inconsistency. Without an efficient system, users may fall victim to scams or experience reduced productivity due to unnecessary email clutter.

Therefore, there is a need for an intelligent and automated system that can accurately classify emails into spam and non-spam categories. By using machine learning and natural language processing techniques, the system can analyze email content, identify patterns, and provide reliable predictions. This will help improve email security, reduce manual effort, and enhance overall user experience.

II. RELATED WORK

This section reviews key studies on email spam detection using machine learning techniques. Pallavi N and Jayarekha (2023) explored algorithms such as Naïve Bayes, Support Vector Machines (SVM), and Decision Trees for spam classification, demonstrating effective results,

though their study was limited to a small set of models. Similarly, Abhila and Delphin (2021) proposed a Naïve Bayes-based framework incorporating preprocessing, feature extraction, training, and testing phases; however, the approach showed limitations in handling complex and evolving spam patterns. In another study, Mansoor and Muhana (2021) emphasized the superiority of machine learning techniques over traditional rule-based methods and highlighted the need for more adaptive and advanced spam detection systems. Furthermore, Narendra Kumar (2022) focused on preprocessing techniques and applied Naïve Bayes for efficient spam detection based on email content, showing promising performance even with limited datasets.

III. WORKFLOW

The proposed workflow presents a well-organized and systematic method for building a hybrid email spam detection system by combining Random Forest and Gradient Boosting algorithms. The process begins with the collection of labeled email datasets, which include both spam and legitimate messages. This data is then preprocessed by cleaning unnecessary information, removing noise, and preparing it for analysis. After preprocessing, the dataset is divided into training and testing sets to ensure proper model development and evaluation.

The next step involves feature extraction, where important characteristics are identified from the email content, such as keywords, patterns, and structural elements. These features are further refined through feature engineering techniques to improve the model's ability to distinguish between spam and nonspam emails.

Following this, both Random Forest and Gradient Boosting models are trained separately using the training data. Their parameters are carefully adjusted to achieve the best possible performance. Once trained, a hybrid approach is applied, where the outputs of both models are combined using ensemble techniques or integrated features, resulting in a more accurate and reliable system.

The hybrid model is then evaluated using performance metrics like accuracy, precision, recall, and F1score, and its results are compared with those of individual models to measure improvement. Finally, the system is deployed for real-world applications, where it continuously monitors incoming emails and updates itself to handle new and evolving spam techniques. Overall, this workflow ensures the development of a strong, flexible, and efficient spam detection system.

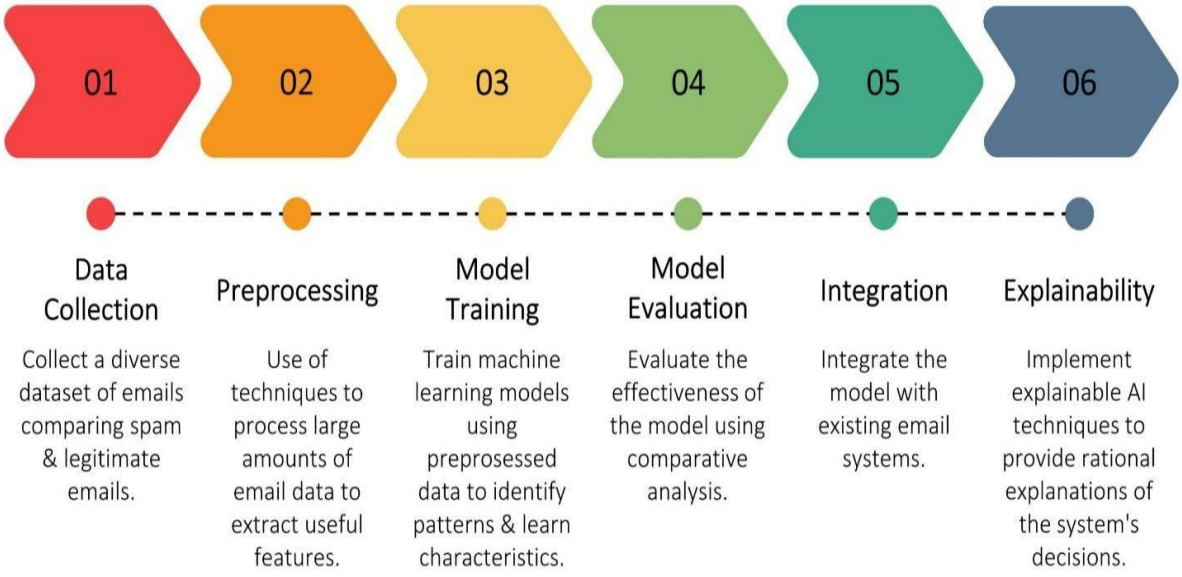

***Figure2**: Workflow of Proposed Sysyem*

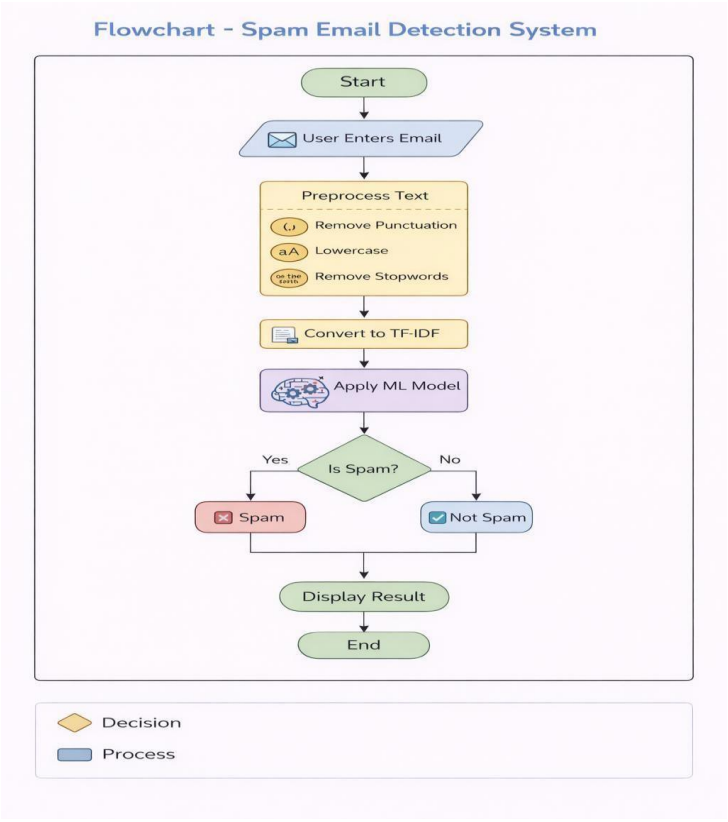

***Figure 3:** Flow chart of Spam Email Detection System*

IV. PROPOSED WORK

The proposed hybrid email spam detection system combines the capabilities of Random Forest and Gradient Boosting algorithms to overcome the limitations of conventional spam filtering techniques. By integrating these two powerful classifiers, the system aims to achieve higher accuracy, better

generalization, and improved robustness in identifying spam emails. The overall architecture is divided into several functional modules:

1. **Data Collection and Preprocessing Module:**

This module focuses on acquiring labeled email datasets that include both spam and legitimate messages. The collected data is cleaned by eliminating duplicate records, irrelevant content, and formatting errors. After preprocessing, the dataset is systematically split into training and testing subsets, enabling efficient model development and performance evaluation.

2. **Feature Extraction and Engineering Module:**

In this stage, meaningful features are derived from email data, including textual content, structural patterns, and metadata attributes. These features are further enhanced using feature engineering techniques to increase their relevance and discriminatory capability, ultimately improving the model's ability to differentiate between spam and non-spam emails.

3. **Model Training and Hybridization Module:**

Both Random Forest and Gradient Boosting models are trained independently using the prepared training dataset. Their outputs are then combined using ensemble strategies such as voting or stacking to create a hybrid model. Hyperparameter tuning and feature optimization are also performed to enhance model performance and ensure reliable spam detection.

4. **Evaluation and Validation Module:**

The system assesses the hybrid model using various performance metrics on the testing dataset. It also compares the results with those obtained from individual classifiers. Techniques like cross-validation and sensitivity analysis are applied to verify the stability, consistency, and effectiveness of the model.

5. **Deployment and Monitoring Module:**

Once validated, the hybrid model is deployed for practical use in detecting spam emails. The system continuously monitors performance to adapt to new spam patterns and evolving data. A user-friendly interface allows users to interact with the system and adjust detection settings, ensuring long-term efficiency and adaptability.

V. ARCHITECTURE

The Spam Email Detection System using Machine Learning follows a simple layered architecture:

1. User Interface Layer

- Accepts email text input from the user
- Displays classification result (Spam / Not Spam)

2. Application Layer

- Handles request processing
- Connects UI with backend logic (Streamlit / Flask)

3. Processing Layer

- Performs text preprocessing (cleaning, tokenization)
- Converts text using **TF-IDF vectorization**

4. Model Layer

- Trained ML model (Naive Bayes / Logistic Regression)
- Predicts whether the email is spam or not

5. Data Layer

- Stores dataset (CSV file)
- Contains trained model and vectorizer files

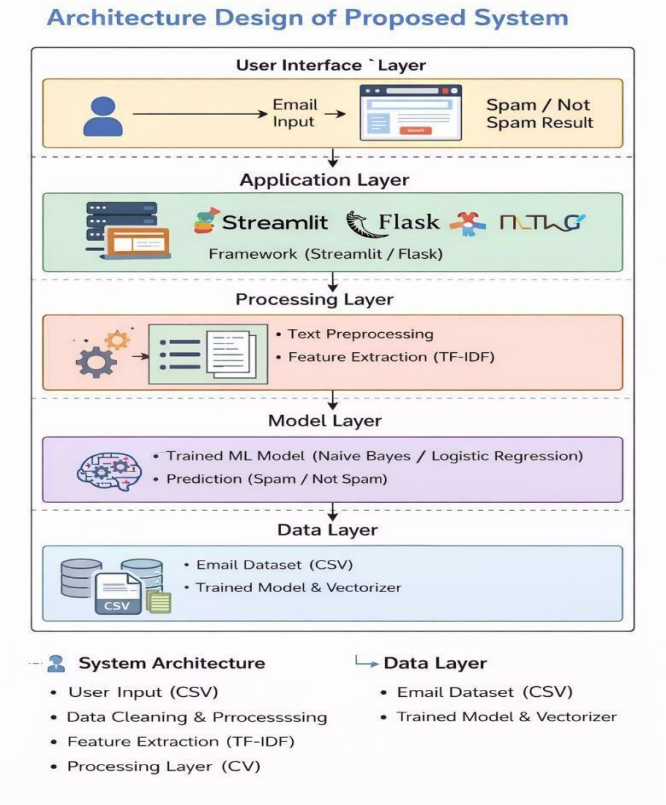

Figure 3: Architectural Design of the Proposed System

VI. RESULT AND DISCUSSIONS

As a result, we have successfully developed a Spam Email Detection System that integrates a machine learning model with a Streamlit-based web application. This intelligent system is capable of analyzing email text and classifying it as Spam or Not Spam in real time.

Built using Python, Scikit-learn, Pandas, NumPy, and Streamlit, the project aims to help users identify unwanted or fraudulent emails efficiently. The system uses a trained Naive Bayes classifier along with TF-IDF vectorization to achieve accurate predictions. Below is a breakdown of the results and discussion:

1.Accurate Spam Detection:

- **Model Performance:** The implemented machine learning model demonstrates high accuracy on the dataset, effectively distinguishing between spam and legitimate (ham) emails.
- **Real-time Prediction:** The system provides instant classification results when the user inputs email text, ensuring quick decision-making.
- **Keyword-Based Learning:** The model successfully identifies common spam patterns such as promotional words (“win”, “free”, “urgent”), improving detection reliability.

2. **Efficient System Functionality:**

- **Automated Email Classification**: The system automatically processes and classifies user input without manual intervention, increasing efficiency.
- **End-to-End Pipeline:** From user input → text preprocessing → TF-IDF transformation → model prediction → result display, the entire workflow is seamless.
- **Lightweight and Fast:** The use of Naive Bayes ensures low computational cost and fast execution, making the system suitable for real-time applications.

3. **User-Friendly Interface:**

- **Interactive Web Application:** The Streamlit interface allows users to easily enter email text and check results with a single click.
- **Clear Output Display:** Results are displayed using visual indicators:
  - Spam Email
  - Not Spam Email

4.Limitations and Future Scope:

- The model may misclassify emails with very complex or ambiguous language.
- Future improvements can include:
  - Deep learning models (LSTM, BERT)
  - Larger and more diverse datasets
  - Multi-language spam detection

VII. CONCLUSION

The Spam Email Detection System developed using machine learning successfully demonstrates the ability to classify emails as **Spam** or **Not Spam** with good accuracy. By utilizing techniques such as **TF-IDF vectorization** and the **Naive Bayes algorithm**, the system effectively analyzes textual data and identifies patterns commonly associated with spam messages.

The integration of the model with a **Streamlit web application** provides a simple and interactive interface, enabling users to input email content and receive instant predictions. The system is lightweight, fast, and efficient, making it suitable for real-time applications.

Overall, this project highlights how machine learning can be applied to enhance email security, reduce unwanted messages, and improve user experience.

VIII. LIMITATION

- The model is trained on a limited dataset, which may reduce its ability to generalize to all types of spam emails.
- It may fail to detect new or advanced spam techniques (e.g., cleverly worded phishing emails).
- The system mainly focuses on text-based analysis and does not consider attachments, images, or links.
- Accuracy may decrease for very short or ambiguous messages.
- It does not support multi-language detection effectively (if trained on English-only data).

IX. FUTURE SCOPE

- Use deep learning models such as LSTM or BERT for higher accuracy.
- Train the system on larger and more diverse datasets to improve performance.
- Implement multi-language spam detection.
- Extend detection to include email attachments, URLs, and images.
- Deploy the system as a browser plugin or email client integration (e.g., Gmail filter).
- Add continuous learning (online learning) to adapt to new spam patterns.
- Improve UI/UX with advanced dashboards and analytics.

REFERENCES

- **P. N. Pallavi and Jayarekha,** "Email Spam Classification Using Machine Learning Techniques," *International Journal of Computer Applications*, 2023.

- **R. Abhila and J. Delphin**, "Spam Email Detection Using Naïve Bayes Algorithm," *International Journal of Engineering Research & Technology (IJERT)*, 2021.
- **Mansoor and M. Muhana,** "Machine Learning Approaches for Spam Detection: A Review," *Journal of Information Security*, 2021.
- **N. Kumar**, "Efficient Spam Detection Using Naïve Bayes Classifier," *International Journal of Advanced Research in Computer Science*, 2022.
- Sharma, A., Amrendra, K., & Ranjan, P. (2025). Comparative analysis of ensemble classifiers over machine learning classifiers for early software quality prediction. In Proceedings of the Recent Advances in Artificial Intelligence for Sustainable Development (RAISD 2025) (pp. 351–366). Atlantis Press. https://doi.org/10.2991/978-94-6463-787-8_29
- **M. Bishop,** *Pattern Recognition and Machine Learning*. **New York, USA:** Springer, 2006.
- **D. Manning, P. Raghavan, and H. Schütze,** *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008.
- Scikit-learn, "Machine Learning in Python," [Online]. Available: https://scikit-learn.org
- **Streamlit**, "The fastest way to build and share data apps," **[Online]. Available:** https://streamlit.io
- Amrendra, K., & Ranjan, P. (2020). Emerging trends and applications in mobile ad hoc networks (MANETs). In S. Prasad (Ed.), Advances in Science & Technology (pp. 10–18). Empyreal Publishing House. ISBN: 978-81-946375-0-9.
- Amrendra, K., Sharma, A., & Ranjan, P. (2021). Challenges, attacks and security issues in MANET (mobile ad hoc networks). International Journal of Advance and Innovative Research, 8(4), 137–144. ISSN 2394-7780.
- UCI Machine Learning Repository, "Datasets for Machine Learning," **[Online]. Available**:
- https://archive.ics.uci.edu
- Natural Language Processing (NLP) concepts for text preprocessing and feature extraction.