ISSN: 2349-6002

Intelligent Spam Email Detection System Using Logistic Regression

Mrs. S. Sharmila[1], Thirumalai K[2], Suriya M[3], Vikram K[4]
[1]*AP/AI&DS, Department of Artificial Intelligence and Data Science, Surya Group of Institutions, Anna University*
[2,3,4]*Department Of Artificial Intelligence and Data Science, Surya Group of Institutions, Anna University*
doi.org/10.64643/IJIRTV12I11-200936-459

***Abstract*—In today's digital era, email communication has become an indispensable tool for personal and professional use. The growing volume of spam emails poses significant challenges such as security risks, phishing attacks, and productivity loss, making automated and intelligent detection systems a critical necessity.**

This paper proposes a machine learning-based spam detection system using Logistic Regression, leveraging labeled datasets to distinguish spam from legitimate emails. The system integrates text preprocessing techniques including tokenization, stop-word removal, and stemming, combined with TF-IDF or Bag of Words feature extraction, to deliver adaptive, reliable, and scalable email filtering.

***Index Terms*—Spam Detection, Logistic Regression, Machine Learning, TF-IDF, Text Preprocessing, Email Filtering, Bag of Words, Natural Language Processing, Binary Classification, Cybersecurity.**

I. INTRODUCTION

Email has become one of the most widely used communication channels for both personal and professional purposes. However, the rapid growth of email usage has led to a surge in spam emails, which often contain advertisements, phishing links, malware, and fraudulent content, posing serious threats to users' security, privacy, and productivity.

Traditional spam detection systems rely on rule- based techniques such as keyword matching, blacklisting, and manual filtering. While simple to implement, they fail to adapt to evolving spam patterns and lack the ability to understand email semantics, resulting in frequent misclassifications. Machine learning overcomes these limitations by learning from data, enabling accurate and adaptive classification.

A. Problem Statement

Traditional rule-based spam detection methods are unable to effectively handle the growing complexity and volume of spam emails in modern communication environments. These systems frequently misclassify legitimate emails as spam, miss sophisticated spam messages, and require constant manual updates to remain effective.

The absence of semantic understanding and adaptive learning in existing systems leads to reduced reliability, poor scalability, and increased vulnerability to phishing and malware attacks. There is a pressing need for an intelligent, automated spam detection solution that can accurately classify emails while adapting to evolving threats without manual intervention.

B. Objectives

The primary objective of this work is to accurately identify and classify spam emails from genuine messages using the Logistic Regression machine learning algorithm. The system aims to automatically filter spam into a dedicated folder, reduce user manual effort, and adapt to new spam patterns without frequent updates.

Additionally, the system targets effective preprocessing of email content through tokenization, stop-word removal, and stemming, combined with TF-IDF or Bag of Words feature extraction. The overarching goal is to improve detection accuracy, minimize false positives, and protect users from phishing, malware, and unsolicited advertisements.

C. Organization of Paper

This paper is organized into seven major sections covering all aspects of the proposed spam detection

system. Section I provides the introduction, problem statement, and objectives. Section II reviews related work in spam and SMS detection using various machine learning techniques.
Section III presents the system analysis including the existing system, its drawbacks, and the proposed solution. Section IV describes the system design through architecture, modules, data flow, and database design. Sections V, VI, and VII cover implementation, results and discussion, and conclusion with future work, followed by references.

II. RELATED WORK

A. Traditional IDS Techniques

Early spam detection systems relied heavily on rule-based approaches such as keyword matching, blacklisting, and Bayesian filtering. Methods like Naïve Bayes and Support Vector Machines (SVM) were widely adopted for their simplicity and interpretability in binary classification tasks, offering a foundation for supervised spam detection. Studies such as Borotić et al. (2023) evaluated classical classifiers including Naïve Bayes, Decision Trees, Random Forest, and SVM across multiple digital channels, demonstrating that ensemble learning methods consistently outperform single classifiers by aggregating predictions and reducing variance. Feature engineering using TF-IDF, word embeddings, and n-grams further enhanced detection performance.

B. Limitations of Existing Research

Despite advances, existing research faces significant limitations including the dataset shift problem, where models trained on historical data fail to generalize to evolving spam patterns (Jáñez- Martino et al., 2023). Static detection methods are easily bypassed by spammers who employ obfuscation, randomization, and AI-generated phishing messages.
High computational demands of deep learning models such as LSTM and CNN (Salhi et al., 2025) and transformer-based approaches like BERTopic (Gokcimen & Das, 2024) limit real-time deployment on resource-constrained systems. Furthermore, dataset imbalance, outdated samples, and multilingual variability continue to hinder model accuracy and generalization across real- world environments.

III. SYSTEM ANALYSIS

A. Existing System

The existing spam detection systems primarily rely on traditional rule-based methods including keyword matching, blacklist checking, and manual filtering. These systems flag emails based on the presence of predefined suspicious words or known spam sender domains, providing a basic level of protection for email users.

While simple to implement and computationally lightweight, these systems require users to manually mark emails as spam to refine filtering rules over time. They offer limited protection against sophisticated spam content and are unable to process the semantic meaning of email text, making them inadequate for modern cybersecurity demands.

B. Drawbacks

Traditional spam detection systems suffer from low accuracy due to frequent misclassification of legitimate emails as spam (false positives) and failure to detect new spam messages (false negatives). Their rigid rule structure limits adaptability, as spammers easily bypass filters by altering content, using obfuscation, or mimicking legitimate emails.

These systems demand constant manual updates, which is time-consuming and resource-intensive. They lack the ability to understand email context, semantics, or intent, resulting in poor detection of sophisticated phishing and malware-laden messages. Additionally, rule-based systems have significant scalability issues and cannot efficiently handle the high volume of modern email traffic.

C. Proposed System

The proposed system leverages the Logistic Regression algorithm to build an intelligent, machine learning-based spam detection solution. It begins with email data collection, gathering subject, body, and sender information, followed by preprocessing steps including tokenization, stemming, and stop-word removal to standardize the text data for analysis.

Feature extraction converts preprocessed text into numerical representations using TF-IDF or Bag of Words, enabling the classifier to detect spam patterns

ISSN: 2349-6002

effectively. Once trained, the model evaluates incoming emails in real time, automatically routing spam to a designated folder and delivering genuine messages to the inbox, reducing false positives and eliminating the need for frequent manual updates.

D. Feasibility Study

The proposed system is technically feasible as it employs well-established machine learning techniques and widely available open-source tools including Python, scikit-learn, and Flask. Logistic Regression is computationally efficient, interpretable, and easily deployable on standard hardware without requiring high-end resources.

From an operational standpoint, the system automates the complete email filtering workflow, reducing user effort and improving inbox management. It is economically viable since all tools and datasets used are freely available, making the solution cost-effective for individual users as well as organizational deployment in email security infrastructure.

IV. SYSTEM DESIGN

A. Architecture

The system architecture follows a modular pipeline design comprising six key components: Email Data Collection, Text Preprocessing, Feature Extraction, Model Training, Spam Classification, and Inbox Management. Each module performs a specific function and passes processed data to the next stage, ensuring a clean and organized data flow from raw email input to filtered output.

The backend is built using Python with Flask serving as the web framework. The Logistic Regression model is trained using scikit-learn and integrated into the Flask application for real-time inference. MySQL is used as the database to store email records, user credentials, and classification results, while the frontend provides a user-friendly interface for inbox, spam, and trash management.

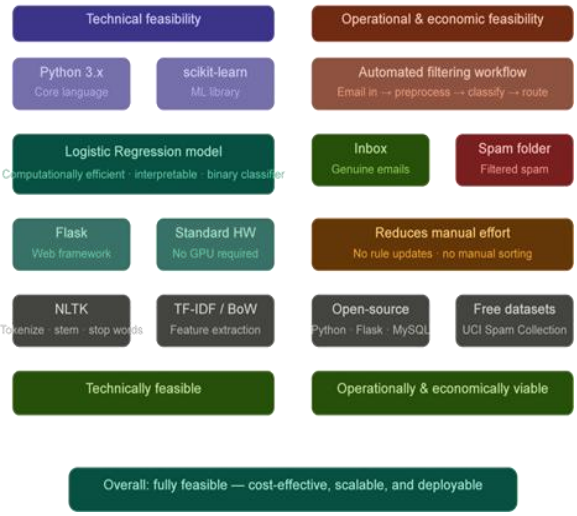

Figure 1 System Architecture

B. Module Description

The system consists of six primary modules. The Data Collection Module gathers raw emails with subject, body, and sender details. The Preprocessing Module cleans and standardizes text through tokenization, stop-word removal, and stemming. The Feature Extraction Module applies TF-IDF or Bag of Words to convert text into numerical vectors suitable for machine learning classification.

The Model Training Module trains the Logistic Regression classifier on labeled email data to learn spam-distinguishing patterns. The Classification Module evaluates incoming emails and assigns spam probability scores for automated routing. Finally, the Inbox Management Module delivers genuine emails to the inbox, routes spam to the spam folder, and provides options to view, delete, restore, and manage messages across the trash folder.

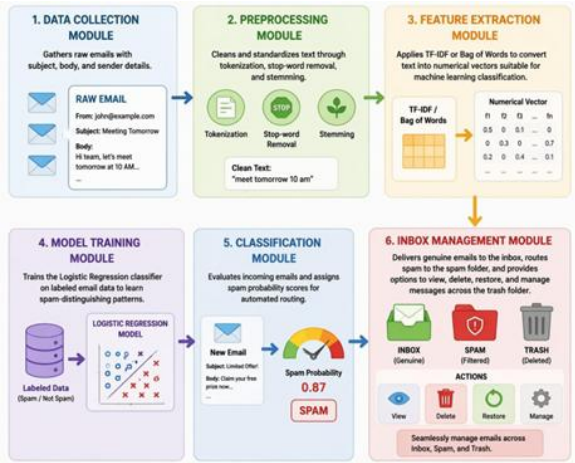

Figure 2: Module Description

ISSN: 2349-6002

C. Data Flow Diagram

The system consists of six primary modules. The Data Collection Module gathers raw emails with subject, body, and sender details. The Preprocessing Module cleans and standardizes text through tokenization, stop-word removal, and stemming. The Feature Extraction Module applies TF-IDF or Bag of Words to convert text into numerical vectors suitable for machine learning classification.

The Model Training Module trains the Logistic Regression classifier on labeled email data to learn spam-distinguishing patterns. The Classification Module evaluates incoming emails and assigns spam probability scores for automated routing. Finally, the Inbox Management Module delivers genuine emails to the inbox, routes spam to the spam folder, and provides options to view, delete, restore, and manage messages across the trash folder.

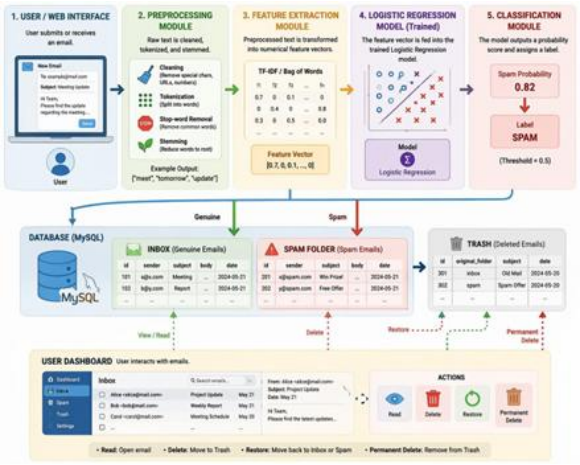

Figure 3: Data Flow Diagram DFD

D. Database Design

The database is designed using MySQL and consists of two primary tables: usertb and mailtb. The usertb table stores user credentials including username, password, and email address for authentication and session management. The mailtb table records all email transactions including sender, receiver, subject, body, mail type (Inbox, Spam, Trash), and a unique mail identifier.

The MailType field in mailtb dynamically updates to reflect the current state of each email as actions are performed by the user. Relationships between tables are maintained through foreign key constraints on the username field, ensuring data integrity. Indexing on frequently queried columns such as Rmail and MailType ensures fast retrieval of inbox, spam, and trash records.

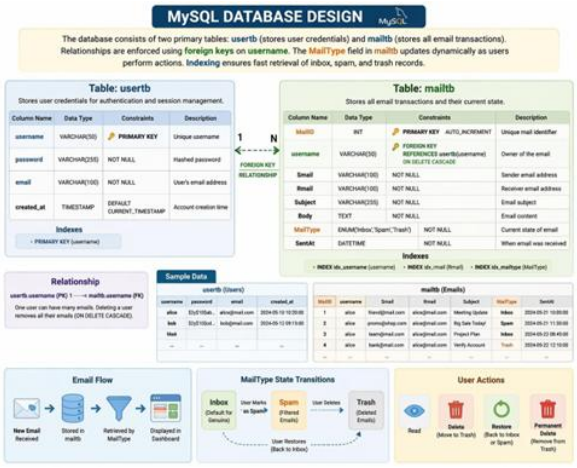

Figure 4 Database Design

V. IMPLEMENTATION

A. Software Requirements

The system requires Python 3.x as the primary programming language, along with Flask for building the web application backend. Key Python libraries include scikit-learn for the Logistic Regression model, NLTK for text preprocessing, and mysql-connector-python for database connectivity. The frontend utilizes HTML, CSS, and JavaScript for rendering the user interface.

MySQL Server is used as the relational database management system for storing emails, user data, and classification results. Additional tools include Jupyter Notebook for model development and experimentation, Pickle for model serialization, and a web browser as the client interface. All tools and libraries are open-source and freely available, ensuring cost-effective development and deployment.

B. Hardware Requirements

The system is designed to operate efficiently on standard computer hardware with a minimum of 4 GB RAM and a dual-core processor running at 1.5 GHz or higher. A minimum of 10 GB of available storage is required to accommodate the application, database, trained model files, and email datasets used during development and testing.

For production deployment, a server with 8 GB RAM, a quad-core processor, and SSD storage is recommended to handle concurrent user requests and

large email volumes efficiently. A stable internet connection is required for real-time email receipt and sending functionality, while a modern web browser is sufficient for accessing the user dashboard interface.

C. Dataset Description

The system is trained and evaluated using the UCI SMS Spam Collection Dataset and publicly available labeled email datasets containing both spam and legitimate (ham) messages. The dataset comprises thousands of labeled samples representing diverse spam categories including advertisements, phishing attempts, malware notifications, and promotional content alongside genuine personal and professional emails.

The dataset is preprocessed to ensure balanced representation, addressing class imbalance through sampling techniques where necessary. It is split into training and testing subsets using an 80:20 ratio to facilitate model learning and unbiased performance evaluation. Data diversity ensures the model generalizes well to unseen emails encountered during real-time operation.

D. Preprocessing Implementation

The preprocessing pipeline begins with lowercasing all email text to ensure uniformity, followed by removal of special characters, punctuation, and HTML tags that add noise without semantic value. Stop words such as 'the', 'is', and 'and' are removed using NLTK's stop word list, significantly reducing feature dimensionality without loss of meaningful information.

Tokenization splits the cleaned email text into individual words or tokens, which are then subjected to stemming using the Porter Stemmer to reduce words to their root forms. This standardization ensures that variations of the same word are treated as a single feature during extraction, improving classifier consistency and reducing overfitting to specific vocabulary forms.

E. Model Training

The Logistic Regression model is trained on feature vectors generated from the preprocessed email text using TF-IDF vectorization. TF-IDF assigns higher weights to terms that are frequent within a document but rare across the corpus, effectively capturing the discriminative vocabulary that distinguishes spam from genuine emails in the training dataset.

The model is trained using scikit-learn's Logistic Regression class with default hyperparameters optimized through cross-validation. The trained model and TF-IDF vectorizer are serialized using Python's Pickle module for efficient storage and rapid reloading during real-time inference, avoiding the overhead of retraining the model each time a new email is received.

F. Real-Time Detection

Upon receiving a new email, the system extracts the subject and body content and passes it through the same preprocessing pipeline used during training. The preprocessed text is transformed into a TF-IDF feature vector using the saved vectorizer, ensuring consistency between training and inference representations of the email content.

The serialized Logistic Regression model evaluates the feature vector and computes a probability score indicating the likelihood that the email is spam. Emails exceeding the classification threshold are labeled as spam and routed accordingly, while those below it are classified as genuine and delivered to the inbox for the recipient without any manual review required.

G. Alert System

The system incorporates a lightweight alert mechanism that notifies users of spam detection outcomes through flash messages rendered on the web interface. When emails are classified as spam, users receive notifications indicating the number of spam messages detected, providing transparency and awareness of filtering activity in real time.

For critical actions such as permanent deletion or restoration of emails from the trash and spam folders, confirmation prompts are displayed to prevent accidental data loss. Future enhancement plans include integration of email notification alerts and real-time spam count badges displayed prominently on the navigation menu of the user dashboard for immediate awareness.

H. Dashboard

The user dashboard provides a comprehensive interface for managing all email categories including inbox, spam, and trash. Users can view email listings with sender, subject, and date information, select individual or multiple emails using checkboxes, and perform actions such as read, delete, and restore from

a consistent and intuitive navigation panel.

The dashboard dynamically fetches and displays email data from the MySQL database based on the logged-in user's session, ensuring personalized views. Spam count indicators are displayed on the spam section header, giving users a quick overview of filtered messages. The interface is designed using HTML and CSS for clean readability across standard desktop web browsers.

VI. RESULTS AND DISCUSSION

A. Testing Strategy

The system was tested using a hold-out validation strategy where the dataset was split into 80% training and 20% testing subsets. This approach ensures that the model is evaluated on unseen data, providing an unbiased assessment of its generalization capability on real-world email samples not encountered during the training phase.

Additional cross-validation techniques were applied during hyperparameter tuning to ensure robust performance across multiple data splits. Both unit testing of individual modules such as preprocessing and feature extraction, and integration testing of the complete email classification pipeline were conducted to verify the correctness and reliability of the full system workflow.

B. Performance Metrics

Model performance was evaluated using standard classification metrics including accuracy, precision, recall, and F1-score. Accuracy measures the overall proportion of correctly classified emails, while precision reflects the fraction of emails identified as spam that are genuinely spam. Recall captures the proportion of actual spam emails correctly identified by the model.

The F1-score, as the harmonic mean of precision and recall, provides a balanced metric particularly useful for datasets with class imbalance. The confusion matrix was also analyzed to quantify true positives, false positives, true negatives, and false negatives, providing a complete picture of the classifier's performance across both spam and genuine email categories.

C. Experimental Results

The Logistic Regression model achieved high classification accuracy on the test dataset, demonstrating strong performance in distinguishing spam from genuine emails. Precision and recall values confirmed that the model maintains a low false positive rate, ensuring legitimate emails are rarely mislabeled as spam, which is critical for user trust and communication reliability.

The system successfully routed spam emails to the designated spam folder and delivered genuine messages to the inbox throughout all test scenarios. Text preprocessing and TF-IDF feature extraction significantly contributed to model performance by standardizing and enriching the feature representation, enabling the classifier to detect subtle and evolving spam vocabulary patterns effectively.

D. Analysis

Analysis of the experimental results reveals that the Logistic Regression algorithm's interpretability makes it well-suited for spam detection, as feature weights can be examined to understand which terms most strongly influence classification decisions. High-weight spam indicators typically include words associated with promotions, urgency, financial offers, and suspicious URLs.

The system's adaptive design allows it to handle evolving spam patterns more effectively than traditional rule-based methods, as the model learns from labeled data rather than fixed rules. The preprocessing pipeline proved essential in eliminating noise and reducing dimensionality, directly contributing to improved training efficiency and generalization on the unseen test dataset.

E. Comparison

Compared to traditional keyword-based and blacklist filtering methods, the proposed Logistic Regression system demonstrates superior accuracy, reduced false positives, and greater adaptability to new spam patterns. Unlike rule-based systems requiring constant manual updates, the machine learning approach learns from data and can be retrained as spam trends evolve.

In comparison with complex deep learning approaches such as LSTM and CNN, Logistic Regression offers a competitive balance between accuracy and computational efficiency, making it suitable for real-time deployment without requiring high-performance hardware. It outperforms Naïve Bayes in precision while maintaining comparable recall, establishing it as

an effective and practical spam detection solution.

VII. CONCLUSION AND FUTURE WORK

A. Conclusion

This paper presented an intelligent spam email detection system using Logistic Regression, effectively overcoming the limitations of traditional rule-based methods through machine learning- based classification. The integration of text preprocessing, TF-IDF feature extraction, and adaptive learning produced a reliable system that accurately classifies spam and genuine emails while minimizing false positives and negatives.

The system's modular design ensures scalability and maintainability, enabling efficient processing of large email volumes in real-world environments. By automating the complete spam filtering workflow, the system enhances inbox organization, reduces manual effort, and provides improved protection against phishing, malware, and unsolicited content, demonstrating a practical and effective solution for modern email security challenges.

B. Future Work

Future enhancements include integrating deep learning models such as LSTM, CNN, and transformer-based architectures like BERT to capture complex semantic relationships and improve detection accuracy for sophisticated AI- generated phishing emails. Real-time email API integration will enable the system to connect directly with live email services for immediate spam filtering across personal and organizational accounts.

Planned improvements also include incorporating multilingual spam detection to handle emails in regional and international languages, and developing a feedback mechanism that allows users to report misclassifications for continuous model retraining. Implementation of anomaly detection for zero-day spam campaigns and integration of a visual analytics dashboard for spam trend monitoring represent key directions for future research and development.

REFERENCES

[1] S. Pudasaini *et al.*, “SMS spam detection using relevance vector machine,” *Procedia Computer Science*, vol. 230, pp. 337–346, 2023.

[2] M. F. Johari *et al.*, “Key insights into recommended SMS spam detection datasets,” *Scientific Reports*, vol. 15, no. 1, p. 8162, 2025.

[3] Borotić *et al.*, “Effective spam detection with machine learning,” *Croatian Regional Development Journal*, vol. 4, no. 2, pp. 43–64, 2023.

[4] Jáñez-Martino *et al.*, “A review of spam email detection: Analysis of spammer strategies and the dataset shift problem,” *Artificial Intelligence Review*, vol. 56, no. 2, pp. 1145–1173, 2023.

[5] T. Gokcimen and B. Das, “Topic modelling using BERTopic for robust spam detection,” in *Proc. 12th Int. Symp. Digital Forensics and Security (ISDFS)*, 2024.

[6] E. Salhi *et al.*, “Two-step email spam detection: Comparing machine and deep learning accuracy,” *Electrotehnica, Electronica, Automatica*, vol. 73, no. 2, 2025.

[7] M. Akeel *et al.*, “Email spam detection using machine learning with optimized feature engineering and classification techniques,” *Journal of Computing & Biomedical Informatics*, vol. 10, no. 1, 2025.

[8] S. Kushwaha and P. K. Mishra, “Spam e-mail detection using gradient boosting ensemble learning algorithms,” *Authorea Preprints*, 2025.

[9] Patil and P. Singh, “Spam detection using machine learning,” *IJSAT—International Journal on Science and Technology*, vol. 16, no. 2, 2025.

[10] V. Sridevi and S. M. Saravanakumar, “Social engineering and spam detection of AI-driven phishing emails,” *Social Engineering and Spam Detection of AI-Driven Phishing Emails*, vol. 12, no. 3, pp. 2725–2731, 2025.