



International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Comparative Analysis of Machine Learning Algorithms for Email Spam Detection

Jenifer Mary , Dr. Nancy Jasmine Goldena

Stud Regno: 25081205300112017, Department of Computer Application and Research Centre, Sarah Tucker College(Autonomous), Tirunelveli.

^bAssociate Professor, Department of Computer Application and Research Centre, Sarah Tucker College(Autonomous), Tirunelveli.

ABSTRACT:

The rapid growth of electronic communication has led to a significant increase in spam emails, which may contain unwanted advertisements, fraudulent content, phishing attempts, and potentially harmful links. Traditional rule-based spam filtering methods may not effectively identify changing spam patterns. This paper, "Comparative Analysis of Machine Learning Algorithms for Email Spam Detection," focuses on developing a machine learning-based system to classify emails as spam or ham. The email dataset is first cleaned and preprocessed, followed by TF-IDF (Term Frequency–Inverse Document Frequency) feature extraction to convert textual email content into numerical features. Four machine learning algorithms, namely Multinomial Naive Bayes, Logistic Regression, Support Vector Machine (SVM), and Random Forest, are implemented and compared. Their performance is evaluated using accuracy, precision, recall, and F1-score, along with confusion matrices. Based on the experimental results, SVM achieved the highest accuracy of 98.03% and showed the best overall performance among the compared algorithms. The developed system also supports email analysis through a user interface and Gmail integration for analyzing inbox messages.

Keywords: Email Spam Detection, Machine Learning, TF-IDF, Multinomial Naive Bayes, Logistic Regression, Support Vector Machine, Random Forest, Text Classification, Precision, Recall, F1-Score.

1. Introduction:

Electronic mail remains one of the most widely used modes of digital communication, both for personal and professional purposes. Alongside its growth, the volume of spam emails has increased sharply, ranging from unsolicited advertisements to phishing attempts and malware-laden messages. Left unfiltered, spam consumes storage, wastes user time, and exposes users to financial and security risks. Traditional spam filters rely on static, rule-based techniques such as blacklists and keyword matching. While simple to implement, these methods struggle to keep pace with spammers who continuously alter message content and structure to evade detection. This limitation motivates the use of machine learning, which can learn discriminative patterns directly from labelled email data and generalize to previously unseen messages. This paper, Comparative Analysis of Machine Learning Algorithms for Email Spam Detection, implements and compares four widely used classification algorithms — Multinomial Naive Bayes, Logistic Regression, Support Vector Machine (SVM), and Random Forest — for the task of classifying emails as spam or ham (legitimate). Email text is converted into numerical form using TF-IDF feature extraction, after which each algorithm is trained and evaluated on the same dataset to allow a fair, direct comparison. Beyond the comparative study, the project also delivers a practical spam-detection system with a user interface and Gmail integration, allowing a user's inbox to be analyzed directly using the best-performing trained model. [1]

2. Related Work:

Email spam detection has been an active research area for over two decades. Early approaches relied on rule-based and blacklist filtering, where messages were flagged based on sender reputation or the presence of predefined keywords. While easy to deploy, such systems required constant manual updates and were easily bypassed by minor changes in spam wording. The introduction of statistical and machine learning methods significantly improved spam detection. Naive Bayes classifiers were among the earliest to be widely adopted for text classification because of their simplicity and strong performance on word-frequency features. Subsequent studies explored discriminative models such as Logistic Regression and Support Vector Machines, which often outperformed Naive Bayes on larger and noisier datasets by learning more flexible decision boundaries. Ensemble methods, particularly Random Forest and boosting-based classifiers, were later applied to spam detection to reduce overfitting and improve robustness by aggregating the predictions of multiple weak learners. More recent work has explored deep learning architectures, including recurrent neural networks and transformer-based language models, which can capture contextual and semantic relationships between words rather than relying solely on frequency statistics. Despite these advances, TF-IDF combined with classical machine learning algorithms remains a strong and computationally efficient baseline for spam detection, particularly in resource-constrained or real-time deployment settings such as the Gmail-integrated system developed in this project. This paper builds on that foundation by systematically comparing four widely used classical algorithms under identical preprocessing and feature-extraction conditions, providing a clear, reproducible benchmark before considering more computationally expensive deep-learning alternatives [1], [3]

3. Methodology:

The proposed methodology consists of dataset collection, data cleaning and preprocessing, TF-IDF feature extraction, model training using four machine learning algorithms, performance evaluation, and deployment through a user interface with Gmail integration.

i) Dataset Description:

A labelled email dataset containing both spam and ham (legitimate) messages was used for training and evaluation. Each record consists of the raw email/message text along with its corresponding class label. The dataset was split into training and testing sets to evaluate the generalization performance of each model. The dataset contains 4,825 ham messages and 747 spam messages, resulting in a total of 5,572 messages, as shown in Fig. 1.

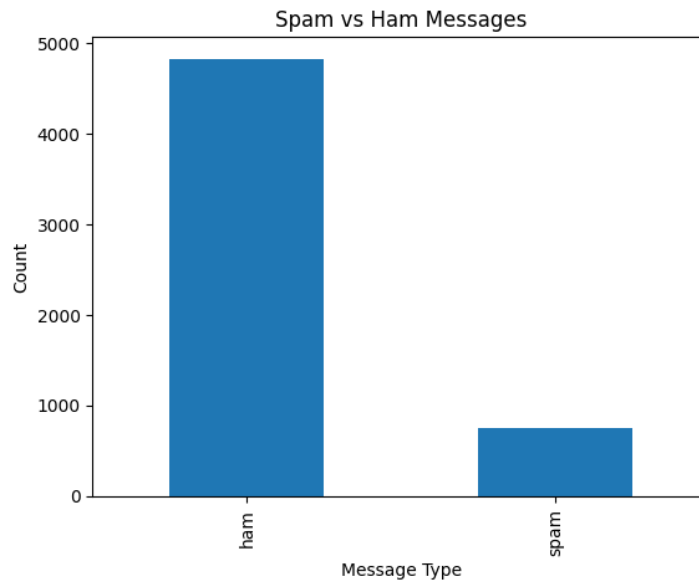


Fig. 1. Distribution of spam vs. ham messages in the dataset.

ii) Data Preprocessing:

Raw email text was cleaned before feature extraction. Preprocessing steps included conversion to lowercase, removal of punctuation, numbers, and special characters, removal of common stop-words, and tokenization of the remaining text. Word stemming was additionally applied to reduce related words (e.g., “offer”, “offering”, “offered”) to a common root form, which helps consolidate sparse vocabulary and improves the reliability of the TF-IDF weights computed in the next step. This step reduces noise and ensures that the extracted features capture meaningful vocabulary rather than formatting artifacts.

iii) Feature Extraction — TF-IDF:

The cleaned text was converted into numerical feature vectors using TF-IDF (Term Frequency–Inverse Document Frequency), which weights each term by how frequently it occurs in a given email relative to how common it is across the entire dataset. This gives higher importance to terms that are distinctive to spam or ham messages while down-weighting terms that occur in almost every email. [5]

iv) Machine Learning Algorithms Used:

Four supervised classification algorithms were selected for comparison, representing a probabilistic model, a linear discriminative model, a margin-based model, and an ensemble model:

- **Multinomial Naive Bayes** — a probabilistic classifier based on Bayes' theorem, which assumes conditional independence between features. It estimates the probability of a message being spam or ham from the frequency of each TF-IDF term, and is well suited to text data due to its simplicity and low computational cost.
- **Logistic Regression** — a linear model that applies the sigmoid function to a weighted combination of TF-IDF features to estimate the probability of an email being spam, with the decision boundary learned by minimizing a log-loss objective during training.
- **Support Vector Machine (SVM)** — finds an optimal separating hyperplane that maximizes the margin between spam and ham classes in the TF-IDF feature space. A linear kernel was used, which is typically effective for high-dimensional, sparse text data.
- **Random Forest** — an ensemble method that builds multiple decision trees on bootstrapped subsets of the training data and averages their predictions, reducing variance and improving robustness compared to a single decision tree.

v) Model Training and Evaluation:

Each of the four algorithms was trained on the same TF-IDF feature representation and the same training split, ensuring a consistent basis for comparison. The dataset was divided using an 80:20 train–test split, with model hyperparameters kept at standard default settings to focus the comparison on the

underlying algorithmic behaviour rather than tuning effort. Model performance was evaluated using accuracy, precision, recall, and F1-score, supported by confusion matrices to examine class-wise prediction behaviour. Precision indicates how many messages predicted as spam were actually spam, while recall indicates how many actual spam messages were correctly detected — a distinction that is particularly important in spam filtering, where both false positives (genuine mail marked as spam) and false negatives (spam reaching the inbox) carry real costs. [2], [5]

v) System Interface and Gmail Integration:

The best-performing trained model was integrated into a user interface that allows a user to input or fetch email content and receive a spam/ham classification. Gmail integration extends this capability to directly analyze messages in a connected inbox, enabling practical, real-time spam screening beyond the offline dataset.

4. Results and Discussion:

The four algorithms — Multinomial Naive Bayes, Logistic Regression, SVM, and Random Forest — were trained and evaluated on the same TF-IDF feature set. Their performance was compared using accuracy, precision, recall, and F1-score.

A. Performance Comparison of Algorithms

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Multinomial Naive Bayes	97.13	100.00	78.52	87.97
Logistic Regression	95.78	100.00	68.46	81.27
Support Vector Machine (SVM)	98.03	99.22	85.91	92.09
Random Forest	97.31	100.00	79.87	88.81

Table 1. Performance comparison of the four machine learning algorithms. Among the four algorithms, SVM achieved the highest overall accuracy of 98.03%, along with the best recall (85.91%) and F1-score (92.09%), indicating that it most effectively identifies spam messages while maintaining a very low false-positive rate. Naive Bayes, Logistic Regression, and Random Forest all achieved 100% precision, meaning they never misclassified a genuine (ham) email as spam, but their comparatively lower recall shows that a larger proportion of spam messages went undetected. Logistic Regression, in particular, had the lowest recall (68.46%) among the four models, missing the largest share of spam.

B. Confusion Matrix Analysis

Confusion matrices for each classifier were examined to study class-wise performance. Fig. 2 shows the confusion matrix obtained for the best-performing model, SVM. Out of 965 ham messages, all 965 were correctly classified with zero ham messages misclassified as spam. Out of 150 spam messages, 113 were correctly detected as spam while 37 were misclassified as ham. This indicates that SVM achieves perfect precision on the ham class and strong overall discrimination, though a portion of spam messages (recall of 85.91%) still goes undetected, which is an area identified for further improvement.

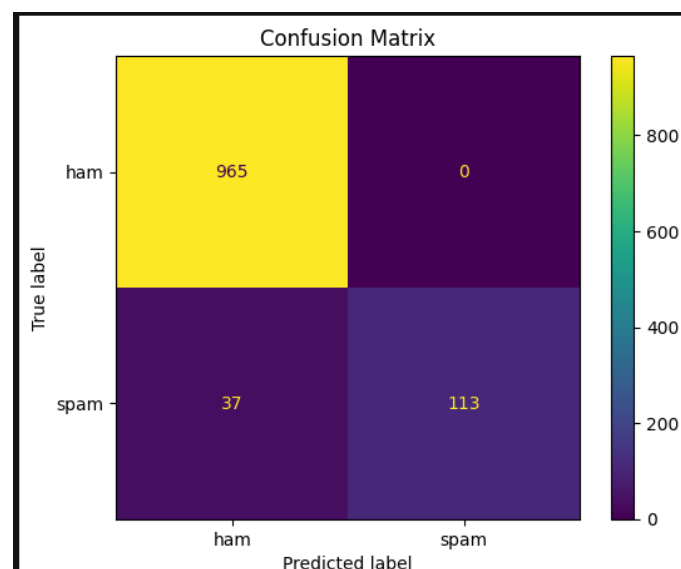


Fig. 2. Confusion matrix of the SVM classifier.

C. Best Performing Model

Based on the comparative results, SVM was identified as the best-performing algorithm for this task and was selected as the underlying model for the deployed spam-detection system.

D. System Interface and Gmail Integration

The trained SVM model was integrated into a user interface that classifies user-provided email text as spam or ham in real time. Fig. 3 shows the home

screen of the deployed Flask application, offering three modes of use: direct email-content analysis, .eml file upload, and Gmail inbox integration.

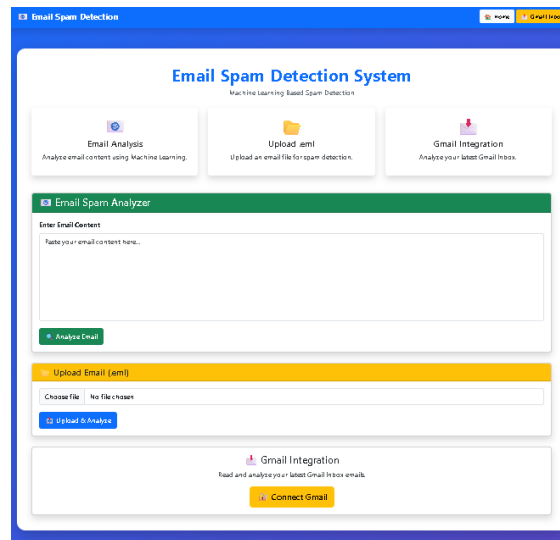


Fig. 3. Home interface of the Email Spam Detection System (Flask application output). Gmail integration was implemented using OAuth-based authentication: when the user selects “Connect Gmail,” the browser opens the standard Google account-selection screen (Fig. 4), allowing the user to securely authorize the application to read their inbox without exposing their credentials to the system directly.

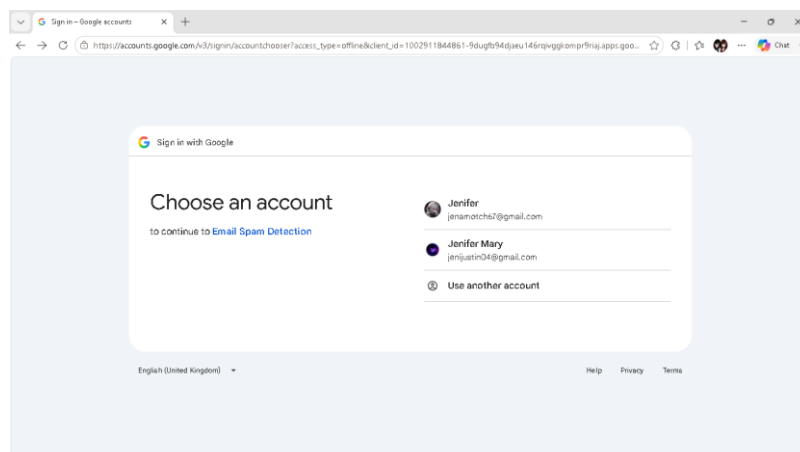


Fig. 4. Google account authorization screen shown on connecting Gmail.

Once authorized, the system fetches and classifies the most recent messages from the connected inbox. Fig. 5 shows the Gmail Inbox Dashboard, which displays the total number of emails scanned, the count classified as ham and spam, the overall spam rate, and a per-message breakdown listing sender, subject, and predicted status — demonstrating the system operating on a real, live inbox rather than only the offline test dataset.

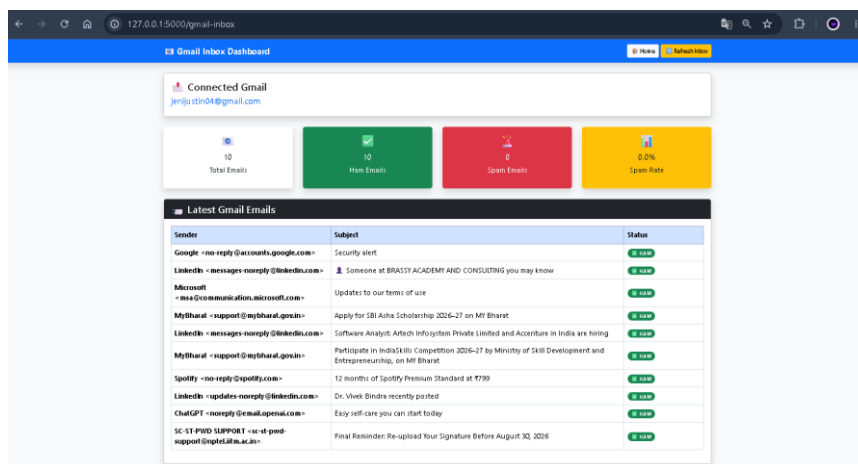


Fig. 5. Gmail Inbox Dashboard showing the latest 10 emails classified by the system.

E. Limitations

While SVM performed best overall, its recall of 85.91% indicates that a portion of spam messages (37 out of 150 in the test set) still evaded detection. The dataset used is also class-imbalanced, with ham messages outnumbering spam by roughly 6:1, which can bias models toward the majority class and inflate precision figures such as the 100% precision observed for Naive Bayes, Logistic Regression, and Random Forest. These factors should be considered when interpreting the reported metrics and when deploying the system on real-world inboxes with different spam characteristics.

5. Conclusion:

This project presented a comparative study of four machine learning algorithms — Multinomial Naive Bayes, Logistic Regression, SVM, and Random Forest — for email spam detection using TF-IDF feature extraction. Among the compared algorithms, SVM achieved the highest accuracy of 98.03%, along with the best recall and F1-score, making it the most suitable model for this task, while Naive Bayes, Logistic Regression, and Random Forest achieved perfect precision at the cost of lower recall. The developed system, supported by a user interface and Gmail integration, demonstrates a practical and effective approach to real-time spam filtering.

6. Future Enhancements:

1. Deep learning approaches such as LSTM and transformer-based models (e.g., BERT) can be explored to capture contextual patterns in email text beyond TF-IDF representations.
2. The dataset can be expanded to include multilingual and image-based (spam-in-image) emails to broaden the system's applicability.
3. Additional features such as sender reputation, email metadata, and embedded-link analysis can be incorporated to further improve detection accuracy and robustness against evolving spam techniques.

Acknowledgment

The author would like to thank the faculty of the Department of Computer Applications for their guidance and support throughout the design, implementation, and evaluation of this project.

REFERENCES

- [1] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: New collection and results," in Proc. ACM Symposium on Document Engineering.
- [2] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research.
- [3] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning.
- [4] L. Breiman, "Random Forests," Machine Learning.
- [5] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Information Processing & Management. [6] M. Karthikai Lakshmi and J. Steffi, "Development of an Interactive Satellite Data Analytics Tool Using Streamlit," International Journal of Research Publication and Reviews, vol. 7, no. 3, pp. 3876–3881, Mar. 2026.