

Spam Email Detection System Using Artificial Intelligence and Natural Language Processing

Bhuvaneshwaran R, Dharshan M, Nandhakumar S, Selvaragavan J, Indhuja N

Department of Artificial Intelligence and Data Science
Sengunthar Engineering College, Tiruchengode, Tamil Nadu, India

Abstract — Email remains a common channel for personal, educational, and business communication, but unsolicited messages and deceptive email campaigns create security and productivity concerns. This paper presents an AI- and natural-language-processing-based workflow for classifying email text as spam or legitimate (ham). The proposed pipeline validates labelled email data, normalizes text, converts messages into numerical features using Term Frequency–Inverse Document Frequency (TF-IDF), and applies a supervised classifier. Multinomial Naive Bayes is used as the baseline model, while Logistic Regression and a linear Support Vector Machine can be considered for comparison. The workflow separates dataset preparation, feature extraction, training, prediction, and evaluation so that the same fitted pipeline is used for new messages. Model quality is assessed using accuracy, precision, recall, F1-score, and a confusion matrix. The project report does not provide verified numerical evaluation results; therefore, performance values should be obtained by running the implementation on the selected labelled dataset and should not be assumed. The design provides a reproducible educational prototype that can be extended with URL analysis, multilingual processing, and email-client integration.

Keywords— Spam detection, natural language processing, machine learning, TF-IDF, Naive Bayes, email classification.

I. INTRODUCTION

Email is widely used for exchanging personal messages, academic information, business records, and service notifications. However, inboxes also receive unwanted advertisements, phishing attempts, fraudulent offers, and messages containing suspicious links. Manual screening is time-consuming and depends on the recipient's attention, while static keyword rules may become ineffective when senders alter wording, spelling, or identity.

Machine-learning-based filtering learns patterns from labelled examples and can apply those patterns to new messages.

Natural language processing (NLP) prepares unstructured email text for analysis, and text vectorization converts it into numerical features suitable for classification. This project develops a modular workflow that predicts whether an email is spam or ham and reports measurable evaluation results.

The main objectives are to preprocess email text consistently, represent messages using TF-IDF features, train a supervised classifier, provide a clear spam/ham prediction, and evaluate the model on held-out data. The system is intended as a prototype and does not replace sender authentication, link scanning, or other security controls.

II. RELATED WORK

Prior research has explored rule-based filtering, statistical text classification, deep learning, and the use of email metadata alongside message content. Phomkeona and Okamura examined feature-based deep-learning methods for investigating zero-day malicious emails, emphasizing the challenge of detecting threats that differ from known examples [1]. Work on email feature extraction has also considered header and body information, while machine-learning studies commonly compare classifiers such as Naive Bayes and Support Vector Machines [2], [3].

Deep-learning and neural approaches can model complex patterns, but their effectiveness depends on representative data, appropriate evaluation, and deployment resources [4], [5]. Classical TF-IDF pipelines remain useful baselines because they are comparatively lightweight, reproducible, and straightforward to evaluate. The present project adopts this baseline approach and emphasizes a consistent training and prediction pipeline rather than claiming unverified superiority over other methods.

The references listed in the source project report provide a starting point for the literature review. Before any formal publication or submission, bibliographic details such as publication year, title, volume, pages, and DOI should be checked against the original publications.

III. PROPOSED SYSTEM

The proposed system receives a labelled dataset containing email text and its class label. The dataset module checks required columns, missing values, empty messages, duplicate records, and supported labels. Messages are then split into training and test subsets. Stratified splitting is preferred when the class distribution allows it, so that both spam and ham are represented in each subset.

The TF-IDF vectorizer learns its vocabulary and inverse-document-frequency weights from training data only. The resulting sparse feature vectors are used to train a Multinomial Naive Bayes classifier. Alternative classifiers, including Logistic Regression or a linear Support Vector Machine, may be compared under the same data split and evaluation procedure. The final test set should not be used repeatedly to tune model parameters.

At prediction time, a new email is passed through the saved vectorizer-classifier pipeline. The system returns a human-readable label, either Spam or Ham. Input validation rejects empty text, and the interface should explain that the prediction is an automated estimate rather than a guarantee that the email is safe.

IV. METHODOLOGY

1. Dataset Preparation

The implementation expects a labelled CSV dataset with a text field and a target label. The project code uses the column names “text” and “label” by default and expects the class labels “spam” and “ham.” These settings must be adapted if the selected dataset uses different column names or label conventions. Missing entries, empty messages, unsupported labels, and duplicate text records are handled before training.

2. Text Preprocessing and Feature Extraction

Text normalization reduces irrelevant variation in message content. The baseline implementation uses TfidfVectorizer with lowercasing, English stop-word filtering, and unigram/bigram features. TF-IDF gives each term a weight based on its frequency in a message and its rarity across the training corpus. Because email URLs, punctuation, and unusual spelling may carry useful signals, any additional cleaning should be tested rather than applied without evaluation.

3. Classifier Training

The source implementation uses an 80:20 train-test split, a fixed random state, stratification by class, and Multinomial

Naive Bayes. The vectorizer and classifier are combined in a scikit-learn Pipeline so that training and inference apply the same transformations. The fitted pipeline is saved with Joblib for later reuse. The saved artifact should be loaded only from a trusted source.

4. Evaluation Metrics

Accuracy measures the overall proportion of correct predictions. Precision measures the proportion of predicted spam messages that are actually spam, while recall measures the proportion of actual spam messages detected. The F1-score balances precision and recall. A confusion matrix reports true positives, true negatives, false positives, and false negatives. These measures are important because false positives can cause legitimate emails to be missed, whereas false negatives allow spam to remain in the inbox.

V. SYSTEM ARCHITECTURE AND IMPLEMENTATION

The architecture is organized into dataset loading and validation, text preprocessing, feature extraction, model training, prediction, and evaluation/reporting modules. This separation makes individual stages easier to test and allows a classifier to be replaced without redesigning the complete application. The training path validates the dataset, splits the records, fits TF-IDF on training text, trains the classifier, evaluates held-out predictions, and saves the pipeline.

The prediction path accepts an email string, checks that it is non-empty, applies the saved pipeline, and displays the predicted class. The project uses Python, pandas, scikit-learn, and Joblib; NLTK may be used for optional additional text processing. A lightweight interface can be built with Flask or Streamlit. The report describes these technologies but does not establish that every optional interface component has been implemented.

The system can be tested using legitimate messages, obvious spam, empty input, mixed-case text, unusual punctuation, and messages containing URLs. Functional tests help confirm correct input handling, but they do not replace quantitative evaluation on an unseen test set.

VI. RESULTS AND DISCUSSION

The source report specifies accuracy, precision, recall, F1-score, and confusion-matrix output as evaluation measures, but it does not provide verified numerical results from a completed dataset run. Consequently, this paper does not claim a particular

accuracy or a performance advantage for the proposed model. The values should be generated by executing the training and evaluation code against the selected dataset and recording the dataset source, class distribution, train-test split, software versions, and model configuration.

The confusion matrix should be examined alongside the summary metrics. A high accuracy value alone may be misleading if the dataset is imbalanced. Precision and recall should be interpreted according to the intended use: a system that flags too many legitimate messages can disrupt communication, while a system that misses spam may leave users exposed to unwanted or deceptive content. Comparisons between classifiers should use the same split and metrics for a fair assessment.

Limitations and Future Work

The baseline text classifier may struggle with new spam campaigns, deliberate misspellings, obfuscated text, multilingual messages, and attacks that rely primarily on sender or URL characteristics. Its predictions depend on the quality and representativeness of the training dataset. The model should therefore be treated as one layer of email screening rather than a complete security solution.

Future work can combine text features with sender-domain reputation, email authentication signals, URL analysis, and attachment-risk indicators. Transformer-based models and multilingual classifiers may be evaluated against the baseline using identical test data. Further development should also consider model versioning, controlled retraining, explainability, privacy-preserving data handling, and a recovery path for incorrectly classified emails.

VII. CONCLUSION

This paper presents a practical AI- and NLP-based workflow for spam email classification using TF-IDF features and a supervised machine-learning classifier. Its modular pipeline covers dataset validation, preprocessing, feature extraction, model training, prediction, and evaluation. The design is reproducible and can serve as a baseline for comparing other algorithms. Reliable conclusions about model performance require measured results on a documented, held-out dataset. Future extensions can improve detection by incorporating additional email signals, multilingual processing, and stronger language models while retaining privacy and security safeguards.

REFERENCES

1. S. Phomkeona and K. Okamura, "Zero-Day Malicious Email Investigation and Detection Using Features with Deep-Learning Approach," *Journal of Information Processing*, vol. 28, pp. 222–229, 2020.
2. F. Jáñez-Martino, E. Fidalgo, S. González-Martínez, and J. Velasco-Mata, "Classification of Spam Emails through Hierarchical Clustering and Supervised Learning," 2020. Bibliographic details should be verified against the original publication.
3. T. M. Ma, K. Yamamori, and A. Thida, "A Comparative Approach to Naïve Bayes Classifier and Support Vector Machine for Email Spam Classification," in *Proc. IEEE 9th Global Conference on Consumer Electronics (GCCE)*, pp. 324–326. Publication year/details require verification.
4. "A Semantic-Based Classification Approach for an Enhanced Spam Detection," *Computers & Security*, vol. 94, Art. no. 101716, 2022. Author and publication metadata should be verified.
5. E. Castillo, S. Dhaduvai, P. Liu, K. S. Thakur, A. Dalton, and T. Strzalkowski, "Email Threat Detection Using Distinct Neural Network Approaches," in *Proc. First International Workshop on Social Threats in Online Conversations*, pp. 48–55, 2023.
6. S. Pandey, "E-Mail Spam Detection Using Machine Learning and Deep Learning," *International Journal for Research in Applied Science and Engineering Technology*, 2023. Volume, issue, and page details should be verified.
7. S. Gibson, B. Issac, L. Zhang, and S. M. Jacob, "Detecting Spam Email With Machine Learning Optimized With Bio-Inspired Metaheuristic Algorithms," *IEEE Access*. Publication metadata should be verified.
8. M. W. Putong and Suharjito, "Classification Model of Contact Center Customers Emails Using Machine Learning," *Advances in Science, Technology and Engineering Systems Journal*. Publication metadata should be verified.
9. "E-Mail Classification with Machine Learning and Word Embeddings for Improved Customer Support," *Neural Computing and Applications*, 2025. Author and publication metadata should be verified.
10. Khaleel Khan Mohammed. "The Future is Cloud: Modernizing Big Data for the Cloud Era