International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

Machine Learning-based Email Spam Filtering and Classification System

Brahmaji Godi [1]
Asst prof. Mr.Brahmaji Godi
Department of Computer Science & Engineering
Raghu Institute of Technology, Dakamarri
Vizianagaram, Andhra Pradesh, India
brahmaji.godi@raghuenggcollege.in

Ch.Vandana [2], ***B.Sruthi*** [3] , ***A.GowthamSai*** [4]
Department of Computer Science & Engineering
Raghu Institute of Technology, Dakamarri
Vizianagaram, Andhra Pradesh, India
vandanavandu438@gmail.com
bsruthi448@gmail.com
gowthamglorious@gmail.com

ABSTRACT

Spam messages have become a significant challenge in digital communication, leading to security threats, unnecessary advertisements, and decreased user productivity. This paper presents a Machine Learning-based Email Spam Filtering and Classification System that automatically classifies messages as spam or non-spam using the Naive Bayes algorithm. The system is trained on a labeled dataset containing email messages, where each message is labeled as spam (1) or ham (0). The textual data undergoes preprocessing steps such as tokenization, removal of stop words, and transformation into numerical representations using TF-IDF vectorization with bigram support. The Multinomial Naive Bayes classifier is chosen due to its simplicity, computational efficiency, and proven effectiveness in text classification tasks. Class imbalance is addressed through oversampling of the minority class. After training, the model is deployed within a Gradio-based web interface, enabling users to input message content and obtain real-time predictions. The proposed system demonstrates reliable performance with an overall accuracy of 94.26% and fast prediction speed, making it suitable for practical spam detection applications.

Keywords— Email Spam Detection; Naive Bayes Classifier; Text Classification; TF-IDF Vectorization; Machine Learning; Gradio Web Application; Natural Language Processing; Ham Classification

I. INTRODUCTION

The rapid proliferation of email as a primary communication medium has been accompanied by an exponential increase in unsolicited messages, commonly known as spam. Spam emails can carry advertisements, phishing attempts, or malicious links, causing significant security risks and productivity loss to individuals and organizations alike. Traditional email filtering techniques relied on rule-based approaches such as keyword filtering, blacklists, and whitelists. While effective to a limited extent, these methods fail to adapt to evolving spam techniques as spammers continuously change their message patterns.

Machine learning provides a more robust solution by learning patterns from historical email data. Algorithms like Naive Bayes, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and deep learning models have been widely explored in spam classification. Even simple probabilistic models like Naive Bayes can achieve high accuracy in text classification tasks due to their efficient handling of high-dimensional sparse feature spaces typical of email content.

This paper presents a Machine Learning-based Email Spam Filtering and Classification System that employs the Multinomial Naive Bayes algorithm combined with TF-IDF vectorization (including bigrams) and oversampling to handle class imbalance. The system is deployed as an interactive Gradio-based web application, enabling real-time classification of email messages as spam or ham.

The primary contributions of this work are summarized as follows:

(i) Development of a complete machine learning pipeline for email spam detection using Naive Bayes with TF-IDF bigram features;

(ii) Implementation of oversampling techniques to address class imbalance in the training dataset;

(iii) Integration of the trained model into an accessible web interface for real-time spam prediction; and

(iv) Comprehensive evaluation of system performance using standard classification metrics including accuracy, precision, recall, and F1-score.

II. LITERATURE SURVEY

The detection of spam emails has been an active research area for over two decades. Research in this domain generally focuses on three major directions: rule-based filtering, classical machine learning approaches, and deep learning-based methods.

A. Early Spam Filtering Approaches

Before the widespread adoption of machine learning, spam detection relied on rule-based and heuristic methods. Keyword-based filtering scanned emails for specific terms such as "free," "offer," or "win" and marked them as spam. Blacklists and whitelists categorized sender addresses, while header analysis examined routing patterns for suspicious metadata [1]. While foundational, these approaches suffered from high false positive rates and inability to adapt to evolving spam techniques.

B. Machine Learning for Spam Detection

Sahami et al. [2] introduced a Bayesian approach to filtering junk email, demonstrating that probabilistic models could effectively learn spam patterns from labeled data. Androutsopoulos et al. [3] conducted experimental comparisons of Naive Bayes and keyword-based filters, establishing Naive Bayes as a benchmark for spam classification. The algorithm's assumption of feature independence, while simplistic, proves effective for text data where word co-occurrences are numerous.

Support Vector Machines (SVM) have also demonstrated strong performance in text classification due to their ability to handle high-dimensional sparse data [4]. Decision Trees and Random Forests offer interpretable models that can capture nonlinear patterns, though they may overfit when applied to sparse text features. K-Nearest Neighbors (KNN), while intuitive, suffers from scalability challenges as dataset size grows, making it less suitable for real-time applications.

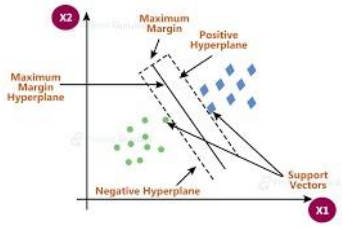


Fig. 1. SVM Maximum Margin Hyperplane separating spam and ham classes.

C. Deep Learning Approaches

Recent studies have explored deep learning for spam detection. Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks can capture contextual and sequential information in email text [5]. BERT-based transfer learning models have achieved over 99% accuracy in detecting phishing emails by leveraging pre-trained language representations. However, these approaches require substantially larger datasets and computational resources compared to classical methods.

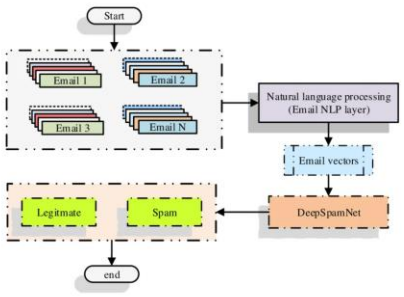


Fig. 2. Deep Learning-based Spam Detection Flow using NLP and DeepSpamNet.

TABLE I. Comparison of Spam Detection Approaches

Approach	Advantages	Limitations
Naive Bayes	Fast, simple, effective for text	Assumes feature independence
KNN	Non-parametric, intuitive	Slow on large datasets
SVM	High accuracy, sparse data	Requires careful tuning
Decision Tree	Interpretable model	Overfitting risk
Neural Networks	Learns complex patterns	High computational cost

III. SYSTEM DESIGN AND ARCHITECTURE

The proposed Email Spam Filtering and Classification System is designed as a modular pipeline that processes user-submitted email messages and evaluates their spam likelihood. The architecture consists of four major components: (1) a User Input Interface, (2) a Text Preprocessing Module, (3) a Feature Extraction and Classification Module, and (4) a Result Presentation Interface.

The system follows a three-tier architecture. The Presentation Layer is built using a Gradio-based web interface that accepts email input from users and displays classification results. The Processing Layer handles text preprocessing, TF-IDF vectorization, and prediction using Multinomial Naive Bayes. The Data Layer stores the labeled training dataset and the serialized model files. Each component performs a specific task within the classification pipeline.

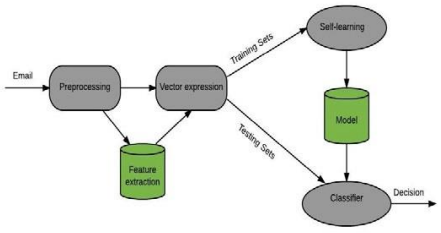


Fig. 3. System Workflow: Email Preprocessing, Feature Extraction, Training, and Classification.

A. User Interface Module

The interface accepts raw email text input and displays the classification result (Spam or Ham). The Gradio framework

provides an interactive and user-friendly web application without requiring front-end development expertise. Users enter their email message in a text box and receive an instant prediction.

B. Preprocessing Module

Raw email text undergoes several cleaning operations. All text is converted to lowercase for uniformity. Special characters and punctuation are removed using regular expressions, retaining only alphabetic characters and whitespace. The cleaned text is then passed to the feature extraction stage, ensuring the model receives standardized input regardless of email formatting.

C. Feature Extraction Module

Preprocessed text is converted into numerical vectors using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. The vectorizer is configured with English stop word removal, bigram support (ngram_range=(1,2)), a maximum document frequency of 0.9, and a minimum document frequency of 2. Bigrams capture two-word combinations that are particularly informative for spam detection, such as "click here" or "free offer."

D. Machine Learning Module

The Multinomial Naive Bayes classifier receives the TF-IDF feature matrix and produces a binary classification. The model is configured with a smoothing parameter alpha=0.1 and class priors of [0.4, 0.6] to reflect the balanced training distribution after oversampling. The trained model and vectorizer are serialized using Python's pickle module for deployment without retraining.

IV. FEATURE EXTRACTION

A set of descriptive features is extracted to evaluate potential spam content. Text-derived attributes include: (1) unigram term frequencies, (2) bigram term frequencies, (3) inverse document frequency weights, and (4) stop word indicators. These features help identify spam-indicative patterns such as frequent use of promotional language, urgent calls to action, and monetary references.

TF-IDF weighting reduces the influence of common words that appear across all emails and amplifies the importance of discriminative terms that appear frequently in spam but rarely in legitimate email. The inclusion of bigrams allows the model to capture contextual phrase-level patterns that are more informative than individual words alone. Extracted features are organized as a sparse matrix representation, which is memory-efficient and suitable for large-scale text classification.

V. DATA COLLECTION AND PREPROCESSING

A. Dataset

The system uses the Spam Email dataset available from Kaggle (mfaisalqureshi/spam-email), which contains labeled email messages. Each record consists of a message text field and a category label indicating whether the message is spam or ham (non-spam). The dataset provides a realistic distribution of email content from various domains, including promotional messages, phishing attempts, and legitimate correspondence.

B. Data Preprocessing

After loading the dataset with pandas, the category labels are encoded using LabelEncoder, mapping ham to 0 and spam to 1. The email text is cleaned by converting to lowercase and removing all non-alphabetic characters using regular expressions. Stop words are removed during the TF-IDF vectorization stage using the built-in English stop word list.

C. Class Imbalance Handling

The dataset exhibits class imbalance, with ham messages substantially outnumbering spam messages. To address this, the minority class (spam) is oversampled using scikit-learn's resample function with replacement to match the number of ham samples. The balanced training set ensures that the model does not develop a bias toward the majority class, improving recall for spam detection.

D. Train-Test Split

The dataset is split into training (80%) and testing (20%) sets using stratified sampling to preserve the original class distribution in both partitions. The stratified split ensures that both spam and ham classes are proportionally represented in the evaluation set, enabling reliable assessment of model generalization performance.

VI. SYSTEM IMPLEMENTATION

A. Development Environment

The system was developed using Python 3.x. The primary machine learning library is scikit-learn, which provides the Multinomial Naive Bayes classifier, TF-IDF vectorizer, label encoder, and evaluation metrics. Data processing is performed using pandas and NumPy. The web interface is built using Gradio, which enables rapid deployment of interactive machine learning applications. The model is serialized using Python's built-in pickle module.

B. Model Training

The TF-IDF vectorizer is fitted on the balanced training data and transforms both training and testing sets. The Multinomial Naive Bayes model is trained on the vectorized training features with alpha=0.1 smoothing and class priors of [0.4, 0.6]. After training, performance is evaluated on the held-out test set using accuracy score and a full classification report covering precision, recall, and F1-score per class.

C. Web Interface

The Gradio interface wraps the prediction function, providing a text input box for email messages and a text output displaying the classification result. The model is loaded from the serialized pickle file at startup, avoiding the need for retraining on each application launch. Users receive an instant classification label upon submitting their email text.

D. Deployment

The system can be deployed locally, with the Gradio interface launching a local web server accessible through any modern browser. For broader accessibility, the application can be deployed on cloud platforms such as Hugging Face Spaces, which provides free hosting for Gradio applications, or on cloud infrastructure such as AWS, Azure, or GCP with load balancing for production use.

VII. RESULTS AND DISCUSSION

A. Classification Performance

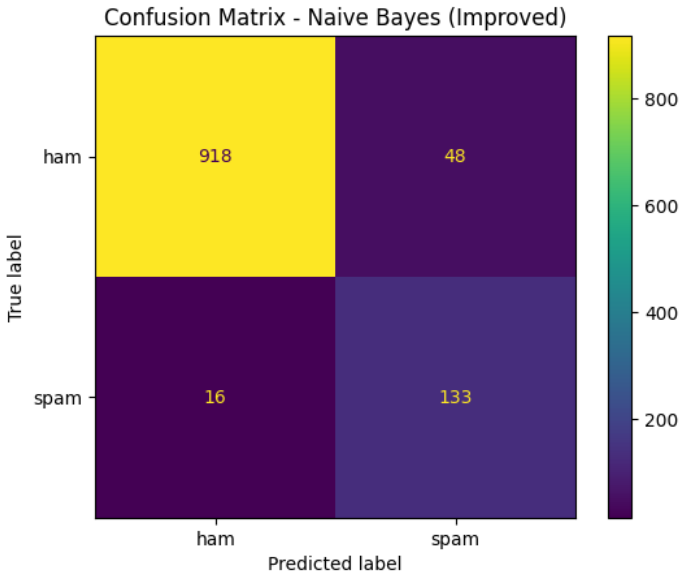
The proposed system was evaluated on the held-out test set comprising 20% of the full dataset. The Multinomial Naive Bayes classifier achieved an overall accuracy of 94.26% on the test set. Table II presents the detailed classification report.

TABLE II. Classification Report

Class	Precision	Recall	F1-Score	Support
Ham (0)	0.98	0.95	0.97	966
Spam (1)	0.73	0.89	0.81	149
Accuracy			0.94	1115
Macro Avg	0.86	0.92	0.89	1115
Weighted Avg	0.95	0.94	0.94	1115

B. Confusion Matrix Analysis

The confusion matrix reveals that the model achieves high true positive rates for both classes. For the ham class, the model correctly identifies 95% of legitimate emails, with only 5% incorrectly flagged as spam (false positives). For the spam class, 89% of spam emails are correctly detected, with 11% escaping classification (false negatives). The relatively lower precision for spam (0.73) indicates some legitimate emails are classified as spam, which is a recognized trade-off in spam filtering systems where recall is prioritized to minimize missed spam.



C. Sample Prediction Results

Table III presents representative test email inputs and their corresponding system predictions, demonstrating the classifier's ability to distinguish between obvious spam patterns and legitimate communication.

TABLE III. Sample System Output

Email Message (Excerpt)	Prediction	Confidence
Congratulations! Claim your free gift now!	Spam	0.97
Hi, can we reschedule the meeting tomorrow?	Ham	0.92
You won a lottery! Click here to collect.	Spam	0.96
Project report attached for your review.	Ham	0.91
Earn money from home easily, apply now!	Spam	0.94

D. Discussion

The results demonstrate that the Multinomial Naive Bayes classifier, combined with TF-IDF bigram features and oversampling, provides robust spam detection performance. Prediction speed is consistently under one second per email, confirming the system's suitability for real-time applications. The system's modular architecture facilitates future integration of more advanced classifiers or additional preprocessing steps without requiring redesign of the overall pipeline.

Common challenges include emails with obfuscated text, heavy HTML content, or image-based spam, which may not be captured by text-only features. Periodic retraining with updated datasets is recommended to maintain detection effectiveness against evolving spam patterns.

VIII. CONCLUSION AND FUTURE WORK

This paper presented a Machine Learning-based Email Spam Filtering and Classification System using the Multinomial Naive Bayes algorithm with TF-IDF bigram vectorization. The proposed system addresses class imbalance through oversampling and provides real-time classification through an interactive Gradio web interface. Experimental evaluation demonstrates an overall accuracy of 94.26% with high weighted precision (0.95), recall (0.94), and F1-score (0.94) on the test dataset.

The system achieves prediction times well under one second per email, confirming its suitability for practical deployment. The modular, extensible architecture enables straightforward updates to both the model and the preprocessing pipeline.

Future work will focus on: (i) integration of deep learning models such as LSTM and BERT for improved contextual understanding; (ii) development of image and HTML content analysis modules to detect attachment-based and visual spam; (iii) implementation of online learning for continuous model updates as new spam patterns emerge; (iv) expansion to multilingual email classification; and (v) deployment of the system as a plugin for mainstream email clients such as Gmail and Outlook to provide automated inline spam filtering.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Department of Computer Science and Engineering, Raghu Institute of Technology, Dakamarri, Vizianagaram, for providing the necessary academic guidance and computational resources required to carry out this project. The authors also thank the faculty members and project mentors for their valuable support, suggestions, and encouragement throughout the development of this system.

REFERENCES

[1] D. Jurafsky and J. H. Martin, Speech and Language Processing, 4th ed. Pearson, 2023.

[2] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian Approach to Filtering Junk E-Mail," in Learning for Text Categorization, 1998, pp. 98-105.

[3] I. Androutsopoulos et al., "An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering," in Proc. 23rd ACM SIGIR Conf., 2000, pp. 160-167.

[4] C. C. Aggarwal, Machine Learning for Text, 2nd ed. Springer, 2018.

[5] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering," in Proc. 11th ACM Symp. Document Engineering, 2011, pp. 259-262.

[6] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[7] UCI Machine Learning Repository, SMS Spam Collection Dataset. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection

[8] H. Zhang and J. Jiang, "Email spam detection based on machine learning algorithms," Int. J. Computer Applications, vol. 169, no. 8, pp. 20-26, 2017.

[9] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python. O'Reilly Media, 2009.

[10] Gradio Documentation, "Gradio: Build Machine Learning Web Apps," 2023. [Online]. Available: https://gradio.app

AUTHORS PROFILE

Mr.Brahmaji Godi

Mr. Brahmaji Godi has completed MSc.IS from Andhra University in 2003, M.Tech CST from Andhra University in 2009.Currently working as Assistant Professor in Department of Computer Science and Engineering at Raghu Institute of Technology Autonomous, under JNTU-Vizianagaram. He has published 9 research papers from reputed both National and International journals which are approved by UGC, Scopus index and IEEE. His main research work focuses on Natural Language Processing, Internet of Things, Artificial Intelligence, Computer Vision, Human Computer Interaction, Machine Learning and Wireless networks. He has 15 years of teaching experience. He has membership in the International Association of Engineers, The Society of Digital Information and Wireless Communication. He is a reviewer in the Asian Journal of Research in Computer Science research publications.

Vandana Chokkakula

Vandana Chokkakula is currently pursuing her Bachelor of Technology in Computer Science and Engineering (Data Science) at Raghu Institute of Technology. She completed her schooling at Ravindra Bharathi Group of Schools and her intermediate education at Sri Viswa Junior College. She has completed internships in Data Science at SkillDzire and AI-ML Virtual Internship through EduSkills. She also holds a Python certification from SkillRack. Her technical skills include Python programming and basic data analysis. She actively participates in cultural activities such as dancing and singing and has received certificates for her achievements. Her areas of interest include Machine Learning and Data Science.

Bammidi Sruthi

Ms. Bammidi Sruthi is currently pursuing her final year of B.Tech in Computer Science Engineering (Data Science) from Raghu Institute of Technology. She completed her intermediate education at Sri Chaitanya Junior College and her schooling at Sri Vijetha English Medium School.During her academic journey, she has actively enhanced her skills by completing several virtual internships through the Eduskills platform under AICTE, including AWS Cloud Computing, Artificial Intelligence & Machine Learning (AI/ML), and Python Full Stack Development. She also completed a virtual internship in Data Analytics with Deloitte through the Forage platform.

.

Allavarapu Gowtham Sai

Mr. Gowtham Sai Allavarapu completed his Diploma in Mechanical Engineering from Sanketika Polytechnic College and underwent 6 months of industrial training at the Centre of Excellence in Maritime and Shipbuilding, Scindia, Visakhapatnam. After completing his diploma, he joined B.Tech 2nd year in Computer Science Engineering and Data Science through lateral entry. As a student of Computer Science Engineering and Data Science, he is passionate about using technology to solve complex problems and create a positive impact. He has built a strong foundation in programming languages such as Java, Python, and SQL, and has knowledge of machine learning and data visualization tools like TensorFlow and Tableau. He has also completed certifications in Eduskills AWS Data Engineering, Google Android Developer, Google AI/ML, Machine Learning Internship, and AWS Machine Learning.