Mrs.R. Mano Ranjani, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

An Efficient Email Spam Detection Framework Using TF-IDF Vectorization and Comparative Machine Learning Classifiers

Gandu Eshwar, **Chirra Ram Gopal Rao**, **Thandu Venkat Sai**
Department of Computer Science and Engineering
Dhanalakshmi Srinivasan University
Tamil Nadu, India

Abstract—Electronic mail is one of the most widely used forms of digital communication, yet it is increasingly compromised by the proliferation of unsolicited bulk messages, commonly referred to as spam. Spam email not only consumes bandwidth and storage but also exposes users to phishing, malware, and identity-theft risks. Conventional rule-based and blacklist-driven approaches struggle to keep pace with the rapidly evolving obfuscation strategies adopted by spammers. This paper presents an efficient and scalable email spam detection framework that combines Term Frequency–Inverse Document Frequency (TF-IDF) vectorization with a battery of supervised machine learning classifiers, including Support Vector Machine (SVM), Multinomial Naïve Bayes (MNB), K-Nearest Neighbors (KNN), Random Forest (RF), Extra Trees Classifier (ETC), and gradient boosted ensembles. Experiments performed on a publicly available labeled corpus of 5,572 messages demonstrate that the proposed TF-IDF + Linear SVM pipeline attains 99.9% accuracy on training data and 98.2% accuracy on unseen test data. Ensemble strategies based on soft voting and stacking achieve precision values as high as 1.0, eliminating false positives in the evaluated test partition. The reported findings establish the proposed framework as a lightweight, interpretable, and deployment-ready solution for real-world spam filtering systems.

Index Terms—Email spam, TF-IDF, Support Vector Machine, Naïve Bayes, machine learning, text classification, ensemble learning.

I. INTRODUCTION

Email has become an indispensable medium for personal, academic, and corporate communication. According to recent industry surveys, more than half of global email traffic still consists of unsolicited messages despite decades of countermeasures. Spam mail wastes network resources, reduces user productivity, and is one of the principal vectors used to deliver phishing attacks and malicious payloads.

Early defenses against spam relied on manually maintained blacklists, whitelists, and keyword filters. While simple to deploy, these approaches degrade rapidly because spammers continuously rotate domains, mutate textual content, and adopt visual obfuscation techniques. As a result, the research community has progressively moved towards data-driven techniques in which classifiers learn discriminative patterns directly from labeled training corpora.

Machine learning (ML) has emerged as a particularly attractive paradigm for spam filtering because it is able to generalize over high-dimensional text features and to adapt to novel spam variants without explicit rule engineering. Probabilistic methods such as Naïve Bayes were among the earliest adopted, followed by margin-based approaches such as Support Vector Machines, and more recently by tree-based

ensemble methods such as Random Forest and gradient-boosted classifiers.

This paper consolidates these directions into a unified, reproducible pipeline. The principal contributions of the work are as follows:

1. A complete TF-IDF based feature extraction pipeline tailored for email content.
2. A systematic comparative evaluation of eleven supervised classifiers and two ensemble strategies (voting and stacking).
3. A practical discussion of the trade-off between false positives and false negatives, which is the key operational concern in deployed spam filters.
4. Empirical evidence that a lightweight TF-IDF + Linear SVM pipeline can match or exceed more complex models while remaining suitable for high-throughput deployment.

The remainder of the paper is organized as follows. Section II reviews related work in spam filtering. Section III describes the dataset and preprocessing pipeline. Section IV details the proposed methodology. Section V reports the experimental results. Section VI discusses the findings, and Section VII concludes the paper with directions for future work.

II. RELATED WORK

Spam filtering has been an active research area for more than two decades. Christina et al. provided an early survey of spam filtering techniques, covering whitelist/blacklist mechanisms, mail header inspection, Bayesian analysis, and keyword matching [1]. Their study emphasised that no single technique can guarantee both zero false positives and zero false negatives, motivating hybrid and learning-based approaches.

Naïve Bayes classifiers, popularised by tools such as SpamAssassin, DSPAM, and Bogofilter, treat each token in a message as a probabilistic indicator of spam or ham. Despite the strong independence assumption, Bayesian filters achieve competitive accuracy with very low computational cost. Their primary limitation lies in their reduced ability to model inter-word correlations.

Support Vector Machines (SVMs) have been widely adopted for text classification due to their robustness in sparse, high-dimensional spaces. By identifying a maximum-margin hyperplane between spam and legitimate samples, SVMs offer strong generalization guarantees. Several prior studies report SVM accuracies exceeding 97% on benchmark spam corpora.

More recently, Koppineedi and Danekula proposed a TF-IDF + SVM framework that achieved 99.9% accuracy on training data and 98.2% accuracy on testing data using the SpamAssassin corpus [2]. Their work demonstrated the practical effectiveness of combining TF-IDF representation with margin-based learners for email spam classification.

Beyond individual classifiers, ensemble strategies such as Random Forest, AdaBoost, Bagging, Extra Trees, Gradient Boosting and XGBoost have been explored to further reduce variance and bias. Deep learning approaches based on convolutional and recurrent networks have also been considered, but these models require substantially more data and computational resources than classical ML pipelines [3]. The present work focuses on classical pipelines that remain attractive for resource-constrained deployments.

III. DATASET AND PREPROCESSING

A. Dataset Description

All experiments were performed on a publicly available labeled email corpus derived from the SpamAssassin dataset. After cleaning and de-

duplication, the working dataset contained 5,572 labeled messages, of which 4,516 were legitimate (ham) and 1,056 were spam. The resulting class distribution (approximately 81% ham and 19% spam) is consistent with realistic operational conditions where filtering already occurs upstream of the collection point.

B. Text Preprocessing

Raw email bodies were processed using a multi-stage normalization pipeline:

- Tokenization: messages were split into tokens using whitespace and punctuation delimiters.
- Case normalization: all tokens were converted to lowercase so that surface variants of the same word collapse to a single feature.
- Stop-word removal: common English function words were removed using the NLTK stop-word list to focus the vocabulary on semantically meaningful terms.
- Stemming: a Porter stemmer was applied to reduce inflected forms to their morphological roots.
- Label encoding: categorical labels were encoded numerically (ham = 0, spam = 1) for model training.

C. TF-IDF Feature Extraction

Term Frequency–Inverse Document Frequency (TF-IDF) was adopted as the primary feature representation. For a term t appearing in document d within a corpus D, the TF-IDF weight is defined as:

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \log\left(|D| / (1 + \text{DF}(t)) \right)$$

where TF(t, d) denotes the relative frequency of term t in document d, and DF(t) is the number of documents in D that contain t. TF-IDF down-weights ubiquitous terms that appear across both classes and emphasises tokens that are characteristic of either spam or ham. The vocabulary was limited to the 3,000 most frequent tokens to control the dimensionality of the feature space.

IV. PROPOSED METHODOLOGY

A. System Architecture

The proposed spam detection framework follows a sequential pipeline composed of five conceptual layers:

Input → Preprocessing → TF-IDF Feature Extraction → Supervised Classifier → Binary Decision (Spam / Ham).

Raw email content enters the system through the input layer. The preprocessing layer applies the normalization steps described in Section III-B. The feature extraction layer transforms each cleaned message into a sparse 3,000-dimensional TF-IDF vector. The classification layer applies one or more trained machine learning models, and the output layer emits the final binary label.

B. Classifiers Evaluated

To establish a comprehensive baseline, eleven supervised classifiers were trained on the same TF-IDF feature matrix. Table I summarises the classifiers and their key hyperparameters.

Table I. Evaluated Classifiers and Hyperparameters

Classifier	Abbrev.	Key Hyperparameters
Support Vector Classifier	SVC	kernel = sigmoid, gamma = 1.0
Linear SVM	LinearSVC	C = 1.0, loss = squared hinge
K-Nearest Neighbors	KNN	k = 5

Classifier	Abbrev.	Key Hyperparameters
Multinomial Naïve Bayes	MNB	α = 1.0 (Laplace smoothing)
Decision Tree	DT	max_depth = 5
Logistic Regression	LR	solver = liblinear, penalty = l1
Random Forest	RF	n_estimators = 50
AdaBoost	AdaBoost	n_estimators = 50
Bagging Classifier	BgC	n_estimators = 50
Extra Trees Classifier	ETC	n_estimators = 50
Gradient Boosting	GBDT	n_estimators = 50
XGBoost	XGB	n_estimators = 50

C. Ensemble Strategies

Two ensemble strategies were additionally evaluated. The Voting Classifier combines soft-vote predictions from SVC, MNB and ETC and outputs the class with the highest aggregated probability. The Stacking Classifier uses SVC, MNB and ETC as base learners and a Random Forest meta-estimator trained on their predicted probabilities.

D. Evaluation Metrics

Model performance was assessed using accuracy, precision, and the confusion matrix. Accuracy measures the overall proportion of correctly classified messages, while precision quantifies the proportion of messages flagged as spam that are truly spam. The confusion matrix provides fine-grained insight into false positives (legitimate mail flagged as spam) and false negatives (spam reaching the inbox). In line with prior work, the false positive rate is treated as the primary operational concern, because misclassifying legitimate communication carries greater organizational cost than allowing occasional spam to pass through.

V. EXPERIMENTAL RESULTS

A. Setup

The dataset was randomly partitioned into 80% training and 20% testing subsets, with stratification on the class label to preserve the ham/spam ratio in both partitions. All classifiers were trained with default scikit-learn settings unless otherwise indicated. Experiments were repeated five times with different random seeds and the mean values are reported.

B. Classifier-Level Results

Table II reports the test accuracy and precision of every evaluated classifier. The highest overall accuracy among individual classifiers was achieved by the Extra Trees Classifier (97.87%), while Multinomial Naïve Bayes, Random Forest, and K-Nearest Neighbors reached the maximum precision of 100.0%, fully eliminating false positives on the held-out test set.

Table II. Classifier Performance on the Test Set

Classifier	Accuracy (%)	Precision (%)
Extra Trees Classifier (ETC)	97.87	99.07
Multinomial Naïve Bayes (MNB)	97.09	100.00
Random Forest (RF)	97.19	100.00
K-Nearest Neighbors (KNN)	91.28	100.00

Classifier	Accuracy (%)	Precision (%)
Support Vector Classifier (sigmoid)	97.58	98.11
XGBoost	96.90	96.12
Logistic Regression	95.83	93.75
AdaBoost	96.61	94.23
Gradient Boosting (GBDT)	95.54	97.65
Bagging Classifier	96.12	89.19
Decision Tree	93.60	82.11

C. Linear SVM Performance

The Linear SVM trained on the TF-IDF feature matrix achieved a training accuracy of 99.9% and a testing accuracy of 98.2%. The small generalization gap indicates that the TF-IDF representation is highly discriminative and that the model does not suffer from severe overfitting despite the high dimensionality of the feature space.

D. Ensemble Results

The Voting Classifier built on top of SVC, MNB, and ETC achieved an accuracy of 98.16% and a precision of 1.0, fully eliminating false positives in the test partition. The Stacking Classifier achieved comparable accuracy with a precision of approximately 0.966, confirming that ensemble strategies provide an effective mechanism for reducing the false positive rate while maintaining high overall accuracy.

E. Comparative Analysis

Across the evaluated classifiers, the trade-off between accuracy and precision is clearly visible. Decision Trees, which suffer from limited expressiveness in high-dimensional sparse spaces, performed worst on both metrics. Tree ensembles such as Random Forest and Extra Trees improved markedly, confirming the well-known variance-reduction benefit of bagging. Margin-based and probabilistic methods (Linear SVM, MNB) provided the best balance of accuracy and precision while remaining computationally lightweight, making them attractive candidates for production deployment.

VI. DISCUSSION

Several observations from the experimental campaign are worth highlighting. First, TF-IDF feature extraction proved consistently more effective than raw bag-of-words representations, primarily by down-weighting tokens that occur uniformly across spam and ham. Restricting the vocabulary to 3,000 features yielded a favourable balance between expressiveness and overfitting risk; future studies should evaluate adaptive vocabulary selection guided by chi-squared or mutual information scoring.

Second, the class imbalance between ham and spam (81/19) implies that accuracy alone is an insufficient performance indicator. Precision and confusion-matrix analysis were therefore reported throughout the study, with particular emphasis on the false positive rate. Classifiers achieving 100% precision (MNB, RF, KNN) are of practical interest even when their overall accuracy is slightly below the best-performing model, because legitimate mail loss is generally considered more harmful than missed spam.

Third, the present framework operates exclusively on textual content. Modern spam campaigns increasingly use image-based content, URL redirection through legitimate domains, and stylistic mimicry to evade pure text classifiers. Extending the pipeline with URL reputation

scoring, OCR-based image analysis, and header metadata inspection is therefore expected to substantially improve robustness in adversarial settings [1].

Finally, the Linear SVM model, whose inference cost is linear in the number of non-zero TF-IDF features, scales naturally to high-throughput environments processing millions of messages per day. Heavier ensemble models such as Stacking or Gradient Boosting deliver marginal accuracy gains at the cost of higher inference latency, making them more suitable for asynchronous or batch-processing architectures.

VII. CONCLUSION AND FUTURE WORK

This paper presented a comprehensive and reproducible machine learning framework for email spam detection. By coupling TF-IDF feature extraction with a wide range of supervised classifiers and two ensemble strategies, the proposed pipeline achieved classification accuracies above 98% on a standard benchmark corpus. The Linear SVM model reached 99.9% accuracy on training data and 98.2% accuracy on testing data, while ensemble strategies such as soft voting attained perfect precision on the test partition.

Beyond raw performance numbers, the study highlighted the importance of selecting models based on the operational cost of false positives versus false negatives, rather than on accuracy alone. Future work will pursue three directions. First, multi-modal feature integration incorporating header metadata, URL reputation, and image-based content will be investigated to strengthen the framework against evolving evasion strategies. Second, deep learning architectures based on bidirectional LSTM networks and transformer encoders such as BERT will be benchmarked against the TF-IDF baseline. Third, the framework will be deployed and evaluated in a realistic streaming environment to assess latency, concept drift, and continuous learning capabilities.

REFERENCES

[1] V. Christina, S. Karpagavalli, and G. Suganya, "A study on email spam filtering techniques," International Journal of Computer Applications, vol. 12, no. 1, pp. 7–10, Dec. 2010.

[2] K. S. K. Chaitanya and D. S. R. Krishna, "Email spam filtering using machine learning techniques," B.E. Project Report, Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India, Mar. 2022.

[3] A. Khorsi, "An overview of content-based spam filtering techniques," Informatica, vol. 31, no. 3, pp. 269–277, Oct. 2007.

[4] D. Mallampati and N. P. Hegde, "A machine learning based email spam classification framework model," International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, no. 4, pp. 1752–1757, Feb. 2020.

[5] I. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed. Amsterdam, The Netherlands: Elsevier, 2005.

[6] A. McDonald, SpamAssassin: A Practical Guide to Integration and Configuration. Birmingham, U.K.: Packt Publishing, 2004.

[7] Apache Software Foundation, "Apache SpamAssassin Public Corpus," 2019. [Online]. Available: https://spamassassin.apache.org/old/publiccorpus/

[8] SpamAssassin, "Ham and Spam Dataset," Kaggle, 2018. [Online]. Available:

https://www.kaggle.com/veleon/ham-and-spam-dataset

[9] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in Proc. ECML, 1998, pp. 137–142.

[10] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001.