

Spam Email Classification Using TF-IDF and Classical Machine Learning on the Enron-Spam Corpus

Nagham Kamil Hadi^{1,*}, Aymen Adil²

¹College of Computer Science and Information Technology, University of Al-Qadisiyah, Al Diwaniyah 58002, Iraq.

²Department of Computer Engineering (DISCA), Universitat Politècnica de València, Spain.

*Corresponding Author: Nagham Kamil Hadi

DOI: <https://doi.org/10.31185/wjcms.562>

Received 25 July 2026; Accepted 06 September 2026; Available online 30 September 2026

ABSTRACT: Unsolicited email is a persistent operational and security burden and many high-performing neural approaches require computational and deployment costs that are not necessary for resource-constrained filtering systems. This paper tries to fill this gap by providing a rigorous and reproducible comparison of lightweight classifiers to discriminate spam from legitimate email without overestimating the corpus as a dedicated phishing or malware benchmark. Using 30,089 cleaned and de-duplicated messages from the Enron-Spam corpus, word-level unigram and bigram term frequency-inverse document frequency (TF-IDF) features were evaluated with Multinomial Naive Bayes, Logistic Regression, Linear Support Vector Machine, and Random Forest classifiers. Hyperparameters were selected by grid search on the training partition; stability was assessed through repeated stratified five-fold cross-validation across five random seeds; and paired Logistic Regression and Linear SVM predictions were compared using an exact two-sided McNemar test. On the held-out test set, Logistic Regression achieved the highest accuracy (99.15%) and F1-score (99.13%), followed by Linear SVM (99.05% accuracy; 99.02% F1). Naive Bayes achieved 98.59% accuracy, whereas Random Forest reached 96.69% despite a spam recall of 99.55%. Repeated cross-validation produced identical rounded mean accuracy and macro-F1 for Logistic Regression and Linear SVM (99.08% $\pm$ 0.11%), and their held-out difference was not statistically significant ($p = 0.2101$). These findings support tuned linear TF-IDF models as accurate, stable, and computationally practical spam-filtering baselines, while cross-corpus validation remains necessary.

Keywords: Spam Detection; Malicious Email Classification; Machine Learning; Text Classification; Naive Bayes; Support Vector Machine; Logistic Regression.



1. INTRODUCTION

The relentless growth of email traffic and the ongoing changes in unwanted messages have made automated filtering an active issue in natural language processing and cybersecurity. The Enron corpus was released after the bankruptcy of Enron Corporation and became an influential resource for email-classification research [1]. The spam/ham extension of Metsis *et al.* [2] provided basic evaluations of Bayesian filters [3], [4]. In this study, “spam” is defined as unsolicited bulk or unwanted email and follows the Enron-Spam label. The corpus does not imply that every spam message is phishing, malware-bearing, or otherwise malicious. Recent advances range from classical sparse-text models to dense embedding, deep neural networks, and transformers. TF-IDF provides almost similar results as used by lightweight classifiers, as seen in comparative studies. Yusupov *et al.* achieved 97.87% accuracy with TF-IDF using Random Forest [5]. Al Tawil *et al.* achieved 98% performance using multilayer perceptron with TF-IDF or Word2Vec representations [6]. Chakravorty *et al.* achieved 95.41% accuracy with unigram/bigram TF-IDF and Logistic Regression [7]. These

*Corresponding author: nagham.kamil@qu.edu.iq.
<https://wjcm.uowasit.edu.iq/index.php/wjcm>

findings motivate a controlled comparison of the classical models on a single feature representation and evaluation protocol, especially when computational economy and interpretability are of concern. However, published comparisons are often different in terms of corpus composition, preprocessing, tuning, metrics, and leakage controls, so direct comparison of model rankings is difficult. The present study thus contributes the following to empirical and reproducibility-oriented research:

(1) A leakage-controlled benchmark of four classifier families (probabilistic, linear discriminative, maximum-margin, and tree-ensemble) under an identical TF-IDF representation, plus full spam-class and confusion-matrix metrics for each model;

(2) Empirical evidence that tuned linear TF-IDF classifiers achieve the best accuracy-efficiency trade-off on the cleaned Enron-Spam corpus, with a quantified explanation of why Random Forest's high spam recall is offset by significantly more false positives;

(3) A robustness analysis showing that the leading Logistic Regression and Linear SVM models are stable across several stratified five-fold evaluations and are not significantly different using an exact paired McNemar test; and

(4) A scope-aware interpretation that combines lexical coefficients, error costs, and corpus limitations, thus distinguishing general spam filtering from dedicated phishing or malware detection. The rest of the paper is organized as follows. We discuss related work in Section 2. Section 3 describes the dataset, representation, models, tuning, and evaluation protocol. Section 4 describes the experimental results. Section 5 discusses implications and limitations. Section 6 concludes the paper.

2. RELATED WORK

This foundational work established the methodological vocabulary and the benchmark corpora used in contemporary spam filtering. In a systematic review [8], Cormack formalized the evaluation issues for spam filters. Guzella and Caminhas categorized the probabilistic, rule-based, and kernel approaches [9]. Blanzieri and Bryl studied the learning-based adaptation and the concept drift [10]. Later, Dada *et al.* synthesized the machine-learning methods and identified the ongoing problems with reproducibility and cross-corpus generalization [11]. We keep some of these references older on purpose, since they provide the Enron benchmark, early Bayesian baselines, or canonical evaluation concepts that the current protocol directly builds upon.

2.1 FOUNDATIONS OF NAIVE BAYES IN SPAM FILTERING

Sahami *et al.* [3] introduced Bayesian spam filtering, and Androutsopoulos *et al.* [4] evaluated it thoroughly. Metsis *et al.* then compared five variants of Naive Bayes on six datasets derived from Enron [2], creating the Enron-Spam benchmark used here. These studies are still relevant as methodological baselines but not as evidence for current threat distributions. Such claims about contemporary deployments should be backed up by recent corpora and cross-domain tests.

2.2 COMPARATIVE STUDIES OF TF-IDF AND CLASSICAL MACHINE LEARNING

Recent comparison studies indicate that model ranking is highly sensitive to representation and corpus. Yusupov *et al.* proved the effectiveness of TF-IDF with traditional learners and achieved an accuracy of 97.87% using Random Forest [5]. Alhuzali *et al.* showed that the different datasets of phishing emails have high variability [12]. Kumari *et al.* presented a lightweight TF-IDF – Logistic Regression application for real-time classification [13]. In sum, the studies validate classical models as useful baselines but do not remove the need for common tuning, metrics, and leakage controls.

2.3 PHISHING AND MALICIOUS-CONTENT DETECTION

The problem of phishing and malicious content is a closely related but different task. Sahoo *et al.* [14] study the lexical, host-based, and content-based features of URLs. Al Tawil *et al.* compared the performance of TF-IDF, Word2Vec and BERT for phishing email detection, achieving an accuracy, precision and recall of 98% with their best multilayer perceptron configurations [6]. Moreover, Yi *et al.* showed that interpretation based on SHAP allowed a mean cross-dataset accuracy of more than 94% [15]. Chakravorty *et al.* achieved 95.41% accuracy and a 94.33% F1-score using full-message TF-IDF features [7]. However, the results cannot be directly compared to Enron spam detection because of the differences in the phishing labels and data distributions.

2.4 POSITIONING OF THE PRESENT STUDY

More recent work has investigated richer feature engineering and model families. Alrammahi *et al.* [16] used TF-IDF with engineered textual features and clustering before the supervised classification step. In the case of phishing emails, Melendez *et al.* [17] compared transformers with classical learners. Dwianti *et al.* reported that TF-IDF with Random Forest got 98.01% accuracy on a balanced phishing dataset [18]. Hosseinzadeh *et al.* applied deep learning with adaptive optimization [19]. These studies improve detection performance or representation capacity, but their different labels, datasets, and validation protocols make it impossible to rank them numerically directly against the present Enron-Spam experiment. Table 1 summarizes the closest studies by task, representation, model, and reported result. What we

fill here is not a new classifier architecture but a transparent and reproducible comparison of four lightweight models on one well-defined corpus, including tuned hyperparameters, complete metrics, repeated multi-seed cross-validation, paired significance testing, and explicit separation of spam from verified phishing or malware.

Table 1. Summary of representative related studies and their positioning relative to the present work

Study	Representation	Classifier(s)	Dataset/task	Evaluation focus	Reported result/relevance
Metsis <i>et al.</i> (2006) [2]	Bag of words	Five NB variants	Six Enron-Spam subsets	Foundational benchmark	Performance varied by variant/subset; established Enron-Spam benchmark
Yusupov <i>et al.</i> (2025) [5]	TF-IDF; embeddings	7 classical; 7 deep	Email spam	Representation–model comparison	Random Forest with TF-IDF: 97.87% accuracy Best MLP: 98%
Al Tawil <i>et al.</i> (2024) [6]	TF-IDF; Word2Vec; BERT	MLP and classical ML	Phishing email	Representation comparison	accuracy, precision, and recall
Alhuzali <i>et al.</i> (2025) [12]	Text features	Classical and deep models	Multiple phishing datasets	Cross-dataset comparison	Model ranking changed substantially by dataset
Meléndez <i>et al.</i> (2024) [17]	TF-IDF; embeddings	Classical ML; transformers	Phishing email	Model-family comparison	Classical models remained competitive baselines
Dwianti <i>et al.</i> (2025) [18]	TF-IDF	Random Forest	Balanced phishing dataset	Content-based RF	98.01% accuracy with 80/20 split
Hosseinzadeh <i>et al.</i> (2025) [19]	Learned representations	Deep learning	Phishing email	Adaptive optimization	High-capacity deep-learning alternative
This work	TF-IDF unigrams+bigrams	NB, LR, Linear SVM, RF	Enron-Spam	Tuned, repeated, paired evaluation	LR: 99.15% accuracy; LR vs SVM $p = 0.2101$

3. METHODOLOGY

Fig. 1 summarizes the pipeline of the end-to-end investigation, from raw messages intake to text cleaning, TF-IDF vectorization, stratified train/test split, parallel training of the four candidate classifiers and final evaluation and model selection. Each stage is described in detail in the following subsections.

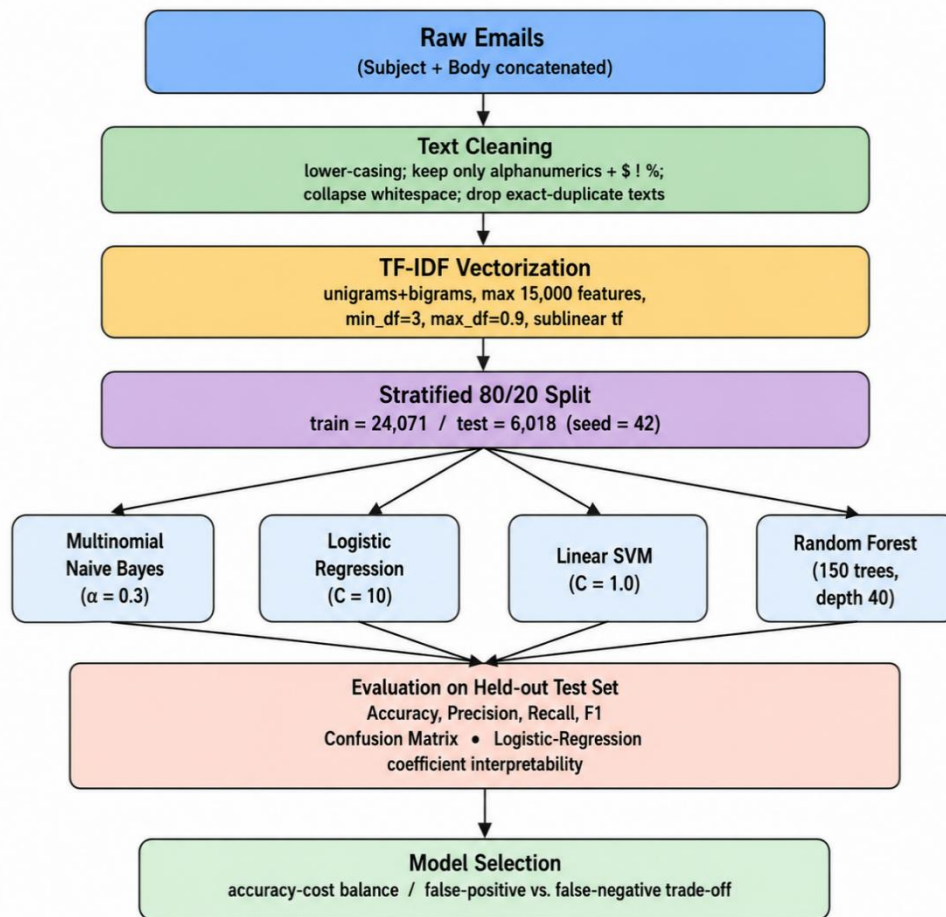


FIGURE 1. Overview of the TF-IDF and machine-learning pipeline used in this study

3.1 DATASET

This study uses a preprocessed, publicly available version of the Enron-Spam corpus, built upon the original Enron corpus [1] and the dataset of Metsis *et al.* [2], and distributed as a single consolidated CSV file [20]. After removing records in which both the subject and message body were empty, and removing messages with an entirely duplicate cleaned text (to prevent data leakage between the training and test sets), the final corpus comprises 30,089 messages; its descriptive statistics are summarized in Table 2 and its class distribution in Figure 2.

Table 2. Descriptive statistics of the Enron-Spam dataset after cleaning and de-duplication

Variable	Count	Percentage
Total messages (after cleaning)	30,089	100.0%
Spam	14,539	48.32%
Ham (legitimate)	15,550	51.68%
Training set (80%)	24,071	—
Test set (20%)	6,018	—

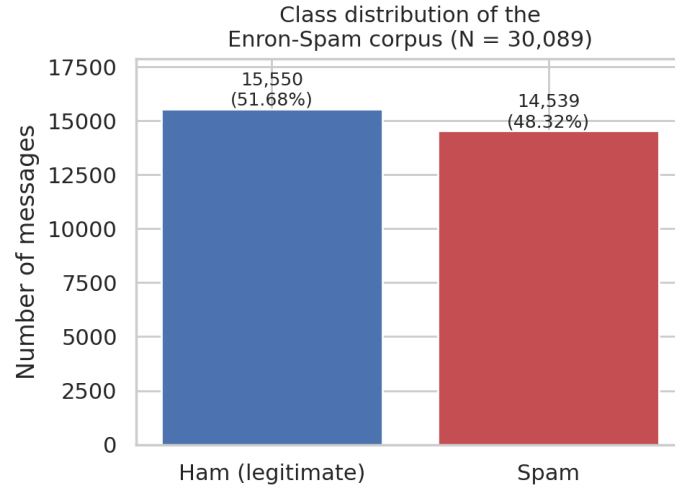


FIGURE 2. Class distribution of the cleaned and de-duplicated Enron-Spam corpus (N = 30,089).

3.2 TEXT PREPROCESSING

The subject and body of each message were concatenated. Text was lower-cased; non-alphanumeric characters were removed except, and %, which were retained as potentially discriminative promotional markers; consecutive whitespace was collapsed; and records with identical cleaned text were de-duplicated before splitting. De-duplication before the train/test division prevents the same normalized message from appearing in both partitions and thereby reduces leakage. The retained symbols were chosen because punctuation and currency markets have been used as engineered spam indicators in recent feature-analysis work [16].

3.3 FEATURE EXTRACTION: TF-IDF

Text features were represented by TF-IDF weighting [21] using scikit-learn's Vectorizer. The configuration used word unigrams and bigrams, a maximum of 15,000 features, min_df = 3, max_df = 0.9, sublinear term-frequency scaling, and English stop-word removal. These settings balance lexical coverage and computational cost in the high-dimensional sparse feature space. With sublinear scaling, the weight assigned to term t in document d is computed as shown in Eq. (1), where $tf(t, d)$ is raw term frequency, N is the number of documents, and $df(t)$ is the number of documents containing t .

$$tfidf(t, d) = (1 + \log tf(t, d)) \times \log \frac{N}{df(t)} \quad (1)$$

3.4 CLASSIFIERS COMPARED

Four classical, lightweight classifiers, all from the scikit-learn library [22], were compared with the following settings:

- Multinomial Naive Bayes: smoothing parameter $\alpha = 0.1$;
- Logistic Regression: $C = 100$, with balanced class weighting;
- Linear SVM: $C = 1.0$, with no class weighting;
- Random Forest: 300 trees, unrestricted maximum depth, with balanced class weighting.

The core decision rule of each classifier is formalized in Eqs. (2)-(5). The Multinomial Naive Bayes classifier assigns the class y that maximizes the posterior probability under the conditional-independence assumption (Eq. 2):

$$\hat{y} = \underset{y \in \{\text{spam}, \text{ham}\}}{\operatorname{argmax}} P(y) \prod_{i=1}^n P(x_i | y) \quad (2)$$

Logistic Regression models the posterior probability of the spam class through the sigmoid $\sigma(\cdot)$ of a linear score, with weight vector w and bias b (Eq. 3):

$$P(y = 1 | \mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + b) = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}} \quad (3)$$

The Linear SVM seeks the maximum-margin separating hyperplane by minimizing the regularized hinge-loss objective below, where C controls the penalty on margin violations (Eq. 4):

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m \max(0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i + b)) \quad (4)$$

Random Forest aggregates the predictions of T decision trees $h_1, \dots, h_T$ (each trained on a bootstrap sample with random feature subsampling) via majority voting (Eq. 5):

$$\hat{y} = \text{mode}\{h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_T(\mathbf{x})\} \quad (5)$$

These four models were selected because they require no GPU or heavy deployment dependencies and cover three major text-classification families: Bayesian, linear discriminative/margin-based, and tree-ensemble methods. Their hyperparameters were not fixed a priori: candidate values were evaluated by stratified five-fold grid search on the training partition only, as detailed below.

3.5 HYPERPARAMETER OPTIMIZATION

To ensure that each classifier is evaluated with systematically chosen rather than ad hoc settings, the main hyperparameters were selected by an exhaustive grid search using stratified five-fold cross-validation on the training partition only, optimizing the macro-averaged F1-score. All tuning was performed strictly within the training set so that the held-out test set played no role in model selection, thereby avoiding information leakage. Table 3 reports, for each classifier, the search space that was explored, the configuration used for the held-out evaluation, and the corresponding cross-validated macro-F1 score. The selected configurations are those used to obtain the test-set results reported in the Experimental Results.

Table 3. Hyperparameter search spaces, selected configurations, and cross-validated macro-F1

Classifier	Hyperparameter search space	Selected configuration	CV macro-F1 (%)
Multinomial Naive Bayes	$\alpha \in \{0.1, 0.3, 0.5, 1.0\}$	$\alpha = 0.1$	98.436
Logistic Regression	$C \in \{0.1, 1, 10, 100\}$; class weight $\in \{\text{none}, \text{balanced}\}$	$C = 100$; balanced	99.073
Linear SVM	$C \in \{0.1, 1, 10\}$; class weight $\in \{\text{none}, \text{balanced}\}$	$C = 1$; none	99.094
Random Forest	trees $\in \{100, 150, 300\}$; max depth $\in \{20, 40, \text{none}\}$; class weight $\in \{\text{none}, \text{balanced}\}$	300 trees; max depth none; balanced	98.349

3.6 EVALUATION PROTOCOL

The data were split, using stratified sampling, into an 80/20 training (24,071 messages) and test (6,018 messages) set; stratification ensures the spam-to-ham ratio is preserved in both sets. The test set was not used at any stage of training or model tuning, and the TF-IDF vectorizer was fit only on the training set. Reported metrics are accuracy, precision, recall, and F1-score, treating the spam class as the positive class, computed from the confusion-matrix counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) as shown in Eqs. (6)-(9):

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (7)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (8)$$

$$F_1 = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

A fixed seed of 42 was used for the held-out split and stochastic model initialization to make that experiment reproducible. This single split was complemented by the repeated multi-seed cross-validation described below. Confusion matrices were reported because false positives and false negatives have different operational costs in email filtering.

3.7 CROSS-VALIDATION AND MULTI-SEED EVALUATION

A single train/test split can give an optimistic or pessimistic estimate depending on which messages happen to fall in the test set. To assess the stability of each model, we therefore complemented the single 80/20 hold-out split with repeated stratified k-fold cross-validation ($k = 5$) on the full corpus, repeated over five independent random seeds. For every model we report the mean and standard deviation of accuracy and macro-F1 across the resulting folds and seeds, as summarized in Table 4.

Table 4. Cross-validation results (mean $\pm$ standard deviation) across five seeds

Classifier	CV accuracy (mean $\pm$ std, %)	CV macro-F1 (mean $\pm$ std, %)
Multinomial Naive Bayes	98.44 $\pm$ 0.16	98.44 $\pm$ 0.16
Logistic Regression	99.08 $\pm$ 0.11	99.08 $\pm$ 0.11
Linear SVM	99.08 $\pm$ 0.11	99.08 $\pm$ 0.11
Random Forest	96.56 $\pm$ 0.25	96.56 $\pm$ 0.25

3.8 STATISTICAL SIGNIFICANCE TESTING

Because the two strongest classifiers differed by only a small held-out margin, their paired predictions were compared using an exact two-sided McNemar test. The test evaluates whether the two discordant error counts are symmetric; $p < 0.05$ was defined as statistically significant.

4. EXPERIMENTAL RESULTS

4.1 OVERALL PERFORMANCE ON THE TEST SET

Table 5 reports held-out performance, and Figure 3 visualizes the same metrics. Logistic Regression achieved the strongest overall result, with 99.15% accuracy and a 99.13% F1-score. Linear SVM followed with 99.05% accuracy and a 99.02% F1-score; Multinomial Naive Bayes reached 98.59% accuracy; and Random Forest reached 96.69%. For Logistic Regression versus Linear SVM, the discordant prediction counts were 11 and 5. The exact two-sided McNemar test yielded $p = 0.2101$, so the observed 0.10-percentage-point accuracy difference was not statistically significant.

Table 5. Classifier performance on the held-out test set (positive class = spam)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Multinomial Naive Bayes	98.59	98.42	98.66	98.54
Logistic Regression	99.15	98.84	99.42	99.13
Linear SVM	99.05	98.94	99.11	99.02
Random Forest	96.69	93.96	99.55	96.68

Accuracy measures the overall correct-decision rate, whereas spam-class precision quantifies the risk of blocking legitimate mail, recall quantifies missed spam, and F1 balances precision and recall. Reporting all four metrics for every classifier is therefore necessary because accuracy alone can conceal operationally important error trade-offs even in this nearly balanced corpus.

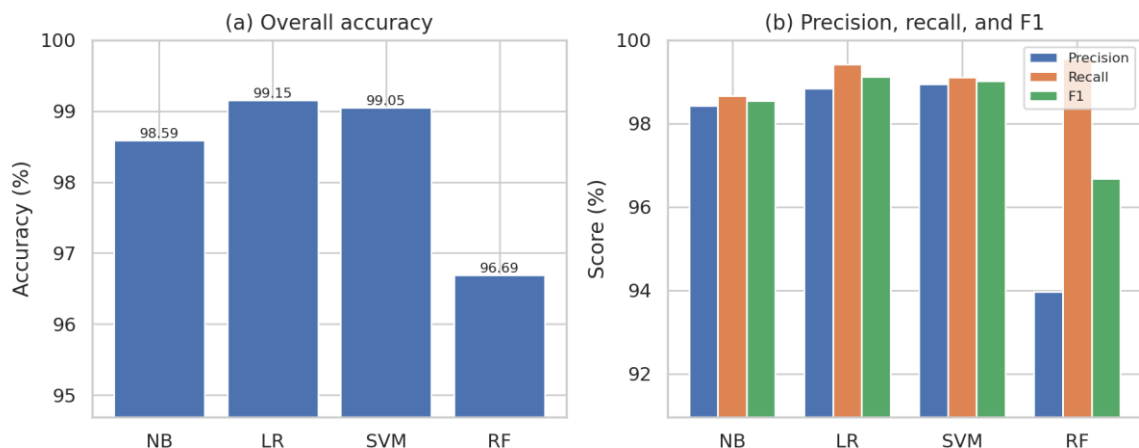


FIGURE 3. Held-out test performance: (a) accuracy; (b) spam-class precision, recall, and F1-score.

4.2 ERROR ANALYSIS: CONFUSION MATRIX

The confusion-matrix counts are shown in Table 6, and the selected Logistic Regression model is shown in Figure 4. Linear SVM produced the fewest false positives (31; 1.00% of 3,110 ham messages), compared with 34 for Logistic Regression. Random Forest produced the fewest false negatives (13) and the highest spam recall (99.55%), but it also generated 186 false positives and consequently the lowest precision (93.96%). Logistic Regression provided the strongest overall balance, with 34 false positives and 17 false negatives. Model selection should therefore reflect the relative costs of blocking legitimate email and allowing spam.

Table 6. Confusion matrix components on the test set (positive class = spam)

Model	True Negative (TN)	False Positive (FP)	False Negative (FN)	True Positive (TP)
Multinomial Naive Bayes	3,064	46	39	2,869
Logistic Regression	3,076	34	17	2,891
Linear SVM	3,079	31	26	2,882
Random Forest	2,924	186	13	2,895

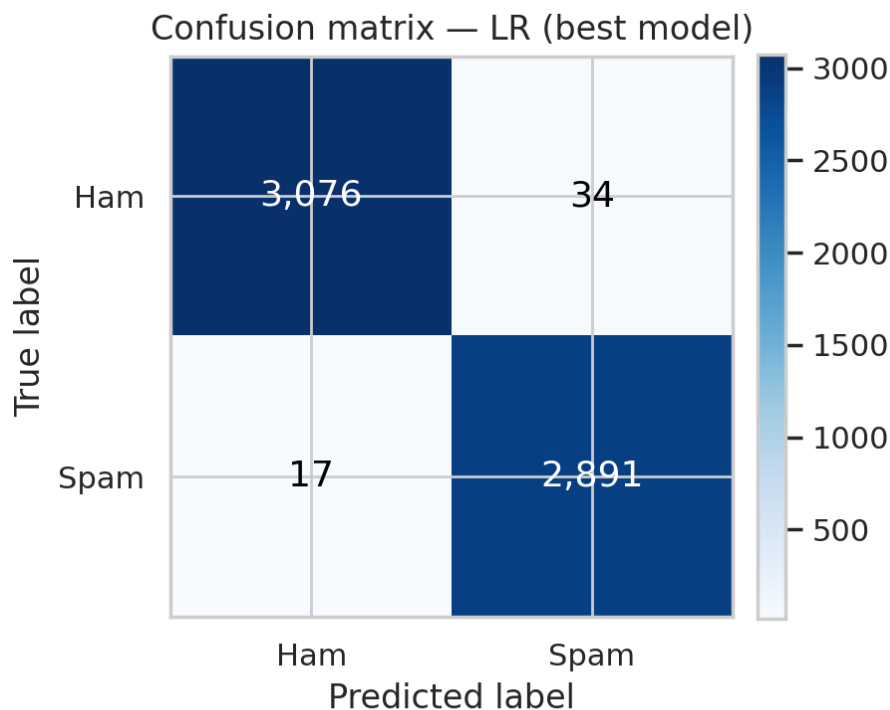


FIGURE 4. Confusion matrix of the selected Logistic Regression model on the held-out test set.

4.3 INTERPRETABILITY: MOST IMPORTANT LEXICAL FEATURES

Extracting the Logistic Regression model's coefficients for each TF-IDF feature, the terms with the largest positive weight (indicative of spam) included words such as “mobile”, “http”, “remove”, “money”, and “software”, consistent with patterns reported in the literature regarding lexical and link-related features of promotional/fraudulent content [14]. Conversely, the terms with the largest negative weight (indicative of ham) were mostly proper names and internal organizational terms specific to the Enron corpus (such as “enron”, “vince”, “attached”, and “thanks”) — a point examined in the Discussion section as an important generalizability limitation.

5. DISCUSSION

Logistic Regression and Linear SVM, separated by 0.10 percentage points in held-out accuracy, provided the strongest accuracy–efficiency trade-off. This is consistent with Chakravorty *et al.* [7], who also found Logistic Regression

stronger than Naive Bayes. In the present study, Logistic Regression exceeded Multinomial Naive Bayes by 0.56 percentage points in accuracy and 0.76 percentage points in spam recall. However, the exact McNemar test found no statistically significant difference between Logistic Regression and Linear SVM ($p = 0.2101$), and repeated cross-validation gave both models the same rounded mean accuracy and macro-F1 ($99.08\% \pm 0.11\%$).

The TF-IDF feature space is high-dimensional and sparse, which is less naturally suited to the axis-aligned splits used by individual decision trees. This helps explain Random Forest's lower accuracy (96.69% versus approximately 99.1% for the linear models) and longer fitting time. The comparison was nevertheless fair at the model-selection level because every classifier, including Random Forest, was tuned by cross-validated grid search. By contrast, linear decision boundaries are well matched to sparse weighted term vectors. The result differs from Yusupov *et al.* [5], who reported Random Forest as their strongest classical model, and from Dwianti *et al.* [18], who reported 98.01% accuracy on a balanced phishing corpus. Differences in labels, preprocessing, vocabulary size, corpus composition, and validation design can change model rankings; numerical results from different studies should not be compared without harmonizing these conditions.

Fit times in the reproduced held-out experiment were approximately 0.03 s for Naive Bayes, 0.73 s for Logistic Regression, 0.36 s for Linear SVM, and 18.92 s for Random Forest on the execution environment used for the notebook. These measurements demonstrate a large within-run fitting-time difference, but they are environment-dependent and should not be interpreted as universal latency guarantees. Hardware specifications and standardized per-message inference benchmarks should be reported in future deployment studies.

From a practical standpoint, the combination of over 99% accuracy and very fast inference makes the linear models a practical and justifiable choice for deployment in real-time filtering systems with limited computational resources particularly in research settings and internal infrastructures where a lightweight, scikit-learn-compatible implementation is the priority.

5.1 LIMITATIONS

First, the Enron corpus reflects the internal communication culture of a single American company in 2001–2002; as the lexical-feature analysis showed, many of the indicators of the “ham” class (such as employees' proper names) are entirely specific to this particular corpus and do not necessarily generalize to today's personal or organizational mailboxes.

Second, the corpus label “spam” primarily denotes unwanted bulk email and does not verify targeted phishing, malicious URLs, or attached malware. Consequently, the results support general spam filtering rather than a dedicated anti-phishing or anti-malware system. Future work should evaluate separate ham, spam, phishing, and malware-bearing categories using dedicated, contemporary corpora.

Third, the static TF-IDF representation cannot detect the semantic rewordings and textual evasion techniques increasingly employed by spammers; approaches based on word embeddings or transformers may be more robust to such variations, albeit at a much higher computational cost and deployment complexity [5, 6].

Fourth, although the present study complements the single 80/20 split with repeated stratified k-fold cross-validation across multiple random seeds and a McNemar test of statistical significance, all experiments are still conducted on a single corpus (Enron-Spam); external validation on other and more recent email collections including corpora that contain verified phishing and malware remains important future work.

6. CONCLUSION

This study compared unigram-and-bigram TF-IDF representations with four classical machine-learning classifiers on the Enron-Spam corpus under a leakage-controlled protocol incorporating hyperparameter tuning, repeated cross-validation, and paired significance testing. Logistic Regression achieved the highest held-out accuracy (99.15%) and F1-score (99.13%), while Linear SVM followed with 99.05% accuracy and a 99.02% F1-score. Their difference was not statistically significant (exact McNemar $p = 0.2101$), and repeated cross-validation produced identical rounded mean performance. Random Forest attained the highest spam recall but lower precision and accuracy, illustrating why complete metrics and confusion-matrix analysis are essential. The findings establish tuned linear TF-IDF models as strong, transparent baselines for lightweight spam filtering, not as validated detectors of phishing or malware. Cross-corpus testing on recent, security-specific email collections is the principal next step.

Funding

None

ACKNOWLEDGEMENT

None

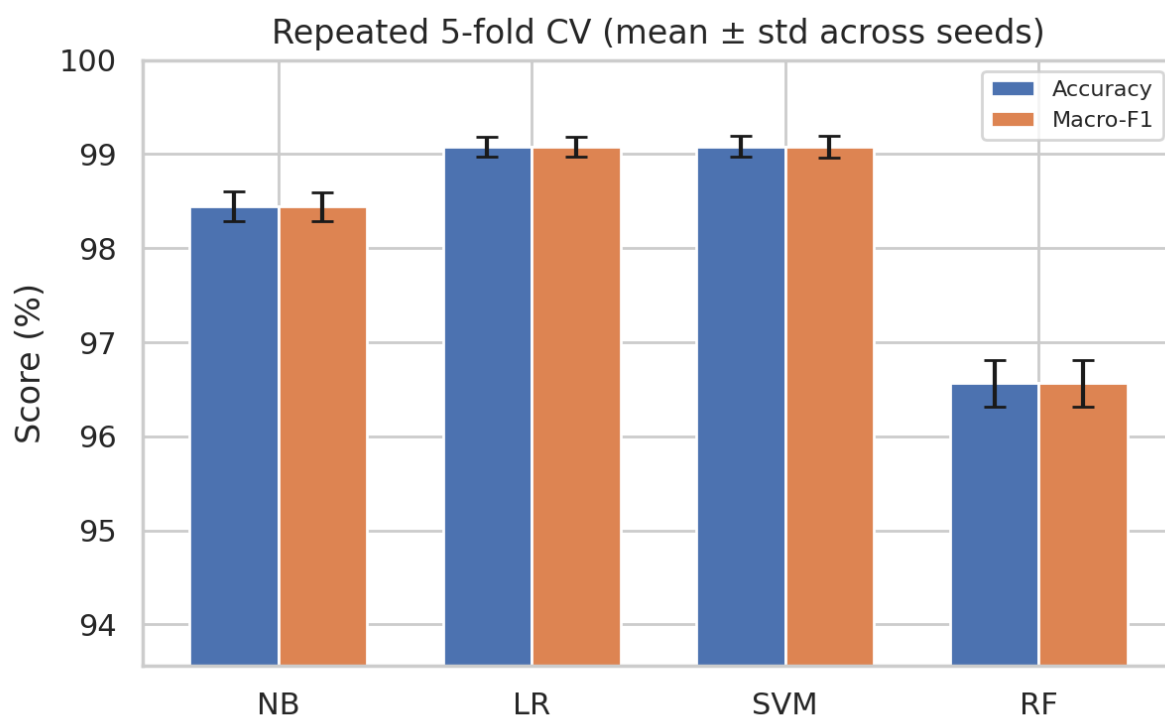
CONFLICTS OF INTEREST

The author declares no conflict of interest.

Code Availability

Code is publicly available at <https://github.com/iraqmoon/enron-spam-classification>.

7. SUPPLEMENTARY MATERIAL



SUPPLEMENTARY FIGURE S1. -Mean accuracy and macro-F1 from repeated stratified five-fold cross-validation across five random seeds; error bars represent standard deviations across folds and seeds.

REFERENCES

- [1] B. Klimt and Y. Yang, "The Enron Corpus: A New Dataset for Email Classification Research," in *Machine Learning: ECML 2004*, J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 217–226. doi: 10.1007/978-3-540-30115-8_22.
- [2] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with Naive Bayes—Which Naive Bayes?" in *Proc. 3rd Conf. Email and Anti-Spam (CEAS)*, Mountain View, CA, USA, Jul. 2006.
- [3] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian approach to filtering junk e-mail," in *Learning for Text Categorization: Papers from the 1998 Workshop*, AAAI Technical Report WS-98-05, Madison, WI, USA, 1998, pp. 55–62.
- [4] I. Androutsopoulos, J. Koutsias, K. V. Chandrinou, G. Paliouras, and C. D. Spyropoulos, "An evaluation of Naive Bayesian anti-spam filtering," in *Proc. Workshop Machine Learning in the New Information Age, 11th European Conf. Machine Learning (ECML)*, Barcelona, Spain, 2000, pp. 9–17.
- [5] K. Yusupov, M. R. Islam, I. Muminov, M. Sahlabadi, and K. Yim, "Comparative Analysis of Machine Learning and Deep Learning Models for Email Spam Classification Using TF-IDF and Word Embedding Techniques," in *Advances on Broad-Band Wireless Computing, Communication and Applications*, L. Barolli, Ed., Cham: Springer Nature Switzerland, 2025, pp. 114–122. doi: 10.1007/978-3-031-76452-3_11.
- [6] A. Tawil, L. Almazaydeh, D. Qawasmeh, B. Qawasmeh, M. Alshinwan, and K. Elleithy, "Comparative Analysis of Machine Learning Algorithms for Email Phishing Detection Using TF-IDF, Word2Vec, and BERT," *CMC*, vol. 81, no. 2, pp. 3395–3412, 2024, doi: 10.32604/cmc.2024.057279.

- [7] A. Chakravorty, M. Price, N. Elsayed, and Z. ElSayed, "Context-Aware Phishing Email Detection Using Machine Learning and NLP," Mar. 28, 2026, *arXiv preprint*: arXiv:2603.27326. doi: 10.48550/arXiv.2603.27326.
- [8] G. V. Cormack, "Email Spam Filtering: A Systematic Review," *Foundations and Trends in Information Retrieval*, vol. 1, no. 4, pp. 335–455, Jun. 2008, doi: 10.1561/15000000006.
- [9] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to Spam filtering," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10206–10222, Sep. 2009, doi: 10.1016/j.eswa.2009.02.037.
- [10] E. Blanzieri and A. Bryl, "A survey of learning-based techniques of email spam filtering," *Artif. Intell. Rev.*, vol. 29, no. 1, pp. 63–92, 2008, doi: 10.1007/s10462-009-9109-6.
- [11] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: review, approaches and open research problems," *Heliyon*, vol. 5, no. 6, p. e01802, Jun. 2019, doi: 10.1016/j.heliyon.2019.e01802.
- [12] A. Alhuzali, A. Alloqmani, M. Aljabri, and F. Alharbi, "In-Depth Analysis of Phishing Email Detection: Evaluating the Performance of Machine Learning and Deep Learning Models Across Multiple Datasets," *Applied Sciences*, vol. 15, no. 6, p. 3396, Jan. 2025, doi: 10.3390/app15063396.
- [13] A. Kumari, S. Kumari, P. Kumari, S. Kumari, and K. Amrendra, "A machine learning and NLP-based approach for efficient email spam detection using TF-IDF and logistic regression," *Int. J. Comput. Techn.*, vol. 13, no. 3, pp. 156–165, May–Jun. 2026.
- [14] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," *arXiv preprint arXiv:1701.07179*, Jan. 2017, doi: 10.48550/arXiv.1701.07179.
- [15] L. Yi, A. Omotosho, and H. Balogun, "Phishing URL Detection and Interpretability With Machine Learning: A Cross-Dataset Approach," *SECURITY AND PRIVACY*, vol. 9, no. 1, p. e70175, 2026, doi: 10.1002/spy2.70175.
- [16] A. A. H. Alrammahi, F. A. O. Sari, Z. A. Muhammad, M. N. Kadhim, D. Al-Shammary, and A. Ibaida, "Enhancing spam detection with advanced feature extraction and unsupervised clustering," *Int. j. inf. tecnol.*, vol. 17, no. 9, pp. 5109–5119, Dec. 2025, doi: 10.1007/s41870-025-02723-6.
- [17] R. Meléndez, M. Ptaszynski, and F. Masui, "Comparative Investigation of Traditional Machine-Learning Models and Transformer Models for Phishing Email Detection," *Electronics*, vol. 13, no. 24, p. 4877, Jan. 2024, doi: 10.3390/electronics13244877.
- [18] I. R. Dwianti et al., "Klasifikasi email phishing menggunakan metode TF-IDF dan algoritma random forest," *ROUTERS: Jurnal Sistem dan Teknologi Informasi*, vol. 3, no. 2, pp. 125–135, Jul. 2025, doi: 10.25181/rt.v3i2.4310.
- [19] M. Hosseinzadeh et al., "Improving phishing email detection performance through deep learning with adaptive optimization," *Sci Rep*, vol. 15, no. 1, p. 36724, Oct. 2025, doi: 10.1038/s41598-025-20668-5.
- [20] M. Wiechmann, "Enron spam dataset," GitHub, 2021. Accessed: Aug. 28, 2026. [Online]. Available: https://github.com/MWiechmann/enron_spam_data.
- [21] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, 1988, doi: 10.1016/0306-4573(88)90021-0.
- [22] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011, [Online]. Available: <http://jmlr.org/papers/v12/pedregosa11a.html>