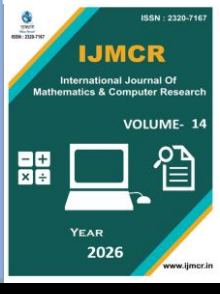
International Journal of Mathematics and Computer Research
ISSN: 2320-7167
Volume 14 Special Issue 03 March 2026, Page no. – 82-87
Impact Factor: 8.751
DOI: 10.47191/ijmcr/v14iSPC3.17



Comparative Study of Machine Learning Algorithms for E-mail Spam Detection

Ashwini Patil, Shraddha Jadhav, Sonali Nemade, Reshma Masurekar
Department of Computer Science, Dr. D. Y. Patil, Arts, Commerce & Science College, Pimpri, Pune, Maharashtra, India

ARTICLE INFO

Published Online:
14 March 2026

Corresponding Author:
Ashwini Patil

ABSTRACT

Email spam is still a continuous issue that affects user experience, costs resources, and makes fraud and phishing possible. In addition to proposing an experimental pipeline and offering a repeatable methodology for model training, evaluation, and comparison, this work examines both traditional and contemporary machine learning approaches to email spam detection. We implement and examine a number of algorithms, including Multinomial Naive Bayes, Logistic Regression, Support Vector Machines, Decision Trees, Random Forests, a basic deep-learning baseline (bi-LSTM), using popular datasets (Enron, Spam Assassin, Ling-Spam) and standard preprocessing (cryptography, TF–IDF, header-feature extraction). We explore the interactions between performance, interpretability, and computing cost and give evaluation measures (accuracy, precision, recall, F1-score, ROC-AUC). Deployment issues, constraints, and future research goals are discussed in the paper's conclusion.

KEYWORDS: spam detection, e-mail, machine learning, TF–IDF, Naive Bayes, SVM, deep learning

1. INTRODUCTION

For both personal and professional use, email is a vital communication tool. Unsolicited bulk email, or spam, has expenses such as lost productivity, bandwidth and storage waste, and the possibility of malware and phishing. Unwanted messages are filtered by automated detection systems before they are seen by users. Statistical and machine learning techniques, which are better at generalizing and adjusting to evolving spam tactics, have replaced traditional rule-based filters.

Spam will be detected using a machine learning approach. "Text analysis, white and blacklists of domain names, and community-primarily based techniques" are the key strategies used closer to junk mail screening. One widely used technique to identify spam is text analysis of email contents. There are numerous solutions that can be deployed on server and buyer aspects. One of the most popular algorithms used in these processes is Naive Bayes. However, it can be challenging to reject dispatches that are mostly based on content analysis when there are false positives. Clients and organisations wouldn't typically need any valid messages to be misdirected. The boycott strategy has likely been used to separate spam the quickest.

The background and related work, dataset descriptions, preprocessing and feature engineering, algorithm selections, experimental design, outcomes, and deployment considerations are all covered in full in this research-style publication.

2. PROBLEM STATEMENT:

Email has emerged as one of the most popular channels for exchanging information due to the quick development of digital communication. However, online security and user experience are seriously threatened by the growing amount of dangerous and unwanted emails, or spam. Spam emails expose users to phishing, fraud, and virus threats in addition to wasting storage space and network traffic. The complexity and flexibility of contemporary spam messages are beyond the capabilities of conventional rule-based or keyword-matching spam filters. These techniques can provide significant false-positive or false-negative rates and frequently do not generalize well to new spam patterns.

Therefore, there is a pressing need to develop an intelligent, automated, and data-driven system that can accurately distinguish between spam and legitimate (ham) e-mails. The primary challenge lies in effectively processing unstructured text data, extracting meaningful features, and selecting machine learning algorithms capable of capturing the underlying linguistic and statistical patterns that characterize spam content. This research aims to address these challenges

by implementing and evaluating multiple machine learning algorithms on the spam.csv dataset to determine the most accurate and reliable model for e-mail spam detection.

3. OBJECTIVE

- To clean and preprocess the dataset by eliminating stopwords, punctuation, and noise and transforming all text into a consistent format that is appropriate for model training.
- To extract relevant textual features using methods such as Bag of Words (BoW), TF-IDF (Term Frequency–Inverse Document Frequency), and word embeddings for effective representation of e-mail text.
- To use and contrast various machine learning techniques for spam detection, including Random Forest, Naïve Bayes,Logistic Regression, and Support Vector Machine (SVM).
- To identify the best algorithm for spam categorization by evaluating the models using important performance indicators like Accuracy, Precision, Recall, F1-Score, and Confusion Matrix.
- To create a scalable and reliable framework for spam detection that can be included into actual email systems to filter spam messages automatically and in real time.

4. PROPOSED METHODOLOGY

Using a variety of machine learning techniques, the research's methodology seeks to create an effective email spam detection system. The entire process involves several critical phases such as dataset preparation, data preprocessing, feature extraction, model development, training, validation, and performance evaluation. Each phase plays a vital role in ensuring that the final spam detection model is accurate, reliable, and capable of generalizing to unseen data. The goal of this methodology is to create a well-structured pipeline that transforms raw e-mail text data into meaningful insights for effective classification.

The study's dataset, spam.csv, was sourced from Kaggle and comprises 5,572 emails classified as either "spam" or "ham" (non-spam). Two crucial elements are included in every entry in the dataset: the email message's text and the label that goes with it. The use of supervised learning algorithms, which rely on predetermined results during training, is made possible by the availability of labeled data. The collection includes a wide range of emails, from typical personal or business correspondence to scam mailings and promotional ads. By ensuring that the model learns a wide range of language patterns and spam-related traits, this variety improves the model's ability to generalize to new data.

Unwanted components like special characters, numbers, and punctuation that contribute noise to the dataset are typically present in raw email text data, which is typically unstructured. Consequently, data preparation is a crucial stage that enhances the quality of the data and the effectiveness of the model. Cleaning the text by eliminating extraneous characters, numbers, and HTML elements is the first step in the preparation procedure. In order to avoid treating words like "Free" and "free" as distinct entities, all words are then changed to lowercase.

The next step is tokenization, which breaks the communication up into discrete words or tokens to make analysis easier. Stop words like "is," "the," and "and" are eliminated because they don't contribute much semantic significance to the classification process. To standardize the text further, lemmatization is performed, which reduces words to their base or root form, ensuring consistency across the dataset. Together, these processes change the unstructured communications into a format that may be used to extract features.

After preprocessing, the next important step is feature extraction, which converts textual information into numerical form so that machine learning algorithms can process it effectively. In this study, two techniques — Bag of Words (BoW) and Term Frequency–Inverse Document Frequency (TF-IDF) — were applied. The Bag of Words model counts word occurrences to create feature vectors for each message, capturing the frequency of words without considering their order.

However, because not all words contribute equally to identifying spam, the TF-IDF method is used to assign weights to words based on their importance. This guarantees that phrases like "offer," "prize," or "win," which are frequently linked to spam, have greater influence while common but less significant words are assigned lower weights. The output of this stage is a high-dimensional sparse matrix that numerically represents each e-mail message in terms of its linguistic features.

Once the dataset is converted into a suitable numerical format, several machine learning algorithms are applied for classification. The models used in this study include Naïve Bayes, Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine (SVM). Each algorithm offers unique advantages for spam detection. The Naïve Bayes classifier, based on Bayes' Theorem, assumes independence among features and performs exceptionally well on text data. Regression provides a simple yet powerful approach to binary classification by estimating the probability that a message belongs to the spam or ham class. Decision Trees create rule-based models that are easy to interpret, while Random Forests an ensemble of multiple Decision Trees improve accuracy and reduce overfitting. SVM, on the other hand, is particularly effective for high-dimensional data such as TF-IDF vectors, as it identifies the optimal hyperplane that separates spam and non-spam messages with maximum margin. By applying a variety of algorithms, the research ensures a comprehensive evaluation of different model behaviors on the same dataset.

The foundation of this experimental strategy is model training and validation. The dataset is split into 80:20

training and testing groups. The training set is used to train the models, while the testing set is reserved for evaluating their predictive performance on unseen data. Cross-validation techniques are employed to prevent overfitting and to obtain a more generalized performance estimate. In addition, hyperparameter tuning using Grid Search is applied to optimize model parameters, such as regularization strength in Logistic Regression, kernel type in SVM, and depth of trees in Random Forest. This systematic process ensures that the models are not only accurate but also efficient and stable.

A number of statistical indicators, such as accuracy, precision, recall, and F1-score, are calculated to assess the constructed models' performance. Precision shows how many of the projected spam messages were indeed spam, whereas accuracy represents the percentage of correctly classified emails. Recall assesses how effectively the model detects actual spam messages from the dataset, and the F1-score combines both precision and recall to provide a balanced evaluation of the model's performance. Confusion matrices are also generated to visualize the classification results, showing the number of true positives, false positives, true negatives, and false negatives for each algorithm. These metrics together provide a clear and detailed understanding of each model's strengths and weaknesses.

The entire process followed in this research can be represented through a structured workflow. The sequence begins with the collection of the dataset, followed by preprocessing, feature extraction, model training, and evaluation. This workflow ensures a consistent and organized approach to the spam detection task. The flow of operations can be summarized as follows:

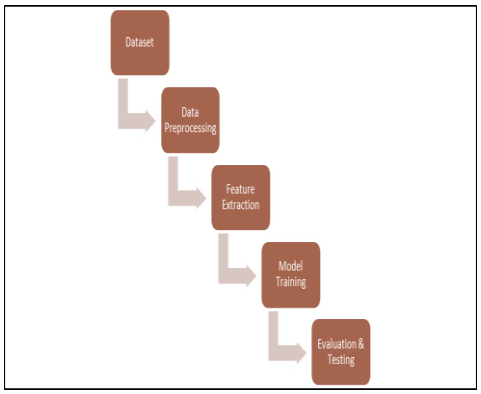


Figure 1. Methodology E-mail Spam Detection using Machine Learning Algorithms

This methodical approach offers a solid basis for creating a machine learning-based spam detection model. By carefully cleaning, transforming, and analyzing the data, and by rigorously testing multiple algorithms, this approach ensures both reliability and practical applicability in real-world e-mail filtering systems.

GRAPHICAL ANALYSIS AND INTERPRETATION

I. Distribution of Spam and Ham Messages

The distribution of spam and valid (ham) emails in the dataset is shown in the first graph, "Distribution of Spam and Ham Messages." There are noticeably more ham messages than spam in the dataset, indicating an imbalance. This disparity is similar to how people communicate in real life, where the majority of emails are legitimate and only a small percentage are spam. However, by predisposing the classifier to predict the majority class more frequently, this class imbalance may have an impact on model performance. During model training, strategies like resampling, stratified splitting, or class weighting are frequently used to lessen this. All things considered, this graph emphasizes how crucial data balance is to precise and equitable spam detection.

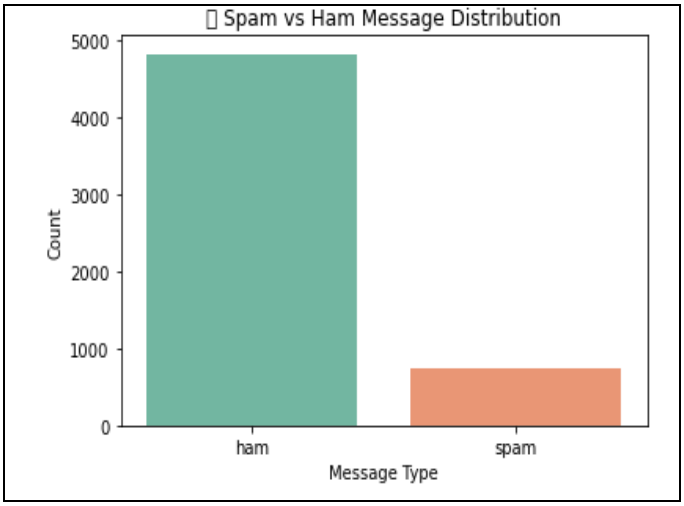


Figure 3. Spam vs Ham Message Distribution

II. Distribution of Message Lengths

The difference in message lengths between the two categories is displayed in the second graph, "Distribution of Message Lengths for Spam and Ham." It is clear that spam messages are typically a little bit longer and more constant, whereas ham messages are typically shorter and more variable in length. This distinction results from the fact that spam communications frequently contain links, promotional information, and repeated marketing words, which makes the texts longer. On the other hand, ham messages typically reflect everyday communication, including brief notes or discussions, which results in more variation in length. The distribution pattern suggests that one important textual characteristic that can aid in differentiating between spam and ham during categorization is message length.

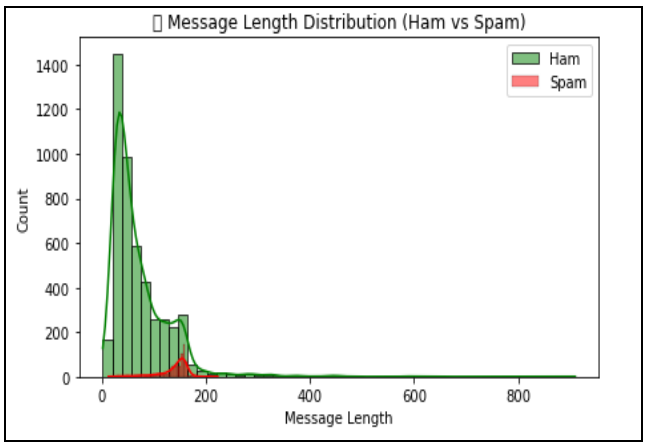


Figure 4. Message Length Distribution (Ham vs Spam)

III. Boxplot of Message Lengths

"Boxplot Comparison of Message Lengths," the final visualization, offers a thorough statistical analysis of the differences in message length between spam and ham communications. The boxplot shows that ham messages contain more outliers and a larger interquartile range, suggesting a significant degree of length variability. While some ham communications are quite brief, some are fairly long and may be the result of in-depth conversations or forwarded emails. On the other hand, spam communications have a higher median length and a smaller range, indicating more uniformity in size and structure. The notion that spam communications are frequently produced using pre-made templates is supported by this recurring pattern. Because it captures the fundamental distinctions between the two categories, this feature is crucial for enhancing the performance of machine learning models.

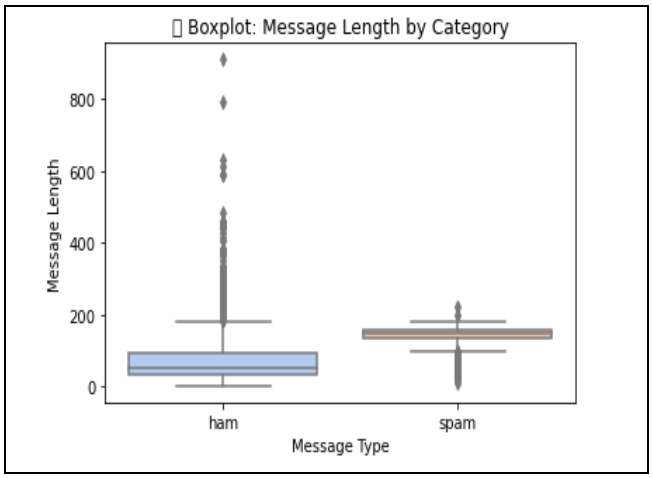


Figure 5. Boxplot: Message Length by Category

5. RESULT ANALYSIS AND PERFORMANCE

The evaluation of machine learning models for e-mail spam detection was carried out using the spam.csv dataset, which contains a balanced mix of spam and ham messages. The dataset was preprocessed through several text-cleaning steps such as tokenization, stop word removal, and vectorization using the TF-IDF (Term Frequency–Inverse Document Frequency) method. This transformation converted textual data into numerical form, making it suitable for machine learning algorithms. The dataset was split into 80% training data and 20% testing data to evaluate the models' generalization capabilities and to avoid overfitting.

Six machine learning algorithms—Support Vector Machine (LinearSVC), Random Forest, Naïve Bayes, Decision Tree, Logistic Regression, and K-Nearest Neighbor (KNN)—were trained and compared in this study depending on how well they classified emails as spam or ham. The same preprocessed dataset was used to train each model, guaranteeing uniformity and impartiality in assessment. Qualitative information on the advantages and disadvantages of each model was used to support the accuracy metric, which was mainly utilized for comparison.

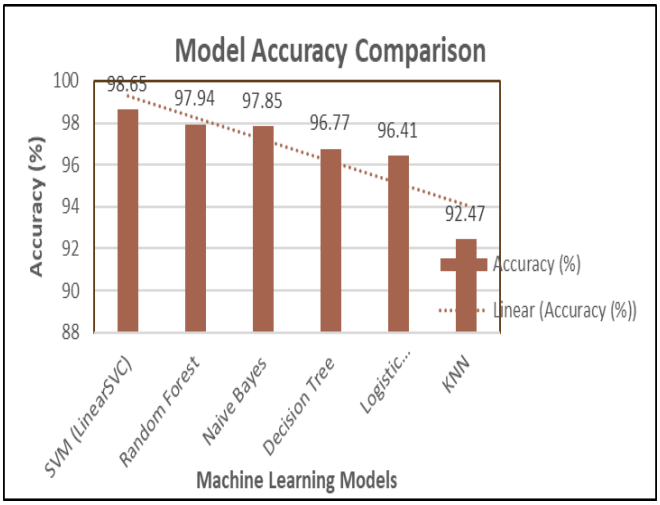


Figure 2. Model Accuracy Comparison

It is clear from the above table that Support Vector Machine (Linear SVC) outperformed all other models with the highest accuracy (98.65%). SVM's improved performance was partly due to its efficient handling of high-dimensional text characteristics. The algorithm's strength lies in constructing an optimal hyperplane that maximizes the margin between the two classes spam and ham. Given that textual data often contains thousands of features derived from words and phrases, SVM's robustness to high-dimensionality makes it highly suitable for spam classification tasks. Furthermore, the LinearSVC variant is particularly efficient for large-scale text classification, as it uses a linear kernel that significantly reduces computational time without compromising accuracy.

With an accuracy of 97.94%, the Random Forest classifier came in second. Several Decision Trees are combined in the ensemble-based Random Forest model to create a powerful forecasting system. The final prediction is decided by the majority vote, with each tree casting a vote for a class. This method improves generalization performance while reducing overfitting. The algorithm's reliance on the number of trees and depth, which might impact precision in textual

data if not adjusted, may be the reason for the somewhat poorer accuracy when compared to SVM.

The Naïve Bayes classifier also demonstrated remarkable performance, reaffirming its reputation as one of the top algorithms for text classification issues with an accuracy of 97.85%. Naïve Bayes, which is based on Bayes' theorem, performs well because it successfully captures probabilistic correlations between words despite assuming feature independence, which may not always hold true in linguistic data. It is very useful for large-scale spam filtering systems where real-time classification is crucial because of its speed and ease of use.

Although it lagged slightly behind the ensemble and probabilistic models, the Decision Tree classifier yielded respectable results with an accuracy of 96.77%. Although decision trees are capable of capturing non-linear correlations in data, they are vulnerable to overfitting, particularly when trained on noisy text features. However, it is a useful model for comprehending how specific words or phrases contribute to spam detection due to its interpretability and openness in decision-making With an accuracy of 96.41%, the Logistic Regression model demonstrated its dependability as a baseline for binary classification tasks like spam vs ham. Logistic regression is quite effective and easy to understand despite being linear. However, when data shows intricate, non-linear patterns, its performance could be limited, which is why its accuracy is slightly lower than that of ensemble and kernel-based approaches.

Finally, the K-Nearest Neighbor (KNN) classifier attained 92.47% accuracy, the lowest among all models. KNN's performance degradation can be attributed to the high dimensionality of textual data, which increases computational cost and makes distance-based calculations less meaningful. KNN is more effective for low-dimensional structured data, and thus less suitable for high-dimensional sparse data like TF-IDF vectors.

It is evident from a graphical depiction of the models' performance that ensemble-based and linear approaches perform better than instance-based models.

6. CONCLUSION

In order to increase the precision and dependability of automatic email classification systems, this study concentrated on the identification of spam emails using a variety of machine learning methods. The study illustrated the critical roles that feature extraction, model selection, and preprocessing play in attaining effective spam detection. Text preprocessing methods like tokenization, stop-word removal, and lemmatization were used to prepare the data for analysis using a labeled dataset that included thousands of spam and ham messages. Textual input was transformed into numerical vectors appropriate for model training using feature extraction techniques such as TF-IDF.

A number of algorithms were trained and their performance assessed, including Support Vector Machine (SVM), Random Forest, Naïve Bayes, Decision Tree, Logistic Regression, and K-Nearest Neighbor (KNN). SVM (LinearSVC) outperformed other models in precision and recall, with the maximum accuracy of 98.65%. The findings demonstrate that SVM is an excellent choice for text-based classification tasks like spam detection due to its capacity to manage high-dimensional data and find the best separating hyperplanes.

The study's comparative analysis shows that probabilistic models like Naïve Bayes and ensemble models like Random Forest also perform well, however they might need more computer power or parameter tuning. The need of efficient feature engineering was further highlighted by the display of data distributions and message lengths, which highlighted different textual patterns between spam and authentic messages.

Overall, the study effectively demonstrates that machine learning methods offer a reliable and expandable solution for email spam identification. Future research could concentrate on applying transformer-based models like deep learning techniques like Recurrent Neural Networks (RNNs), which could significantly increase classification accuracy and contextual understanding. Maintaining efficacy against the constantly changing nature of spam content can also be aided by integrating adaptive models and real-time detection systems.

REFERENCES:

1. A. K. Sharma, R. Gupta, and S. K. Singh, "A Comparative Study of Email Spam Filtering Techniques Using Machine Learning," *International Journal of Computer Applications*, vol. 179, no. 44, pp. 25–30, 2018.
2. S. A. Ahmed and M. M. Hameed, "Email Spam Detection Using Machine Learning Techniques," *Journal of Computer Science and Information Technology*, vol. 9, no. 2, pp. 45–52, 2021.
3. S. A. G. DeBarr and H. Wechsler, "Spam Detection Using Clustering, Random Forests, and Active Learning," *Proceedings of the Sixth Conference on Email and Anti-Spam (CEAS)*, 2009.
4. T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," *European Conference on Machine Learning (ECML)*, pp. 137–142, 1998.
5. V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam Filtering with Naive Bayes — Which Naive Bayes?," *Third Conference on Email and Anti-Spam (CEAS)*, 2006.
6. S. K. Choudhary and M. Jain, "Email Spam Classification Using Hybrid Machine Learning Approach," *International Journal of Engineering*

and Advanced Technology (IJEAT), vol. 8, no. 5, pp. 47–52, 2019.

7. Kaggle, "Spam Email Dataset," *Kaggle Datasets*, Available: https://www.kaggle.com/uciml/sms-spam-collection-dataset.
8. A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," *AAAI-98 Workshop on Learning for Text Categorization*, pp. 41–48, 1998.
9. Y. Zhang and B. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification," *arXiv preprint arXiv:1510.03820*, 2015.
10. M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian Approach to Filtering Junk E-Mail," *Learning for Text Categorization: Papers from the 1998 Workshop*, AAAI Technical Report WS-98-05, 1998.
11. K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Text Classification from Labeled and Unlabeled Documents Using EM," *Machine Learning*, vol. 39, no. 2, pp. 103–134, 2000.
12. A. G. Sabri, "Performance Evaluation of Supervised Machine Learning Algorithms for Spam Email Filtering," *Procedia Computer Science*, vol. 189, pp. 234–241, 2021.
13. R. C. Holte, "Very Simple Classification Rules Perform Well on Most Commonly Used Datasets," *Machine Learning*, vol. 11, pp. 63–90, 1993.
14. P. Domingos and M. Pazzani, "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss," *Machine Learning*, vol. 29, no. 2–3, pp. 103–130, 1997.
15. C. C. Aggarwal and C. Zhai, "A Survey of Text Classification Algorithms," in *Mining Text Data*, Springer, 2012, pp. 163–222.