# Email Spam Detection Using Machine Learning and Natural Language Processing

[1] Fatima Al Mansoori, [2] Zhang Lei
[1] United Arab Emirates University, Al Ain, UAE
[2] Zhejiang University, Hangzhou, China
**Corresponding E-mail:** fatima.mansoori@interviauniversity.com

**Abstract**

Email is one of the most common forms of digital communication used by individuals, businesses, educational institutions, and organizations. At the same time, the growing number of unwanted and potentially harmful emails has become a major problem for both users and email service providers. Spam messages can include advertisements, fake offers, misleading information, phishing attempts, or dangerous links and attachments. Automatically identifying these messages is therefore important for keeping email communication safe and manageable. Machine learning and Natural Language Processing (NLP) provide useful ways to detect spam by learning patterns from email content. This research compares four machine learning algorithms for email spam detection: Naive Bayes, Logistic Regression, Support Vector Machine (SVM), and Random Forest. The proposed process includes collecting the data, preprocessing the email text, extracting features using Term Frequency-Inverse Document Frequency (TF-IDF), dividing the dataset, training the models, and evaluating their performance. Accuracy, precision, recall, and F1-score are used to compare the models. The analysis shows that the quality of text preprocessing and feature representation has a strong effect on classification results. Naive Bayes offers a fast and simple starting point, while Logistic Regression and SVM are well suited to high-dimensional text data. Random Forest provides an ensemble-based alternative that can learn nonlinear patterns. Overall, machine learning can be used effectively for automated spam filtering, although changing spam techniques, misleading messages, and increasingly sophisticated phishing emails continue to create challenges.

**Keywords:** Email Spam Detection, Machine Learning, Natural Language Processing, Naive Bayes, Logistic Regression, SVM, Random Forest, Text Classification

## 1. Introduction

Email has become an essential communication method for personal, professional, academic, and business purposes. People and organizations exchange a huge number of emails every day, making email an important part of modern digital communication. However, the widespread use of email has also led to a continuous increase in unwanted messages, commonly referred to as spam[1].

Spam emails are unsolicited messages that are usually sent to many recipients. They can contain advertisements, fake promotions, fraudulent requests, misleading information, phishing links, or harmful attachments [2]. While some spam messages are simply annoying, others can result in financial loss, identity theft, privacy problems, or security incidents.

Traditional email filtering systems often depend on manually defined rules. For example, a message may be marked as spam if it contains certain keywords, suspicious links, or an unusual sender address [3]. These rules can work well for known spam patterns, but they may become less effective when spammers change their wording or develop new ways to avoid detection.

Machine learning provides a more flexible solution because it can learn patterns from previously labeled emails. A model can be trained using examples of both spam and legitimate messages and then use what it has learned to classify new emails. This means that the system does not need every possible spam rule to be manually written. Instead, it can learn combinations of words and other characteristics that are commonly associated with spam[4].

Natural Language Processing is especially useful in this area because much of the information needed for classification is found in the text of an email. NLP techniques can convert raw email content into structured numerical features that machine learning algorithms can understand.

Several algorithms can be applied to spam classification. Naive Bayes is widely used in text classification because it is simple and efficient. Logistic Regression can work well with large numbers of text features. Support Vector Machine can create effective decision boundaries in

high-dimensional feature spaces, while Random Forest can identify nonlinear relationships by combining multiple decision trees [5].

The main aim of this research is to compare these machine learning techniques for email spam detection. The study focuses on text preprocessing, feature extraction, model development, evaluation metrics, and the main strengths and limitations of the selected algorithms[6].

## 2. Methodology

The research uses a supervised machine learning classification approach. Each email is placed into one of two categories:

- Spam
- Legitimate (Ham)

The overall methodology includes the following stages:

1. Dataset collection
2. Email text preprocessing
3. Feature extraction
4. Dataset splitting
5. Model training
6. Classification
7. Performance evaluation

Four algorithms are compared: Naive Bayes, Logistic Regression, Support Vector Machine, and Random Forest [7].

### 2.1 Research Approach

The proposed system treats spam detection as a binary classification problem. The model learns from emails that have already been labeled as either spam or legitimate and then uses those learned patterns to classify new messages.

The process begins with collecting an appropriate dataset and preparing the email text. After preprocessing, useful numerical features are extracted from the messages. The data is then divided into training and testing portions [8]. The training set is used to build the models,

while the testing set is kept separate so that the models can be evaluated on emails they have not previously seen.

### 2.2 Dataset

A labeled email dataset can be used to evaluate the proposed approach. The dataset should contain examples of both spam and legitimate emails, with each message linked to its corresponding class label.

It is important for the dataset to contain enough examples from both categories. A reasonably balanced dataset can make model comparison easier, although real email systems may contain very different proportions of spam and legitimate messages.

The dataset may include the subject and body of each email. If available, additional information such as sender characteristics, message length, number of links, or attachment details can also be considered.

The data is divided into training and testing subsets. The training data is used to learn patterns from the emails, while the testing data is used to check how well the models classify messages that were not included during training.

### 2.3 Text Preprocessing

Raw email messages often contain HTML code, punctuation, URLs, email addresses, unusual characters, and other information that may not be directly useful for classification. Text preprocessing is therefore used to create a cleaner and more consistent representation of the messages[9].

The preprocessing process can include converting text to lowercase, removing unnecessary punctuation, handling HTML elements, and dealing with irrelevant characters. URLs and email addresses can also be converted into standardized tokens instead of being removed completely, since their presence may provide useful clues when identifying spam.

Tokenization divides an email into individual words or tokens. Stop words may also be removed, although this should be done carefully because some common words can still contribute to the meaning of a message [10].

Stemming or lemmatization can be used to reduce different forms of the same word to a common representation. For example, words such as "buying," "bought," and "buy" may be converted to a common base form.

Preprocessing needs to be handled carefully because removing too much information can actually hurt classification performance. Words related to urgency, financial requests, suspicious offers, or other common spam patterns may provide useful signals and should not be removed without considering their importance[11].

**2.4 Feature Extraction**

Machine learning algorithms work with numerical data rather than raw text. Therefore, the processed email messages need to be converted into numerical features before the models can use them.

A commonly used method is Term Frequency-Inverse Document Frequency (TF-IDF). This technique gives weights to words according to how often they appear in a particular email and how frequently they occur across the entire dataset.

A word that appears often in one message but is relatively uncommon across the dataset may receive a higher weight. This helps the model identify terms that can be useful for distinguishing spam emails from legitimate ones.

Another option is the Bag-of-Words method, which represents an email according to the words it contains. It is simple and useful, but it does not fully represent word order or context[12].

For traditional machine learning approaches to spam detection, TF-IDF is a practical feature representation because it can handle large amounts of text while remaining computationally manageable.

**2.5 Naive Bayes**

Naive Bayes is a probabilistic classification algorithm that is widely used for text-based classification. It estimates the probability that an email belongs to the spam or legitimate category based on the features found in the message [13].

One of the main advantages of Naive Bayes is its speed. It can train and classify large numbers of emails relatively quickly, which makes it useful for systems that need to process many messages.

Its main limitation is the assumption that features are conditionally independent when the class is known. In natural language, this assumption is not always realistic because the meaning of one word can depend on other words around it. Even with this limitation, Naive Bayes can still provide a useful and effective baseline for spam detection.

**2.6 Logistic Regression**

Logistic Regression is a supervised classification algorithm that can be used to estimate whether an email belongs to the spam or legitimate category.

When combined with TF-IDF features, Logistic Regression can handle a large number of text features efficiently. It is also relatively easy to interpret compared with more complicated models, and its coefficients can provide an indication of which features are associated with spam classification.

However, Logistic Regression generally relies on a linear decision boundary. Because spam messages can contain complicated combinations of words and patterns, the model may not capture every relationship unless suitable feature engineering is applied.

**2.7 Support Vector Machine**

Support Vector Machine is another powerful method for distinguishing spam emails from legitimate messages. The algorithm attempts to find a decision boundary that separates the two classes as effectively as possible.

Email datasets can contain thousands of features because each different word or text feature can contribute to the input. SVM is well suited to this type of high-dimensional feature space and can create a strong separation between spam and legitimate messages.

SVM can also be adapted to nonlinear relationships through kernel functions. However, selecting appropriate parameters is important, and the computational cost can increase when working with very large datasets.

**2.8 Random Forest**

Random Forest is an ensemble learning method that combines multiple decision trees. Each tree is trained using different subsets of the observations and features, and their individual predictions are combined to produce the final classification.

This approach allows Random Forest to capture nonlinear relationships between different email features. It may therefore identify patterns that are difficult for a simple linear model to represent.

Random Forest can also provide feature-importance information, which can help researchers understand which words or other characteristics contribute to spam predictions. However, email text can produce a very large number of sparse features, so Random Forest may be less efficient than algorithms such as Naive Bayes or Logistic Regression for very large text datasets.

**3. Results and Discussion**

The comparison shows that machine learning algorithms can successfully distinguish between spam and legitimate emails when suitable preprocessing and feature extraction methods are used.

Naive Bayes provides a useful baseline because it is simple, fast, and well suited to many text-classification problems. Its probabilistic approach allows it to learn relationships between words and spam labels efficiently. However, its assumption that features are independent limits its ability to represent relationships between words.

Logistic Regression is another effective option for text classification. When combined with TF-IDF, it can handle high-dimensional text data efficiently and remains relatively easy to interpret. Its main limitation is that its linear decision boundary may not represent every complex pattern found in spam messages[14].

SVM is particularly useful for high-dimensional text data. It can create a strong separation between spam and legitimate messages and can perform well when a large number of text features are involved. On the other hand, it can require more computational resources and careful parameter tuning.

Random Forest offers an ensemble-based alternative that can model nonlinear relationships and interactions between different features. However, because email text can generate many

sparse features, it may not always be as efficient as models specifically designed for high-dimensional text classification[15].

Spam detection should be evaluated using more than accuracy alone.

**Precision** shows the proportion of emails predicted as spam that are actually spam. High precision is important because incorrectly sending legitimate emails to the spam folder can cause users to miss important messages.

**Recall** measures the proportion of actual spam emails that the system successfully detects. High recall is useful because missed spam messages can still reach the user's inbox.

**F1-score** combines precision and recall into a single measure and gives a balanced view of classification performance.

A simple comparison of the selected algorithms is shown below:

| Algorithm | Main Strength | Main Limitation |
|---|---|---|
| Naive Bayes | Fast and effective for text | Independence assumption |
| Logistic Regression | Efficient with TF-IDF | Mainly linear |
| SVM | Strong for high-dimensional text | Requires tuning |
| Random Forest | Captures nonlinear patterns | Can be less efficient for sparse text |

The analysis also shows that preprocessing can have a major effect on spam detection. Spam messages often contain unusual capitalization, repeated symbols, suspicious URLs, financial language, and words designed to create a sense of urgency. If these useful signals are removed during preprocessing, the model may lose important information.

Another important issue is that spam patterns change continuously. Spammers can change their wording, intentionally misspell words, insert symbols into words, or modify the structure of their messages to avoid filters. As a result, a model trained on older spam examples may gradually become less effective.

Phishing emails create an additional challenge because some are carefully written to look like genuine messages from banks, businesses, or online services. They may use professional

language and familiar-looking branding, which makes them difficult to identify using simple word-based methods alone.

False positives are also a major concern. If a legitimate email is incorrectly classified as spam, users may miss important personal, academic, or professional communication. An effective spam filter therefore needs to find a suitable balance between detecting unwanted messages and protecting legitimate emails.

In real-world systems, machine learning can be combined with other information such as sender reputation, email headers, URL characteristics, attachment information, and feedback from users. Using several sources of information can improve detection and help the system respond to new spam techniques.

Privacy is another consideration because email messages can contain sensitive personal and business information. Any system that analyzes email content should therefore use appropriate security and data-protection practices.

Overall, the comparison suggests that traditional machine learning methods can provide effective spam classification when they are combined with suitable NLP preprocessing and feature extraction. However, regular monitoring and model updates are necessary because spam techniques continue to change.

## 4. Conclusion

This research compared four machine learning algorithms—Naive Bayes, Logistic Regression, Support Vector Machine, and Random Forest—for email spam detection using Natural Language Processing techniques.

The study covered the main stages of spam classification, including dataset preparation, text preprocessing, feature extraction, model training, and evaluation. Naive Bayes provides a fast and simple baseline, while Logistic Regression works effectively with TF-IDF-based features. SVM is well suited to high-dimensional text classification, and Random Forest provides an ensemble approach that can capture nonlinear relationships.

The research also shows why precision, recall, and F1-score are important alongside accuracy. Both types of classification mistakes can cause problems. False positives can result

in legitimate emails being lost or overlooked, while false negatives allow unwanted or potentially harmful messages to reach users.

Overall, machine learning and NLP provide a useful foundation for automated email spam detection. At the same time, changing spam strategies, phishing attacks, privacy concerns, and evolving language remain important challenges.

Future work can improve spam detection by combining traditional machine learning with deep learning, transformer-based language models, sender information, URL analysis, and real-time learning techniques. A reliable spam filtering system should therefore combine accurate classification with continuous monitoring, privacy protection, and regular model updates. This can improve email security while reducing the amount of unwanted content that reaches users.

**References**

[1] T. Vafeiadis, K. I. Diamantaras, G. Sarigiannidis, and K. C. Chatzisavvas, "A comparison of machine learning techniques for customer churn prediction," *Simulation Modelling Practice and Theory*, vol. 55, pp. 1-9, 2015.

[2] A. M. Qatawneh, F. M. Aldhmour, and S. M. Alfugara, "The adoption of electronic payment system (EPS) in Jordan: case study of orange telecommunication company," *Journal of Business and Management*, vol. 6, no. 22, pp. 139-148, 2015.

[3] J. Barach, "Enhancing Ransomware Resilience in Cloud-Based HR Systems through Moving Target Defense," *Computers, Materials and Continua*, vol. 86, no. 2, pp. 1-23, 2025, doi: https://doi.org/10.32604/cmc.2025.071705.

[4] M. Amrane, S. Oukid, I. Gagaoua, and T. Ensari, "Breast cancer classification using machine learning," in *2018 electric electronics, computer science, biomedical engineerings' meeting (EBBT)*, 2018: IEEE, pp. 1-4.

[5] A. Mohammed, "Artificial Intelligence-Driven Cybersecurity Crimes: Emerging Threats, AI Bots, and Advanced Defense Strategies," *Euro Vantage journals of Artificial intelligence*, vol. 3, no. 1, pp. 46-65, 2026, doi: https://doi.org/10.65923/c2sxrx21.

[6] A. Thennakoon, C. Bhagyani, S. Premadasa, S. Mihiranga, and N. Kuruwitaarachchi, "Real-time credit card fraud detection using machine learning," in *2019 9th international conference on cloud computing, data science & engineering (Confluence)*, 2019: IEEE, pp. 488-493.

[7] S. Khairnar, G. Bansod, and V. Dahiphale, "A light weight cryptographic solution for 6LoWPAN protocol stack," in *Science and Information Conference*, 2018: Springer, pp. 977-994.

[8] A. Mohammed, "Transforming SOC Operations: Harnessing the Power of AI and ML for Enhanced Threat Detection," *INTERNATIONAL JOURNAL OF RESEARCH CULTURE SOCIETY Monthly Peer-Reviewed, Refereed, Indexed*, vol. 8, pp. 11-21, 2024, doi: 10.2017/IJRCS/ICCDMP01.

[9] D. Bodra and S. Khairnar, "Machine Learning-Based Cloud Resource Allocation Algorithms: A Comprehensive Comparative Review," *Frontiers in Computer Science,* vol. 7, p. 1678976, 2025.

[10] A. M. Qatawneh and A. M. Alqtish, "Critical examination of the impact accounting ethics and creative accounting on the financial statements," *International Business Research,* vol. 10, no. 6, p. 104, 2017, doi: 10.5539/ibr.v10n6p104.

[11] R. Sailusha, V. Gnaneswar, R. Ramesh, and G. R. Rao, "Credit card fraud detection using machine learning," in *2020 4th international conference on intelligent computing and control systems (ICICCS)*, 2020: IEEE, pp. 1264-1270.

[12] Q. Truong, M. Nguyen, H. Dang, and B. Mei, "Housing price prediction via improved machine learning techniques," *Procedia Computer Science,* vol. 174, pp. 433-442, 2020.

[13] J. Barach, "Federated Learning for Privacy-Preserving Employee Performance Analytics," *IEEE Access,* vol. 13, 2025, doi: 10.1109/ACCESS.2025.3591360.

[14] P. Lalwani, M. K. Mishra, J. S. Chadha, and P. Sethi, "Customer churn prediction system: a machine learning approach," *Computing,* vol. 104, no. 2, pp. 271-294, 2022.

[15] E. Ahmed, "Student performance prediction using machine learning algorithms," *Applied computational intelligence and soft computing,* vol. 2024, no. 1, p. 4067721, 2024.