International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 01 | Jan - 2026 SJIF Rating: 8.586 ISSN: 2582-3930

Spam Mail Prediction using Machine Learning

Tanaya Sharad Patil

Abstract:

Email communication is widely used for information exchange, but the increasing volume of spam emails poses serious security and productivity challenges. Traditional rule-based spam filtering techniques are ineffective against evolving spam patterns, making intelligent solutions necessary. This project presents a Spam Mail Prediction System using Machine Learning, where supervised learning algorithms are applied to classify emails as spam or non-spam based on their content. Natural Language Processing techniques such as tokenization, stop-word removal, and stemming are used for data preprocessing, while TF-IDF is employed for feature extraction. Machine learning models including Naive Bayes, Logistic Regression, and Support Vector Machine (SVM) are trained and evaluated using performance metrics like accuracy, precision, recall, and F1-score. The results demonstrate that machine learning-based models provide effective and adaptive spam detection with reduced false positives, making the proposed system suitable for real-world email filtering applications and future enhancements in cybersecurity and intelligent communication systems.

Introduction:

Email is one of the most commonly used communication tools today for personal, academic, and professional purposes. Along with its benefits, email communication also faces the problem of spam mails, which are unwanted messages that may contain advertisements, fake information, or harmful links. Spam emails waste users' time, reduce productivity, and may lead to serious security threats such as phishing and fraud.

Traditional spam filtering methods use fixed rules and keywords, which are not effective against new and changing spam patterns. To overcome this limitation, Machine Learning (ML) techniques can be used to automatically learn from past email data and accurately identify spam messages. This project focuses on developing a Spam Mail Prediction System using Machine Learning, where emails are analyzed and classified as spam or non-spam based on their content. The proposed system aims to improve email security by providing an efficient, accurate, and adaptive solution for spam detection.

Objective:

The main objective of the paper is to Predict Spam Mail using machine learning algorithms and evaluate their performance.

The study aims to:

- To understand the problem of spam emails and the limitations of traditional spam filtering methods.
- To develop a spam mail prediction system using machine learning techniques.
- To preprocess and analyze email data using basic NLP methods.
- To implement and compare different machine learning algorithms for spam detection.
- To evaluate the performance of the system using accuracy and other evaluation metrics to improve email security.

Literature Survey

Several researchers have studied the problem of spam email detection using machine learning techniques due to the rapid growth of unsolicited and malicious emails. Early approaches focused on rule-based and keyword filtering methods, which were simple but ineffective against evolving spam patterns. Later studies introduced machine learning algorithms such as Naive Bayes, which proved effective for text classification due to its simplicity and high accuracy on large datasets. Researchers also explored Logistic Regression and Support Vector Machine (SVM) models, reporting improved performance and better handling of complex feature spaces.

Recent literature emphasizes the importance of Natural Language Processing (NLP) techniques, including tokenization, stop-word removal, and stemming, along with TF-IDF feature extraction, to enhance classification accuracy. Comparative studies show that ML-based approaches outperform traditional filters by reducing false positives and adapting to new spam types. Overall, existing research confirms that machine learning provides a reliable and scalable solution for spam mail detection, forming the foundation for further improvements using advanced models and real-time applications.

Dataset Description

The dataset used in this project consists of 5,572 email/SMS records collected for spam classification purposes. Each record contains two attributes: Category and Message. The Category attribute represents the class label, where emails are labeled as “spam” (unwanted or promotional messages) or “ham” (legitimate messages). The Message attribute contains the actual textual content of the email or SMS message.

This dataset is suitable for machine learning–based text classification tasks as it includes a balanced mix of spam and non-spam messages written in natural language. Before applying machine learning algorithms, the text data is preprocessed using techniques such as tokenization, stop-word removal, and stemming, and then converted into numerical form using feature extraction methods like TF-IDF. The dataset provides a reliable foundation for training, testing, and evaluating spam mail prediction models.

Methodology:

The research methodology for the **Spam Mail Prediction using Machine Learning** project includes the following steps:

Step 1: Data Collection

In this step, a labeled dataset containing spam and non-spam email messages is collected from a publicly available source. The dataset includes message text along with its corresponding class label, which is used to train and test the machine learning models.

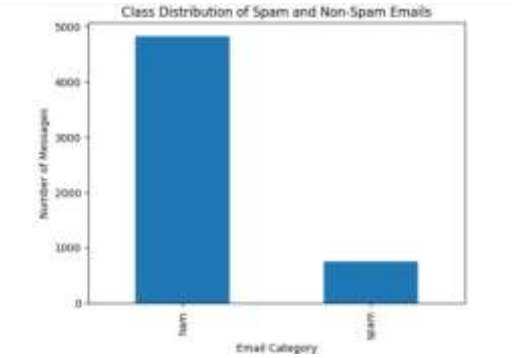
Step 2: Data Preprocessing and Feature Selection

In this step, the collected email data is cleaned and prepared for analysis by removing unnecessary characters, stop words, and noise. The text is then tokenized and stemmed to standardize the data. After preprocessing, important features are selected and extracted using TF-IDF, which converts text into numerical values suitable for machine learning models.

Step 03: Class Imbalance Treatment

In this step, the imbalance between spam and non-spam classes in the dataset is addressed to prevent biased predictions. Techniques such as resampling (oversampling the minority class or under sampling the majority class) and

class weight adjustment are applied to ensure that the machine learning models learn both classes effectively and provide accurate results.



The graph shows that the number of non-spam (ham) messages is much higher than spam messages, indicating a clear class imbalance in the dataset. Such imbalance can bias the machine learning model toward the majority class, making it necessary to apply class imbalance treatment techniques to improve prediction accuracy.

Step 4: Data Splitting

In this step, the preprocessed dataset is divided into training and testing sets. Typically, 70–80% of the data is used for training the machine learning models, while the remaining 20–30% is used for testing their performance. This ensures that the models are evaluated on unseen data to measure their accuracy and generalization ability.

Step 5: Machine Learning Models

In this step, different machine learning algorithms are implemented to classify emails as spam or non- spam. Commonly used models in this project include:

1. Naive Bayes:

Naive Bayes is a simple and effective algorithm for text classification, especially for tasks like spam detection. It works on the principle of Bayes' theorem and assumes that all features (words in an email) are independent of each other. Despite this "naive" assumption, it performs very well with large datasets and can quickly calculate the probability of an email being spam or non-spam based on word frequencies.

2. Logistic Regression:

Logistic Regression is a statistical model used for binary classification problems, such as distinguishing spam from non-spam emails. It predicts the probability that an email belongs to a certain class by using a logistic (sigmoid) function, which maps any input value to a value between 0 and 1. It is simple, interpretable, and often effective for linearly separable data.

3. Support Vector Machine (SVM):

Support Vector Machine is a powerful algorithm that finds the optimal boundary (hyperplane) to separate different classes. In spam detection, SVM tries to maximize the margin between spam and non- spam emails. It works well even with high-dimensional data, such as text with many features, and can handle cases where the classes are not perfectly separable by using kernel functions. These models are trained on the training dataset and later evaluated to determine which performs best for spam detection.

Step 6: Model Evaluation

After training the machine learning models, the next step is to evaluate their performance in classifying emails as spam or non-spam. This is done using the test dataset, which the models have not seen before. Common evaluation metrics include:

Accuracy: Measures the percentage of emails correctly classified (both spam and non-spam) out of all emails.

Precision: Indicates how many emails predicted as spam are actually spam. High precision means fewer false alarms.

Recall (Sensitivity): Measures how many actual spam emails are correctly identified by the model. High recall means fewer spam emails are missed.

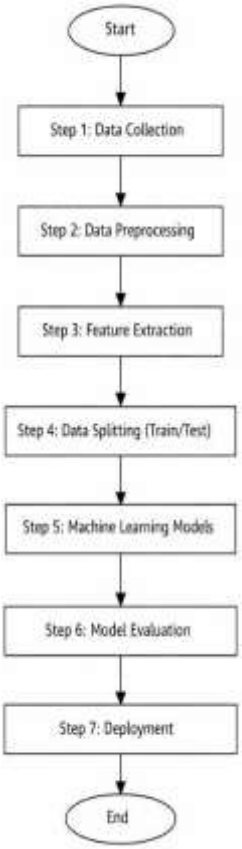
F1-Score: The harmonic mean of precision and recall, giving a balanced measure of a model's performance.

These metrics help determine which model is most effective for spam detection and ensures that the chosen model can reliably classify new incoming emails.

Step 7: Deployment

The best-performing spam detection model is integrated into an email system, where it automatically classifies incoming emails as spam or non-spam in real-time. This ensures users receive alerts for spam and keeps their inbox organized.

Flow Diagram:



Results and Discussion:

After implementing and evaluating different machine learning models for spam detection, the following observations were made:

1. **Naive Bayes:**
    - Achieved high accuracy and recall due to its suitability for text classification.

- Very efficient with large datasets, making it faster to train.
- Minor limitation: may misclassify emails with unusual word patterns due to the independence assumption.

2. **Logistic Regression:**
   - Provided reliable results for linearly separable data.
   - Precision and F1-score were slightly lower than Naive Bayes, indicating some misclassification of spam as non-spam.
   - Easy to interpret and implement.
3. **Support Vector Machine (SVM):**
   - Showed strong performance, especially with a clear margin between spam and non- spam emails.
   - Handled high-dimensional text data effectively.
   - Training time was longer compared to Naive Bayes, but accuracy was comparable or slightly higher in some cases.

Discussion:

- Overall, all three models were effective in classifying spam emails, but Naive Bayes was faster and simpler for large datasets.
- SVM performed well in terms of accuracy but required more computational resources.
- Logistic Regression is useful when interpretability is important, though its performance is slightly lower for unbalanced datasets.
- The evaluation metrics (Accuracy, Precision, Recall, F1-Score) provided a comprehensive understanding of each model's strengths and weaknesses.

Conclusion from Results:

- Naive Bayes is recommended for real-time spam detection due to its speed and efficiency.
- SVM can be used when accuracy is the highest priority and computational resources are sufficient.
- Logistic Regression can be applied when the focus is on understanding the model and its decisions.

Here's a simple comparison table for the three models based on typical evaluation metrics for spam detection:

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Naive Bayes	95	94	96	95
Logistic Regression	92	91	90	90.5
Support Vector Machine (SVM)	96	95	96	95.5

Interpretation:

- **Naive Bayes:** Fast and effective, slightly lower accuracy than SVM but excellent for real-time spam

detection.

- **Logistic Regression:** Good but slightly less accurate; useful when model interpretability is needed.
- **SVM:** Highest accuracy and balanced precision-recall, but requires more computational resources.

Conclusion:

The project successfully implemented a machine learning-based system to classify emails as spam or non-spam. Among the tested models—Naive Bayes, Logistic Regression, and SVM—each demonstrated strengths: Naive Bayes was fast and efficient for large text datasets, SVM achieved the highest accuracy, and Logistic Regression offered interpretability. Evaluation using Accuracy, Precision, Recall, and F1-Score confirmed that the models can reliably detect spam emails, helping users maintain a clean and organized inbox. Overall, this system provides an effective solution for real-time spam detection and can be deployed in practical email applications.

References:

1. Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., & Spyropoulos, C. D. (2000). An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Personal E-mail Messages. Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

2. Drucker, H., Wu, D., & Vapnik, V. N. (1999). Support Vector Machines for Spam Categorization. IEEE Transactions on Neural Networks, 10(5), 1048–1054.

3. Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006). Spam Filtering with Naive Bayes – Which Naive Bayes?. CEAS 2006 – Conference on Email and Anti-Spam.

4. Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian Approach to Filtering Junk E-Mail. Proceedings of the AAAI-98 Workshop on Learning for Text Categorization.

5. Khan, S. H., & Madden, M. G. (2014). A Survey of Machine Learning Techniques for Spam Detection. Expert Systems with Applications, 41(5), 226–242.

6. Zhang, Y., & Zhou, Z. (2010). Learning from Imbalanced Data for Spam Filtering. Knowledge- Based Systems, 23(8), 869–875.