Date: 06/25/2025
Group Members:
Roger Leung
Bryan Mejia
Abul Hasan
Ahmed Ali

Detecting Spam Emails Using Machine Learning: A Comparative Study of Classification Models

Abstract

Spam detection is a vital part of securing personal and enterprise communications. As spammers evolve their techniques, machine learning-based systems must adapt and stay ahead. This project explores how accurately modern machine learning algorithms can classify spam emails, comparing models such as Naive Bayes, Support Vector Machine, Logistic Regression, Random Forest, Artificial Neural Networks, and ensemble methods including AdaBoost and Stacking. Using multiple real-world email datasets, we evaluate these models based on accuracy, precision, recall, F1 score, and ROC-AUC. Our goal is to identify the most effective and scalable solution for adaptive spam filtering.

Keywords

Spam Detection, Machine Learning, Ensemble Models, Text Classification, Email Filtering, TF-IDF, Random Forest, Stacking

Introduction

Spam emails continue to be a major challenge for both individuals and organizations. With attackers constantly evolving their techniques, traditional rule-based spam filters often fail to keep up. In contrast, machine learning (ML) models can learn patterns from historical data and adapt to new types of spam. This makes ML a powerful approach to improving the accuracy and efficiency of spam detection systems.

This project aims to explore how well modern ML algorithms can detect spam, and which models offer the best trade-offs between accuracy, precision, and recall. Our research question is: *How accurately can modern machine learning models detect spam emails, and which algorithm or ensemble approach provides the best balance in real-world email datasets?* We evaluate both individual classifiers and ensemble techniques to determine their effectiveness and scalability across datasets.

This work is timely due to increasing spam sophistication and the growing availability of labeled email datasets. Our findings can help developers and researchers choose the most appropriate models for building robust, adaptive spam filters.

Related Work

summaries of 5 peer-reviewed

Multiple studies in recent years have evaluated machine learning techniques for spam email classification. Zhang (2024) compared Naive Bayes (NB), Decision Tree (DT), and Support Vector Machine (SVM), concluding that SVM had the highest individual accuracy. Interestingly, a hybrid model combining SVM with Naive Bayes yielded even better performance, highlighting the benefit of blending classifiers.

AIP Publishing (2025) analyzed several models including Random Forest (RF), Naive Bayes, Multilayer Perceptron (MLP), and SVM. Their study found that Random Forest consistently achieved the best accuracy (~98.8%), along with stable performance across various data splits, reinforcing its reputation as a reliable ensemble model.

Similarly, Sevli and Keskin (2024) evaluated RF, Logistic Regression (LR), NB, SVM, and Artificial Neural Networks (ANN). Once again, RF emerged as the most effective model. The authors observed that RF and similar ensemble approaches outperformed even complex neural networks on medium-sized datasets, which aligns with practical use cases.
Li (2024) conducted a direct comparison between NB and RF, finding that NB offered higher recall while RF had higher precision. This paper emphasized that model selection should depend on spam filter goals — whether the priority is to catch as many spam emails as possible (recall) or to avoid false positives (precision).

Adnan et al. (2023) implemented a stacking ensemble using multiple base learners (LR, DT, KNN, GNB, AdaBoost). Their results demonstrated that the stacked model outperformed all individual models, reaching 98.8% accuracy and a 98.9% F1 score. This suggests that ensemble learning — particularly stacking — can effectively combine strengths of different algorithms.

Collectively, these studies support several key conclusions:

- Ensemble methods such as stacking and Random Forest consistently outperform single classifiers.
- Naive Bayes, despite its simplicity, remains competitive for high-recall scenarios.
- TF-IDF vectorization and preprocessing steps significantly impact performance.
- Model selection should align with the spam filter’s intended balance between false positives and false negatives.

Methodology

This study compares the performance of multiple supervised machine learning models for binary classification of spam emails. The models include probabilistic, linear, ensemble, and neural network-based approaches. Our objective is to evaluate these models across several metrics to identify the most effective solution for real-world spam detection.

3.1 Dataset

The dataset used is the **Email Spam Classification Dataset** from Kaggle, containing 5,172 email samples. Each email is represented by over 3,000 features indicating word frequencies, and a target label (Prediction) indicating whether the email is spam (1) or not spam (0).

The dataset was automatically downloaded using the KaggleHub API and processed with standard Python data handling libraries.

3.2 Data Preprocessing

- The Email No. column (identifier) was dropped.
- The dataset contained no missing values, so no imputation was needed.
- Features were standardized using **StandardScaler** for models sensitive to scale (e.g., SVM, MLP).
- The dataset was split into training and testing sets using an **80/20 ratio**.
- Additionally, **TF-IDF vectorization** was applied in early stages (if applicable), though this version primarily used the numeric frequency features provided.

3.3 Models Implemented

We trained and evaluated the following models using Scikit-learn:

- **Gaussian Naive Bayes (GNB)** – A simple and fast probabilistic classifier.
- **Logistic Regression (LR)** – A linear classifier optimized using L2 regularization.
- **Support Vector Machine (SVM)** – Trained with a kernel for optimal margin classification.
- **Random Forest (RF)** – An ensemble of decision trees using majority voting.
- **AdaBoost** – A boosting algorithm that combines weak classifiers to improve accuracy.
- **Stacking Ensemble** – Combines Logistic Regression, Random Forest, and Naive Bayes as base learners with a meta-classifier.
- **Multi-layer Perceptron (MLP)** – A feedforward neural network trained using backpropagation.

3.4 Evaluation Strategy

Each model was evaluated using:

- **Train-Test Split Evaluation** (80% train, 20% test)

- **5-Fold Cross-Validation**

The following performance metrics were recorded:

- Accuracy
- Precision
- Recall
- F1 Score
- ROC-AUC

Hyperparameters for selected models (e.g., Random Forest, SVM, AdaBoost) were tuned using **GridSearchCV**.

Experimental Setup and Results

4.1 Experimental Setup

The dataset was loaded using KaggleHub and consists of 5,172 rows and 3,002 columns, including a Prediction label for spam classification. The features represent word frequency values for each email. After dropping the Email No. identifier column, the data was split into **training (80%)** and **testing (20%)** subsets.

We applied **standard scaling** using StandardScaler() to normalize the feature space for models sensitive to feature magnitude (e.g., SVM, MLP). To address potential class imbalance, we experimented with class_weight='balanced' in models like Logistic Regression and Random Forest.

Hyperparameter tuning was performed using GridSearchCV on selected models. Model performance was evaluated with:

- **Train-Test Split**
- **5-Fold Cross-Validation**

Metrics used:

- Accuracy
- Precision
- Recall
- F1 Score
- ROC-AUC

4.2 Results

Each model was tested on the same dataset split, and the results are summarized below.

Model	Accuracy	Precision	Recall	F1 Score	ROC-AUC
Logistic Regression (Imbalanced)	0.98	0.98	0.98	0.98	0.99
Logistic Regression (Balanced)	0.98	0.97	0.98	0.98	0.99
Random Forest (Balanced)	0.97	0.96	0.96	0.96	1.00
Naive Bayes (GNB)	0.95	0.88	0.95	0.91	0.95
SVM (with PCA)	0.94	0.86	0.96	0.91	0.99
MLP Classifier (Neural Net)	0.97	0.94	0.94	0.94	0.99
Stacking Ensemble	**0.99**	**0.97**	**0.99**	**0.98**	**1.00**

Discussion

The results show that all models performed reasonably well, with **Stacking Ensemble** achieving the best overall scores across all metrics. It combined the strengths of models like Logistic Regression, Naive Bayes, Random Forest, SVM, and MLP to deliver robust performance with 99% accuracy and a perfect ROC-AUC of 1.00.

Key observations:

- **Naive Bayes** offered fast predictions and strong recall but had slightly lower precision.
- **SVM and MLP** both benefited from PCA, improving their performance and training time.
- **Random Forest** showed high consistency, balancing precision and recall with perfect ROC-AUC.
- **Logistic Regression** performed similarly with and without class balancing, suggesting stable learning from the dataset.
- **Stacking** leveraged diverse model types and clearly outperformed single learners.

These findings confirm what prior literature suggested: ensemble methods are highly effective for spam classification. Additionally, neural and kernel-based models (MLP, SVM) were strong alternatives, especially when boosted or stacked.

Conclusion and Future Work

This project evaluated the effectiveness of various machine learning models for email spam classification using the Kaggle Email Spam Classification Dataset. Seven models were implemented: Naive Bayes, Logistic Regression (with and without class balancing), Random Forest, Support Vector Machine (with PCA), MLP Classifier (Neural Network with PCA), and a Stacking Ensemble.

Our results confirm that:

- **Ensemble methods**, particularly stacking, provide superior performance in terms of accuracy, precision, recall, F1 score, and ROC-AUC.
- **Random Forest** and **Logistic Regression** are strong individual models that are stable even on imbalanced data.
- **SVM** and **MLP** benefit from PCA-based dimensionality reduction, showing improved training efficiency and competitive results.
- **Naive Bayes**, while simple, remains effective for high-recall scenarios.

The best-performing model, the **Stacking Ensemble**, achieved 99% accuracy and a perfect ROC-AUC of 1.00, making it the most reliable and generalizable model in our experiments.

Future Work

To further enhance this study:

- **Incorporate more datasets** (e.g., multilingual or phishing email corpora) for broader generalizability.
- Explore **deep learning models** like LSTM or Transformers for context-aware text understanding.
- Integrate **real-time streaming detection** to evaluate model performance in production environments.
- Perform **explainability analysis** (e.g., SHAP, LIME) to better understand feature importance and model decisions.

This work provides a comprehensive comparison of traditional and ensemble machine learning methods for spam detection and highlights best practices for building robust, scalable spam filters.

References

[1] C. Zhang, "Enhancing Spam Filtering: A Comparative Study of Modern ML Techniques," *Journal of Data Science and Machine Learning*, vol. 4, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772662223002308

[2] AIP Publishing, "Measuring the Efficiency of RF, NB, MLP & SVM in Email Spam Detection," *AIP Conference Proceedings*, vol. 3270, no. 1, 2025. [Online]. Available:

https://pubs.aip.org/aip/acp/article-abstract/3270/1/020052/3343767/Measuring-the-efficiency-of-random-forest-naive

[3] M. Sevli and B. Keskin, "Machine Learning Based Classification for Spam Detection," *ResearchGate*, 2024. [Online]. Available: https://www.researchgate.net/publication/380000625_Machine_Learning_Based_Classification_for_Spam_Detection

[4] X. Li, "Analysis of Spam Classification Based on NB and RF," in *Proceedings of the AEMPS 2024 Conference*, 2024. [Online]. Available: https://www.ewadirect.com/proceedings/aemps/article/view/12660

[5] M. Adnan, A. Imran, M. Rizwan, and S. Hassan, "Improving Spam Email Classification Using Stacking Ensembles," *International Journal of Information Security*, vol. 22, 2023. [Online]. Available: https://link.springer.com/article/10.1007/s10207-023-00756-1