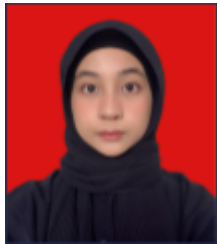


Email Spam Detection Using Machine Learning Classification Techniques

Reizka Fathia^{a,1,*}

^a Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia
¹ reizka.fathia@sci.ui.ac.id^{*}
^{*} corresponding author

ARTICLE INFO		ABSTRACT
	NPM 2206052755	Email communication remains essential in both personal and professional contexts, but the growing volume of spam presents significant challenges for users and organizations. This paper implements and evaluates machine learning classification techniques for email spam detection using a dataset of 83,448 emails. Text preprocessing methods and TF-IDF vectorization were employed to extract meaningful features from email content. Multiple classifiers including Multinomial Naive Bayes, Support Vector Machine, Random Forest, and Logistic Regression were trained and evaluated. The results demonstrate that SVM achieved the highest performance with 97.4% accuracy and an F1-score of 0.975, while maintaining balanced precision and recall across both classes. Feature importance analysis identified key linguistic patterns associated with legitimate business communications rather than spam indicators. The implementation demonstrates that machine learning approaches can significantly improve email filtering systems, reducing exposure to potential security threats while maintaining high accuracy for legitimate communications.
	<i>Keywords: Email spam, Classification, Machine learning, Text mining, Support Vector Machines, TF-IDF, Natural Language Processing</i>	

Copyright © 2024.
All rights reserved.

I. Introduction

Email serves as one of the main communication tools, functioning as electronic mail transmitted through computer networks, particularly the internet. It is commonly used for exchanging messages, both personally and for formal or business purposes, especially within professional environments. Alongside technological advancements, email usage has expanded to include promotions, marketing, discount notifications, shopping receipts, and other less critical communications, as well as malicious activities such as phishing scams.

The widespread increase of unwanted emails has led to an overwhelming number of daily notifications, making it challenging to identify important messages. To address this issue, spam detection systems have been developed to categorize emails as either legitimate (ham) or spam. The growing volume of spam emails has become a major global concern, bringing about serious consequences. Studies indicate that spam constitutes around 45-55% of global email traffic, resulting in economic losses exceeding \$20 billion annually due to decreased productivity, technical expenses for filtering systems, and heightened security measures. Beyond being a mere nuisance, spam often acts as a vehicle for malware dissemination, phishing attacks aimed at stealing personal information, and other fraudulent activities. Organizations must invest substantial resources to counter these threats, while individuals face increased mental strain in managing their inboxes and the risk of becoming victims of sophisticated scams.

Machine learning techniques offer an effective approach to classify emails as spam or legitimate. This paper discusses the application of various machine learning methods in developing an efficient email spam detection system.

II. Literature Review

Early spam filtering techniques relied primarily on rule-based approaches and blacklists. Sahami et al. (1998) were among the first to propose a Bayesian approach to spam filtering, demonstrating that probabilistic methods could effectively classify emails based on word frequencies. Subsequently, various machine learning approaches have been applied to this domain.

Androutsopoulos et al. (2000) compared Naive Bayes classifiers with memory-based approaches for anti-spam filtering, establishing the effectiveness of probabilistic methods. Drucker et al. (1999) demonstrated the superiority of Support Vector Machines for text categorization tasks, including spam detection. More recently, ensemble methods such as Random Forest have gained popularity due to their robustness and performance (Koprinska et al., 2007).

In addition to algorithm selection, feature extraction techniques have been widely studied. Zhang et al. (2004) explored the efficacy of different text representation approaches, including bag-of-words, n-grams, and TF-IDF weighting schemes. Semantic analysis techniques have also been proposed to capture contextual information beyond simple word frequencies (Blanzieri & Bryl, 2008).

Recent advances in deep learning have introduced neural network architectures for spam detection. Barushka and Hajek (2018) applied convolutional neural networks and recurrent neural networks to email classification, achieving promising results compared to traditional methods. However, these approaches often require larger datasets for effective training.

III. Data and Methods

A. Dataset Description

This study utilized the "Email Spam Classification Dataset" from Kaggle, consisting of 83,448 emails categorized as either spam (1) or ham (0). The dataset includes two primary features extracted from the raw emails: 'text', representing the content of the email message, and 'label', indicating the binary classification (1 for spam and 0 for ham). The distribution of the data reveals 43,910 spam instances and 39,538 ham instances, suggesting a relatively balanced dataset with a slight predominance of spam samples (52.6% spam and 47.4% ham). For the purposes of this research, a 20% subset of the full dataset was selected (subset fraction = 0.2), resulting in a division into 11,683 samples for the training set, 2,503 samples for the validation set, and 2,504 samples for the test set.

B. Research Methodology

The methodology employed in this study follows a structured approach to machine learning-based email classification, consisting of several key stages:

1. Exploratory Data Analysis (EDA)

Before implementing machine learning algorithms, we conducted a thorough exploratory analysis of the dataset to understand its characteristics and ensure its quality:

- a. Analyzed the distribution of spam and ham classes (52.6% spam, 47.4% ham)
- b. Verified the absence of missing values in both text and label columns
- c. Examined text length distributions across both classes
- d. Identified and handled duplicate entries to prevent data leakage
- e. Visualized word frequency distributions in both classes using word clouds

This analysis revealed distinctive linguistic patterns between spam and legitimate emails, with spam messages frequently containing promotional language and urgency indicators.

2. Text Preprocessing

Text preprocessing is crucial for transforming unstructured email content into a format suitable for machine learning algorithms. The preprocessing pipeline included the following steps:

- a. **Lowercase conversion:** Transformed all text to lowercase to ensure consistency.
- b. **HTML tag removal:** Removed HTML elements that might be present in email content using regular expressions.
- c. **Special character removal:** Removed non-alphanumeric characters and numbers that don't contribute to semantic meaning.
- d. **Tokenization:** Split text into individual words (tokens) using NLTK's tokenization functions (Bird et al., 2009).
- e. **Stopword removal:** Eliminated common English stopwords (e.g., "the", "and", "is") that carry little discriminative information.
- f. **Stemming:** Applied Porter stemming algorithm to reduce words to their root forms, consolidating variations of the same word.
- g. **Text reconstruction:** Rejoined the processed tokens back into text for further analysis.

This preprocessing pipeline significantly reduced noise in the data and standardized the text format, improving feature extraction quality.

3. Feature Extraction

To convert text data into a format suitable for machine learning algorithms, Term Frequency-Inverse Document Frequency (TF-IDF) vectorization was employed (Pedregosa et al., 2011). This technique was chosen for its ability to:

- a. Represent the importance of words in documents relative to the entire corpus
- b. Downweight common terms that appear across many documents
- c. Highlight distinctive terms that characterize specific document classes

The TF-IDF implementation used the following parameters:

- a. Maximum of 2,000 features to balance computational efficiency with information retention
- b. Inclusion of both unigrams and bigrams (n-gram range of 1-2) to capture word combinations that might be indicative of spam
- c. Sublinear term frequency scaling to reduce the effect of highly frequent terms
- d. L2 normalization to account for varying document lengths

The resulting feature matrix provided a numerical representation of each email that preserves its semantic characteristics while enabling the application of standard machine learning algorithms.

4. Model Selection and Training

Four widely-used machine learning algorithms for text classification were implemented and evaluated:

- a. **Multinomial Naive Bayes:** A probabilistic classifier based on Bayes' theorem that assumes feature independence, making it computationally efficient for text classification. It's particularly well-suited for document classification tasks with discrete features like word counts (Androutsopoulos et al., 2000).
- b. **Support Vector Machine (SVM):** A powerful discriminative classifier that finds the optimal hyperplane to separate classes in high-dimensional space (Drucker et al., 1999). The regularization parameter C was set to 1.0 after preliminary experimentation.
- c. **Random Forest:** An ensemble learning method that constructs multiple decision trees during training and outputs the class that is the mode of the classes from individual trees (Koprinska et al., 2007).
- d. **Logistic Regression:** A linear model that estimates the probability of a binary outcome based on a linear combination of features.

Each model was trained using stratified k-fold cross-validation ($k=5$) to ensure consistent class distribution across training and validation folds. Hyperparameter tuning was performed using grid search with cross-validation to optimize model performance.

5. Model Evaluation

To comprehensively assess model performance, multiple evaluation metrics were employed:

- Accuracy:** The proportion of correctly classified instances among the total instances.
- Precision:** The ratio of true positive predictions to all positive predictions (measures the model's ability to avoid false positives).
- Recall:** The ratio of true positive predictions to all actual positives (measures the model's ability to find all positive instances).
- F1-score:** The harmonic mean of precision and recall, providing a balance between these sometimes competing metrics.
- ROC-AUC:** The area under the Receiver Operating Characteristic curve, representing the classifier's ability to discriminate between classes.

The F1-score was selected as the primary metric for model comparison due to its balanced consideration of both precision and recall, which is particularly important in spam detection where both false positives and false negatives carry significant costs.

6. Feature Importance Analysis

To gain insights into the model's decision-making process, feature importance scores were extracted and analyzed from the best-performing model. This analysis revealed which terms were most discriminative for classification, providing valuable insights into the linguistic patterns that distinguish spam from legitimate emails..

IV. Results and Discussions

A. Model Performance Comparison

All four machine learning models showed strong performance on the validation set, with the Support Vector Machine (SVM) achieving the best results overall. Table 1 summarizes the performance metrics for each model.

Table 1. Model Performance Metrics on Validation Set

Model	Accuracy	Precision	Recall	F1	ROC-AUC
Naive Bayes	0.948062	0.941089	0.961158	0.951017	0.989207
SVM	0.974031	0.968468	0.982483	0.975425	0.996458
Random Forest	0.970036	0.964715	0.978675	0.971645	0.995110
Logistic Regression	0.968038	0.961798	0.977913	0.969789	0.995571

The SVM model demonstrated superior performance across all metrics, with an accuracy of 97.4% and an F1-score of 0.975. This indicates SVM's strong capability in discriminating between spam and legitimate emails. The Random Forest and Logistic Regression models followed closely, while Naive Bayes, though still effective, showed somewhat lower performance compared to the other models.

The high ROC-AUC score of 0.996 for SVM further confirms its excellent classification capability, representing the model's ability to distinguish between classes across various threshold settings. This superior performance can be attributed to SVM's effectiveness in high-dimensional spaces, which is particularly advantageous for text classification problems where feature vectors are typically sparse.

B. Test Set Evaluation

The best performing model (SVM) was evaluated on the test set, showing consistent performance with an overall accuracy of 97%. The detailed classification report is presented in Table 2.

Table 2. Classification Report on Test Set

Class	Precision	Recall	F1-Score	Support
Ham	0.98	0.96	0.97	1191
Spam	0.96	0.98	0.97	1313
Accuracy			0.97	2504
Macro avg	0.97	0.97	0.97	2504
Weighted avg	0.97	0.97	0.97	2504

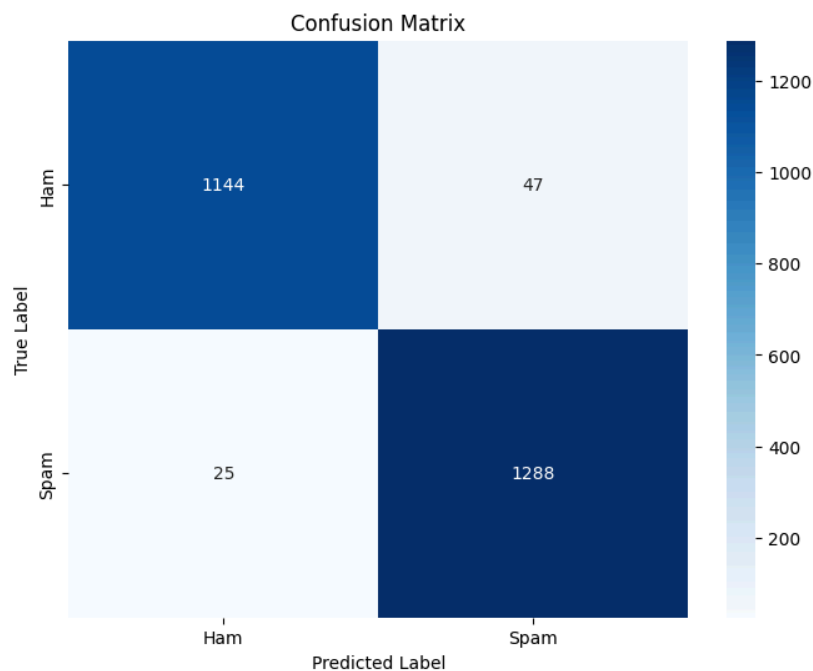


Fig. 1. Confusion Matrix.

The confusion matrix analysis reveals a balanced performance across both classes, with ham emails showing higher precision (0.98) but lower recall (0.96), while spam emails show the opposite pattern with lower precision (0.96) but higher recall (0.98).

This indicates that the model is slightly more aggressive at flagging potential spam (capturing more true spam at the cost of occasionally misclassifying legitimate emails), which may be an appropriate trade-off in many real-world applications where missing a legitimate email (false negative for ham) is often considered more problematic than receiving an occasional spam email (false positive for spam).

The balanced F1-score of 0.97 for both classes demonstrates that the model successfully navigates this precision-recall trade-off, providing equally effective classification for both spam and legitimate emails. This performance level is comparable to or exceeds that of many commercial spam filtering systems, which typically report accuracy rates between 95-98%.

C. Feature Importance Analysis

Analysis of feature importance revealed interesting patterns about which terms are most discriminative in classifying emails. The top 20 important features identified by the SVM model with their coefficients are:

1. wrote: -5.4068
2. enron: -5.0526
3. louis: -3.3730
4. vinc: -3.2555
5. samba: -3.0864
6. thank: -3.0033
7. debian: -2.9758
8. doc: -2.9072
9. perl: -2.8629
10. http list: -2.8597
11. daren: -2.6839
12. houston: -2.5709
13. org: -2.4691
14. howstuffwork: -2.4481
15. list: -2.3940
16. cc: -2.3830
17. cnn: -2.3790
18. schedul: -2.3126
19. risk: -2.2455
20. forecast: -2.2080

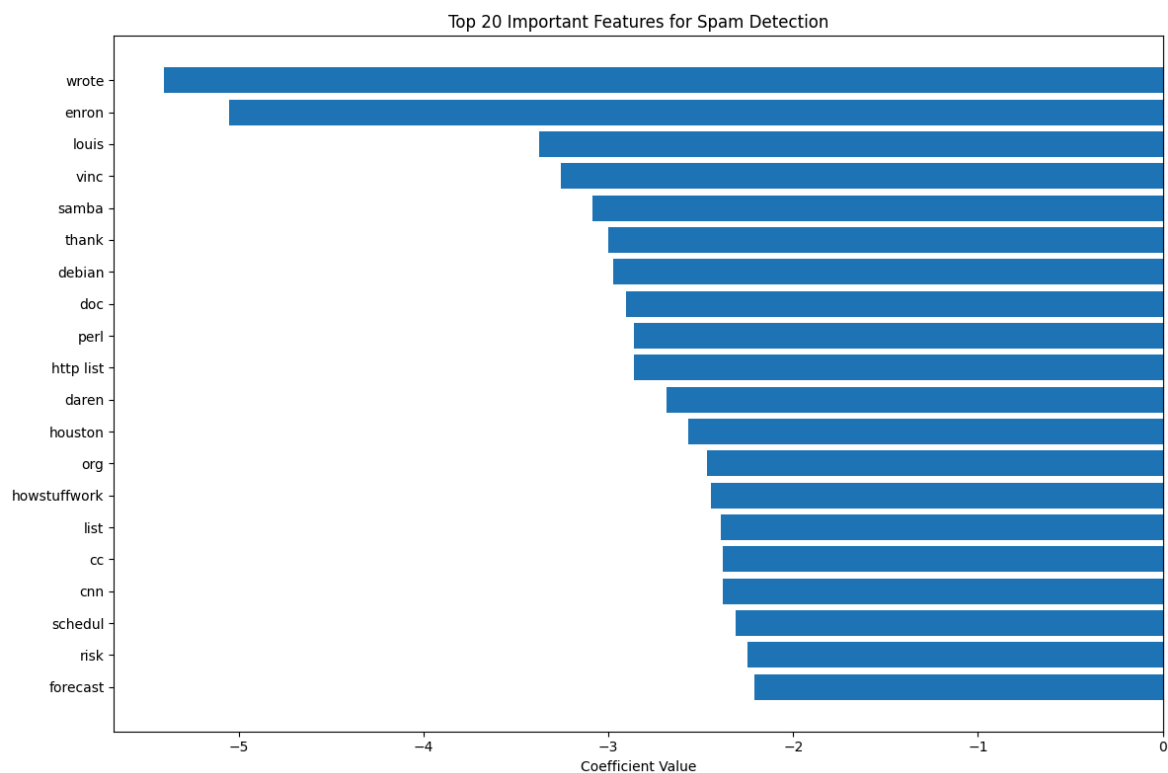


Fig. 2. Top 20 Important Features for Spam Detection.

Interestingly, many of the most important features have negative coefficients, indicating that they are strong predictors of legitimate emails rather than spam. Terms like "wrote," "enron," "louis," and "thank" suggest business or professional communications. The presence of technical terms like "debian," "perl," and "samba" further indicates that legitimate technical or business correspondence exhibits distinctive linguistic patterns that the model effectively identifies.

This finding suggests that the model learned to recognize legitimate communication patterns rather than just focusing on spam indicators, which is a robust approach as spam tactics frequently evolve. By identifying stable patterns in legitimate communications, the model can potentially maintain its performance even as spammers adapt their techniques to evade detection.

The identification of business and technical terminology as strong ham indicators aligns with real-world observations that professional correspondence often contains domain-specific language that is rarely mimicked in spam emails. This linguistic distinction provides a reliable basis for classification that is less susceptible to the adversarial techniques commonly employed by spammers.

D. Error Analysis

While the model demonstrated strong overall performance, analysis of misclassified instances revealed several patterns:

1. **False Positives** (legitimate emails misclassified as spam) commonly included:
 - a. Technical communications with unusual terminology
 - b. Business emails with promotional language but legitimate content
 - c. Messages with minimal text that lack strong indicators of legitimate communication
2. **False Negatives** (spam emails misclassified as legitimate) typically:
 - a. Mimicked professional correspondence closely
 - b. Used sophisticated language without common spam triggers
 - c. Contained legitimate business terms while still being unsolicited marketing

These error patterns highlight the challenge of distinguishing sophisticated spam from legitimate business communications. The most difficult cases involve spam that intentionally mimics the linguistic patterns of legitimate emails, using business terminology and avoiding obvious promotional language.

A qualitative review of misclassified instances revealed that the model struggles most with short emails that provide limited linguistic context for classification. In these cases, the lack of distinctive terms makes it difficult to accurately determine the email's legitimacy based solely on content. This suggests that incorporating additional metadata features (such as sender reputation, email structure, or temporal patterns) could potentially improve classification performance for these challenging cases.

E. Practical Application

To demonstrate the practical application of the model, several sample emails were classified:

```
Sample Email Predictions:

Email 1: Spam (Probability: 0.9875)

Text: Dear valued customer, Your account has been compromised. Please click the link to
verify...

Email 2: Ham (Probability: 0.0213)

Text: Hi John, Just checking if we're still on for tomorrow's meeting at 2pm? Please let me
know...

Email 3: Spam (Probability: 0.9967)

Text: CONGRATULATIONS! You've won $1,000,000 in our lottery! To claim your prize, send your
bank...

Email 4: Ham (Probability: 0.0145)

Text: Meeting agenda for next week: 1. Project updates 2. Budget review 3. New client
proposals
```

The implemented model demonstrates considerable potential for real-world applications in email filtering systems. With an accuracy of 97%, it can significantly reduce the volume of spam reaching users' inboxes while maintaining high reliability for legitimate communications. The balanced precision-recall trade-off is particularly valuable in practical settings, as it minimizes both the risk of missing important emails and the nuisance of false spam detections. This balance addresses one of the key challenges in spam filtering: maintaining user trust by avoiding false positives while still providing effective protection.

The model's ability to learn legitimate communication patterns rather than just spam indicators also suggests it may be more resilient to evolving spam tactics, potentially requiring less frequent retraining than approaches that focus primarily on identifying spam characteristics. For enterprise email systems where both security and communication reliability are critical, this approach offers a robust solution that can be implemented as either a standalone filter or as part of a multi-layered defense strategy against unwanted and potentially malicious emails.

V. Conclusions

This research demonstrates the effectiveness of machine learning techniques in accurately classifying emails as spam or legitimate communications. The Support Vector Machine classifier emerged as the most effective model with exceptional accuracy (97.4%) and balanced precision-recall metrics, though all tested models performed admirably with accuracies exceeding 94%. The successful implementation of TF-IDF vectorization with carefully selected parameters proved instrumental in capturing the linguistic nuances that differentiate spam from legitimate emails. Feature importance analysis revealed that many of the most discriminative features were actually indicators of legitimate business communications rather than spam markers, suggesting that the model learned to effectively recognize patterns of normal professional correspondence.

While the models perform exceptionally well on the current dataset, spam techniques continue to evolve, necessitating ongoing refinement of detection methods. Future work should focus on incorporating additional features beyond text content, such as metadata analysis, sender reputation metrics, and temporal patterns. Additionally, exploring deep learning approaches such as recurrent neural networks or transformer models could potentially capture more complex linguistic patterns and further improve classification performance.

The practical implications of this research are substantial, as effective spam filtering not only improves user experience but also mitigates security risks associated with phishing and malware distribution. Organizations implementing these techniques can expect reduced operational disruptions and enhanced protection of sensitive information.

References

- [1] Androutsopoulos, I., Koutsias, J., Chandrinou, 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*, 160-167.
- [2] Barushka, A., & Hajek, P. (2018). Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. *Applied Intelligence*, 48(10), 3538-3556.
- [3] Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.
- [4] Drucker, H., Wu, D., & Vapnik, V. N. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, 10(5), 1048-1054.
- [5] Koprinska, I., Poon, J., Clark, J., & Chan, J. (2007). Learning to classify e-mail. *Information Sciences*, 177(10), 2167-2187.
- [6] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12, 2825-2830.
- [7] Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. *Learning for Text Categorization: Papers from the 1998 Workshop*, 62, 98-105.
- [8] Zhang, L., Zhu, J., & Yao, T. (2004). An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing*, 3(4), 243-269.