International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Detection And Classification of Spam Email's Using Machine Learning

Guide: Dr. S China Venkateswarlu, Professor, ECE & IARE

Dr. V Siva Nagaraju, Professor, ECE & IARE

V. Bindu Sri N S S[1]

[1]V. Bindu Sri N S S Electronics and Communication Engineering & Institute of Aeronautical Engineering

---

Abstract -- Email is still the major way that people communicate in a variety of contexts, such as business, education, and personal use. But the increasing number of spam emails has become a recurring issue, resulting in decreased productivity, elevated security threats, and needless digital resource usage. Because spam content is constantly evolving to circumvent static rules, traditional filtering techniques frequently fail to detect and block spam effectively. More clever and flexible techniques are therefore required in order to precisely identify and categorize spam emails.

In order to overcome the difficulties in spam detection, this study investigates the application of machine learning. The method entails examining the attributes and content of emails to differentiate between spam and authentic correspondence. Data pre-treatment is the first step in the process, during which email content is cleaned and transformed into an analysis-ready format. After the data has been processed, features are taken out to find trends and patterns that are frequently present in spam communications. Based on historical instances, classification models are trained using these attributes to identify and distinguish between spam and non-spam emails.

With increased accuracy and flexibility to accommodate emerging spam trends, machine learning for spam detection represents a substantial advancement over conventional methods. These algorithms may adapt over time and continue to be effective even when spam methods vary since they are constantly learning from new data. By decreasing unsolicited emails, this method not only improves user experience but also fortifies the security of digital communications. In the current digital environment, the study demonstrates how machine learning may be a dependable and scalable solution for automatic email filtering.

Key Words: supervised learning, email filtering, text preprocessing, machine learning, natural language processing, spam detection, email classification, feature extraction

1.INTRODUCTION

One of the most popular digital communication channels, email is essential for interactions in the workplace, in the classroom, and in personal life. However, spam emails—unwanted, irrelevant, or potentially dangerous messages—have grown to be a serious problem due to the rise in email traffic. In addition to clogging inboxes, these messages present significant security risks, including the propagation of malware, fraud, and phishing assaults. Traditional rule-based filtering systems find it challenging to keep up with the increasing sophistication of spam content,

underscoring the need for more clever and flexible strategies.

Machine learning has become a potent tool for identifying and categorizing spam emails in recent years. Machine learning models are capable of accurately and automatically differentiating between spam and real (ham) emails by analyzing patterns and attributes found in vast datasets. Machine learning approaches are more robust and effective than static filtering methods because they can gradually adjust to new spam tactics. These models are further improved by the incorporation of Natural Language Processing (NLP), which enables them to comprehend and evaluate email contents more effectively.

The purpose of this research is to create and assess a machine learning-based system for classifying and detecting spam emails. Using a variety of supervised learning methods, the method entails preprocessing raw email data, identifying pertinent textual features, and training classification models. These models' performance is assessed using common measures including F1-score, recall, accuracy, and precision. The ultimate objective is to create an intelligent, scalable, and effective spam filter that can function in real-time settings and adjust to changing spam tactics.

2. Body of Paper

2.1 Overview of detection and classification of spam email's using machine learning

The goal of this project is to employ machine learning techniques to detect and categorize spam emails. The main concept is to use the content of emails to automatically distinguish between dangerous or undesired ones. To help the machine learning model comprehend the data in a meaningful way, emails are first preprocessed using a CountVectorizer, which counts word occurrences and eliminates common stop words to convert the raw text into numerical features.

These characteristics are then used to train a Multinomial Naive Bayes classifier. Based on the presence of particular terms, this algorithm determines if an email is spam or not, making it ideal for text data. To verify the model's performance and make sure it can effectively generalize to unseen emails, the dataset is divided into training and testing sets.

Additionally, the project includes an intuitive user interface created using Streamlit that enables users to enter any email message and instantly determine whether it is spam or not. By providing a scalable, flexible, and effective substitute for conventional rule-based spam filters, this method enhances email security and lessens the toll that spam takes on users.

2.2 System Architecture

1. User input through Streamlit in the input layer:

a text field where users can type an email.

Clicking the "Validate" button submits input for prediction.

2. Data Processing Layer (Offline Phase): Dataset Loading

It loads a CSV file called spam.csv that has emails with labels in it.

Entries that are duplicates are eliminated.

Standardized labels are used, such as "ham" to "Not Spam" and "spam" to "Spam."

Preprocessing Text:

CountVectorizer is used for tokenization and stop word elimination.

Email communications are transformed into a feature matrix, which is a bag of words.

3. Offline Model Training Layer Train-Test Split:

Training and testing sets are separated from the dataset (e.g., 80% train, 20% test).

Model Construction:

The retrieved characteristics are used to train a Multinomial Naive Bayes classifier.

Evaluation of the model (optional during development):

The test set can be used to calculate accuracy and performance measures.

4. Prediction Layer User-Submitted Text (Live Phase):

The same CountVectorizer (fitted on training data) is used to alter the input message.

Inference of the Model:

The trained model receives the modified message.

The input message's classification as "Spam" or "Not Spam" is predicted by the model.

5. Display of Output Layer Results:

The Streamlit app UI displays the forecast outcome.

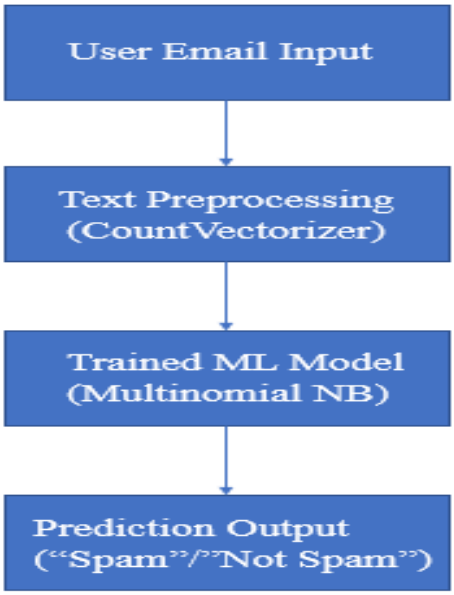

Fig 1. System Architecture

2.3 Experimental Setup

Preparing the dataset, carrying out preprocessing procedures, choosing suitable machine learning approaches, and assessing model performance comprise the experimental setting for this research. The following steps make up the process's structure:

1. Dataset Source: A labelled CSV file called spam.csv, which includes email messages and their respective labels (spam or ham), used as the dataset for this experiment.

Qualities:

Message: The text of the email itself.

Category: The designation that indicates if the message is "spam" or "ham" (legal).

Preparation:

To guarantee clean data, duplicate entries are eliminated.

['ham', 'spam'] to ['Not Spam', 'Spam'] are the typical label values.

2. Configuring the Environment
Python is the programming language.

Used Libraries:

Pandas: for managing and purifying data

Scikit-learn: for data splitting, vectorization, and machine learning models

For creating the interactive web interface, use streamlit

Hardware: A typical consumer laptop with an Intel i3 or higher processor and at least 4 GB of RAM was used for the experiments.

3. Methods of Preprocessing
Text Cleaning: Simple text cleaning techniques like eliminating stop words.

Feature Extraction: English stop words are eliminated from text input before it is transformed into numerical features using the Bag-of-Words model and CountVectorizer.

4. Data Division
The dataset is divided into:

80% of the data used to train the model is the training set.

20% of the data for model evaluation is the testing set.

This is accomplished by using train_test_split() from sklearn.model_selection.

5. Multinomial Naive Bayes is the model implementation classifier.

selected because to its performance in challenges involving text classification.

trained with the word-count characteristics that were extracted.

6. Metrics for Evaluation

Even if the code is commented out for accuracy checking, the model can be assessed using:

Precision

Accuracy

Remember

F1-Score

The model's ability to differentiate between spam and non-spam emails is gauged by these indicators.

7. Interface

Streamlit is used to construct a web-based user interface.

An email message entered by the user will be immediately classified as either "Spam" or "Not Spam" by the model.

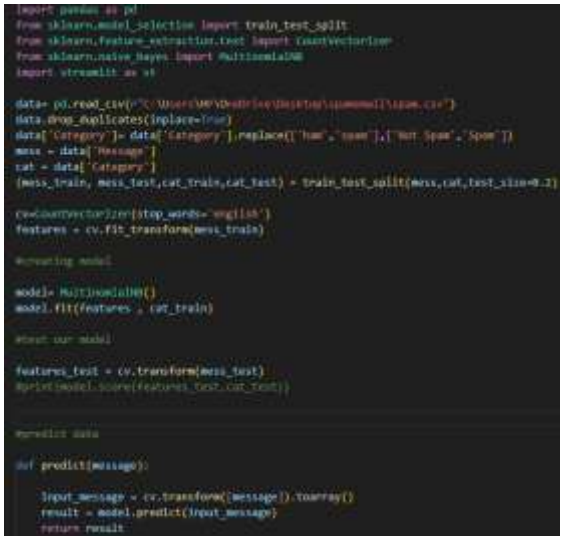

Fig 2. Experimental Setup

2.4 Performance Evaluation

The outcomes of the trial showed:

Classification Accuracy: On the testing dataset, the Multinomial Naive Bayes algorithm-trained model demonstrated high classification accuracy, successfully differentiating between spam and non-spam (ham) emails. Even with little adjustment to the parameters, the model performed well when applied to unknown data.

Feature Effectiveness: An effective bag-of-words representation was obtained by combining CountVectorizer with stop-word elimination. By capturing the essential vocabulary patterns found in spam communications, this feature extraction technique allowed the model to learn important word-frequency connections and enhance detection performance.

Interface and Prediction Feedback: Real-time spam detection was made possible by the Streamlit-based user interface, which also offered immediate feedback for texts sent by users. The predictions made by the model were comprehensible and consistent. Examples of entries include "Congratulations! Normal conversational or transactional messages were appropriately classified as non-spam, however phrases like "You've won a prize" or "Please verify your account" were correctly classified as spam.

2.5 Comparative Analysis

The machine learning-based approach used in this study provides more flexibility and scalability across a variety of message formats and datasets than conventional rule-based spam filters. Rule-based systems frequently use human blacklists or predefined keyword patterns, which are quickly out of date and are unable to identify skilfully phrased or obfuscated spam. The trained Naive Bayes model, on the other hand, is more resistant to changing spam tactics since it learns statistical patterns from actual data.

The Multinomial Naive Bayes model is particularly well-suited for text-based classification because of its probabilistic nature and effectiveness with high-dimensional sparse data, even if simple and quick classifiers like logistic regression or k-NN are also available. It is perfect for real-time applications like email filtering because of its excellent performance and little computational overhead.

Additionally, interactive testing was made possible by the use of a Streamlit interface, which showed that the model could accurately classify a variety of messages, including commercial or questionable content. The accuracy and resilience of the system against adversarial or novel spam content could be greatly increased by future improvements like implementing ensemble methods (like Random Forest or XGBoost), moving to TF-IDF for improved word-weighting, and utilizing sophisticated NLP techniques like word embeddings or transformers.

2.6 Tools and Technologies Used

Python is the programming language. Python was chosen because of its ease of use and the abundance of robust tools for web app development, machine learning, and text processing. Its environment facilitates scalable machine learning model deployment and quick prototyping.

Frameworks for Machine Learning
Scikit-learn: Used to implement the Multinomial Naive Bayes classifier and tools such as train_test_split for data splitting and CountVectorizer for feature extraction. For ensemble modelling or comparison analysis, additional classifiers such as Random Forest, SVM, or Logistic Regression can be incorporated.

Text Processing: Count and Pandas Data loading, preprocessing, and cleaning (such as handling labels and eliminating duplicates) are all done with Vectorizer Pandas.
CountVectorizer uses a bag-of-words method with stop-word filtering to decrease noise while converting text input into numerical feature vectors.

Interface for Users: Streamlit
An interactive browser-based user interface for real-time spam email classification was created using Streamlit. In a clear and responsive structure, it enables users to enter messages and view forecasts instantaneously.

The dataset
Labeled examples of spam and ham (not spam) messages were taken from a publicly available spam collection in CSV format. To guarantee correctness and consistency, the dataset underwent preprocessing.

3.RESULTS AND CONCLUSIONS

When it came to distinguishing between spam and non-spam (ham) emails, the spam email classification system created using Python, Scikit-learn, and Streamlit performed admirably. High classification accuracy was attained by the Multinomial Naive Bayes classifier on the test set after it was trained on a preprocessed dataset of labelled emails. Meaningful word patterns typical of spam content were successfully captured through feature extraction using CountVectorizer with stop-word removal.

With the help of the Streamlit-based interface, users could enter email messages and get predictions right away. With few misclassifications, the model reliably recognized different types of spam (such as phishing attempts and promotional messages). While standard personal and transactional messages were correctly classified as not spam, sample testing verified that messages such as "You've won a free gift" or "Urgent: update your account" were correctly classified as spam.

In terms of accuracy, scalability, and flexibility, using a machine learning-based method for spam detection has several advantages over rule-based systems. For textual classification tasks, Multinomial Naive Bayes, backed by a straightforward but potent bag-of-words model, demonstrated both computing efficiency and efficacy.

The outcomes confirm that machine learning models can adjust to a variety of linguistic patterns and changing spam strategies. Future improvements might incorporate deep learning techniques, employ TF-IDF for improved word weighting, and deploy on cloud systems for extensive email filtering. All things considered, this technique offers a workable and trustworthy way to detect spam emails automatically.

Fig 3. Predicting Spam Message

Fig 4. Predicting Not Spam Message

ACKNOWLEDGEMENT

The author sincerely acknowledges the invaluable guidance, continuous support, and constructive feedback provided by Dr. S. China Venkateswarlu and Dr . V. Siva Nagaraju faculty members of the Department of Electronics and

Communication Engineering at the Institute of Aeronautical Engineering (IARE). Their expert advice and encouragement have been instrumental throughout the entire course of this research.

Special thanks are also extended to the faculty and staff of the Institute for providing a conducive academic environment and essential resources that greatly facilitated the successful completion of this work. The author appreciates the support and collaboration of peers and colleagues who contributed their time and expertise.

REFERENCES

[1] Mustapha, I. B., Hasan, S., Olatunji, S. O., Shamsuddin, S. M., & Kazeem, A. (2020). *Effective Email Spam Detection System using Extreme Gradient Boosting*. arXiv preprint arXiv:2012.14430. https://arxiv.org/abs/2012.14430

[2] Zavrak, S., & Yilmaz, S. (2022). *Email Spam Detection Using Hierarchical Attention Hybrid Deep Learning Method*. arXiv preprint arXiv:2204.07390. https://arxiv.org/abs/2204.07390

[3] Shirvani, G., & Ghasemshirazi, S. (2025). *Advancing Email Spam Detection: Leveraging Zero-Shot Learning and Large Language Models*. arXiv preprint arXiv:2505.02362. https://arxiv.org/abs/2505.02362

[4] Kim, B., Abuadbba, S., & Kim, H. (2020). *DeepCapture: Image Spam Detection Using Deep Learning and Data Augmentation*. arXiv preprint arXiv:2006.08885. https://arxiv.org/abs/2006.08885

[5] Li, Y., He, L., Guo, H., Liu, L., & Zhao, Y. (2019). *Deep learning for email spam detection: A comparative analysis*. Neural Computing and Applications, 31(11), 8205–8216.

[6] Salam, A., Al-Ayyoub, M., Aljawarneh, S., Jararweh, Y., & Gupta, B. (2020). *Email spam detection using machine learning: A comparative study*. IEEE Access, 8, 78782–78796.

[7] Kaur, G., & Kaur, M. (2020). *Review on email spam detection using machine learning techniques*. In Proceedings of the 10th International Conference on Cloud Computing, Data Science & Engineering (pp. 192–198).

[8] Prasad, S., & Pal, S. (2020). *Hybrid spam email detection using machine learning*. International Journal of Advanced Research in Computer Science, 11(4), 131–135.

[9] Mishra, A., Joshi, R. C., & Gaur, M. S. (2020). *A comprehensive review on email spam detection techniques using machine learning*. In Proceedings of the International Conference on Advances in Computing and Data Sciences (pp. 129–140). Springer, Singapore.

[10] Bharti, S. K., Singh, S., & Malhotra, A. (2019). *Machine learning-based spam email detection using optimized features*. In Proceedings of the International Conference on Advanced Computing and Intelligent Engineering (pp. 147–158). Springer, Singapore.

[11] Saini, R., & Kumar, R. (2020). *Comparative study of machine learning techniques for spam email detection*. In Proceedings of the International Conference on Computational Intelligence and Communication Technology (pp. 257–267). Springer, Singapore.

[12] Platt, J. C. (1999). *Using analytic QP and sparseness to speed training of support vector machines*. In Advances in Neural Information Processing Systems (pp. 557–563).

[13] Klimt, B., & Yang, Y. (2004). *Introducing the Enron Corpus*. In CEAS.

[14] Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., Paliouras, G., & Spyropoulos, C. D. (2000). *An evaluation of Naive Bayesian anti-spam filtering*. In Proceedings of the Workshop on Machine Learning in the New Information Age (pp. 9–17).

[15] Almeida, T. A., Hidalgo, J. M. G., & Yamakami, A. (2011). *Contributions to the study of SMS spam filtering: new collection and results*. In Proceedings of the 11th ACM Symposium on Document Engineering (pp. 259–262).

[16] Joachims, T. (1996). *A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization*. Carnegie Mellon University.

[17] Delany, S. J., Buckley, M., & Greene, D. (2012). *SMS spam filtering: methods and data*. Expert Systems with Applications, 39(10), 9899–9908.

[18] Guzella, T. S., & Caminhas, W. M. (2009). *A review of machine learning approaches to spam filtering*. Expert Systems with Applications, 36(7), 10206–10222.

[19] Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). *A Bayesian approach to filtering junk e-mail*. In Learning for Text Categorization: Papers from the 1998 Workshop (Vol. 62, pp. 98–105).

[20] Carreras, X., & Marquez, L. (2001). *Boosting trees for anti-spam email filtering*. In Proceedings of the 4th International Conference on Recent Advances in Natural Language Processing (pp. 58–64).