Kadali Devi Sindhuja, 2025, 13:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

# Email Spam Detection with Machine Learning

**Rajnish Kumar Chauhan, Praveen Yadav, Vishakha Kashyap,**
**Assistant Professor Dr. Chhaya Singh**
**MCA Final Year Student, Galgotias University,**
**GB Nagar, Greater Noida**

**Abstract- Email remains one of the most widely used communication tools, but the increasing volume of spam messages has become a persistent issue for both individuals and organizations. Traditional rule-based filtering methods struggle to keep up with the ever-changing techniques used by spammers, leading to inefficiencies in detection. To address this challenge, this study explores a machine learning-based approach to improve spam classification and enhance email security.**

**The research applies algorithms such as Random Forest, Logistic Regression, and K-Nearest Neighbors (KNN) to differentiate between spam and legitimate emails. By analyzing key features like email content, subject lines, and sender details, the model learns to identify patterns commonly found in spam messages. Performance evaluation using standard datasets demonstrates that machine learning significantly improves detection accuracy, speed, and adaptability compared to conventional methods. The findings suggest that machine learning offers a robust and scalable solution to the growing problem of email spam.**

**Keywords: Email, Spam Classification, Machine Learning, KNN, Term Frequency, Inverse Document Frequency, Natural Language Processing.**

## I. INTRODUCTION

Email has become an essential communication tool in both personal and professional settings, enabling quick and efficient information exchange. However, the rise of spam emails has led to challenges such as inbox overload, security threats, and potential data breaches. Many of these emails contain phishing links, malware, or fraudulent content, making them more than just an inconvenience—they pose real risks to users and organizations.

Traditional spam filters rely on predefined rules, but spammers constantly evolve their techniques, making it harder to detect and block unwanted messages effectively. This study explores the use of machine learning to enhance spam detection and improve email security. By implementing algorithms such as K-Nearest Neighbors (KNN) and analyzing email content using Term Frequency (TF) and Inverse Document Frequency (IDF), the system can recognize patterns and distinguish between legitimate and spam emails with greater accuracy. Furthermore, natural language processing (NLP) techniques are integrated to refine how the system interprets email content, ensuring it adapts to emerging spam trends. The ultimate goal of this research is to develop a smart and reliable spam detection model that helps users manage their emails efficiently while reducing security risks associated with spam.

- **Problem Statement :** The Growing Challenge of Spam Emails

Spam emails have become a persistent issue for users worldwide, cluttering inboxes, wasting time, and posing serious security risks. Many of the

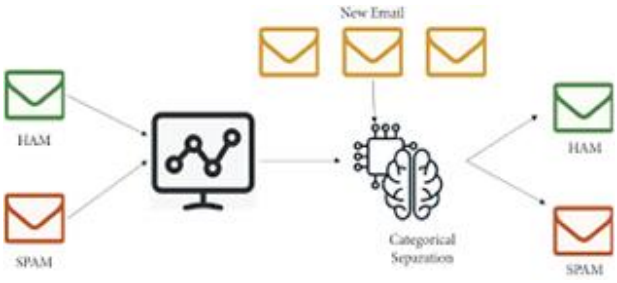


Figure: 1

messages contain phishing scams designed to steal sensitive information or malware that can damage devices and compromise data. Although traditional spam filters attempt to block these threats, they often fall short in providing accurate and reliable protection.

A major drawback of existing spam detection systems is their lack of accuracy. They sometimes misclassify important emails as spam (false positives) or fail to detect actual spam messages (false negatives), allowing harmful content to reach

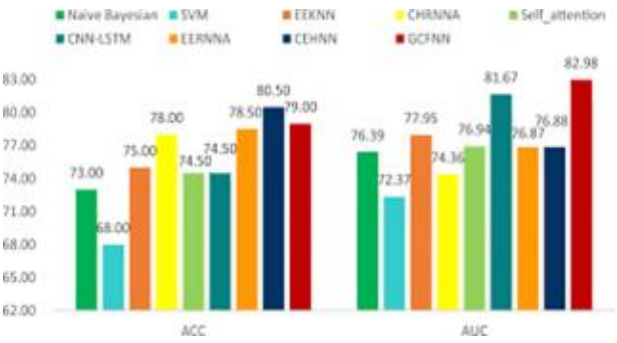


Figure: 2

users. Spammers continuously adapt their tactics, making it difficult for standard filters—many of which rely on static, rule-based methods—to keep up. These traditional approaches, such as blocking emails based on specific keywords, quickly become outdated as spammers develop new ways to bypass detection. This inability to adapt has led to growing security concerns and user frustration.

To overcome these challenges, this research introduces a machine learning-driven solution for spam detection. By leveraging algorithms like K-Nearest Neighbors (KNN), the system can analyze email content, identify patterns, and accurately classify messages as spam or legitimate. Unlike traditional filters, machine learning models continuously learn and improve, making them more effective at detecting evolving spam tactics without requiring constant manual updates. By integrating natural language processing (NLP) and advanced data analysis techniques, this approach aims to develop a more intelligent, adaptive, and highly accurate spam filter, ultimately enhancing email security and improving the user experience.

**Related Work**

A lot of work has already been done to fight spam emails, and many different methods have been tried. In the past, most spam filters were based on simple rules—like looking for specific words or phrases that were often found in spam emails. However, this approach didn't always work well because spammers are constantly changing their tactics, and some real emails would still get flagged as spam.

Recently, researchers have started using machine learning to improve spam detection. Machine learning models, like Naive Bayes and Support Vector Machines, can learn from past examples and get better at recognizing spam over time. These models often use something called Term Frequency-Inverse Document Frequency (TF-IDF), which helps them understand which words in an email are important and could indicate spam.

Other approaches are also exploring deep learning, which is a more advanced technique, though it can be more complicated and resource-heavy. Some studies also focus on analyzing email headers, metadata, and who sent the email, in addition to just the content inside.

Even though these methods have made spam detection better, there are still some challenges. For instance, it's tough to keep up with new spam strategies, and false positives (real emails getting flagged as spam) are still a problem. This paper builds on previous work by using machine learning

algorithms, like K-Nearest Neighbors (KNN), to create a smarter, more adaptable spam filter.

## II. LITERATURE REVIEW

Researchers have explored various methods to enhance spam email detection over the years. In the early stages, spam filtering was primarily based on rule-based systems, where emails were flagged as spam if they contained certain words or phrases commonly found in unwanted messages. While this approach was somewhat effective, it had significant limitations. Spammers could easily manipulate their messages by slightly altering words or using misleading formats, making these traditional methods unreliable. Additionally, rule-based filters often misclassified legitimate emails as spam, leading to unnecessary inconveniences for users.

With advancements in technology, machine learning emerged as a more efficient solution for spam detection. Various studies have demonstrated the effectiveness of algorithms such as Naïve Bayes, Support Vector Machines (SVM), and Decision Trees in classifying emails. Among these, Naïve Bayes has been widely used due to its simplicity and efficiency. However, it operates under the assumption that words in an email are independent of each other, which is not always the case, sometimes affecting its accuracy.

One of the most promising techniques in modern spam detection research is Term Frequency-Inverse Document Frequency (TF-IDF). This method improves spam classification by analyzing the occurrence of specific words in a dataset and determining their relevance. By identifying frequently used words in spam emails that rarely appear in legitimate messages, TF-IDF enhances the ability to detect spam more effectively.

Beyond content analysis, some researchers have explored email metadata, such as sender information and email headers, as additional indicators of spam. These elements provide valuable clues that can strengthen the detection process. Despite these advancements, challenges remain, including minimizing false positives (where genuine emails are mistakenly classified as spam) and ensuring the system can adapt to evolving spam techniques.

This review underscores the evolution of spam detection methods, highlighting how machine learning approaches, particularly TF-IDF and K-Nearest Neighbors (KNN), contribute to developing more accurate and adaptable spam filters. These techniques continue to refine spam detection, offering improved accuracy and efficiency in distinguishing legitimate emails from unwanted messages.

## III. RESEARCH METHODOLOGY

This study aims to develop a machine learning-based system capable of automatically detecting spam emails. The methodology follows a structured approach, beginning with data collection and concluding with model evaluation to ensure the system's effectiveness.

The first step in the process is collecting a dataset that consists of both spam and legitimate emails.

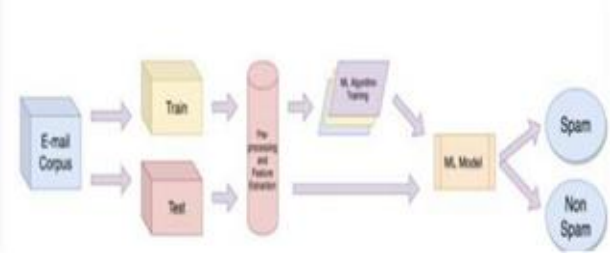


Figure: 3
Shows the training data phase

This dataset is essential because it allows the system to learn the characteristics of spam messages. Once the data is gathered, it undergoes a preprocessing stage, where unnecessary elements such as common words (e.g., "a," "the," "and") and special characters are removed.

This step is crucial in ensuring that only meaningful content is analyzed, preventing distractions that could impact the model's accuracy.

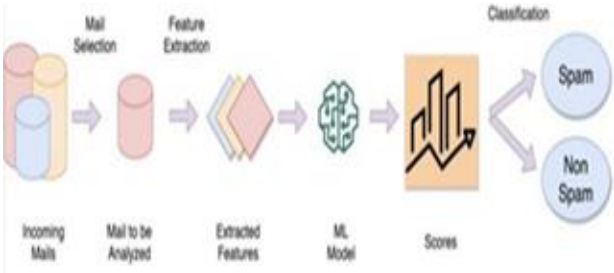


Figure: 4 Shows the Proposed New E-mail Classification

Following data cleaning, the next step is feature extraction, which involves identifying important words and patterns that help differentiate spam emails from legitimate ones. A widely used technique for this purpose is Term Frequency-Inverse Document Frequency (TF-IDF). This method assigns importance to words based on how frequently they appear in an email compared to their occurrence across the entire dataset. Words that frequently appear in spam emails but are rare in regular messages serve as strong indicators for spam classification.

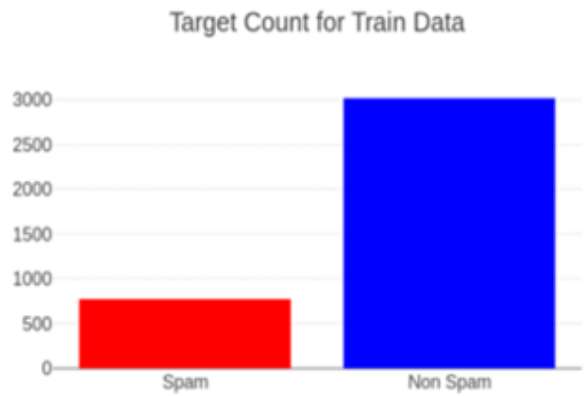


Figure: 5 Shows the target data set

After extracting relevant features, the next phase is applying machine learning algorithms. This study employs the K-Nearest Neighbors (KNN) algorithm, which determines whether an email is spam or not by comparing it with similar examples in the dataset. The model is first trained on a subset of the dataset, allowing it to recognize spam characteristics. It is then tested using a different portion of the dataset that it has not encountered before, ensuring that the system is capable of making accurate predictions on new emails.

The final step in the methodology is evaluating the model's performance.

Several metrics are used to assess its effectiveness, including accuracy (how often the model correctly classifies emails), precision (how many of the emails flagged as spam are actually spam), and recall (how many spam emails are correctly identified by the system). These performance indicators help determine the strengths and weaknesses of the spam detection system, allowing for further improvements.

By following this systematic approach, the research aims to create a highly accurate and reliable spam detection system that can adapt to evolving spam tactics, ensuring enhanced security and efficiency in email communication.

## IV. PROPOSED SYSTEM

To effectively address the issue of spam emails, this system implements a spam classification mechanism that distinguishes between spam and non-spam messages. Since spam emails are frequently sent multiple times, manually identifying them each time becomes inefficient. To enhance detection, the system not only identifies spam content but also tracks the IP address from which spam emails originate. If a spam message is detected, the associated IP address is blacklisted, allowing future emails from the same source to be automatically flagged as spam.

**System Implementation**

The system is developed as a web-based application using .NET, incorporating machine learning techniques for spam detection. It consists of the following modules:

**User Management**

- New users must register to create an account before accessing the system.
- Registered users can log in using their credentials.
- If incorrect login details are provided, an error message is displayed.

**Email Composition**

- Users can draft a new email by entering the recipient's email address, subject, and message.
- Once sent, the email is delivered to the specified recipient.

**Inbox**

- Stores all received emails, arranging them in chronological order.
- Users can access and read incoming messages.

**Sent Items**

- Maintains a record of all emails sent by the user.
- Users can review previously sent messages.

**Trash Management**

- Emails that are deleted by the user are moved to the trash folder.
- Deleted emails remain in the trash bin for retrieval if needed.

**Voice Messaging Feature**

- Emails are sent as text-based messages, with an option for recipients to listen to them using a voice note feature.

**Offline Notification System**

- If the recipient is offline, they receive a text message (SMS) notification about the new email.

**Delete for Everyone**

- Allows the sender to delete an email even after it has been sent.
- Once deleted, the email is erased for both the sender and the recipient.

**Read Receipt**

- When a recipient reads an email, the sender receives a notification confirming that the message has been read.
- Received emails are exported to a dataset for spam classification.
- The system applies the Naïve Bayes Classifier to determine whether an email is spam or legitimate.
- Before classification, the model undergoes training, as explained in the subsequent section.

## V. SPAM DETECTION USING MACHINE LEARNING

For training the algorithm dataset from Kaggle is used which is shown below.

2. It has many fields, some of these columns of the dataset are not required. So remove some columns which are not required. We need to change the names of the columns.

```
In [28]: mails = pd.read_csv('spam.csv', encoding = 'latin-1')
         mails.head()
```

Out[28]:

| | v1 | v2 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 |
|---|---|---|---|---|---|
| 0 | ham | Go until jurong point, crazy.. Available only ... | NaN | NaN | NaN |
| 1 | ham | Ok lar... Joking wif u oni... | NaN | NaN | NaN |
| 2 | spam | Free entry in 2 a wkly comp to win FA Cup fina... | NaN | NaN | NaN |
| 3 | ham | U dun say so early hor... U c already then say... | NaN | NaN | NaN |
| 4 | ham | Nah I don't think he goes to usf, he lives aro... | NaN | NaN | NaN |

Figure:6 Dataset

```
In [30]: mails.rename(columns = {'v1': 'labels', 'v2': 'message'}, inplace = True)
         mails.head()
```

Out[30]:

| | labels | message |
|---|---|---|
| 0 | ham | Go until jurong point, crazy.. Available only ... |
| 1 | ham | Ok lar... Joking wif u oni... |
| 2 | spam | Free entry in 2 a wkly comp to win FA Cup fina... |
| 3 | ham | U dun say so early hor... U c already then say... |
| 4 | ham | Nah I don't think he goes to usf, he lives aro... |

Figure:7 Classification dataset

With the help of NLTK (Natural Language Tool Kit) for the text processing, Using Matplotlib you can plot graphs , histogram and bar plot and all those things ,Word Cloud is used to present text data and pandas for data manipulation and analysis, NumPy is to do the mathematical and scientific operation. The packages used in the proposed model are shown below.

```
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from math import log, sqrt
import pandas as pd
import numpy as np
import re
%matplotlib inline
```

Figure: 8 Packages

3. Split the data into training and testing sets as shown below. Some percentage f the data set is used as train dataset and the rest as a test dataset.

```
totalMails = 4825 + 747
trainIndex, testIndex = list(), list()
for i in range(mails.shape[0]):
    if np.random.uniform(0, 1) < 0.75:
        trainIndex += [i]
    else:
        testIndex += [i]
trainData = mails.loc[trainIndex]
testData = mails.loc[testIndex]
```

Figure: 9 Train dataset

4. Whenever there is any message, we must first preprocess the input messages. We need to convert all the input characters to lowercase.

5. Then split up the text into small pieces and also removing the punctuations. So the Tokenization process is used to remove punctuations and splitting messages.

6. The Porter Stemming Algorithm is used for stemming. Stemming is the process of reducing words to their root word.

7. We need to find the probability of the word in spam and ham messages.

$$P(word) = \frac{Total\ occurrences\ of\ word}{Total\ number\ of\ words}$$

Figure: 10 Frequency of word

Then spam word frequency is calculated as follows:

$$P(word|spam) = \frac{Total\ Occurences\ of\ spam\ message}{Total\ number\ of\ words\ in\ the\ spam\ message}$$

Figure: 11 Spam word frequency

8. Tf –idf(term frequency-inverse document frequency) has to be calculated. TF: Term Frequency, which measures how many times a term occurs in a document. TF(t) = (Number of times t appeared in a document) / (Total terms in the document). IDF: Inverse Document Frequency, which measures the significance of the term. IDF(t) = loge(Total documents / documents with term t in it).

9. See how well the model performed by evaluating Naïve Bayes Classifier and showing the accuracy score.

## VI. RESULTS AND DISCUSSIONS

When we receive message in the inbox ,that message will be exported to dataset as shown below. This message will be detected as spam or not.

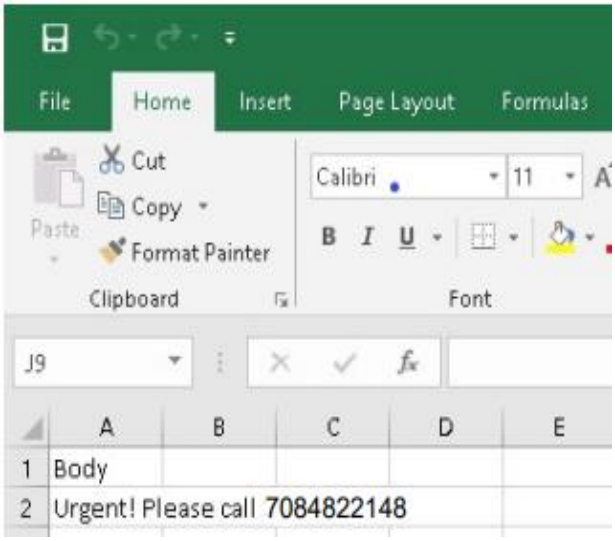


Figure: 12 Exported Dataset

The exported message will be detected as spam or not using Bayes' theorem and Naive Bayes' Classifier following all the steps discussed above along with finding probability of words in spam and ham messages to detect it as spam or not. The below figures shows message which got detected as spam and ham. If "Urgent! Please call 7084822148" is an exported message from the inbox to the dataset then based on trained dataset and using Bayes' theorem and Naive Bayes' Classifier, the above message is detected as Spam.

## VII. CONCLUSION

Email has become an essential part of our daily communication, allowing messages to be shared instantly across the world. However, with over 270 billion emails sent every day, nearly 57% are spam—cluttering inboxes, posing security threats, and sometimes even leading to financial loss. These unwanted emails often contain phishing links or malicious attachments designed to steal sensitive information, making them more than just a nuisance.

To tackle this growing problem, our system is designed to detect and filter out spam emails effectively, helping users keep their inboxes clean and secure. By reducing the number of unsolicited messages, this system enhances both personal and organizational security. Looking ahead, future improvements could include more advanced algorithms and additional features to make spam detection even more accurate and adaptable to evolving threats.

## REFERENCES

1. Shukor Bin Abd Razak, Ahmad Fahrulrazie Bin Mohamad "Identification of Spam Email Based on Information from Email Header" 13th International Conference on Intelligent Systems Design and Applications (ISDA), 2013.
2. Mohammed Reza Parsei, Mohammed Salehi "E-Mail Spam Detection Based on Part of Speech Tagging" 2 nd International Conference on Knowledge Based Engineering and Innovation (KBEI), 2015.
3. Sunil B. Rathod, Tareek M. Pattewar "Content Based Spam Detection in Email using Bayesian Classifier", presented at the IEEE ICCSP 2015 conference.
4. Aakash Atul Alurkar, Sourabh Bharat Ranade, Shreeya Vijay Joshi, Siddhesh Sanjay Ranade, Piyush A. Sonewa, Parikshit N. Mahalle, Arvind V. Deshpande "A Proposed Data Science Approach for Email Spam Classification using Machine Learning Techniques", 2017.
5. Kriti Agarwal, Tarun Kumar "Email Spam Detection using integrated approach of Naïve Bayes and Particle Swarm Optimization", Proceedings of the Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018.
6. Cihan Varol, Hezha M.Tareq Abdulhadi "Comparison of String Matching Algorithms on Spam Email Detection", International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism Dec, 2018.
7. C.P. Lueg, from spam filtering to information retrieval and back: seeking conceptual foundations for spam filtering, Proc. Assoc. Inf. Sci. Technol. 42 (1) (2005).
8. Emmanuel Gbenga Dada, Joseph Stephen Bassi, Machine learning for email spam filtering: review, approaches, and open research problems.
9. Loredana Fire, Camelia Lemnaru, Spam Detection Filter using KNN Algorithm and Resampling.
10. D.M. Fonseca, O.H. Fazzion, E. Cunha, I. Las-Casas, P.D. Guedes, W. Meira, M. Chaves, measuring characterizing, and avoiding spam traffic costs, IEEE Int. Comp. 99 (2016).