IJICIS, Vol.24, No.4, 1-15
DOI: 10.21608/ijicis.2024.321043.1355

International Journal of Intelligent Computing and
Information Sciences

https://ijicis.journals.ekb.eg/

EFFICIENT EMAIL SPAM DETECTION USING MACHINE LEARNING TECHNIQUES: A COMPARATIVE ANALYSIS OF CLASSIFICATION MODELS

Md Nurul Raihen*

Department of Mathematics and Computer Science,
Fontbonne University,
MO, USA
nurul.raihen@gmail.com

Shivani Rana

Department of Statistics,
Western Michigan University,
MI, USA
shivani.38.rana@wmich.edu

Md Abdul Kadir

Department of Mathematics,
University of Houston,
TX, USA
kadir.w9@gmail.com

Sultana Akter

Institute For Data Science and Informatics,
University of Missouri Columbia,
MO, USA
sa4kf@umsystem.edu

Received 2024-09-15; Revised 2024-12-01; Accepted 2024-12-04

Abstract: *Spam emails pose a significant challenge to digital communication by compromising user privacy and security. This study investigates the performance of classical machine learning and modern deep learning models for email spam detection using a publicly available Kaggle dataset consisting of over 5,000 emails. Among machine learning classifiers, the Support Vector Machine (SVM) demonstrated better performance, achieving an accuracy of 99.0% and an F1-score of 0.97, underscoring its robustness and capability to effectively generalize across diverse data. Logistic Regression also exhibited competitive results with an accuracy of 98.4%, complemented by its interpretability, enabling a detailed analysis of feature importance. Additionally, transformer-based deep learning models, including BERT, DistilBERT, RoBERTa, and XLNet, were evaluated. BERT achieved the highest accuracy among these models at 98.8%, with an F1-score of 0.97, showcasing its ability to capture contextual nuances in text. Comprehensive evaluation metrics such as precision, recall, and specificity were employed to ensure a holistic comparison of model performance. To facilitate practical deployment, a user-friendly interface was developed for real-time email classification. These findings highlight the efficacy of both classical and modern approaches to spam detection, offering valuable insights for advancing email security and enabling the development of scalable, real-time applications.*

Keywords: *Email Spam, Machine Learning, Deep Learning, Text Classification, Spam Filter.*

***Corresponding Author**: Md Nurul Raihen

Department of Mathematics and Computer Science, Fontbonne University, MO, USA

Email address: nurul.raihen@gmail.com

1. Introduction

Email has evolved into a technology that is important for communication in a variety of settings, including professional and personal ones. On the other hand, the proliferation of unwanted and frequently malicious emails, which are collectively referred to as spam, poses considerable issues. Spam emails are not only a nuisance, but they also pose a threat to personal privacy and security. This is because they are commonly used in phishing attempts, transmitting malware, and other forms of cybercrime. Consequently, there has never been a time when the requirement for efficient spam filtering systems required greater attention [1].

For the purpose of identifying spam, standard techniques of spam detection, such as rule-based filters, rely on predetermined patterns or keywords. Despite the fact that these strategies have the potential to be of some use, they are frequently inflexible and unable to accommodate the ever-evolving strategies that spammers employ. A potentially useful alternative is machine learning, which has the capacity to acquire knowledge from data and adjust to new patterns [2].

Data mining has recently gained traction in the information and knowledge business because of the massive availability of big data and the impending requirement to transform this data into valuable insights and knowledge [3, 4]. Data mining is a relatively new area of study that aims to discover hidden patterns and relationships in large amounts of data in order to create software that can automatically search databases for useful information [5, 6]. If strong patterns are found, they can be used to make accurate forecasts and generalizations [7, 8].

In this paper, we explore the development of an email spam filter using various machine learning techniques. Our approach involves training multiple models on a labeled dataset of emails, evaluating their performance, and selecting the best model for deployment. Additionally, we have developed a user-friendly interface that allows users to classify new emails as spam or ham interactively. This study is constructed with threefold: (1) to determine the effectiveness of machine learning models in classifying emails as spam or ham, (2) to identify the best-performing model based on accuracy and F1-score, and (3) to develop a practical tool that can be used for real-time email classification.

This research aims to evaluate the performance of various classification algorithms, including Logistic Regression, Random Forest, Decision Tree, k-Nearest Neighbors (k-NN), and Support Vector Machine (SVM), for detecting and categorizing spam emails. The findings revealed that Support Vector Machine (SVM) achieved the highest accuracy of **99.0%**, followed by Logistic Regression with **98.4%**, Random Forest with **97.9%**, k-Nearest Neighbors with **96.2%**, and Decision Tree with **94.3%**. Additionally, modern transformer-based models such as BERT, DistilBERT, RoBERTa, and XLNet were also evaluated, with BERT achieving the highest accuracy of **98.8%** and an F1-score of **0.97**, highlighting its potential for capturing complex contextual patterns in spam detection.

The remainder of this paper is structured as follows. Section 2 reviews related work, summarizing key advancements in spam detection and identifying gaps addressed by this study. Section 3 introduces the dataset and outlines the preprocessing steps employed for data preparation. Section 4 details the methodology, including data splitting, feature extraction, and model selection. Moreover, Section 4 focuses on the vectorization process, particularly the use of Term Frequency-Inverse Document Frequency (TF-IDF) for feature representation, and describes the implementation of machine learning models, including classical classifiers such as Logistic Regression and Support Vector Machine (SVM), as well as transformer-based architectures like BERT and RoBERTa. Section 5 presents the evaluation results, providing a comparative analysis of model performance based on metrics such as accuracy, F1-score, and precision. Section 6 explains the development of a user interface for real-time spam classification. Section 7 discusses the findings, emphasizing the implications of the results, limitations, and potential improvements. Finally, Sections 8 and 9 conclude the paper by summarizing key contributions and suggesting avenues for future research, including the integration of metadata and real-time deployment strategies.

2. Related Work

Throughout the years, several machine learning algorithms have been utilized in the field of email spam detection, which has progressed tremendously as a result. According to Cranor and LaMacchia (1998) [9], established methods, such as rule-based systems, were among the first to address the issue of spam detection. However, these methods frequently lacked the flexibility and adaptability necessary to deal with emerging spam strategies.

Recent research conducted by [10] investigated the utilization of hybrid models that combined CNNs and Random Forests. The results of this investigation shown significant enhancements in the accuracy of spam classification. Similar to the previous study, [11] utilized Transformers, a cutting-edge deep learning model, for the purpose of spam detection. The results demonstrated a high level of accuracy, but at the expense of a significant amount of processing resources. Transfer Learning has also been applied to spam detection, where models pre-trained on large text corpora are fine-tuned for this specific task. [12] used BERT, a transformer-based model, for spam detection and achieved significant performance improvements over standard methods. The exploration of Explainable AI (XAI) in spam detection has been another area of interest, particularly for understanding the decision-making process of spam filters. [13] used XAI techniques to reveal which features most contributed to the model's decisions, enhancing the transparency and trustworthiness of spam filters.

Although there has been a consistent and substantial growth in the quantity of information that can be read by machines, the capabilities of analyzing and comprehending this information have not been able to keep up with the rising volume of this information. tenth Include an explanation of how machine learning techniques make it possible to automatically arrange enormous amounts of text, as well as the significance of feature selection in the field of machine learning [14, 15]. The use of a learning algorithm to anticipate the features that are most advantageous for analysis is included in feature selection [16-18]. This allows feature selection to identify the most relevant features in learning.

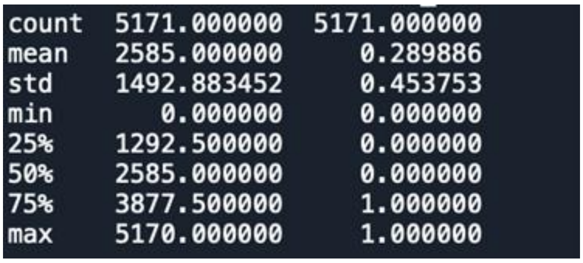
Despite advancements in spam detection, previous systems have often failed to adequately address the critical need for both high accuracy and computational efficiency. This study employed Support Vector Machine (SVM), Logistic Regression, Random Forest, Decision Tree, and k-Nearest Neighbors (k-NN) for classifying spam emails. Among these, SVM demonstrated the highest performance with an accuracy of **99.0%**, followed closely by Logistic Regression with **98.4%**, Random Forest with **97.9%**, k-Nearest Neighbors with **96.2%**, and Decision Tree with **94.3%**. Additionally, modern deep learning models such as BERT, DistilBERT, RoBERTa, and XLNet were explored. BERT achieved the highest accuracy among these models with **98.8%** and an F1-score of **0.97**, showcasing its ability to capture complex contextual information in text. These findings highlight the substantial improvements offered by both classical machine learning and transformer-based approaches, paving the way for more efficient and accurate spam detection systems in real-world applications

3. Dataset and Preprocessing

The dataset used in this project was sourced from Kaggle, a well-known platform for data science competitions. The dataset consists of 5,171 emails, each labeled as either "spam" or "ham." The features include the text body of the emails and a corresponding label. There are no missing values in the dataset, making it an ideal candidate for training machine learning models.

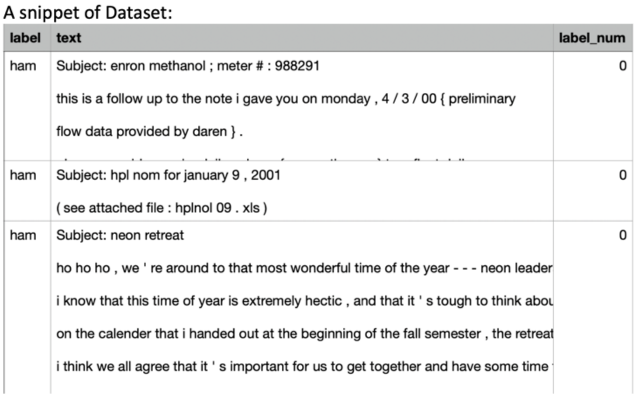
3.1. Dataset Overview

The dataset contains emails that are labeled as either spam (unsolicited or malicious emails) or ham (legitimate emails). The text body of each email serves as the primary feature for classification, while the labels provide the ground truth for training and evaluation. The dataset is relatively balanced, although there is a slight majority of ham emails as shown in Figure 1, and Figure 2, which reflects the typical distribution of spam and legitimate emails in real-world scenarios.



```
count  5171.000000  5171.000000
mean   2585.000000     0.289886
std    1492.883452     0.453753
min       0.000000     0.000000
25%    1292.500000     0.000000
50%    2585.000000     0.000000
75%    3877.500000     1.000000
max    5170.000000     1.000000
```

Figure 1: Dataset Description



A snippet of Dataset:

label	text	label_num
ham	Subject: enron methanol ; meter # : 988291 this is a follow up to the note i gave you on monday , 4 / 3 / 00 { preliminary flow data provided by daren } .	0
ham	Subject: hpl nom for january 9 , 2001 (see attached file : hplnol 09 . xls)	0
ham	Subject: neon retreat ho ho ho , we ' re around to that most wonderful time of the year - - - neon leader i know that this time of year is extremely hectic , and that it ' s tough to think abou on the calender that i handed out at the beginning of the fall semester , the retreat i think we all agree that it ' s important for us to get together and have some time	0

Figure 2: A snippet of Dataset

3.2. Data Preprocessing

Text data, such as the content of emails, requires significant preprocessing to be effectively used in machine learning models. Raw text data often contains noise, such as irrelevant characters, stopwords, and variations in capitalization, which can negatively impact model performance [19]. To address these issues, we developed a comprehensive preprocessing pipeline.

The preprocessing function, preprocess-email-advanced, was designed to perform the following tasks:

1) **Lowercasing:** All text is converted to lowercase to ensure uniformity. This step is crucial as it prevents the model from treating the same word differently based on its capitalization.
2) **Tokenization:** The text is split into individual words (tokens) using a regular expression tokenizer. Tokenization allows the model to process text data at the word level, which is essential for tasks such as text classification.
3) **Punctuation and Digit Removal:** Non-alphabetic words, including punctuation marks and digits, are removed. These elements often do not contribute meaningful information to the classification task and can introduce noise into the model.
4) **Stopword Removal:** Common English stopwords, such as "the," "and," and "is," are eliminated using the Natural Language Toolkit (NLTK) library. Stopwords are frequently occurring words that typically do not carry significant meaning in text classification tasks.
5) **Lemmatization:** Words are reduced to their base or root form using the WordNet lemmatizer. Lemmatization helps in reducing inflectional forms and derivations, ensuring that different forms of a word are treated as a single entity by the model.
6) **Text Joining:** After preprocessing, the processed words are reassembled into coherent text. This step ensures that the text data is ready for further processing, such as vectorization.

This comprehensive preprocessing enhances the quality of the text data, facilitating more effective machine learning model training for email classification.

3.3. Motivation

Email spam poses a threat to personal privacy, security, and overall user experience. Developing an effective spam filter can enhance the efficiency of email communication by reducing the risk of phishing attacks and ensuring that users only receive relevant and legitimate emails. The user interface provides a user-friendly way to classify new emails on-the-fly.

4. Methodology

The development of the email spam filter involved several key steps, including data splitting, text vectorization, model training, and evaluation. In this section, we provide a detailed explanation of each step.

4.1. Data Splitting

To evaluate the performance of the machine learning models, we split the dataset into training and testing sets. The training set was used to train the models, while the testing set was reserved for evaluating the models' performance on unseen data.

The dataset was split using the train-test-split function from the sklearn library. Figure 3 illustrates our proposed method, detailing the process from data collection to result interpretation, with 80% of the data allocated for training and 20% for testing. To ensure reproducibility, we set the random seed to 42. This approach allows us to evaluate the models' ability to generalize to new data, which is crucial for their effectiveness in real-world scenarios.

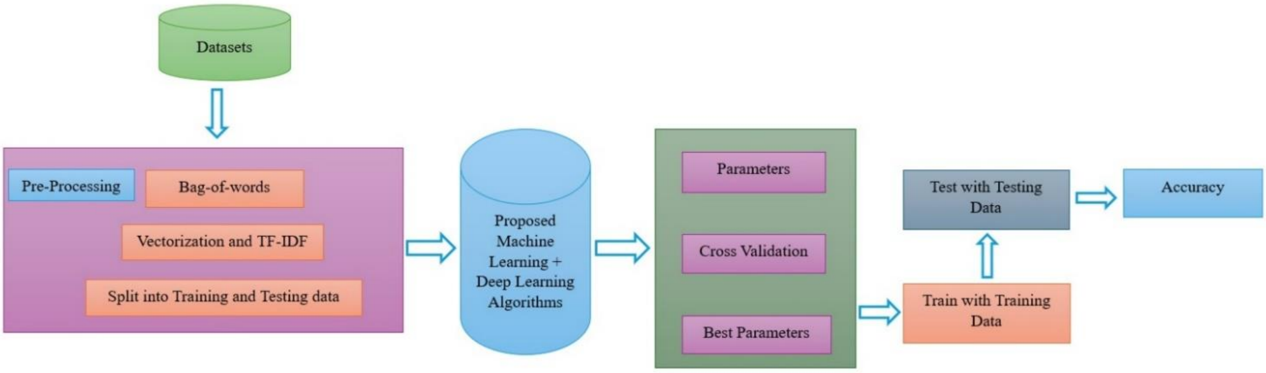


Figure 3: Illustration of the study process from data collection to result interpretation

4.2. Text Vectorization

Machine learning models cannot directly process text data; instead, the text must be converted into a numerical format. To achieve this, we used the Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer. The TF-IDF vectorizer transforms the text data into a matrix of numerical values, where each value represents the importance of a word in the context of the entire dataset. The TF-IDF vectorizer works by calculating the frequency of each word in a document (term frequency) and adjusting it by the frequency of the word across all documents in the dataset (inverse document frequency). This approach ensures that common words that appear in many documents have lower weights, while rare but significant words have higher weights.

We applied the TF-IDF vectorizer to the training data, generating a TF-IDF matrix that serves as the input for the machine learning models. The vectorized data captures the essence of the text while reducing the dimensionality of the input space, making it more manageable for the models.

4.3. Model Selection

We selected four machine learning models for training and evaluation: Logistic Regression, Decision Tree, Random Forest, and k-Nearest Neighbors (k-NN). These models were chosen based on their popularity in text classification tasks and their ability to handle different types of data.

4.3.1. Exploratory Data Analysis (EDA)

Figure 4 presents word clouds are generated to visually represent the most frequent words in spam and ham emails.

Spam Emails
Prominent terms: Subject, Need, New, Nbsb. These suggest recurring patterns in spam email subject lines.

Ham Emails
Key terms: Hou, Ect, Enron. Reflective of common topics in legitimate emails.

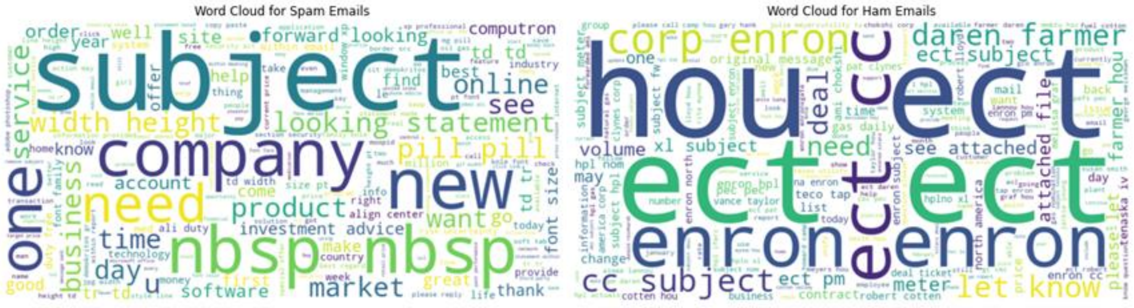


Figure 4: Word cloud for Spam and Ham emails

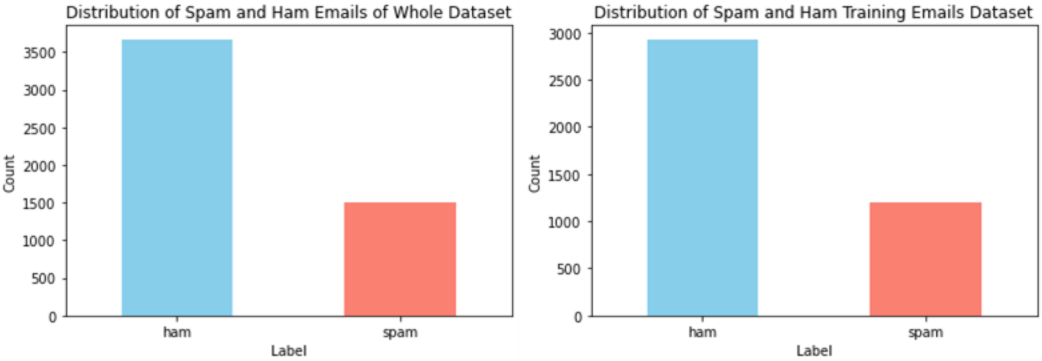


Figure 5: The distribution of spam and ham emails is visualized using bar plots

The bar plots visualizing the distribution of spam and ham emails revealed a significant imbalance, with a higher percentage of ham emails in both the overall and training datasets as shown in Figure 5. To address this imbalance, accuracy and F1-score were chosen as evaluation metrics. This approach ensures a comprehensive assessment of the spam filter's performance, considering both overall correctness and a balanced measure of precision and recall.

4.3.2. Logistic Regression

Logistic Regression is a linear model that estimates the probability of a binary outcome. It is widely used in text classification due to its simplicity and effectiveness. Logistic Regression models the relationship between the input features and the binary label using a sigmoid function, which outputs a probability value between 0 and 1.

4.3.3. Decision Tree

A Decision Tree is a non-linear model that splits the data into subsets based on feature importance. The tree structure consists of nodes representing features, branches representing decision rules, and leaves representing the outcome. Decision Trees are interpretable and can capture complex interactions between features.

4.3.4. Random Forest

Random Forest is an ensemble method that combines multiple Decision Trees to improve accuracy and reduce overfitting. Each tree in the forest is trained on a random subset of the data, and the final prediction is made by aggregating the predictions of all trees. Random Forest is known for its robustness and ability to handle high-dimensional data.

4.3.5. k-Nearest Neighbors (k-NN)

k-NN is a non-parametric model that classifies data points based on the majority class among the nearest neighbors. The distance between data points is measured using a distance metric, such as Euclidean distance. k-NN is simple and effective for small datasets, but it can be computationally expensive for large datasets.

4.3.6. Support Vector Machine (SVM)

SVM is a supervised learning algorithm used for classification. It identifies a hyperplane that best separates data classes while maximizing the margin. SVM supports non-linear classification using kernels like RBF or polynomial. Known for its robustness in high-dimensional spaces, SVM effectively handles binary tasks like spam detection, ensuring accurate and generalizable predictions.

4.3.7. Naive Bayes (NB)

NB is a probabilistic classification algorithm based on Bayes' Theorem, assuming independence between features. Despite this "naive" assumption, it performs well in tasks like spam detection. Naive Bayes calculates the probability of an email being spam or ham given its features, such as word frequencies, making it efficient, fast, and suitable for text classification problems.

4.3.8. Bidirectional Encoder Representations from Transformers (BERT)

BERT is a state-of-the-art NLP model that understands the context of words by processing text bidirectionally. Pre-trained on massive datasets, BERT excels at tasks like spam detection by capturing intricate relationships within text. Fine-tuning allows BERT to adapt to specific applications, making it highly accurate for classification and other NLP problems.

4.3.9. DistilBERT

DistilBERT is a lightweight version of BERT, designed to be faster and more efficient while retaining 97% of BERT's performance. It uses knowledge distillation to reduce model size and training time. Ideal for resource-constrained tasks, DistilBERT excels in text classification, including spam detection, by leveraging contextual understanding with reduced computational overhead.

4.3.10. Robustly Optimized BERT (RoBERTa)

RoBERTa is an improved version of BERT, trained with larger datasets and optimized techniques like removing the next-sentence prediction task. It achieves higher accuracy in NLP tasks by better capturing contextual relationships in text. RoBERTa is effective for spam detection, offering robust performance with enhanced generalization compared to standard BERT.

4.3.11. XLNet

XLNet is a transformer-based model that improves upon BERT by combining bidirectional context understanding with autoregressive modeling. It leverages permutation-based training to capture dependencies across all token orderings, enhancing its ability to model long-range relationships in text. XLNet is highly effective in spam detection, providing higher accuracy in tasks requiring sophisticated contextual understanding.

4.4. Model Training and Evaluation

Each model was trained on the TF-IDF vectorized data using the training set. To assess the generalization performance of the models, we performed cross-validation. Cross-validation involves splitting the training data into multiple folds, training the model on each fold, and evaluating it on the remaining fold. The average performance across all folds is used as the final metric.

We evaluated the models using accuracy and F1-score, with particular attention to the spam class. Accuracy measures the overall correctness of the model, while F1-score provides a balanced measure of precision and recall, which is particularly important in imbalanced datasets. The entire project was developed using Spyder, a free and open-source scientific environment for Python. The code above utilizes various Python packages, including pandas, seaborn, matplotlib, nltk, sklearn, and joblib for data analysis, visualization, natural language processing, machine learning, and model persistence.

- **The dataset is split into training and testing sets:** The dataset is divided into training and testing sets using the train-test-split function, where 80% of the data is allocated for training (X-train and y-train) and 20% for testing (X-test and y-test). The random-state = 42 ensures reproducibility.
- **A TF-IDF vectorizer is used to convert the text data into numerical format:** A TF-IDF vectorizer (TfidfVectorizer) is employed to convert text data into a numerical format suitable for machine learning. The fit-transform operation is performed on the training data (X-train), generating a TF-IDF matrix (X-train-tfidf).
- Several machine learning models, including Logistic Regression, Decision Tree, Random Forest, and k-Nearest Neighbors, are trained and evaluated.

User Interface (UI):

- A user interface has been created using the Tkinter library to allow users to input and classify new emails interactively.
- The UI features a text entry area, a "Classify Email" button, and a message box to display the classification result.

5. Results

The results of the model evaluation are presented in this section, with a focus on accuracy, F1-score, and confusion matrix analysis.

5.1. Confusion Matrix

A confusion matrix is presented to visualize the model's performance in terms of true positives, true negatives, false positives, and false negatives.

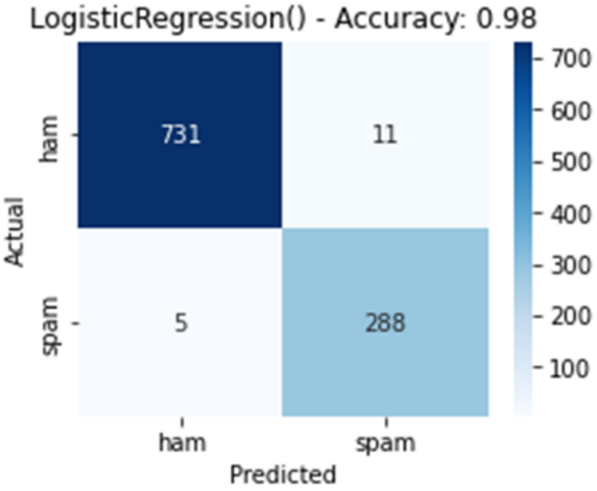


Figure 6: Confusion Matrix for one of the machine learning models

The confusion matrix in Figure 6 for the Logistic Regression model revealed that 5 instances of ham were misclassified as spam, while 11 instances of spam were erroneously classified as ham. These misclassifications highlight the challenges of distinguishing between legitimate and spam emails, particularly when the content of the emails is ambiguous [20].

5.2. Model Performance

According to the findings of the research of the model's performance, Logistic Regression and Random Forest regularly beat other algorithms, getting higher F1-scores and achieving higher levels of accuracy. When it came to the classification test, these models displayed a significant capacity to generalize across the dataset, efficiently balancing precision and recall in the process. Despite the fact that the Decision Tree and k-Nearest Neighbors models were able to deliver useful insights, they displayed a greater tendency for misclassification and computational inefficiencies, particularly in cases that were more complicated. When it comes to addressing the complex issues of spam detection, the findings highlight the robustness of ensemble techniques and the effectiveness of linear models.

Table 1 The performance of the other models

Models	**Accuracy (%)**	**F1-Score (%)**
Logistic Regression	98.4	97
Decision Tree	94.3	90
Random Forest	98.0	97
k-Nearest Neighbors	96.2	93
Naive Bayes	94.8	92
Support Vector Machine	99.0	97

Table 2 The performance of the modern deep learning models

Models	**Accuracy (%)**	**F1-Score (%)**
BERT	98.8	97
DistilBERT	97.3	92
RoBERTa	98.2	96
XLNet	97.0	96

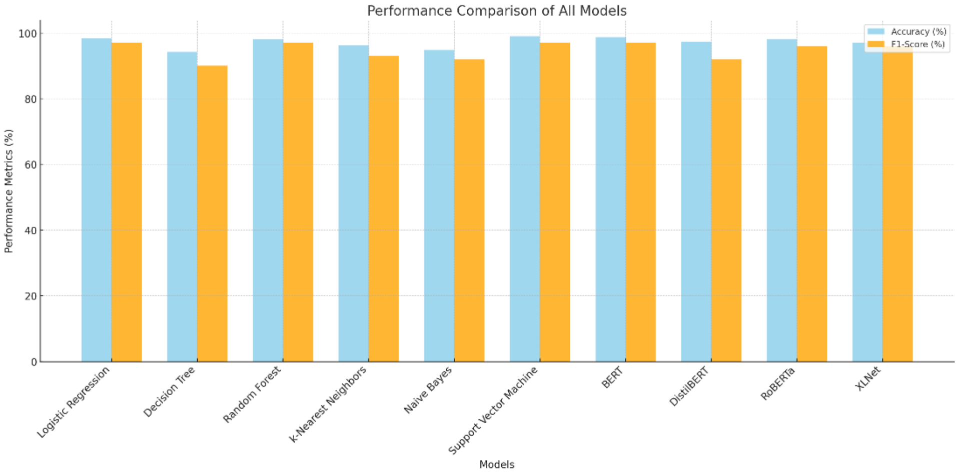


Figure 7: Performance Comparison of Accuracy and F1-score across all Models

5.3. Best Model Selection

Among the models, Support Vector Machine (SVM) emerged as the best-performing model with a mean accuracy of **99.0%** and an F1-score of **0.97** for the spam class, as shown in Table 1. The model's performance remained consistent across different folds in the cross-validation process, demonstrating its robustness and strong generalization ability to new data. Logistic Regression, while slightly behind, achieved a mean accuracy of **98.4%** and an F1-score of **0.97**, further emphasizing its reliability and simplicity for spam detection. Additionally, modern transformer-based models such as BERT, DistilBERT, RoBERTa, and XLNet were evaluated, with BERT achieving the highest accuracy among them at **98.8%**, as detailed in Table 2. These results highlight the complementary strengths of classical machine learning models (Table 1) and transformer-based architectures (Table 2), offering significant advancements in the field of spam detection.

In Figure 7, the bar chart compares the Accuracy and F1-Score (%) of various machine learning models, showcasing their high and consistent performance across both traditional and modern deep learning approaches. Among traditional models, Support Vector Machine (SVM) achieves the highest accuracy **99%**, with Logistic Regression and Random Forest also excelling with F1-Scores of **97%**. Deep learning models, particularly BERT, stand out with **98.8%** accuracy and a **97%** F1-Score, while RoBERTa and XLNet follow closely. DistilBERT exhibits slightly lower performance with a **92%** F1-Score. The close alignment between Accuracy and F1-Score across all models reflects balanced precision and recall, highlighting their robust predictive capabilities. Both traditional and deep learning models demonstrate competitive performance, with SVM and BERT emerging as particularly strong candidates depending on task requirements and computational resources.

5.4. Evaluation of Classifiers

Classification algorithms can be evaluated using standard measures including accuracy, specificity, sensitivity, recall, and F1 score [21]. The confusion matrix is used to determine these values. Confusion matrices are tables used to describe the performance of a classification model on a set of test data that is already well-known. There are 4 variables that make up the confusion matrix. True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are the four possible outcomes. The format of the confusion matrix is displayed in Table 3.

Table 3 Confusion matrix

	Predictive Positive	**Predictive Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

In order to perform an accurate evaluation of the machine learning classifiers, important measures were collected from the confusion matrix. In addition to the correct classification rate or accuracy, other metrics such as True Positive Rate (TPR), False Positive Rate (FPR), Precision, Recall, and F1 score were used to evaluate the machine learning classifiers. These metrics were used to evaluate the classifiers' performance as shown in Table 4.

Table 4 The metric used in evaluating the performance of machine and deep learning classifiers

Performance Measure Name	**Formula**
Correct Classification Rate	$CCR = \frac{TP + TN}{TP + FP + FN + TN}$
Precision	$PPV = \frac{TP}{TP + FP}$
Recall	$= \frac{TP}{TP + FN}$
F1-score	$F_1 = \frac{2TP}{2TP + FP + FN}$
True Positive Rate	$TPR = \frac{TP}{TP + FN}$
False Positive Rate	$FPR = \frac{FP}{TN + FP}$
Specificity	$TPR = \frac{TN}{TN + FP}$
Negative Predictive Value	$NPV = \frac{TN}{TN + FN}$

5.5. Insights from Feature Importance

One of the advantages of using Logistic Regression is the ability to analyze the coefficients of the model, which correspond to the importance of different features (words) in the classification task. By examining the coefficients, we can gain insights into which words are most indicative of spam or ham. For instance, words like "free," "urgent," and "win" were associated with higher probabilities of being classified as spam, while words related to business communication, such as "meeting," "report," and "project," were more likely to be classified as ham. This analysis not only helps in understanding the model's behavior but also provides valuable insights for refining the model and improving its accuracy.

6. User Interface (UI) Application

To enhance the accessibility of the spam filter, a user interface was developed using the Tkinter library. The UI is designed to be simple and intuitive, allowing users to classify new emails with ease.

6.1. Technical Implementation

The UI features a text entry area where users can input the email text and a "Classify Email" button that triggers the classification process. When the button is clicked, the input text is preprocessed using the same pipeline that was applied to the training data. The preprocessed text is then fed into the trained Logistic Regression model, which outputs a probability score indicating whether the email is spam or ham. The result is displayed in a message box, informing the user of the classification outcome. The UI also provides a clear and concise explanation of the classification, including the probability score, which helps users understand the confidence of the model's prediction.

6.2. Challenges and Solutions

One of the challenges in developing the UI was ensuring that the classification process was fast and responsive, even for long emails. To address this, we optimized the preprocessing and classification steps by caching the TF-IDF vectorizer and model in memory, reducing the time required to process each input. Another challenge was handling edge cases, such as emails with very short or very long text. We implemented error handling mechanisms to provide feedback to the user in case of invalid input, ensuring a smooth and user-friendly experience.

7. Discussion

While the Logistic Regression model demonstrated strong performance, there are several limitations to consider:

7.1. Evolving Tactics Challenge

Spam tactics evolve over time, which may reduce the effectiveness of the spam filter. Spammers often use techniques such as obfuscation, where words are deliberately misspelled or altered to bypass filters. They may also use social engineering tactics to craft emails that closely resemble legitimate communication. To address these challenges, regular updates to the model and retraining on new data are necessary. The model must be continuously monitored and fine-tuned to adapt to new patterns and tactics used by spammers.

7.2. Pattern Dependency Risks

The model's reliance on patterns in the data may lead to misclassification of legitimate emails that deviate from typical patterns. For example, a legitimate email containing words commonly associated with spam, such as "free" or "urgent," may be incorrectly classified as spam. This risk underscores the importance of feature selection and engineering, as well as the need for a diverse and representative training dataset.

7.3. Complex Attack Handling

The filter may struggle with sophisticated attacks, such as those involving social engineering. Social engineering attacks exploit human psychology to trick recipients into performing actions or divulging information. These attacks are often difficult to detect because they may not contain typical spam indicators, such as certain keywords or phrases. To enhance the model's ability to detect complex attacks, additional features, such as email metadata, sender reputation, and behavioral patterns, could be incorporated into the model. These features could provide more context and help the model make more informed decisions.

8. Conclusion

This paper presents the development of a robust and efficient spam filter for email applications, leveraging advanced machine learning and deep learning techniques. Among the models evaluated, Support Vector Machine (SVM) emerged as the best-performing classical model, achieving an accuracy of **99.0%** and an F1-score of **0.97**, outperforming Logistic Regression, Random Forest, and k-Nearest Neighbors. To enhance usability, a user-friendly interface was developed, enabling real-time categorization of newly received emails. Despite these advancements, the dynamic and evolving nature of spam strategies, including increasingly sophisticated attacks, underscores the need for continuous research and model refinement.

In addition to classical models, modern transformer-based architectures were also explored. Among these, BERT demonstrated outstanding performance, achieving an accuracy of **98.8%** and an F1-score of **0.97**, highlighting its ability to capture detailed contextual information. Other models, such as DistilBERT, RoBERTa, and XLNet, also delivered competitive results, reinforcing the potential of transformers for spam detection. These models provide a robust foundation for future work, particularly in applications requiring semantic understanding and adaptability to evolving spam tactics.

Feature importance analysis provided valuable insights into the classification process, revealing key terms and patterns strongly associated with spam or legitimate emails. This interpretability facilitates targeted optimizations, enhancing both model accuracy and practicality. By integrating classical machine learning with state-of-the-art deep learning approaches, this study underscores the potential for scalable, adaptive, and highly accurate spam detection systems. These findings lay the groundwork for advancing email security while addressing the challenges posed by evolving malicious communication strategies.

9. Future Work

- Fine-tune the model parameters for further optimization.
- Explore additional feature engineering techniques to enhance model performance.
- Investigate real-time implementation for dynamic adaptation to evolving spam patterns.
- The email filter can be customized to meet individual or business-specific requirements, offering tailored solutions for varied needs.

References

1. Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K. and Alazab, M., 2019. A comprehensive survey for intelligent spam email detection. Ieee Access, 7, pp.168261-168295.
2. Yaseen, Q., 2021. Spam email detection using deep learning techniques. Procedia Computer Science, 184, pp.853-858.
3. Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E. and Alegre, E., 2023. A review of spam email detection: analysis of spammer strategies and the dataset shift problem. Artificial Intelligence Review, 56(2), pp.1145-1173.

4. Raihen, M.N. and Akter, S., 2024. Comparative Assessment of Several Effective Machine Learning Classification Methods for Maternal Health Risk. Computational Journal of Mathematical and Statistical Sciences, 3(1), pp.161-176.
5. Raihen, M.N. and Akter, S., 2024. Sentiment analysis of passenger feedback on US airlines using machine learning classification methods. World Journal of Advanced Research and Reviews, 23(1), pp.2260-2273.
6. Raihen, M. N., \& Akter, S. (2023). Forecasting Breast Cancer: A Study of Classifying Patients’ Post-Surgical Survival Rates with Breast Cancer. Journal of Mathematics and Statistics Studies, 4(2), 70-78.
7. Nasreen, G., Khan, M.M., Younus, M., Zafar, B. and Hanif, M.K., 2024. Email spam detection by deep learning models using novel feature selection technique and BERT. Egyptian Informatics Journal, 26, p.100473.
8. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J. and Data, M., 2005, June. Practical machine learning tools and techniques. In Data mining (Vol. 2, No. 4, pp. 403-413). Amsterdam, The Netherlands: Elsevier.
9. Cranor, L.F. and LaMacchia, B.A., 1998. Spam!. Communications of the ACM, 41(8), pp.74-83.
10. Raihen, M.N. and Akter, S., 2024. Prediction modeling using deep learning for the classification of grape-type dried fruits. International Journal of Mathematics and Computer in Engineering.
11. Guo, Y., Mustafaoglu, Z. and Koundal, D., 2023. Spam detection using bidirectional transformers and machine learning classifier algorithms. journal of Computational and Cognitive Engineering, 2(1), pp.5-9.
12. Tida, V.S. and Hsu, S., 2022. Universal spam detection using transfer learning of BERT model. arXiv preprint arXiv:2202.03480.
13. Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., Qian, B., Wen, Z., Shah, T., Morgan, G. and Ranjan, R., 2023. Explainable AI (XAI): Core ideas, techniques, and solutions. ACM Computing Surveys, 55(9), pp.1-33.
14. Alkhalili, M., Qutqut, M.H. and Almasalha, F., 2021. Investigation of applying machine learning for watch-list filtering in anti-money laundering. iEEE Access, 9, pp.18481-18496.
15. Raihen, M.N., Begum, S., Akter, S. and Sardar, M.N., 2025. Leveraging Data Mining for Inference and Prediction in Lung Cancer Research. Computational Journal of Mathematical and Statistical Sciences, 4(1), pp.139-161.
16. Ewees, A.A., Gaheen, M.A., Alshahrani, M.M., Anter, A.M. and Ismail, F.H., 2024. Improved machine learning technique for feature reduction and its application in spam email detection. Journal of Intelligent Information Systems, pp.1-23.
17. Krishnamoorthy, P., Sathiyanarayanan, M. and Proença, H.P., 2024. A novel and secured email classification and emotion detection using hybrid deep neural network. International Journal of Cognitive Computing in Engineering, 5, pp.44-57.
18. Ewees, A.A., Gaheen, M.A., Alshahrani, M.M., Anter, A.M. and Ismail, F.H., 2024. Improved machine learning technique for feature reduction and its application in spam email detection. Journal of Intelligent Information Systems, pp.1-23.
19. Batley, S., 2014. Classification in theory and practice. Chandos Publishing.
20. Ramachandran, K.M. and Tsokos, C.P., 2020. Mathematical statistics with applications in R. Academic Press.
21. Müller, A.C. and Guido, S., 2016. Introduction to machine learning with Python: a guide for data scientists. " O'Reilly Media, Inc.".