A Machine Learning Framework for Automated Spam E-mail Classification

Pranav Patil[1], Sonu H.T.[2], Prajwal S.[3], Sachin B.[4]

Department of Computer Science and Engineering- Data Science, Dayananda Sagar Academy of Technology & Management, Bangalore, India.

E-mail: [1]1dt23cd034@dsatm.edu.in, [2]1dt23cd051@dsatm.edu.in, [3]1dt23cd033@dsatm.edu.in, [4]1dt23cd043@dsatm.edu.in

Abstract

Spam mails are some of the critical issues in cybersecurity. In this regard, they have adverse effects on the use of resources and productivity of users. This study suggests an artificial intelligence technique in classifying spam mail by utilizing machine learning and deep learning methods. Such include decision tree, naive bayes, support vector machine, and deep learning known as LSTM. Performance analysis of the model will be dependent on aspects such as accuracy, precision, recall, and F1 score among others. Based on the results in this study, the deep learning techniques such as LSTM provide high accuracy rates of about 98.5%. Similarly, the integration of the Convolutional Neural Network and LSTM offers high accuracy of about 99%. However, Naive Bayes shows low accuracy in the short time frame of training.

Keywords: Spam Detection, Email Classification, Decision Tree, SVM, Naive Bayes, LSTM, Machine Learning, Deep Learning, Keyword Filtering.

1. Introduction

The fast advancement in digital communication has led to a significant rise in the number of uses of emails in the exchange of information. Unfortunately, the rise in the usage of emails has been accompanied by an equal rise in spamming emails. These spamming emails pose significant threats to security in addition to causing wastage of network resources. Examples of these threats include phishing attacks, malware dissemination, financial frauds among others. According to estimates, more than half of the global email exchanges are spam. From this perspective, there is a need for efficient methods for detecting and preventing spams.

DOI: https://doi.org/10.36548/rrrj.2026.1.001
Received: 18.02.2026, received in revised form: 21.03.2026, accepted: 08.04.2026, published: 21.04.2026

Traditionally, rules and keywords were used in the detection of spams. While these methods are quite simple and easy to develop, they lack flexibility and generalization. In recent times, with the development of intelligent methods based on big data analytics, machine learning techniques such as Naïve Bayes, Decision Trees, and Support Vector Machines have been used in detecting spams. They utilize statistical features of text documents such as term frequency and TF-IDF in spam detection. Such techniques are efficient in discriminating spams from legitimate emails; however, they cannot detect context information.

Furthermore, advances made in the domain of deep learning techniques are also beneficial in enhancing the effectiveness of techniques used in spam detection. LSTM-based techniques are known to have the ability to learn dependencies within the sequence of data, which is quite useful in identifying dependencies within textual data. Even though such techniques are extremely successful in identifying various patterns within the data, they are also quite computationally expensive.

Given the advantages and disadvantages of both kinds of techniques, a requirement has been identified wherein both of these approaches can be combined to create a framework that will allow us to adopt a more balanced approach to spam email detection. In light of this requirement, the current paper aims at creating a unified framework for spam email classification through the use of classical classification techniques like Decision Tree, Naive Bayes, and SVM along with the use of LSTM-based deep learning techniques.

2. Literature Review

Email spam classification is a problem that has been extensively explored using various machine learning and deep learning models. Initially, supervised learning algorithms were used for this purpose. In this context, Naive Bayes, Decision Trees, and SVM were used to classify spam and non-spam emails using text features. These models were found to perform well for this problem because of their simplicity and ease of implementation [2], [3]. Moreover, they were able to perform well with high-dimensional text features. However, they were limited to certain assumptions that could not capture relationships between features.

Recently, various machine learning frameworks were explored for this problem. These frameworks were able to improve the accuracy of the classification model. Moreover, feature engineering techniques were used with machine learning models to improve the accuracy of the model. These frameworks were able to address various limitations that can affect the

performance of a spam detection model [4]. These limitations include data imbalance, feature sparsity, and changing nature of spams. In addition to this, various NLP techniques have been used with machine learning models to improve feature representation. This is because NLP can help in better understanding of text semantics [5].

However, recent research has also focused on the application of deep learning models, especially the use of recurrent neural networks like Long Short-Term Memory (LSTM), to capture sequential dependencies in the context of emails. The results have also been promising in terms of better accuracy in comparison to the earlier models, as these models are capable of learning the patterns in the text data with greater accuracy [6].

In addition to that, the development of transformer models and multimodal models is also considered while studying spam emails for increasing accuracy in detecting the same, highlighting how advanced this area has become in the realm of spam classifiers [1]. Nevertheless, some challenges exist concerning the application of such models, including the expense incurred by computations, the requirement of massive labeled data, and the difficulty in understanding the results derived from deep learning models.

As stated in the comparative study performed in the latest research paper, despite the superior results achieved via deep learning models in terms of precision, the significance of the conventional models based on their efficiency cannot be overlooked [7].

3. Proposed Work

3.1 Overview of the Proposed Framework

The proposed approach seeks to develop a system which would bring together machine learning and deep learning for purposes of automating the process of spam email classification. The purpose of the framework in the proposed system is to bring together the benefits of classical machine learning methods and deep learning methods within one framework. This should help improve the overall performance of the process of email classification [3]. The framework involved in the proposed system comprises of four main steps, namely; data preprocessing, feature extraction, model training, and classification. To begin with, it will be necessary to preprocess the initial email data set after which it will be necessary to extract important features from the emails. After the feature extraction process, the next step would

involve using these features to train the model before finally carrying out the classification exercise to determine spam emails and ham emails.

The pipeline shown in Figure 1 showcases the data preprocessing and model training phases, emphasizing the logical progression from input data to classification output.

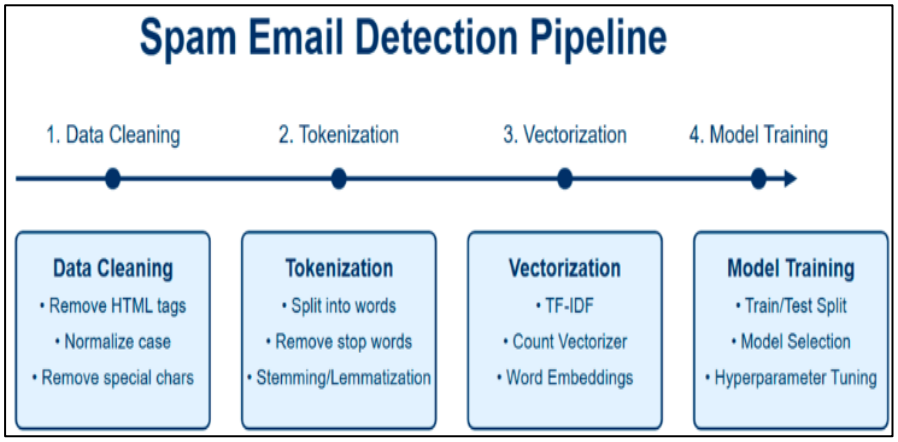

Figure 1. Spam Email Detection Pipeline Showing Preprocessing and Model Training Stages

3.2 Data Preprocessing and Feature Engineering

Data Preprocessing plays a crucial role in the suggested system. It is necessary for the email text to be preprocessed before feeding it to the machine learning model to maintain high-quality data. First, the text should undergo cleaning by eliminating any unnecessary elements such as HTML tags, special characters, and noise. Next, the text should be tokenized, which means dividing the text into tokens. Stop-word removal and text normalization (stemming/lemmatization) are the following preprocessing techniques.

The proposed framework uses two different techniques to perform the feature extraction:

- TF-IDF (Term Frequency-Inverse Document Frequency): This technique can be used for classical models like Decision Trees, Naive Bayes, SVM to transform text into numerical form based on the importance of words.
- Word Embedding: This technique can be used for LSTM models to understand the meaning in the text.

3.3 Decision Tree-Based Classification

The Decision Tree model is used as a classification model that is based on decision-making using rules. This model works on the principle of recursively partitioning the feature space using the best features. These features are usually selected using information gain and Gini index. In this model, features such as keyword detection, word frequency, and metadata are used for training [4]. This model works on splitting the feature space using thresholds until the final class labels are obtained. In this model, it is observed that the decision-making process is based on evaluating various spam indicators using a flow diagram (Figure 2). In this model, pruning is done to avoid overtraining.

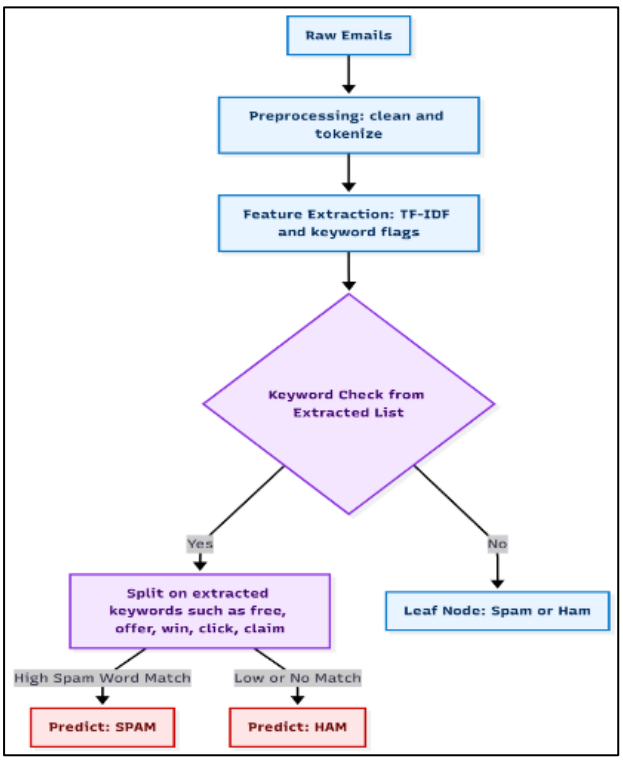

Figure 2. Flow Diagram of Decision Tree Model Process

3.4 Naive Bayes Classification

The Naive Bayes classifier is used in a probabilistic manner by employing Bayes' theorem. Despite this assumption, it works well for high-dimensional text data. It calculates the posterior probability of a given email being a spam given a set of features. The class with the highest posterior probability is chosen for the final output. There are two types of naive Bayes algorithms: Bernoulli naive Bayes for binary features and multinomial naive Bayes for word frequencies, both of which are applicable in this context. As depicted in Figure 3, this

classifier works on a set of features by calculating the probability of a spam email. Naive Bayes is a good classifier due to its efficiency in terms of computational cost and training time.

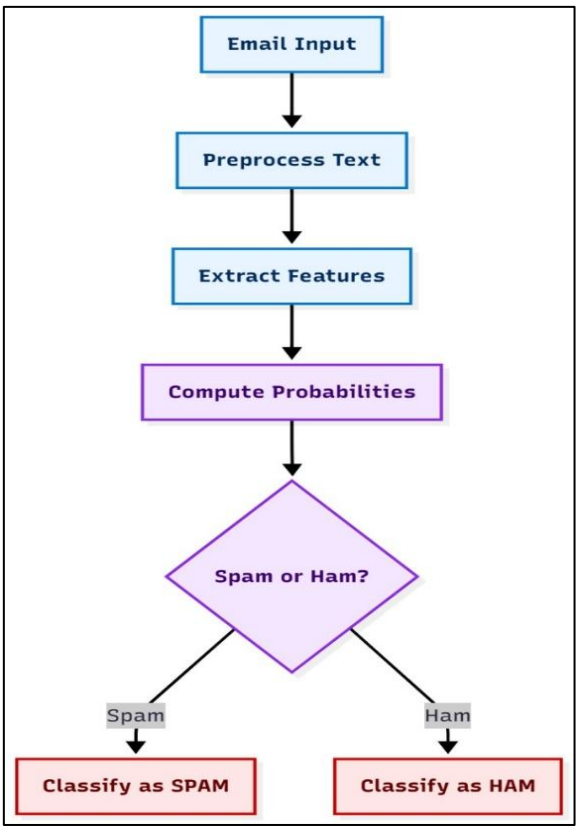

Figure 3. Flow Diagram of Naive Bayes Model Process

3.5 LSTM-Based Deep Learning Model

The Long Short-Term Memory (LSTM) model is used to incorporate sequential dependencies and context relationships within the text of the email. Unlike other models, this model treats input information as a sequence of tokens and also maintains long-term dependencies using gated memory cells. In this model, the text of the email is mapped to word embeddings and is used as input to the LSTM model [10]. This model is able to learn various patterns such as word order and context phrases like “click here to claim.” These are usually indicative of spam emails. In this model, as depicted in Figure 4, the LSTM model is able to capture context information and is therefore able to achieve better accuracy.

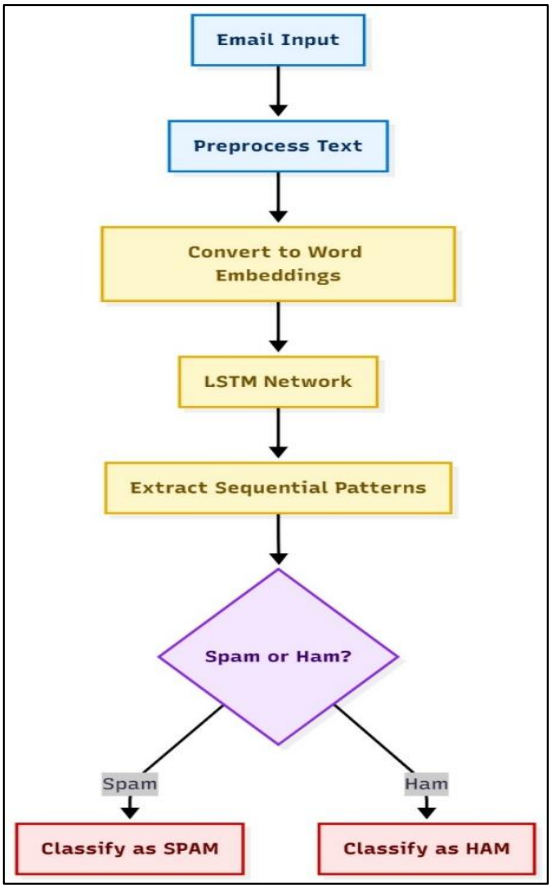

Figure 4. Flow Diagram of LSTM Model Process

3.6 Support Vector Machine (SVM) Classification

The Support Vector Machine (SVM) is utilized as a strong supervised classifier using the optimal hyperplane concept to effectively separate the spam and ham classes in the high-dimensional space. The SVM classifier uses TF-IDF vector inputs and tries to achieve the maximum margin between the two classes. SVM uses both linear and nonlinear kernels, such as the Radial Basis Function (RBF), to effectively tackle the data distribution complexity [5]. As shown in Figure 5, the SVM classifier processes the input features to generate the output based on the learned decision boundary. SVM is considered a powerful classifier in handling the sparsity of the data, although it is computationally expensive for handling large datasets.

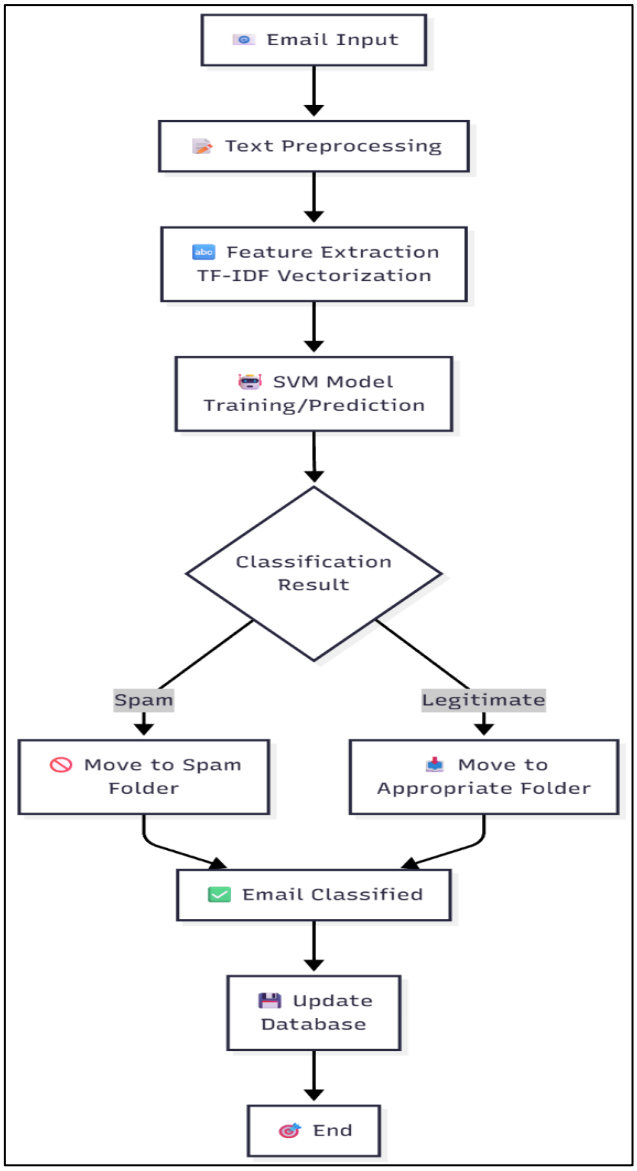

Figure 5. Flow Diagram of SVM Model Process

3.7 Integrated Classification Strategy

The objective is to develop a model that will integrate all the four models into one pipeline (Figure 6). While the Naive Bayes model works well in terms of computation, Decision Trees are great for interpretation, SVM is excellent for creating strong classification boundaries, and LSTM is great in identifying context relationships. All of these models can be evaluated separately and can be integrated through an ensemble method [6].

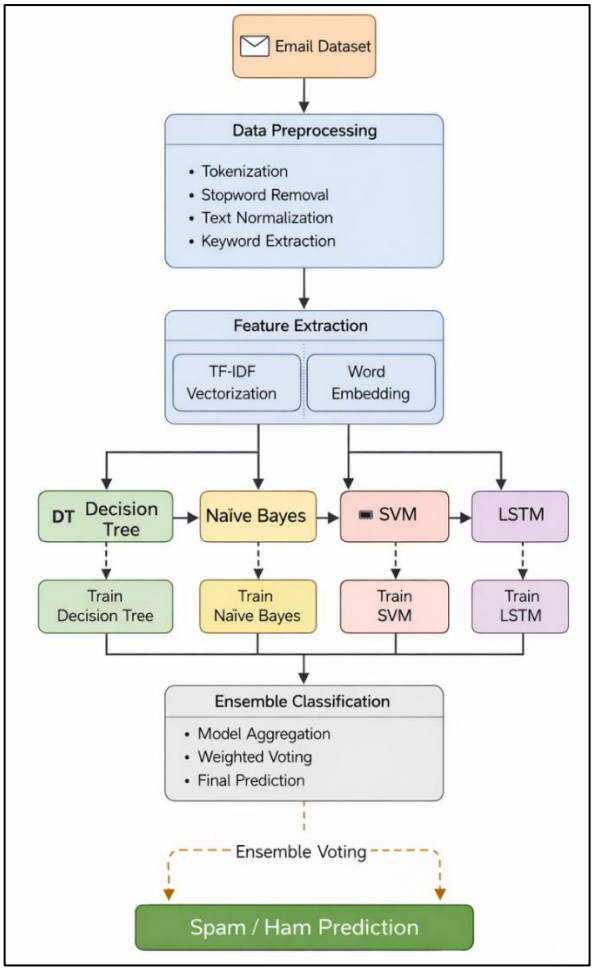

Figure 6. Integrated Classification Architecture

4. Results

The performance of the suggested framework for spam email classification is evaluated using various machine learning and deep learning models. The results obtained using various models are compared and are presented in Table 1. It can be concluded that the performance of each model is better in terms of accuracy, precision, recall, and F1-score. Among various models used for evaluation, Naive Bayes results in an accuracy of 87%, with an F1-score of 0.86. This is because Naive Bayes is a fast classifier that assumes features are independent. Decision Tree results in better performance in terms of accuracy and F1-score. In this model, accuracy is 94%, with an F1-score of 0.93. This is because this model is based on decision theory and can handle non-linear decision boundaries. In addition, the SVM model results in better performance in terms of accuracy and F1-score. In this model, accuracy is 92%, with an F1-score of 0.92. This is because this model has better generalization performance in a high-dimensional feature space.

Table 1. Comparative Performance Metrics of Common Spam Classification Algorithms

Model	Accuracy (%)	Precision	Recall	F1 Score
Naive Bayes	87.00	0.85	0.88	0.86
Decision Tree	94.00	0.93	0.94	0.93
SVM	92.00	0.91	0.93	0.92
LSTM	98.50	0.98	0.98	0.98
CNN–LSTM Hybrid	99.00	0.99	0.99	0.99

On the other hand, deep learning approaches show a high level of superiority compared to traditional approaches. In this case, the LSTM model shows a high level of accuracy of 98.5%, and the F1 score is 0.98. Moreover, the CNN-LSTM hybrid approach shows the highest performance in terms of accuracy and F1 score at 99% and 0.99, respectively. From this case, it is evident that combining these approaches is more beneficial than using a single approach. Figure 7 shows that the comparative accuracy of using deep learning approaches is higher than that of using traditional approaches.

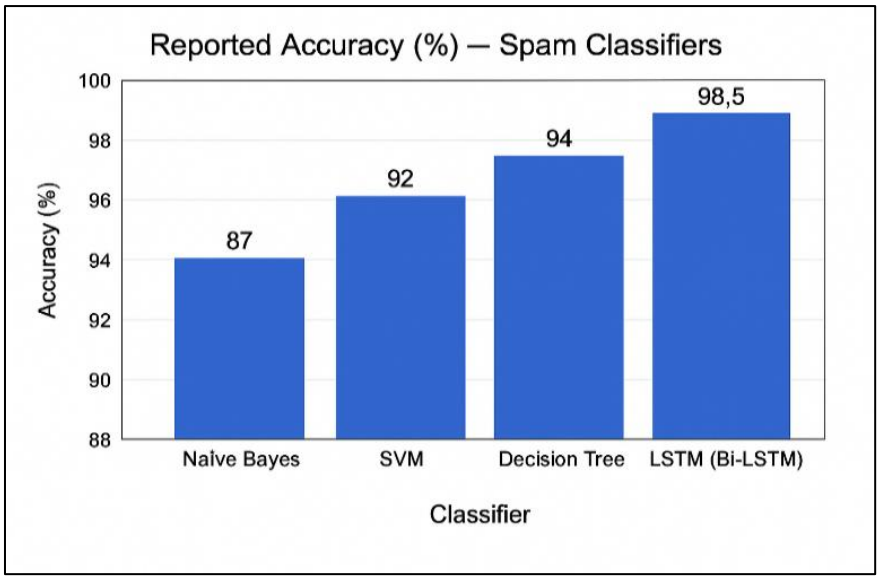

Figure 7. Comparative Accuracies of Spam Classifiers

The differences in the performances of the models can be attributed to the inherent characteristics of the models used in the experiments. Machine learning algorithms such as Naive Bayes, Decision Tree, and SVM are primarily based on the statistical representation of

the words, i.e., TF-IDF, which only considers the importance of words but not the relationship between words in the context of the sentence. However, the LSTM-based models take the text as a sequence and learn the relationship between words, making it possible to identify the complex patterns of spam messages such as the structure of the sentence and the semantics used in the message. The CNN-LSTM hybrid model also improves the performance by combining the local and sequential features.

The applicability of the proposed framework is further shown through the development of the web-based interface for the classification of spam emails. As shown in the dashboard in Figure 8, the applicability of the proposed system is further shown through the development of the web-based interface for the classification of spam emails. In the same way, the applicability of the proposed system is further shown through the development of the spam classification interface as shown in Figure 9.

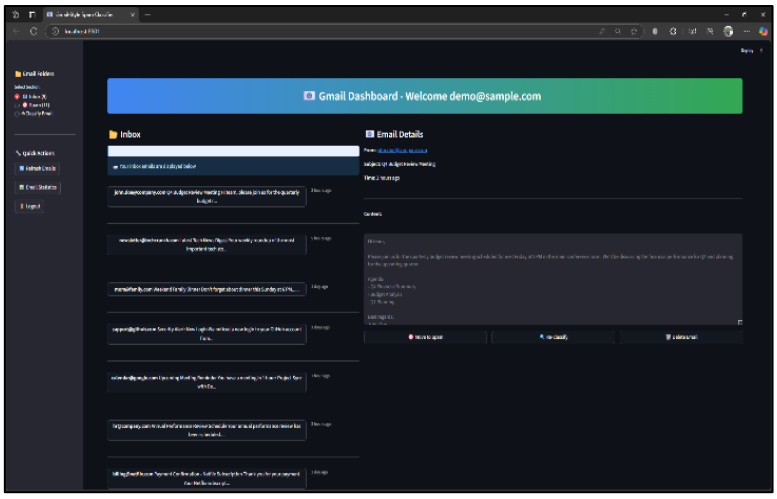

Figure 8. Spam Email Inbox Interface

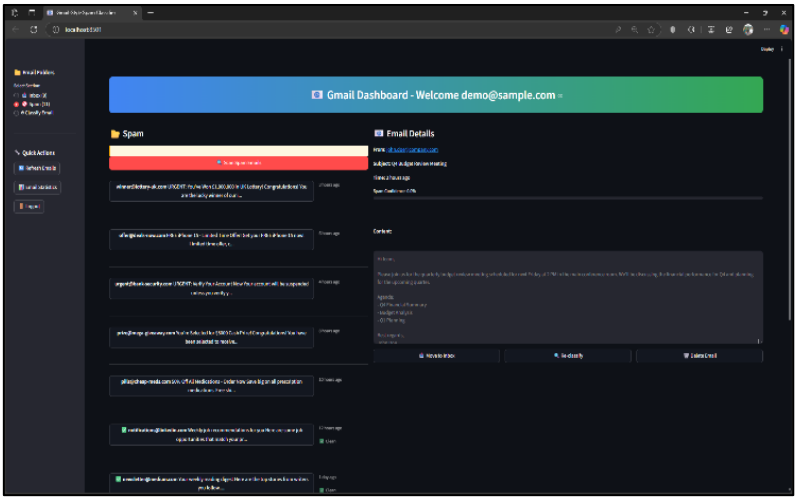

Figure 9. Email Spam Interface

Despite the fact that deep learning models achieve better results, it is essential to understand that there is a trade-off between accuracy and complexity. In this regard, Naive Bayes has the least complexity in terms of training and testing. Decision Trees also have an advantage in terms of interpretability. SVM is another model that strikes a balance between accuracy and robustness. On the other hand, LSTM and CNN-LSTM models have high complexity in terms of training time. In this regard, it is essential to understand that these models might not be suitable for real-time systems without adequate infrastructure.

From the above results, it is clear that the proposed framework is quite effective in its use for classifying data. It is important to note that using more than one model in the same framework has its advantages, especially where the individual strengths of each model, including efficiency, understandability, and contextual learning, can be leveraged in the process.

5. Conclusion

The suggested research proposes a common framework for the classification of spam emails using a combination of different machine learning techniques such as Decision Tree, Naïve Bayes, SVM and LSTM. The main novelty of the research is that it uses an appropriate combination of several methods within one framework, thus facilitating a comparative analysis of the performance of the different machine learning models. According to the results of the experiment, deep learning techniques outperformed the traditional machine learning techniques in terms of accuracy due to their ability to capture contextual data from the emails. Nonetheless, the traditional techniques also had certain advantages in terms of efficiency. The hybrid model used in the proposed research managed to find a compromise between the efficiency and accuracy of the approaches, which was achieved by incorporating the strengths of both traditional machine learning techniques and deep learning models. Thus, the experimental results confirmed the high efficiency of the hybrid models in the task of classifying spam emails. The latter demonstrated significantly higher results when compared to the performance of the other approaches in this task.

References

[1] Asliyuksek, Halim, Ozgur Tonkal, and Ramazan Kocaoglu. "A Comparative Evaluation of a Multimodal Approach for Spam Email Classification Using DistilBERT and Structural Features." Electronics 14, no. 19 (2025): 3855.

[2] Lakshmi, R. Deepa, and N. Radha. "Spam Classification Using Supervised Learning Techniques." In Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India, 2010, 1-4.

[3] Awad, Wael Abou, and S. M. ELseuofi. "Machine Learning Methods for Spam E-Mail Classification." International Journal of Computer Science & Information Technology (IJCSIT) 3, no. 1 (2011): 173-184.

[4] Mallampati, Deepika, and Nagaratna P. Hegde. "A Machine Learning Based Email Spam Classification Framework Model: Related Challenges and Issues." International Journal of Innovative Technology and Exploring Engineering 9, no. 4 (2020): 3137-3144.

[5] Junnarkar, Akash, Siddhant Adhikari, Jainam Fagania, Priya Chimurkar, and Deepak Karia. "E-mail Spam Classification via Machine Learning and Natural Language Processing." In 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 693-699.

[6] Siddique, Zeeshan Bin, Mudassar Ali Khan, Ikram Ud Din, Ahmad Almogren, Irfan Mohiuddin, and Shah Nazir. "Machine Learning‐Based Detection of Spam Emails." Scientific Programming 2021, no. 1 (2021): 6508784.

[7] Dada, Emmanuel Gbenga, Joseph Stephen Bassi, Haruna Chiroma, Shafi'I. Muhammad Abdulhamid, Adebayo Olusola Adetunmbi, and Opeyemi Emmanuel Ajibuwa. "Machine Learning for Email Spam Filtering: Review, Approaches and Open Research Problems." 2019, Heliyon 5, no. 6.

[8] Rayan, Alanazi. "Analysis of E‐Mail Spam Detection Using a Novel Machine Learning‐Based Hybrid Bagging Technique." Computational Intelligence and Neuroscience 2022, no. 1 (2022): 2500772.

[9] Mansoor, R. A. Z. A., Nathali Dilshani Jayasinghe, and Muhana Magboul Ali Muslam. "A Comprehensive Review on Email Spam Classification Using Machine Learning

Algorithms." In 2021 International conference on information networking (ICOIN), 327-332.

[10] Srinivasan, Sriram, Vinayakumar Ravi, Mamoun Alazab, Simran Ketha, Ala'M. Al-Zoubi, and Soman Kotti Padannayil. "Spam Emails Detection Based on Distributed Word Embedding with Deep Learning." In Machine intelligence and big data analytics for cybersecurity applications, Cham: Springer International Publishing, 2020, 161-189.