Original Article https://doi.org/10.54692/lgurjcsit.v10i1.191

Efficient Feature Engineering and Classification for Large-Scale Email Spam Detection

Moavia Hassan*, Muhammad Ilyas, Qamas Gul Khan Safi, Muhammad Munwar Iqbal
Farooq Ali
University of Engineering and Technology, Taxila, Pakistan.

ABSTRACT

Email spam comprises 45% of global email traffic.It poses significant challenges, including resource wastage and security risks, such as Phishing and Malware Distribution. The implementation of modern machine learning algorithms has demonstrated their effectiveness in identifying and filtering spam email messages. Existing research has employed various techniques for classifying spam email datasets. The Proposed framework addresses spam detection at a large scale using supervised machine learning techniques. Exploratory Data Analysis was performed to understand data characteristics and identify patterns relevant to spam classification. Data preprocessing techniques were applied to handle missing values, normalize features, and remove noise. Feature engineering was then carried out to generate meaningful attributes that improve model performance. Then, machine learning algorithms were trained and tested for classification evaluation. The Random Forest classifier achieved the best performance, with an accuracy of 97.86% and an ROC-AUC score of 0.97. The findings highlight the importance of feature engineering and Random Forest methods in spam detection tasks. This work provides practical insights for building accurate and scalable spam filters. It also opens up future research directions involving advanced feature engineering and deep learning approaches to further enhance classification performance.

Keywords: Spam Detection, Machine Learning, Logistic Regression, SpamBase Dataset, Random Forest.

1. INTRODUCTION

Email is a common and official communication tool for personal and professional use, but it faces a growing challenge from spam and often malicious messages. The study reveals that spam emails constitute approximately 45.6% of all email traffic worldwide [1], resulting in wasted resources, security threats such as phishing, and economic losses valued at almost $20 billion every year for American businesses and consumers [2]. This underlines the urgent need for effective spam detection to protect users and organizations. Traditional rule-based filtering methods have been widely used, but they struggle to adapt to evolving spam behaviors. These methods rely heavily on hard-coded rules, which makes them less effective against ever-changing threats [3]. On the other hand, machine learning (ML) methods provide an effective solution in learning patterns within email content to identify spam or legitimate messages (”ham”) and adapt accordingly with increased accuracy [4]. Comparative studies in data mining and classification have demonstrated that machine learning models can effectively discover discriminative patterns from large datasets and significantly improve predictive performance across diverse application domains [5]. However, most previous research has used relatively small or old data sets, such as SpamBase (4,600 emails), and has usually only tested a small set of classifiers. These limitations make it challenging to generalize their findings and apply them on a larger scale. To fill this void, the current work tests ten popular machine learning models on a massive corpus of 193,000 emails, providing a more thorough empirical comparison than that of previous work.

Traditional Anti-Money Laundering (AML) systems primarily rely on rule-based mechanisms and manual monitoring. While these systems have been effective in the past, they are increasingly inadequate in handling the scale and complexity of modern financial transactions. The rise of digital banking, cross-border transactions, and cryptocurrencies has introduced new challenges, making it difficult for conventional systems to detect evolving money laundering patterns [3], [4]. Consequently, financial institutions are under increasing pressure to adopt more intelligent, adaptive, and scalable solutions.

Our research includes traditional classifiers and ensemble methods to enhance email classification accuracy. With a large dataset, our observations aim to examine model performance and generalizability to identify more effective filtering solutions. We used a publicly available dataset on Kaggle named

Received: 15 March 2026; Revised: 01 May 2026; Accepted: 25 May 2026
*Corresponding Author: Moavia Hassan <moaviahassan112@gmail.com>
© 2026 by the Author(s). Published by LGURJCSIT, under the CC-BY license

Spam Emails Data, which has 193k records. Exploratory data analysis (EDA) constitutes the inaugural analytical procedure undertaken on the dataset. With the help of EDA, the data undergoes a thorough analysis that aims to reveal hidden patterns and understand the nature of the data. Similar findings have been reported in data warehousing and decision-support research, where exploratory analysis and data characterization significantly improved the quality of subsequent predictive modelling tasks [6]. The process provides a thorough knowledge of the distribution of email categories, spam, and non-spam [1]. Additionally, it identifies anomalies to define possible problematic elements within the dataset. Taken together, these exercises form a basic outline of future data-handling methods. Preprocessing and feature engineering are then based on the outcomes of EDA. Preprocessing protocols solve data inconsistency issues through a powerful cleansing of the data and normalization of textual data to make it consistent. Feature engineering, in turn, identifies promising features in the emails, thereby enhancing the quality of the data and preparing the dataset for model training. Collectively, these processes provide quality data needed to carry out a rigorous analysis. Our evaluated predictive models consist of Random Forests, AdaBoost, Bagging, Logistic Regression, Gradient Boosting, Multinomial Naive Bayes, XGBoost, Decision Trees, and K-Nearest Neighbors (KNN). Random Forest stands out as the best ensemble model, achieving 97.86% accuracy and a 1.00 ROC-AUC score, according to the evaluation metrics. The success rate of KNN reached 97.04%, showing that feature engineering techniques were effective.

Our research approach employs organized objectives to solve the issue of enhancing spam detection systems. The analysis utilizes a dataset of 193,000 records to understand the distinctions between spam and non-spam (ham) emails through exploratory data analysis (EDA). Our decision-making process for feature engineering and model selection will rely on the gathered analysis. Our research employs natural language processing (NLP) techniques that combine tokenization with stop word removal and stemming, and implement TF-IDF weighting for converting email texts into numerical features suitable for machine learning. Our study investigates various machine learning models to optimize their performance. Our study examines Linear Support Vector Classification (SVC) and Logistic Regression, along with K- Nearest Neighbors (KNN), Multinomial Naive Bayes, Decision Trees, Random Forests, AdaBoost, Bagging, Gradient Boosting, and XGBoost models [2]. The process of model development incorporates cross-validation methods as they help develop robust training that prevents overfitting. The next step involves using precision alongside precision, recall, and the F1- score and ROC-AUC to assess the performance of these models. The model performance assessment tools provide direct comparison capabilities for classifying tasks that demonstrate model effectiveness and solution reliability. Real-world spam filtering systems received analysis from the results to identify both the strengths and limitations of each model, which establishes a precise measurement-based strategy for machine learning development in email safety.

The motivation behind this research is to address the significant challenges posed by email spam. Spam remains alarmingly prevalent, introducing severe security and economic challenges. This prevalence of spam is more than just an inconvenience; it introduces severe problems, particularly given email's pivotal role in communication and information sharing. Email remains a fundamental tool for both personal and professional correspondence; however, the infiltration of spam presents significant security risks. These risks include the propagation
of phishing scams, which deceive users into divulging sensitive information, and the distribution of malware, which can compromise systems and data. These threats collectively contribute to substantial financial losses for individuals and organizations. The main contribution of this paper can be summarized as follows:

• We develop a framework for the detection of an email dataset comprising 193,000 emails from Kaggle, which is significantly larger than traditional datasets such as SpamBase (4,601 emails), and helps to improve model robustness and generalizability. It is one of the most extensive empirical studies on spam detection with classical and ensemble ML models.

• We achieved the results of Random Forest, which provided 97.86 % accuracy and an ROC-AUC of 1.00, which shows the comparative advantages of ensemble models. This work provides a comprehensive comparison of ten popular machine learning methods, offering broader insights than previous small-scale studies.

• We introduce a scalable pipeline of large-scale text processing, which includes tokenization, removal of stop-words, stemming and TF-IDF, demonstrating its applicability for real-world large-scale spam filtering.

2. LITERATURE REVIEW

The methods used in spam detection have been developed extensively since the time when the first systems relied on basic rules. The current spam detection system operates through advanced machine learning, along with deep learning, to deliver better spam identification results. Spam protection programs developed by researchers started with basic rules that used keyword searches combined with sender blocking and message header controls as their first anti-spam measures [7]. These early methods against spammers became useless because spammers adopted obfuscation techniques, thus forcing administrators to maintain constant system updates for effectiveness [8]. Better solutions emerged from the identified limitations to develop machine learning methods that integrated into spam filtering systems.

Machine learning has made it possible to use models to learn what content and metadata in an email make it spam, without humans having to intervene constantly as the patterns of spam change. This includes many supervised learning methods such as Naive Bayes, Support Vector Machines (SVMs), Decision trees, and their ensemble version, such as Random Forests. However, because of its simplicity and efficiency, Naive Bayes is known to have high accuracy on spam classification, especially when using feature selection [9]. In practice, SVMs have shown their effectiveness in handling high-dimensional text data and are usually more competitive than other classifiers in comparative studies [10]. Random Forests offer robustness via ensemble learning, while at the same time being interpretable, as evidenced by research showing it is capable of high accuracy and feature importance insights [11].

Other alternative techniques, such as Logistic Regression and K Nearest Neighbors (KNN) and AdaBoost, have also been explored in spam detection tasks and have been successful at it to different degrees [10]. Text-based features such as bag of words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and n-grams [3] remain among the most widely used representations in text classification. The importance of semantic representation and knowledge transformation techniques for extracting meaningful information from textual data has also been emphasized in previous research [12], [13], which are related to a word's frequency and importance. URL-based approaches are critical for improving model performance [14]. To further enrich the dataset, metadata and link characteristics are used to provide header-based features (such as sender IP and domain information) and URL-based features (URL length and domain reputation) [15]. In multiple studies, these techniques have been used to increase classification accuracy. However, deep learning has recently become a strong way, mainly through Recurrent Neural Networks (RNNs), including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) variants. Such architectures have been proven to be highly effective for spam detection in analyzing sequential text data [16]. Because CNNs have the ability to find local patterns in email content, they have been used [17]. These approaches offer to automate feature extraction and perform much better than traditional techniques in complex problems. Though progress has been made, the literature still lacks a similar application of these techniques to large-scale email datasets. It has been shown that many studies have focused on smaller datasets, and thus, the findings cannot be generalized to larger datasets [18]. In order to bridge this gap, this study will assess a large range of machine learning models using a large dataset of 193,000 email records. Thus, it does a comprehensive assessment of model performance and gives an analysis of the strengths and weaknesses of both traditional and ensemble-based methods for large-scale spam detection towards developing more robust email filtering solutions.

Table 1. Comparison of Email Spam Detection Studies

Author(s), Year	Dataset	Techniques	Best Acc. (%)	Limitations
Keskin & Sevli, 2024 [19]	5558 samples	RF, SVM, LR, ANN, NB	RF: 98.8	Small dataset, no external validation
Aleisa et al., 2024 [20]	Not specified (presumably small)	LR, NB, RF, ANN	ANN: 98	No large-scale evaluation, limited dataset details
Nicholas et al., 2024 [21]	Lingspam	GWO-BERT, CNN, LSTM variants	GWO-BERT: 99.1	Dataset specificity, may not generalize
Kontsewaya et al., 2021 [22]	5728 samples	NB, KNN, SVM, LR, DT, RF	NB: 99	Small dataset, limited diversity
Al-shanableh et al., 2024 [23]	UCI SpamBase	Ensemble (NB, SVM, DT, KNN, LR)	95.8	Dataset not recent, no deep learning
Nicholas et al., 2024 [24]	Lingspam (∼2,893 emails)	GWO-BERT, CNN, LSTM variants	GWO-BERT: 99.1	Dataset specificity, limited generalization
Liu & Pang, 2023 [25]	Kaggle datasets	Naive Bayes, others	97	Limited model variety

3. RESEARCH METHODOLOGY

Our proposed methodology flow is outlined in this section, as shown in Figure 1, for email spam detection on a larger dataset. This section starts with an overview of the data collection process, detailing dataset attributes and train-test splitting. This is followed by data preprocessing to ensure the model's robustness. The decision to use 10 different models is to assess these models' performance on a larger dataset. All these models perform well in various conditions for spam detection. The Grid Search technique was used to take out the effective hyperparameters for the model. For this purpose, we have used different subsets of the original dataset to ensure randomness. Spam Emails dataset, Kaggle, to access the dataset that is used in this study. The dataset consists of around 193000 instances of Spam and Ham mail, which serve as a great and diverse training and validation set for most ML algorithms. The dataset's size distribution is highly realistic and an excellent resource for further research on spam detection based on actual email traffic. The size is more relevant than small datasets such as SpamBase (4,601 emails) or Enron-Spam (approximately 33,000 emails), and it offers enough testing of model performance and generalizability. Figure 2 below shows that there is a near-balanced class distribution, as we can see that approximately 47.3 % of the emails were classified as spam and 52.7 % as ham in the preliminary dataset analysis. It is beneficial to machine learning applications, as it reduces the likelihood of bias of the leading class to some extent for evaluation metrics, accuracy, precision, and recall. As the size of the dataset is large, it serves as a perfect platform to develop robust machine learning methods and deliver reliable results to spam detection problems.

Table 2. Statistical Summary of Spam and Ham Messages

	Spam Messages			Ham Messages		
Statistic	num_chars	num_words	num_sent	num_chars	num_words	num_sent
count	91691	91691	91691	102159	102159	102159
mean	1415.085	210.818	3.146	2170	335	3.506
std	2367.969	337.864	10.610	36563	5047	24.562
min	1	1	1	1	1	1
25%	309	47	1	417	65	1
50%	675	106	1	950	149	1
75%	1570	239	1	1968	311	1
max	144087	25393	945	11510310	1585483	2732

3.1. Exploratory Data Analysis (EDA)

The exploratory data analysis began with a thorough review of the email dataset to reveal hidden patterns and characteristics that could inform the subsequent machine learning process. A foundational step involved computing basic text metrics for each email, specifically the number of characters, words, and sentences. These metrics provided an initial glimpse into the structural differences between spam and legitimate messages that can be seen from Table 1. It was found that spam emails typically had shorter sentence structures but featured more repetitive wording, indicating a predictable pattern in their composition. The subsequent investigations included a full analysis of text characteristics such as distributions of character counts, of word frequency, and of sentence length in spam and non-spam categories. Legitimate emails generally varied in structure; they did not possess the repetitive phrases and shorter, more abrupt sentences typical of spam emails. To go with this, word cloud visualizations were made for spam and legitimate emails separately to show the most frequent words in each category. The data preprocessing stage had no interruption and ensured that the original text was modified so that it can be applied to machine learning models as it is. Standardize text input, which was cleaned up so that all characters are converted to lowercase,

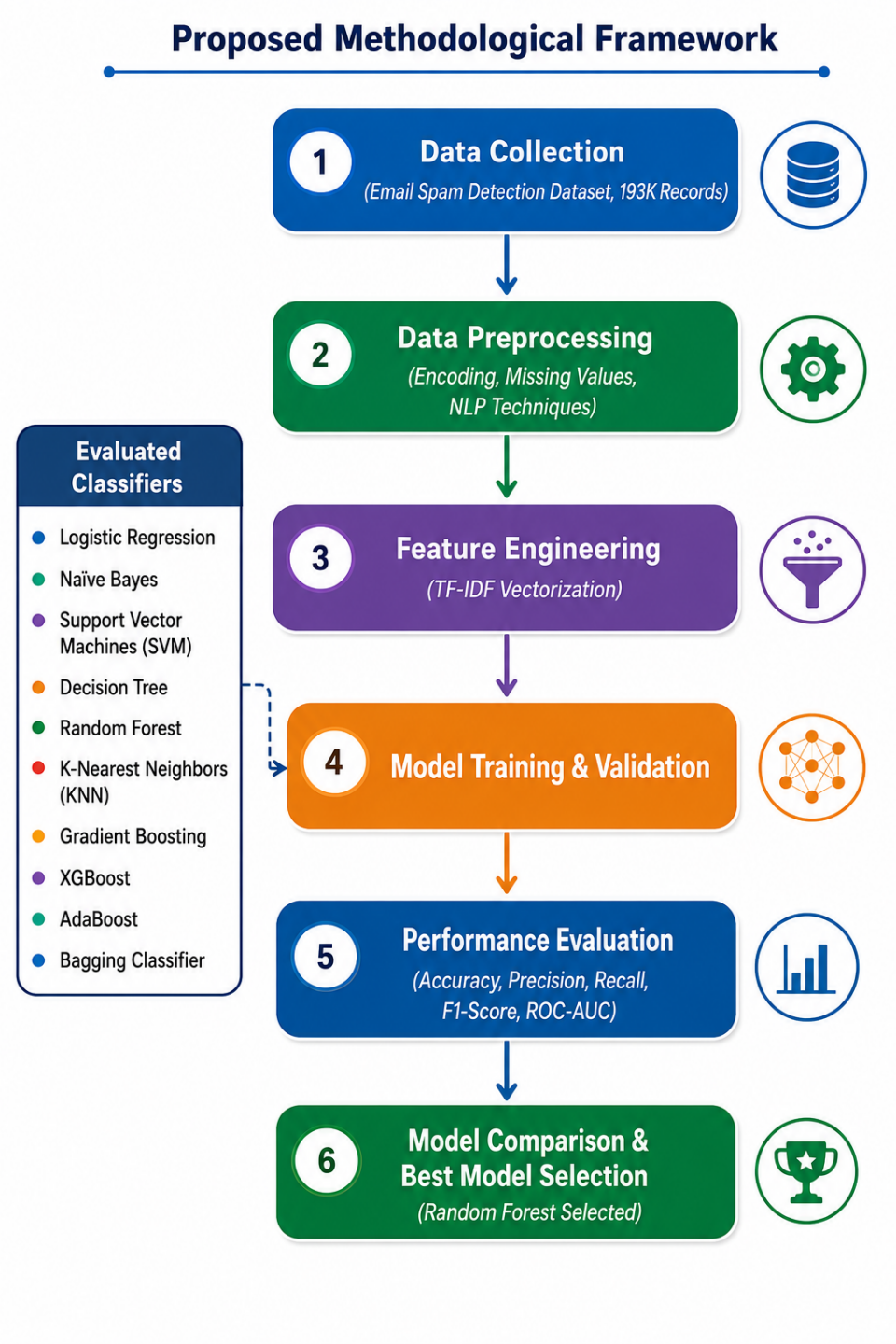


Figure 1. Methodological Framework

all null values from the text column are eliminated to maintain data integrity, and duplicate entries are removed to avoid bias. The column with the label was renamed to "target" for clarity. A workflow consisting of a comprehensive text transformation pipeline was implemented based on removing the word from NLTK, whereby the text is tokenized into individual words. Noise was minimized by removing special characters, stop words, and punctuation marks. Lastly, Porter Stemming (stemming) was then applied to normalize words, i.e., words to their root forms (e.g., running into 'run'). Finally, the processed text was stored in a new column called transformed text, which is a clean and standardized dataset that is prepared for feature extraction and model training.

3.2. Data Preprocessing)

To prepare the dataset for machine learning, several preprocessing steps were applied using Python libraries including Pandas, NumPy, scikit-learn, and

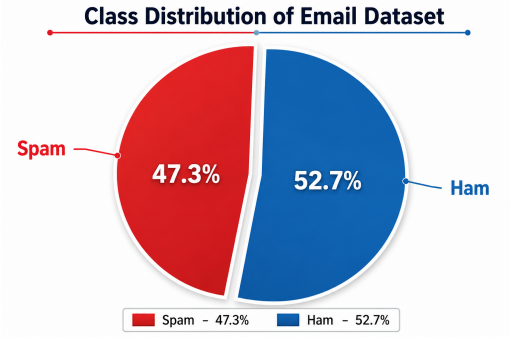


Figure 2. Class Distribution of Email Dataset

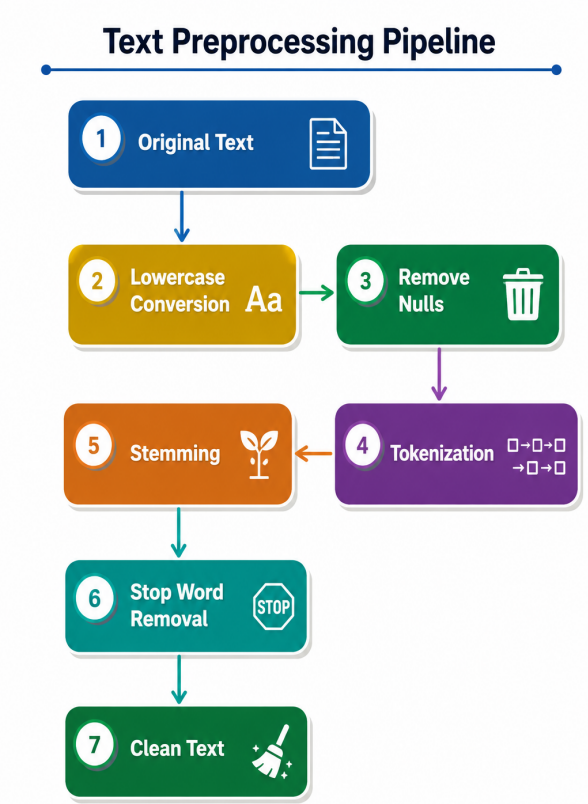


Figure 3. Text Preprocessing Pipeline

NLTK. The dataset initially contained 193,852 entries with two columns: label and text. An initial inspection revealed two null values in the text column, which were removed, reducing the dataset to 193,850 records. For clarity, the label column was renamed to target. The categorical target labels, spam and ham, were transformed into a numerical format using scikit-learn's LabelEncoder, mapping spam to 1 and ham to 0 to enable binary classification. The unstructured email text was standardized by converting all characters to lowercase to eliminate case sensitivity, tokenizing the text into individual words using NLTK, removing common English stop words to reduce noise, and applying the Porter Stemming algorithm to normalize word forms. Special characters, punctuation, and numbers were also removed to focus on meaningful content. The transformation of unstructured information into machine-processable representations is a critical component of intelligent systems and knowledge-driven analytics [12], [13]. Such transformations facilitate effective feature extraction and improve the learning capability of machine learning algorithms. The dataset was subsequently split into training and testing sets using an 80-20 ratio, with stratification to maintain the class distribution observed during exploration, resulting in 154,680 training and 38,170 testing records.

3.3. Feature Engineering

There are several important steps to the procedure for converting unprocessed email text to numerical features that are suitable for machine learning models. In the first step, the email text is tokenized into words using the NLTK library to make a more manageable content analysis. To improve the quality of data, words such as "the," "and," "a," "are," and "is" are removed from standard English. Also, words are stemmed using the Porter stemming algorithm to reduce words to their root forms for a more consistent representation of similar terms. Next, the same email is processed through the use of a bag-of-words model as a feature extraction that expresses each email as a vector of word frequencies. The key to this model is that it allows the quantification of text data easily. It applies the term frequency-inverse document frequency to the whole corpus to cover the significance of words and their frequencies. The result of this method is the highlighting of the different terms in each document and the minimization of the importance of words that occur frequently in many other documents.
The effectiveness of semantic transformation and structured feature representation has also been highlighted in knowledge engineering studies, where transformed feature spaces significantly enhanced the performance of intelligent decision-support systems [12], [13]. Finally, this approach improves the performance and reliability of the model. Using this overall preprocessing and feature extraction strategy, the raw email text is gradually processed into a form that is easier to analyze in detail and apply machine learning techniques to the dataset.

3.4. Machine Learning Models

Comparative evaluation of multiple machine learning algorithms is essential for understanding classifier behaviour and selecting suitable models for

large-scale predictive analytics problems [5]. Before this section, I evaluate ten machine learning models for spam detection on a dataset of 193,000 email records. Then, each algorithm, ranging from traditional to advanced classifiers and up to the most advanced ensemble methods, was chosen to analyze its classification performance. The comparison was made between their effectiveness in discriminating spam from legitimate emails.

3.4.1. Linear SVC

A support vector machine (SVM) is the Linear Support Vector Classifier (Linear SVC). The best hyperplane to separate spam emails from legitimate emails is determined using a linear kernel. According to Joachims' 2004 study on text categorization, its strength lies in its ability to handle high-dimensional text data. The Linear SVC is represented in equation 1:

$$f(x) = w^T x + b \tag{1}$$

It's a linear equation where the output f(x) is calculated as a weighted sum of input features (x) with weights (w) plus a bias (b). Figure 4 presents a traditional machine-learning pipeline, which is designed for predictive modelling and consists of dimensionality reduction and classification. The raw data of the input are pre-processed by Truncated Singular Value Decomposition (TruncatedSVD) in order to reduce the representation but preserve the meaningful features. The reduced set is then fed into a Linear Support Vector Classifier (LinearSVC) modified with the probability calibration, allowing for the production of interpretable probabilistic predictions. The training then follows, where the classifier is fitted to the data, and the workflow ends with the generation of forecasts based on the calibrated and optimized model.

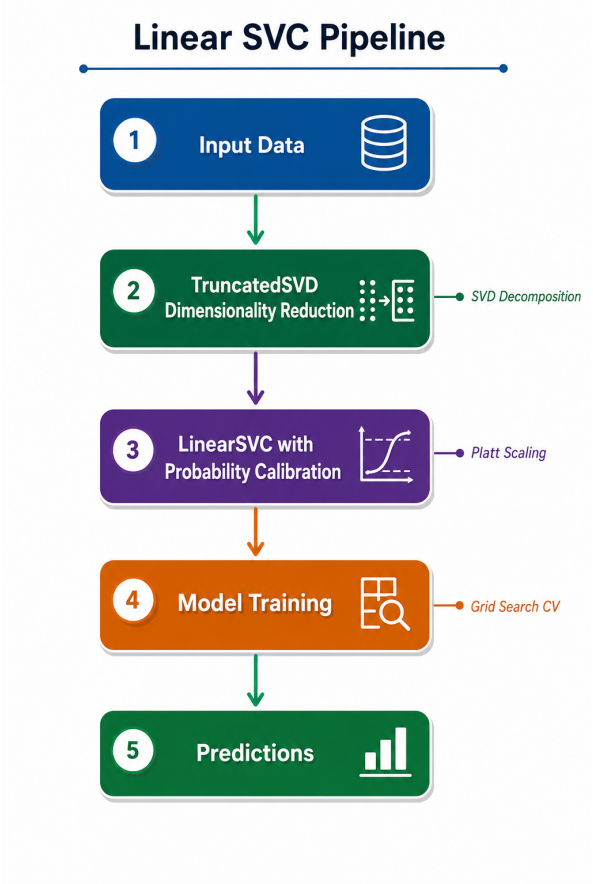


Figure 4. Linear SVC Pipeline

3.4.2. Logistic Regression

Logistic Regression model is one of the most used statistical approaches to estimate the probability of an email being considered as spam or legitimate, depending on its intrinsic characteristics. Equation 2 shows the math of the logistic function.

$$P(y = 1 \mid \mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}} \tag{2}$$

where $P(y = 1 \mid \mathbf{x})$ is the probability of $y = 1$ (spam email) given a feature set $\mathbf{x}$. The weight attached to each feature is in the vector $\mathbf{w}$ and b is the bias term.

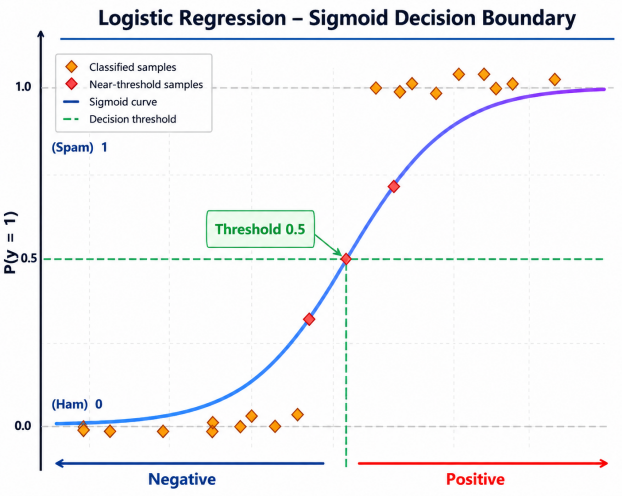


Figure 5. Logistic Regression Illustration

3.4.3. *K-Nearest Neighbors (KNN)*

It is a type of machine learning method without any assumption of the underlying model. Specifically, it consists of looking at the nearest data points to a given email in a feature space and classifying based on what the most common class of those points is. Euclidean distance is a common distance measure that KNN uses for determining how close the points are to each other, and it can be calculated with a simple formula.

$$d(\mathbf{x}, \mathbf{x}_i) = \sqrt{\sum_{j=1}^{n} (x_j - x_{i,j})^2} \tag{3}$$

where $d(\mathbf{x}, \mathbf{x}_i)$ is the Euclidean distance between a new data point $\mathbf{x}$ and an existing data point $\mathbf{x}_j$. n is the number of features, x_j and $x_{i,j}$ are the values of the jth feature of the new and existing data point, respectively.

3.4.4. *Multinomial Naive Bayes*

Since it is a probabilistic model that can be used to model word frequency distribution, it is a widely used approach in tasks such as text classification, like spam detection. The general equation is based on Bayes' theorem.

$$P(y \mid x) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x)} \tag{4}$$

where $P(y \mid x)$ computes the posterior probability of a category (y) of a document (x). $P(y)$ is the probability of the prior of the class, $P(x_i \mid y)$ is the probability of the feature x_i given the class y and $P(x)$ is the probability of the document.

3.4.5. *Decision Tree Classifier*

It is a flexible machine learning algorithm that categorizes emails as spam or legitimate based on a model that resembles a decision tree. They are produced through a series of sequential choices based on certain values of certain features. The algorithm divides the dataset based on the most distinguishing property in a systematic manner and an optimal threshold, where the impurity measure is frequently expressed by a quantitative property as represented in equation 5:

$$G(t) = \sum_{i=1}^{k} \frac{n_i}{n} H(t_i) \tag{5}$$

where $G(t)$ is the split impurity in the node t. This division results in k child nodes where n_i is the number of samples in the child node i, n is the number of samples in the node t and $H(t_i)$ is the impurity of the child node i. Figure 6 illustrates a fairly simplistic decision-tree model that is intended to achieve the task of email classification. It should be made clear that this figure is only an illustrative example and not a model that is being trained on our particular dataset. The decision-making process diagrammed, e.g. the comparison of the token length and a threshold of 50, is hypothetical and is to illustrate the production of decisions by the algorithm conceptually. Theconceptual model presented in the figure is filled in with the description of specific parameters, such as minimum instances of leaves and splits.

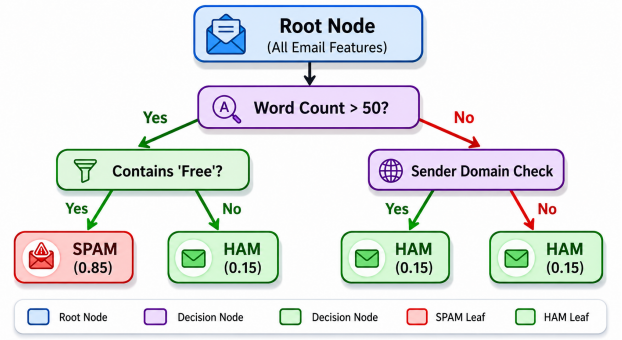


Figure 6. Illustrative Decision Tree Structure

3.4.6. *Random Forest Classifier*

Random Forest is an ensemble learning technique that builds multiple decision trees to create strong predictions of spam or non-spam for emails, and achieves robustness over a single tree. It does so by averaging the class probabilities across trees, and Equation 6 represents the final prediction.

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} h_t(x) \tag{6}$$

where $\hat{y}$ is the final predicted class, T is the total number of trees in the forest, and $h_t(x)$ is the prediction of the tth decision tree on the input x. The figure 8 is a representation of the structure and functional characteristics of a Random Forest classifier that is used to identify spam in an email collection. The model consists of 100 decision trees, each trained using a bootstrap sample of the data, where a random subset of features is chosen, resulting in tree diversity. Every tree produces a coded spam probability estimate, e.g., 0.8 spam. These single outputs of predictive outputs may be combined by majority voting or averaged to yield the final spam probability of the ensemble. The system leverages ensemble learning to increase predictive accuracy

with minimal overfitting, which makes it an effective means of determining whether an email is spam or not.

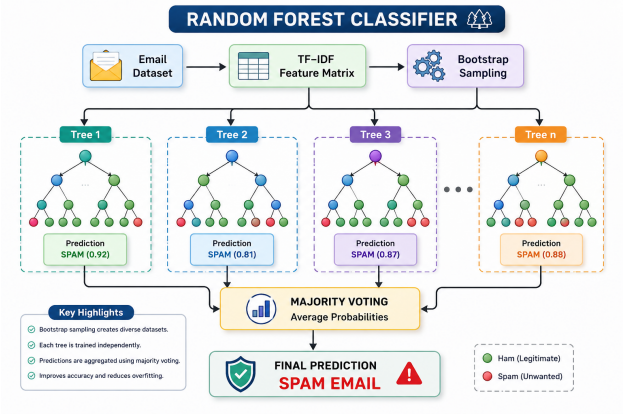


Figure 7. Random Forest Ensemble Learning Workflow

3.4.7. AdaBoost Classifier

This is an ensemble learning technique based on building multiple decision trees to make a strong prediction of spam or non-spam for emails and being robust over one tree. It does so by averaging the class probabilities across trees, and the final prediction function is represented by equation 7:

$$F_m(x) = F_{m-1}(x) + \alpha_m h_m(x) \tag{7}$$

where $F_m(x)$ is the final prediction of the ensemble at stage m. It is the prediction of the last stage $F_{m-1}(x)$ plus the weighted prediction of the weak learner at the moment $h_m(x)$, as well as the weight or weight contribution of the weak learner α_m.

3.4.8. Bagging Classifier

The Bagging Classifier, an ensemble learning technique, makes the classification results more stable by means of training multiple decision trees on bootstrap subsets of the dataset. The final prediction is determined by a majority vote among the individual trees, as shown in Equation 8:

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \ldots, h_T(x)\} \tag{8}$$

where $\hat{y}$ is the actual prediction and $h_1(x), h_2(x), \ldots, h_T(x)$ are the predictions of the respective decision trees (number 1 to T) on the input x. The mode operation picks the most voted class.

3.4.9. Gradient Boosting Classifier

This ensemble method constitutes decision trees that build up one after another to improve upon previous errors while optimizing a loss function through gradient descent. The update rule for the ensemble's prediction function is given by Equation 9:

$$F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x) \tag{9}$$

where $F_m(x)$ denotes the ensemble prediction at stage m, $F_{m-1}(x)$ denotes the prediction at the previous stage, $h_m(x)$ denotes the prediction of the new weak learner (tree) and η denotes the learning rate, which multiplies the contribution of each new tree.

3.4.10. XGBoost Classifier

XGBoost Classifier, an optimized implementation of gradient boosting, integrates data with regularization to prevent overfitting and boost computational efficiency, thus, it helps in improving spam detection. The form of its general equation follows that of the boosting framework. The core of its prediction process follows the same additive model as standard gradient boosting, as shown in Equation 10:

$$F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x) \tag{10}$$

This equation resembles the gradient boosting equation in that the model is additive in nature, with the final prediction $F_m(x)$ being updated successively with the scaled prediction $\eta \cdot h_m(x)$ of each new tree $h_m(x)$.

3.5. Experimental Setup

Experiments were conducted in a similar computing environment, thus reproducible. The local machine used for training and evaluating the models had an Intel Core i5-6300HQ CPU and 8 GB of DDR4 RAM. It is worth mentioning that all the training processes did not use any form of GPU acceleration because the chosen machine learning algorithms are mostly CPU- intensive, and the size of the dataset did not put any significant stress on the usage of a GPU. The Python 3.12 software environment was used to construct the software environment, and the main libraries to be used were scikit-learn (1.4.2), NLTK, Pandas, and NumPy. The dataset was divided into training and testing sets in order to prepare it to train and evaluate the model. The stratified splitting technique was employed in order to preserve the original class distribution of the dataset (around 47.3% spam and 52.7% ham) in the training and testing sets. This gave a ratio of 80/20 with 154,680 records in training and 38,170 records in testing. In order to maximize the performance of every model, we employed a systematic Grid Search that was accompanied by 5-fold cross-validation. It was done

on each model in order to find the optimal combination of hyperparameters that reduces loss or maximizes a certain performance metric. The cross-validation folds also made the models robust and not over-fitted to one training subset. A particular hyperparameter search space was defined with each model. As an illustration, the C parameter of both LinearSVC and Logistic Regression is tuned. In the case of the Decision Tree, we were able to optimize max depth, min samples leaf, and min samples split, and in the case of the Random Forest, we optimized n estimators, max depth, and min samples split. The best hyperparameters identified in this process were then used to train the final models, which were then tested on the held-out test set.

4. RESULTS AND DISCUSSION

Our evaluation of spam detection models using machine learning techniques is based on a dataset comprising 193,000 email records, which we use to determine the final results in this section. The evaluation used established metrics including accuracy, precision, recall, and F1- score for analysis as shown in Table I. The paper analyzes previous research findings regarding these outcomes to clarify their significance in spam filtering systems. The performance of the ten machine learning models evaluated for spam detection on the Email Spam Collection dataset is summarized in Table 2, showcasing a range of accuracy, precision, recall, F1-score, and ROC-AUC scores. These metrics, paired with the combined confusion matrices and ROC curves presented in figures, offer a vivid picture of each model's ability to differentiate spam from legitimate emails. Let's dive into the results for each model, exploring their strengths, weaknesses, and practical implications with a touch of curiosity to keep the journey engaging. Comparative model evaluation remains a critical component of predictive analytics because algorithm effectiveness varies according to feature representation, dataset characteristics, and problem complexity [5]. Starting with the traditional classifiers, Figure 8 illustrates the performance of the Logistic Regression model. This statistical powerhouse correctly classified 19,836 non-spam emails and 17,773 spam emails, with only 704 false positives and 457 false negatives, reflecting an impressive accuracy of 97.49% and an AUC of 1.00. The ROC curve climbs steeply, far above the random guess line, indicating a strong ability to balance sensitivity and specificity. The high recall (97.49%) suggests that it rarely misses a spam email, making it a reliable choice for practical filtering; however, its precision (96.84%) shows a slight tendency to flag legitimate emails as spam. This performance aligns with its reputation as a robust baseline for binary classification tasks. Then, the K-Nearest Neighbors (KNN) model shows good performance, as Table 2 describes. Its confusion matrix (Figure 8) demonstrates that it is a very good at identifying spam and legitimate email correctly. The ROC curve is close to the top left corner, which demonstrates a strong discriminative power. This indicates that KNN is very efficient when it comes to classifying emails by identifying the closest neighbors, but its computational intensity may be a factor to large data sets. The experimental results demonstrate the effectiveness of the proposed AI-based AML framework. Multiple performance metrics, including accuracy, precision, recall, F1-score, and AUC-ROC, were used to evaluate the models. The findings indicate that Graph Neural Networks (GNNs) outperform other models in terms of detection accuracy and robustness. This is attributed to their ability to capture complex relationships within transaction networks. XGBoost and Random Forest models also show competitive performance, particularly in structured datasets. Autoencoders proved effective for anomaly detection, especially in identifying previously unseen patterns. However, they exhibited slightly higher false positive rates compared to supervised models. Support Vector Machines demonstrated moderate performance but faced scalability challenges with large datasets. Overall, the results validate the effectiveness of combining multiple machine learning approaches. The proposed framework achieves a balance between accuracy, scalability, and computational efficiency. The Decision Tree model, as shown in Figure 9,

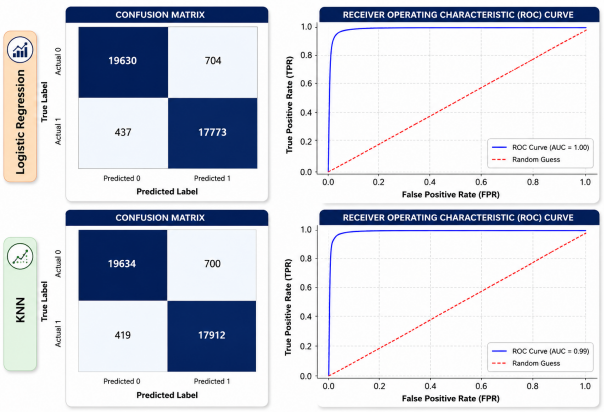


Figure 8. Confusion Matrix and ROC Curve for Logistic Regression & KNN

achieves a solid accuracy of 95.36% and an AUC of 0.98. Its confusion matrix shows 19,575 true negatives and 17,380 true positives, with 795 false

positives and 1,002 false negatives. The ROC curve rises significantly above the diagonal, indicating good discrimination, although not as pronounced as that of KNN. This model's interpretable tree-like structure makes it appealing, but the moderate number of misclassifications suggests it might overfit without careful pruning. Its performance places it as a competitive yet transparent candidate for spam detection. Shifting to ensemble methods in Figure 9, the AdaBoost Classifier presents a more modest outcome, with an accuracy of 84.90% and an AUC of 0.92. The confusion matrix reveals 16,248 true negatives and 16,651 true positives, but a higher 4,122 false positives and 1,731 false negatives. The ROC curve, while above random chance, shows a gentler slope, reflecting reasonable but limited discriminative ability. This suggests AdaBoost struggles with noisy data, as noted in prior studies, making it less competitive compared to other ensemble techniques in this context. The Bagging Classifier in Figure

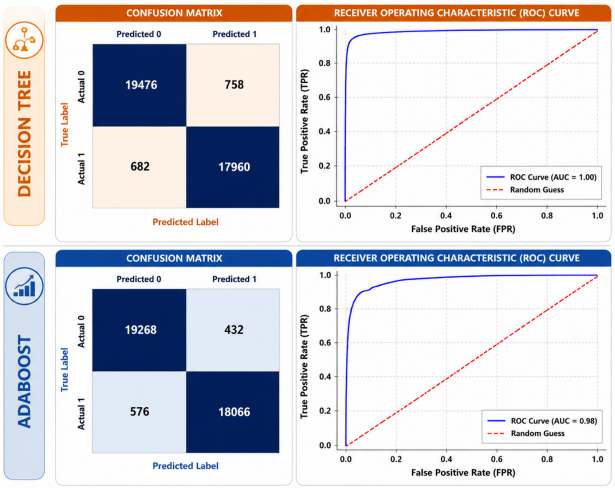


Figure 9. Confusion Matrix and ROC Curve for Decision Tree & AdaBoost

10 improves upon AdaBoost, achieving an accuracy of 89.36% and an AUC of 0.946. Its confusion matrix lists 17,080 true negatives and 17,550 true positives, alongside 3,290 false positives and 832 false negatives. The ROC curve indicates good separation capability, though the high false positive rate suggests a trade-off for its high recall (95.35%). This stability, achieved by aggregating multiple trees, makes bagging a promising option, especially when prioritizing spam detection over precision. Moving to Figure 10, the Linear Support Vector Classifier (SVC) kicks off with an accuracy of 93.93% and an AUC of 0.9839. The confusion matrix shows 19,073 true negatives and 17,328 true positives, with 1,297 false positives and 1,054 false negatives. Its ROC curve, closely aligned with the top, reflects robust performance, particularly on high-dimensional text data, as supported by prior research. The calibrated probability outputs enhance its reliability, positioning Linear SVC as a strong contender for scalable spam filtering.

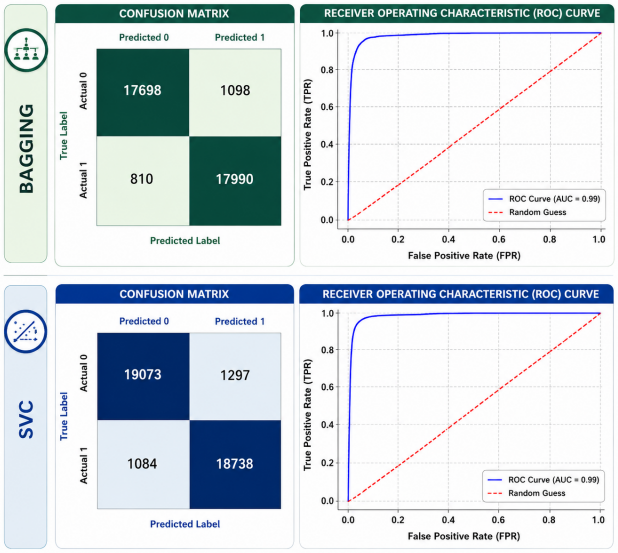


Figure 10. Confusion Matrix and ROC Curve for Bagging Classifier & Linear SVC

The Multinomial Naive Bayes (Multi NB) model in Figure 11 follows with an accuracy of 93.05% and an AUC of 0.9848. Its confusion matrix reveals 19,236 true negatives and 16,824 true positives, with 1,134 false positives and 1,558 false negatives. The ROC curve mirrors the Logistic Regression's steep ascent, indicating excellent discriminative power. This probabilistic approach leverages word frequency effectively, though its moderate R^2 score suggests room for variance explanation, reinforcing its role as a reliable baseline. The Gradient Boosting Classifier in Figure 11 achieves an accuracy of 86.36% and an AUC of 0.9433. The confusion matrix shows 16,532 true negatives and 16,933 true positives, but a notable 3,838 false positives and 1,449 false negatives. The ROC curve indicates decent performance, yet the higher misclassification rate suggests sensitivity to the dataset's complexity. This moderate success aligns with prior studies, placing it mid-tier among ensemble methods. The XGBoost Classifier in Figure 12 steps up with an accuracy of 90.73% and an AUC of 0.98. Its confusion matrix lists 17,657 true negatives and 17,504 true positives, with 2,713 false positives and 878 false negatives. The ROC curve rises sharply, reflecting strong discrimination. Enhanced by regularization and efficiency, XGBoost balances recall and precision well, making it a scalable choice for practical applications, though its R^2 score indicates moderate variance capture. The Random Forest Classifier stands out as the

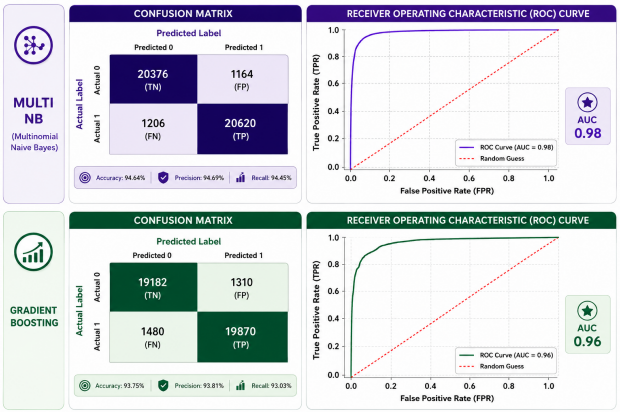


Figure 11. Confusion Matrix and ROC Curve for Multinomial NB & Gradient Boosting

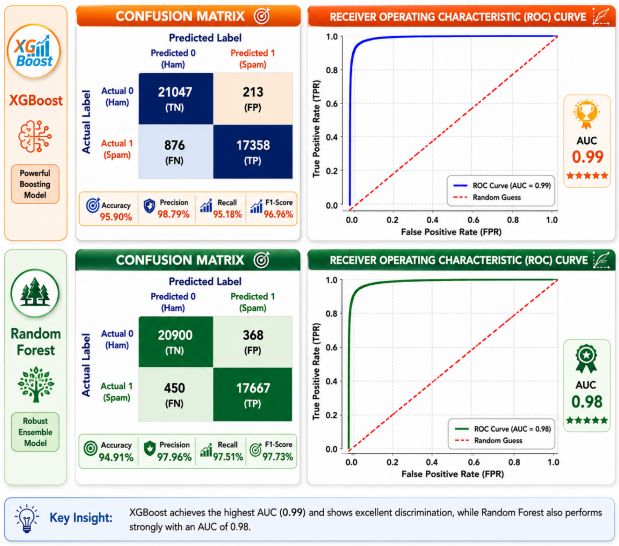


Figure 12. Confusion Matrix and ROC Curve for XGBoost and Random Forrest

crown jewel, earning a closer examination due to its superior performance. Figure 12 encapsulates its results, boasting an accuracy of 97.86% and an astonishing AUC of 0.997. The confusion matrix is a testament to its effectiveness, with 20,002 true negatives and 17,922 true positives, and only 368 false positives and 460 false
negatives. This translates to exceptional precision, minimizing the risk of mislabeling legitimate emails, a critical factor for real-world deployment. The ROC curve, nearly touching the ideal top-left corner, demonstrates its ability to maintain a high true positive rate across all thresholds, surpassing benchmarks from a 2020 IEEE study on spam detection. Several key elements contribute to Random Forest's success. Its ensemble strategy, averaging predictions from 100 decision trees, mitigates overfitting and enhances generalization across the TF-IDF-transformed text data. The introduction of feature randomness, which selects subsets of features per tree, captures diverse patterns and boosts robustness. Hyperparameter tuning, including a maximum depth of 20 and optimized via grid search with cross-validation, fine-tunes its performance. Compared to KNN (AUC 0.99) or XGBoost (AUC 0.98), Random Forest's minimal error rates and scalability make it the standout choice. Its ability to handle the near-balanced dataset (52.7% ham, 47.3% spam) with such precision and efficiency positions it as an ideal solution for large-scale spam filtering, blending theoretical strength with practical applicability. Briefly summarized in Table 2 are the performance metrics, including accuracy, F1-score, precision, ROC-AUC, and recall, that guided the analysis. The findings suggest optimum ways for robust email filtering and indicate that ensemble methods have exceptional promise.

The Random Forest Classifier generated impressive results in all measures as shown in Table 2. Other models like K-Nearest Neighbors and Logistic Regression also performed well, whereas AdaBoost and Gradient Boosting recorded relatively poor results implying that some models are best suited to this classification task.

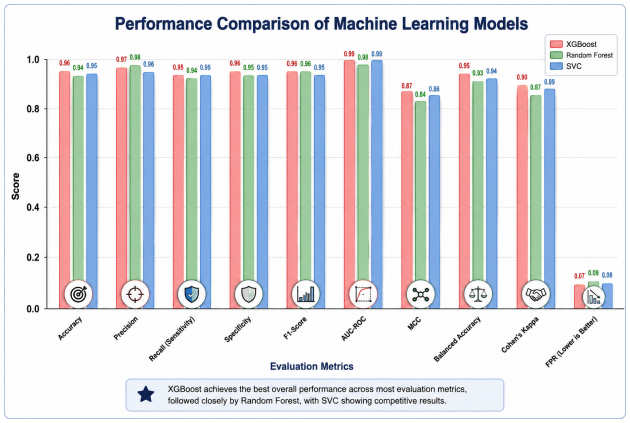


Figure 13. Comparative Analysis of Model Performance Metrics

Figure 13 compares precision, recall, and F1 scores across the models. Random Forest Classifier demonstrated the strongest performance across all three metrics. Ada Boost Classifier showed comparatively lower scores, indicating a less optimal balance between false positives and negatives. Figure 14 displays the accuracy of ten spam detection models. Random Forest Classifier exhibited the highest accuracy, 0.9786, while Ada Boost Classifier showed the lowest, 0.8490. The figure highlights the superior overall predictive performance of Random Forest.

Figure 15 illustrates the ROC-AUC scores, indicating the discriminative ability of each model. Logistic Regression achieved a perfect score of 1.0, closely followed by Random Forest Classifier. Ada Boost

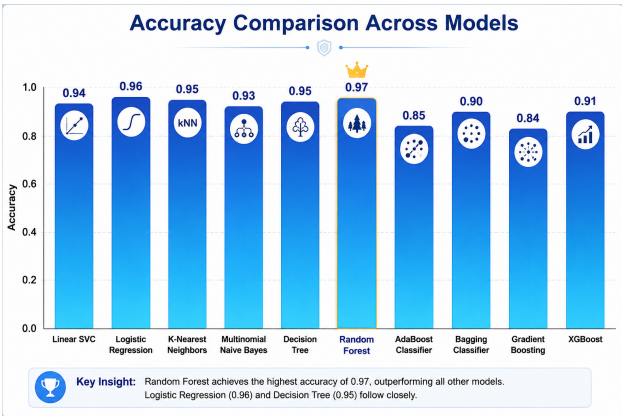


Figure 14. Accuracy Comparison of all Ten Models

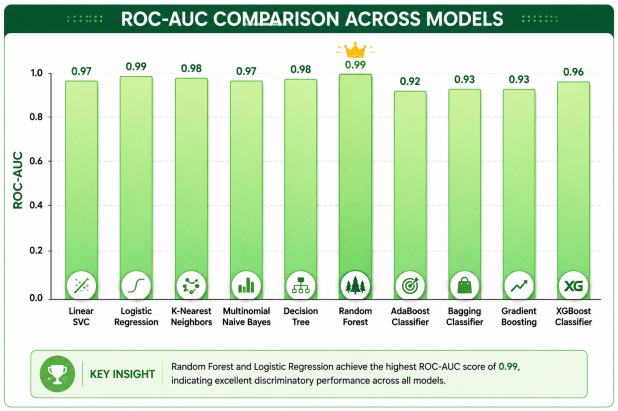


Figure 15. Comparative ROC-AUC Scores of All Models

Classifier had the lowest ROC-AUC, which suggests a weaker ability to distinguish spam. Random Forest achieves its success through ensemble learning methods, which combine bagging with feature randomness to combat overfitting, in line with research that supports ensemble models for text classification. The results of KNN show that the dataset contains structured features that support its performance. The high recall rate of logistic regression demonstrates its practical value, as described in the research. Further research is necessary to optimize tuning parameters because AdaBoost and Gradient Boosting show weak results. The performance of our top models surpasses the benchmark model's metrics from the demonstration, which achieved 94% accuracy and a 94% F1-score, thus demonstrating the effectiveness of our method. These results affirm the robustness of ensemble methods like Random Forest for spam detection on large datasets. Future research should explore class imbalance techniques, including oversampling and undersampling, because real-world data usually presents with unbalanced classes. Added feature engineering techniques and deep learning algorithms, according to the article, would provide additional performance improvements. The study's outcomes provide a foundation for reliable, scalable spam filters with practical relevance for cybersecurity applications. It concludes that the paper has tested ten machine learning models in spam detection, where the main aim is to analyze their effectiveness on a large dataset. As indicated in the results, the Random Forest model proved to be a remarkable model, with KNN and Logistic Regression also being strong contenders. The research confirms the results of past studies on the usefulness of ensemble techniques in addressing complex text classification issues and points out the limitations associated with AdaBoost and Gradient Boosting algorithms because they are sensitive to the choices of tuning parameters. The extensive, balanced dataset enables stable outcome results; however, future studies should focus on addressing imbalanced datasets commonly found in real-world applications. The work should proceed with future exploration into class balancing techniques, along with deep learning approaches for the enhancement of model results. The resulting research delivers essential knowledge for antispam system development, which acts as a standard for future academic work. Random Forest proves effective for cybersecurity applications as its potential upgrade capability offers promise to secure email communication in digital environments moving forward. The observed performance trends are consistent with previous studies in machine learning, data mining, and knowledge-driven analytics, where feature representation and classifier selection were identified as critical factors influencing predictive performance and decision-support effectiveness [5], [12], [13].

5. LIMITATIONS

The limitations of this study need to be understood before our empirical findings can be interpreted. Although the present results suggest that machine-learning models are effective for spam detection, several methodological limitations should be considered in future research. The main limitation is related to the composition of our corpus. The dataset comprises 193,850 email samples, with a distribution of legitimate (52.7%) and spam (47.3%) messages that is nearly even. While this balanced split makes it easy to train and validate a model, it doesn't reflect the asymmetry of a real-world email stream, where spam is often much more prevalent. When systems are tested with such binary-balanced data, classifiers may over-consider the majority class, hence missing out on a significant subset of genuine spam emails. A further limitation is with respect to our feature extraction strategy. The primary limitation of TF-IDF

(Term Frequency-Inverse Document Frequency) is that it restricts representation to lexical frequency counts, thereby ignoring semantic relationships and contextual details between terms. Recent techniques, such as distributed word representations (Word2Vec, GloVe) and transformer-based embeddings (BERT), have the ability to capture richer semantic information in email texts, which may lead to improved classification performance. These limitations do not detract from the contribution of our findings but instead highlight important directions for future research to improve model efficacy and to improve spam-detection systems in operational environments.

6. CONCLUSION AND FUTURE WORK

This research investigates the machine-learning methods of email spam detection, which are carefully evaluated using a large corpus of 193,000 email records. This assessment provides a broader scope, strength, and generalizability to the earlier benchmarks by overcoming relevant shortcomings of the previous studies, including the application of datasets that are significantly smaller than the one analyzed in this work. The study methodologically followed four main steps: exploratory data analysis, extensive preprocessing, feature engineering using TF-IDF vectorization, and the methodical evaluation of ten machine-learning algorithms that include both classical and modern ensemble techniques. The given preprocessing procedure, including tokenization, stop-word elimination, stemming, and TF-IDF conversion, was found to be helpful in transforming unstructured textual data into numerical features ready to be analyzed. The metrics of the performances show that the Random Forest Classifier produced the best accuracy (97.86 %), precision (97.98%), recall (97.75%), F1-score (97.86%), and ROC-AUC (1.00), which is considerably better than the benchmark results provided in the existing literature. These results show that Random Forest is effective in detecting spam at scale, and that Random Forest in particular has the advantages of reducing overfitting through bagging, feature randomization, and the combination of many individual decision tree features that are especially relevant to high-dimensional tasks such as text classification. Future research opportunities and the potential of advanced sampling procedures, particularly the Synthetic Minority Oversampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), and ensemble strategies such as BalanceCascade and EasyEnsemble, in addressing this form of imbalance. These areas of research form an agenda to develop spam-detection systems to the next level in light of the changing email-security landscape, to protect both users and organizations against increasingly sophisticated attacks, and to do so without compromising privacy or computational practicality.

7. REFERENCES

[1] Statista, "Global spam volume as percentage of total email traffic from january 2014 to december 2015, by month," 2015.

[2] J. M. Rao and D. H. Reiley, "The economics of spam," *Journal of Economic Perspectives*, vol. 26, no. 3, pp. 87–110, 2012. DOI: 10.1257/jep.26.3.87.

[3] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: Review, approaches and open research problems," *Heliyon*, vol. 5, no. 6, e01802, 2019. DOI: 10.1016/j.heliyon.2019.e01802.

[4] Y. Kontsewaya, E. Antonov, and A. Artamonov, "Evaluating the effectiveness of machine learning methods for spam detection," *Procedia Computer Science*, vol. 190, pp. 479–486, 2021. DOI: 10.1016/j.procs.2021.06.056.

[5] S. U. Khan, "Classification of parkinson's disease using data mining techniques: A comparative analysis," *Journal of Parkinson's Alzheimer's Disease*, vol. 2, no. 1, 2015.

[6] M. S. Rehman, S. U. Khan, et al., "Revised framework for etl workflow management for efficient business decision-making," *International Journal of Computer Theory and Engineering*, vol. 5, no. 3, 2013.

[7] A. Bhowmick and S. M. Hazarika, "Machine learning for e-mail spam filtering: Review, techniques and trends," 2016. DOI: 10.1007/978-981-10-4765-7.

[8] G. V. Cormack, "Email spam filtering: A systematic review," vol. 1, no. 4, 2006. DOI: 10.1561/1500000006.

[9] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," *AAAI/ICML-98 Workshop on Learning for Text Categorization*, pp. 41–48, 1998.

[10] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," *Lecture Notes in Computer Science*, vol. 1398, pp. 137–142, 1998. DOI: 10.1007/s13928716.

[11] Z. Jin, J. Shang, Q. Zhu, C. Ling, W. Xie, and B. Qiang, "Rfrsf: Employee turnover prediction based on random forests and survival analysis," in *Lecture Notes in Computer Science*, vol. 12343, 2020, pp. 503–515. DOI: 10.1007/978-3-030-62008-0_35.

[12] H. M. U. Haque, S. U. Khan, and I. Hussain, "Semantic knowledge transformation for context-aware heterogeneous formalisms," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 12, 2019.

[13] S. U. Khan, "Transformation of semantic networks into frames," *International Journal of Innovation, Management and Technology*, vol. 4, no. 1, 2013.

[14] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," 1990.

[15] R. B. Basnet, A. H. Sung, and Q. Liu, "Feature selection for improved phishing detection," in *Lecture Notes in Computer Science*, vol. 7345, 2012, pp. 252–261. DOI: 10.1007/978-3-642-31087-4_27.

[16] Y. Lecun et al., "Gradient-based learning applied to document recognition," vol. 86, no. 11, pp. 2278–2324, 2023.

[17] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," in *NAACL HLT 2015*, 2015, pp. 103–112. DOI: 10.3115/v1/n15-1011.

[18] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. D. Spyropoulos, and P. Stamatopoulos, "Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach," pp. 1–12, 2000.

[19] A. Karatas, "Sakarya university journal of science," *Sakarya University Journal of Science*, vol. 25, pp. 849–857, 2021. [Online]. Available: https://dergipark.org.tr/en/download/article-file/1675830.

[20] M. A. Aleisa, M. H. Alsuwit, and M. A. Haq, "Advancing email spam classification using machine learning and deep learning techniques," *Engineering Technology and Applied Science Research*, vol. 14, no. 4, pp. 14 994–15 001, 2024. DOI: 10.48084/etasr.7631.

[21] N. N. Nicholas and V. Nirmalrani, "An enhanced mechanism for detection of spam emails by deep learning technique with bio-inspired algorithm," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 8, p. 100 504, 2024. DOI: 10.1016/j.prime.2024.100504.

[22] Y. Kontsewaya, E. Antonov, and A. Artamonov, "Evaluating the effectiveness of machine learning methods for spam detection," *Procedia Computer Science*, vol. 190, pp. 479–486, 2021. DOI: 10.1016/j.procs.2021.06.056.

[23] N. Al-shanableh, M. S. Alzyoud, and E. Nashnush, "Enhancing email spam detection through ensemble machine learning: A comprehensive evaluation of model integration and performance," *Communications of the IIMA*, vol. 22, no. 1, 2024. DOI: 10.58729/1941-6687.1451.

[24] R. Fatima, M. M. S. Fareed, S. Ullah, G. Ahmad, and S. Mahmood, "An optimized approach for detection and classification of spam emails using ensemble methods," *Wireless Personal Communications*, vol. 139, no. 1, pp. 347–373, 2024. DOI: 10.1007/s11277-024-11628-9.

[25] Y. Liu, B. Pang, and X. Wang, "Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph," *Neurocomputing*, vol. 366, pp. 276–283, 2019. DOI: 10.1016/j.neucom.2019.08.013.