International Journal of Information Security (2024) 23:505–517
https://doi.org/10.1007/s10207-023-00756-1

REGULAR CONTRIBUTION

Improving spam email classification accuracy using ensemble techniques: a stacking approach

Muhammad Adnan[1] · Muhammad Osama Imam[2] · Muhammad Furqan Javed[2] · Iqbal Murtza[2]

Accepted: 2 September 2023 / Published online: 20 September 2023

Abstract

Spam emails pose a substantial cybersecurity danger, necessitating accurate classification to reduce unwanted messages and mitigate risks. This study focuses on enhancing spam email classification accuracy using stacking ensemble machine learning techniques. We trained and tested five classifiers: logistic regression, decision tree, K-nearest neighbors (KNN), Gaussian naive Bayes and AdaBoost. To address overfitting, two distinct datasets of spam emails were aggregated and balanced. Evaluating individual classifiers based on recall, precision and F1 score metrics revealed AdaBoost as the top performer. Considering evolving spam technology and new message types challenging traditional approaches, we propose a stacking method. By combining predictions from multiple base models, the stacking method aims to improve classification accuracy. The results demonstrate superior performance of the stacking method with the highest accuracy (98.8%), recall (98.8%) and F1 score (98.9%) among tested methods. Additional experiments validated our approach by varying dataset sizes and testing different classifier combinations. Our study presents an innovative combination of classifiers that significantly improves accuracy, contributing to the growing body of research on stacking techniques. Moreover, we compare classifier performances using a unique combination of two datasets, highlighting the potential of ensemble techniques, specifically stacking, in enhancing spam email classification accuracy. The implications extend beyond spam classification systems, offering insights applicable to other classification tasks. Continued research on emerging spam techniques is vital to ensure long-term effectiveness.

Keywords Spam · Email · Classification · Machine learning · Ensemble · Stacking method

1 Introduction

With the significant growth in Internet usage for communication, email has become a reliable and efficient method. However, it has also become a likely target for marketing firms and cyber threat actors. Spam email, also known as junk email, refers to unwanted email projected to a larger number of receivers without their consent [1]. These emails are often sent by marketers or other entities to promote their goods or services, but they can also originate from individuals or attack groups with malicious intent, such as phishing scams or attempts to spread spyware or adware. The prevalence of spam and phishing has posed significant challenges to individuals and companies, leading to financial losses and privacy breaches due to the lack of cyber awareness and robust email filtering methods.

Figures indicate that as of 2022, 55% of all emails are categorized as spam [2], quantity to roughly 15.4 billion emails per day with estimated $355 million per year Internet users roughly [3]. However, numerous email providers have applied spam filters to recognize and block spam emails, and it is still necessary for consumers to implementation restraint and thoroughly examine emails before opening them or clicking on any links they may comprise.

Conversely, spam and phishing emails remain to be a key issue owing to advancing methods and capability. Spammers persistently modify the manner they send spam, forming it further possible that their emails will go pass the spam filters. These tactics include developing new and updated methods and techniques, such as using sender email addresses that look legitimate [4]. Furthermore, personalized spam, which includes the recipient's name, occupation and other private information in the body of the message or the subject line,

✉ Muhammad Adnan
muhammad.adnan@uit.no

[1] Department of Technology and Safety, UiT the Arctic University of Norway, Tromsø, Norway

[2] Faculty of Computing & AI, Air University Islamabad, Islamabad, Pakistan

creates extra problems for spam filters to precisely detect and block such messages [5].

To address this problem, this study proposes an ensemble framework that combines the predictions of five base classifiers into a stacking method. The goal of the proposed study was to enhance the precision of spam detection using an ensemble method as compared to individual classifiers alone. The five base classifiers used in this study are logistic regression, decision tree, Gaussian Naive Bayes (NB), AdaBoost and K-nearest neighbors. The technique of stacking classifiers makes it possible to joint the outputs of several distinctive classifiers into a single system that is more accurate than any one in isolation.

The next paper parts are compromised with section-2 related to the literature review part after brief introduction, section-3 describes material and method including the dataset, section-4 then provides a detailed description of the proposed methodology, section-5 deals with results and discussion followed by the presentation of our results, and section-6 describes conclusion and future work.

2 Review of literature

Spam email classification is an evolving and challenging problem, and many machine learning techniques have been widely explored to improve its precision and accuracy. Several past studies have investigated different aspects of spam email classification, including application of machine learning approaches, adversarial approach, use of ensemble methods and unsupervised learning.

Nikhil Kumar et al. in 2020 study provided a contrast of various machine learning algorithms in the field of spam classification [6]. They used support vector classifier, K-nearest neighbor, Naïve Bayes, decision tree, random forest, AdaBoost classifier and Bagging classifier. In their study, support vector classifier achieved 0.92 precision, K-nearest neighbor reached 0.92, Naïve Bayes attained 0.87, decision tree achieved 0.94, random forest scored 0.90, AdaBoost classifier reached 0.95, and Bagging classifier attained 0.94 precision. In our study, we utilized a different dataset, and our base models demonstrated precision values closely aligned with their reported results, often surpassing 0.92

Akash Junnarkar et al. (2021) conducted a series of experiments on Enron dataset by applying four classification algorithms [7]. They applied SVM, RF, NB, DT and KNN with achieved accuracies as 97.83%, 97.60%, 95.48%, 90.90% and 95.29%, respectively. SVM emerged as standout performer closely followed by random forest classifier. The authors also proposed potential research direction about further refining accuracy through the adoption of computationally expensive yet highly precise ensemble techniques like xgboost.

In a study conducted by W.A. Awad et al., the performance of six machine learning methods in the context of spam classification was summarized using SpamAssasin dataset [8]. In terms of accuracy, for the Naïve Bayes (NB) method, accuracy stood at 99.46%. The SVM achieved an accuracy of 96.90%, and KNN algorithm showed an accuracy of 96.20%. In same study, neural network (NN) approach had accuracy of 96.83%. The artificial immune system (AIS) achieved an accuracy of 96.23%. Lastly, the rough sets (RS) method had an accuracy of 97.42%.

In their study, Zhang et al. reviewed the adversarial methods used to evade spam email classification methods and discussed the methods proposed to counter these attacks [9]. They also highlighted the constraints of presented methods and techniques and suggested some guidelines for potential research in the field of spam email classification.

In their study published in 2020, Shaukat et al. evaluated the working of various ML methods for spam email classification—comprising DT, SVM and NB classifiers [10]. They observed that support vector machines showed similar performance to decision trees. The researchers also found that these two methods were effective when it came to handling email with large amounts of content—such as those emails with more than 10,000 words. In another study, researchers utilized different techniques such as multilayer perceptron, SVM, KNN and RF for classification problems [11, 12].

Hajek et al. anticipated a deep learning model that used feature representations, such as character n-grams and word embeddings. They also used unsupervised topic modeling technique for the similar problem [13]. Their study presented promising results compared to publicly available baseline machine learning models. But, Ramanathan et al. proposed an unsupervised topic modeling technique for spam email classification but achieved near similar results. They proposed the use of latent Dirichlet allocation model to generate features from the training set and used these features for a deep learning model [14].

In a hybrid approach, Ghourabi et al. proposed a combination of CNN and LSTM techniques for email classification [15]. Their proposed hybrid model outperformed several frequently used methods such as GNB and decision trees.

In a comprehensive study comprising strengths and weakness of several machine learning models, Madhavan et al. experimented on spam email dataset, using multiple approaches such as hyperparameter tuning [16]. They also identified future scope and challenges, pointed out limitations and suggested directions for further research including use of hybrid or ensemble frameworks. Parallel to this, Rayan et al. combined DT and RF classifiers to improve classification accuracy [17]. Their proposed model demonstrated improved performance compared to some baseline methods. Similarly, Suborna et al. enhanced the accuracy of spam online reviews by applying the stacking approach and achieved significant results [18].

In study published by Isvani Frias et al. [19], they proposed a fast adaptive stacking of ensembles method (FASE) for learning non-stationary data streams. Their algorithm processed real-time input in constant time and space complexity. Their experiments showed improved predictive accuracy as contrasted to several another traditional machine learning methods. Moreover, El-Kareem et al. [20] employed a stacking approach that combined Naive Bayes, SVM, decision trees and a meta-classifier for email spam classification, reaching a precision of 95.67%. Besides, Madichetty et al. utilized a stacking-based CNN for detecting fake or spam tweets [21].

Oh et al. [22] proposed a method for identifying spam remarks on YouTube video streaming website, addressing the need for more effective spam detection despite YouTube's existing spam blocking system. The writers organized tests using six different ML methods and two ensemble models on remark data from prevalent videos. The results contributed to the performance of spam detection on YouTube and addressing associated challenges.

Zhao et al. [23] focus on spam recognition in social media networks and suggest a heterogeneous stacking-based ensemble learning architecture to mitigate the effect of class inequality. They utilize six different base classifiers in the base module and introduce cost-sensitive learning in the combining module. Experimental results demonstrate improved spam detection on imbalanced datasets, enhancing information security in social networks.

Liu et al. [24] address the class inequality challenge in Twitter spam recognition. They suggest a fuzzy-based oversampling method called FOS and develop an ensemble learning method involving adjusting the class distribution, building classification models on redistributed datasets and combining predictions through majority voting. Experimental results show significant improvement in spam detection rate for imbalanced class distribution, mitigating Twitter spam.

Omotehinwa et al. [25] focus on spam email detection and classification, a significant cybersecurity threat. They develop standard models using random forest and XGBoost ensemble algorithms and employ hyperparameter optimization techniques. The adjusted XGBoost model outperforms the RF model, achieving high accuracy, sensitivity and F1 scores. The improved XGBoost model proves efficient and well organized for spam email recognition, contributing to cybersecurity efforts. Researchers also emphasized that maintaining software and code reliability is essential for quality research in classification problems [26–28].

In conclusion, the studies reviewed above demonstrate the diverse approaches and advancements in spam email classification using machine learning techniques. Compared to existing approaches, our proposed model offers accuracy improvement in spam email classification. By focusing on enhancing accuracy and addressing evolving spam techniques, we introduce a stacking ensemble method that combines predictions from multiple base classifiers. Our experimental evaluations using distinct datasets, along with additional experiments, validate the effectiveness and generalizability of our approach. The model demonstrates higher precision, recall and F1 scores, addressing limitations of individual models and improving performance. The proposed research provides renewed comparisons of classifier performances, considering the combination of diverse datasets, showcasing the potential of our model to enhance spam email classification accuracy.

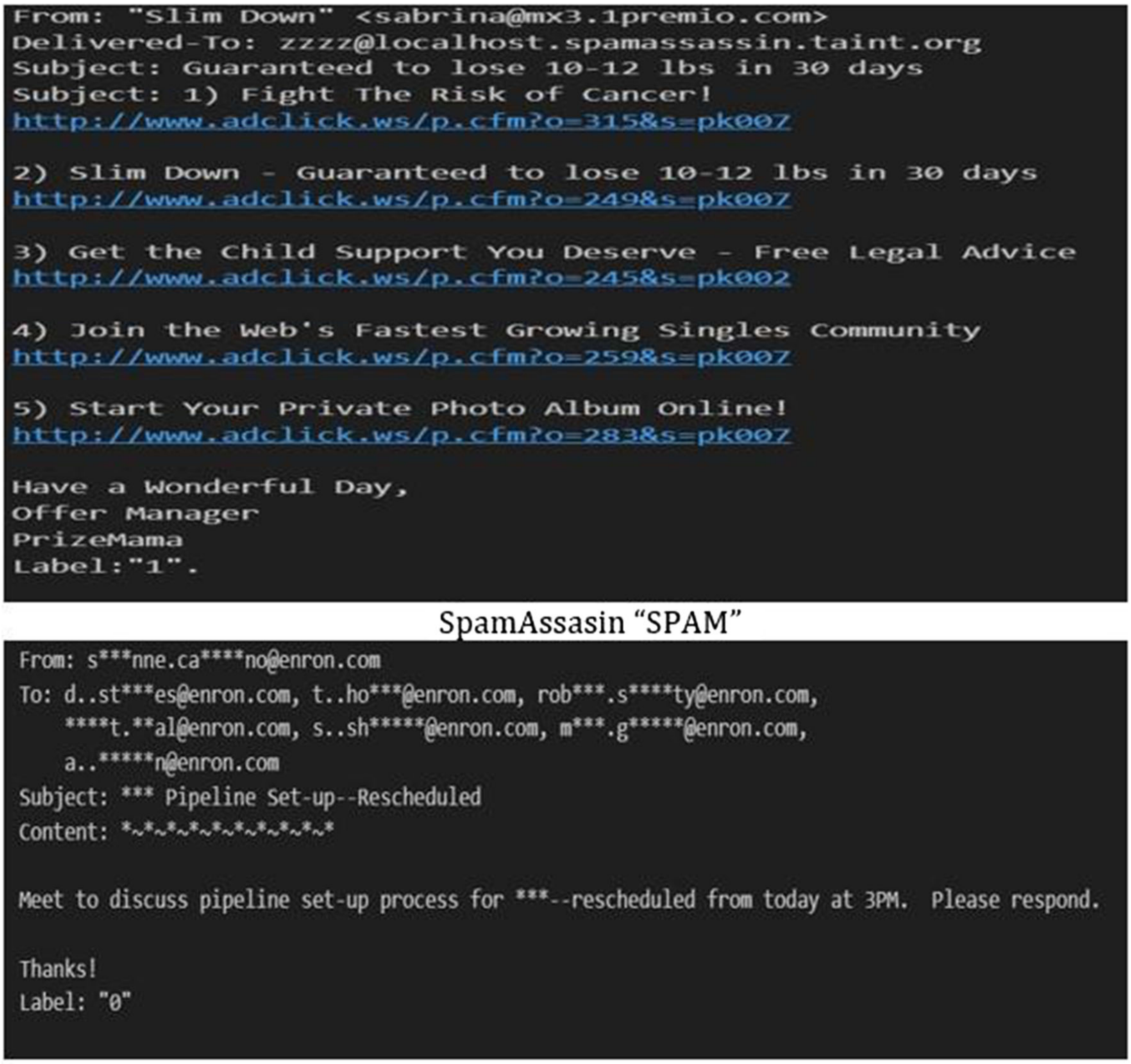
3 Material (dataset description)

To enhance the diversity and robustness of our spam email classification model, we combined two publicly accessible datasets: the SpamAssassin (SA) dataset [29] and the Enron-Spam dataset processed form [30]. The SA dataset consisted of 6047 messages, of which 31.37% (1897) were categorized as "spam" and 4150 were labeled as "Ham." On the other hand, the Enron dataset contained 0.5 million email messages; however, we created a subsampled version of 7582 messages, with a spam ratio of 41%, to integrate it with the SA dataset. Figure 1 shows example message of both datasets. To ensure consistency and modeling of data, features "label," "Subject" and "Body" in the Enron dataset were aligned with those in the SA dataset, and subsequently, a CSV file was generated utilizing data frames. The resulted csv file consisted of 13629 rows; each row contained an individual email.

3.1 Preprocessing

The combined dataset had imbalanced class distributions, and we increased the number of spam emails to balance the dataset and prevent overfitting toward the majority class. Specifically, we replicated the spam emails in the dataset to increase their count to match that of the ham emails. This over sampling approach ensured that our model had equal representation of both classes, which is essential for accurate classification performance. Figure 2 shows balance of the dataset achieved after over sampling.

Text column contained Subject and Content of that email. We performed a series of text data preprocessing operations on the dataset using Python and the Natural Language Toolkit (nltk). The first operation converts all the text to lowercase and removes special characters. Then, the text is tokenized into individual words using the Natural Language Toolkit (nltk). The next step is to eliminate stop words from the text using a predefined set of stop words from the nltk library.



```
From: "Slim Down" <sabrina@mx3.1premio.com>
Delivered-To: zzzz@localhost.spamassassin.taint.org
Subject: Guaranteed to lose 10-12 lbs in 30 days
Subject: 1) Fight The Risk of Cancer!
http://www.adclick.ws/p.cfm?o=315&s=pk007

2) Slim Down - Guaranteed to lose 10-12 lbs in 30 days
http://www.adclick.ws/p.cfm?o=249&s=pk007

3) Get the Child Support You Deserve - Free Legal Advice
http://www.adclick.ws/p.cfm?o=245&s=pk002

4) Join the Web's Fastest Growing Singles Community
http://www.adclick.ws/p.cfm?o=259&s=pk007

5) Start Your Private Photo Album Online!
http://www.adclick.ws/p.cfm?o=283&s=pk007

Have a Wonderful Day,
Offer Manager
PrizeMama
Label:"1".
```

SpamAssasin “SPAM”

```
From: s***nne.ca****no@enron.com
To: d..st***es@enron.com, t..ho***@enron.com, rob***.s****ty@enron.com,
    ****t.**al@enron.com, s..sh*****@enron.com, m***.g*****@enron.com,
    a..*****n@enron.com
Subject: *** Pipeline Set-up--Rescheduled
Content: *~*~*~*~*~*~*~*~*

Meet to discuss pipeline set-up process for ***--rescheduled from today at 3PM.  Please respond.

Thanks!
Label: "0"
```

Enron “HAM”

Fig. 1 Sample emails of datasets

Furthermore, textual data are transformed to numerical structure using the term frequency-inverse document frequency (TF-IDF) method [31]. Using this technique, a weight is assigned to each word in the document based on its frequency and rarity across all documents in the dataset. The resulting vectorized format prepares the data for further analysis or modeling. The result was a matrix in which each unique word was represented by a column of that matrix and each sample text was a row.

Additionally, we applied grid search with cross-validation (GSCV) to fine-tune hyperparameters for k-nearest neighbors (kNN), logistic regression (LR), decision tree (DT), AdaBoost, Gaussian Naive Bayes (GNB) and the stacking meta-classifier. This involved segmenting the hyperparameter space into a predefined grid and conducting fivefold cross-validation. The optimal hyperparameter combinations for each model were then used for further experimentation.

4 Proposed methodology

We proposed the use of a stacking ensemble method for spam classification, which involves training multiple classifiers which in our case was LR, DT, GNB, KNN and AdaBoost on the training data and afterward using their estimates as inputs to “meta-classifier” that makes the final prediction. The framework illustrated in Figure 3 demonstrates that two datasets are merged, preprocessing and balancing operations are performed on them before we forward them into base classifiers, and their output is then aggregated as an input to the stacking-based meta-classifier.

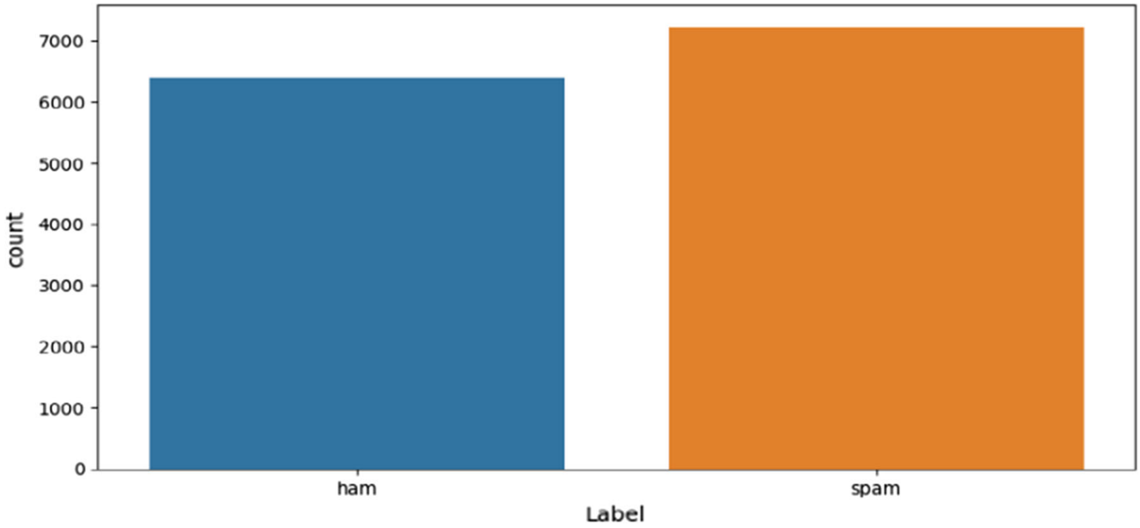


Fig. 2 Combined dataset distribution

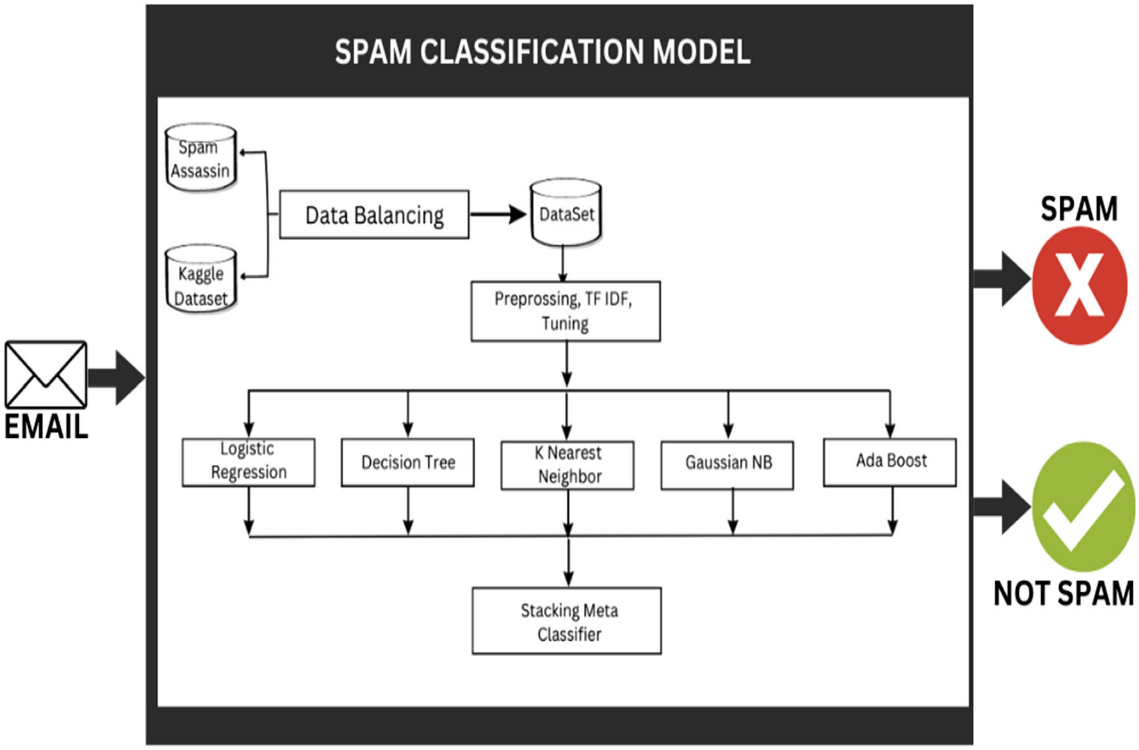


Fig. 3 (Basic framework of stacking classifier)

Regarding our choice of base classifiers, we aimed to select a diverse set of classifiers that perform well on different types of data and classification tasks. We selected logistic regression, decision tree, GNB, KNN and AdaBoost classifiers as they are commonly used and have shown good performance in similar spam classification tasks.

Logistic regression is a linear model that works well with large datasets and can be easily interpreted. Decision trees are a nonlinear model that can handle datatypes of both categorical and numerical in nature and can capture complex relationships between features. GNB is a probabilistic model that works well with high-dimensional data and assumes independence between features. KNN is a lazy learning algorithm that works well with small datasets and can capture local patterns in the data. AdaBoost which is also an ensemble method combines a group of weak classifiers to mold a

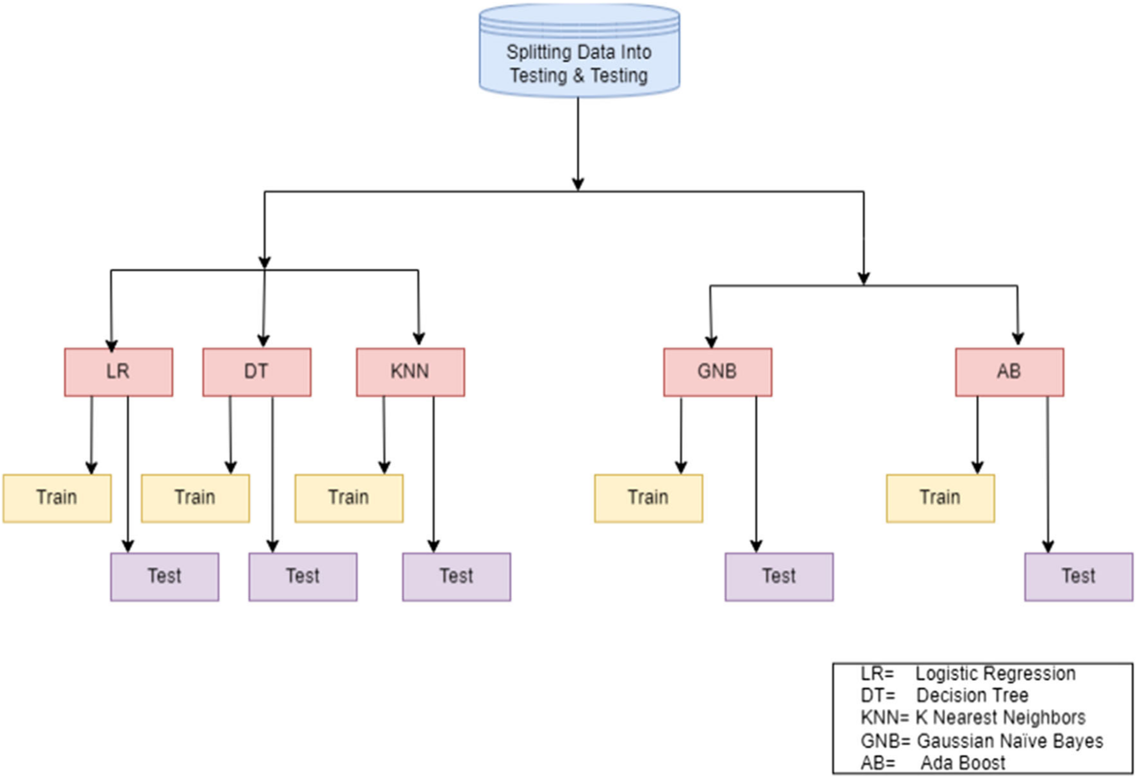


Fig. 4 (Flowchart of base classifiers)

stronger classifier and similarly can handle both numerical and categorical data.

By selecting a mix of weak and strong classifiers, we aimed to take advantage of the strengths of individual classifier and increase the comprehensive performance of the aggregate model. The testing and training on selected base classifiers work in the way demonstrated in Figure 4. The dataset was divided and distributed among base classifiers as follows:

The working and use of base classifiers on our dataset is deeply discussed in ensuing paragraphs.

4.1 Logistic regression

First logistic regression was used for spam classification. In this application, the dependent variable was binary, which in our case has a value of "spam" or "not spam." The predictor variables include features of the email, sender, the subject line and content of the message. The logistic regression model was trained on our dataset, using the predictor variables to learn the patterns that distinguish spam emails from non-spam emails, given the values of the predictor variables for that email. This prediction was represented by the following (Equation 1) [32]:

$$p(y = 1|x) = \text{sigmoid}(w_0 + w_1x_1 + w_2x_2 + \ldots + w_nX_n) \tag{1}$$

In this equation, y is the dependent variable, to which a value of 1 is assigned if the email was spam and 0 if it was not spam. **x** is a vector of predictor variables, which include features the sender, subject line and contents of the email. $\mathbf{W_0}$ was the intercept term, which represents the long odds of the dependent variable being 1 when all the predictor variables are 0. $\mathbf{W_1, w_2, ..., w_n}$ are the coefficients aimed at the predictor variables $\mathbf{x_1, x_2, ..., x_n}$, respectively. These coefficients denote the changeover in long odds of the dependent variable staying 1 for a one-unit raise in the subsequent predictor variable, sharing all other predictor variables constant. sigmoid was the logistic function, which maps the input to a value between 0 and 1. It was defined mathematically as (Eq. 2) [33].

$$\text{sigmoid}(x) = \frac{1}{(1 + \exp(-x))} \tag{2}$$

For classifying of an email either spam or not spam using logistic regression, the expected probability **p(y=1|x)** was compared to a threshold value. The email was classified as spam, if the probability was above the threshold and not spam, if it was below the threshold. The threshold value was set to 0.5, but it can be adjusted based on the specific requirements of the classification task. Gradient descent an optimization algorithm was used to learn the coefficients $\mathbf{w_0, w_1, w_2, ..., w_n}$ and the threshold value from the training data [34]. The objective was to determine the parameter values that reduce the difference between the forecasted probabilities and actual labels in the training dataset to a minimum.

4.2 Decision tree

During the training of a decision tree model, the predictor variables or features are used to recursively partition the data into slighter and lesser subsets until a final decision is made at a leaf node. At the root node, the model splitting the information into two subdivisions based on the sender of the email, categorizing emails from known spammers in one subset and emails from non-spammers in the other subset [35]. At each subsequent node, the model further splits the data based on the value of the subject line of the email, identifying subsets of emails with suspicious or benign subject lines. This process continues until the data are partitioned into pure subsets containing only spam or non-spam emails. These pure subsets, known as leaf nodes, yield the final predictions for the corresponding subset of emails.

- If sender in spam list:
- Predict spam.
- Else: If subject line has suspicious words:
- Predict spam Else: If email content has spammy words:
- Predict spam Else: Predict not spam.

4.3 K-Nearest neighbor

In KNN, the distance between the new email and training set emails was used to make a prediction about whether the new email was spam or not spam. The email was represented as a feature vector, which consists of the values of the predictor variables for that email. These features included the sender, the subject line, the content and other characteristics of the email. The distance between the new email and other training emails was calculated using a distance metric, Euclidean distance.

The size of k was set to 11. The K training emails with the minimum distances to the new email are chosen as the closest neighbors. The majority label (i.e., “spam” or “not spam”) among the K-nearest neighbors was used as the prediction for the new email. If there was a tie, the prediction was based on the median of the labels of the nearest neighbors, or a random label was chosen. This prediction was represented by the following equation:

Given a new email x and a training set $\{(\mathbf{x_1}, \mathbf{y_1}), (\mathbf{x_2}, \mathbf{y_2}), ..., (\mathbf{x_n}, \mathbf{y_n})\}$, where x_i is the attribute vector for the $\mathbf{i^{th}}$ training email and $\mathbf{y_i}$ is the label (“spam” or “not spam”) for that email, the prediction for **x** using KNN is (Eq. 3):

$$y = \text{majority label}(Y_i)\text{for } i \text{ in nearest neighbours}(x, k) \quad (3)$$

In this equation, **x** is the feature vector for the new email, **yi** is the label for the **ith** training email and **k** is the number of nearest neighbors to consider. **Nearest_neighbors (x, k)** returns the indices of the **k** training emails that are nearest to **x**, and **majority_label(**$\mathbf{y_i}$**)** returns the majority label among the labels $\mathbf{y_i}$ for the indices in **nearest_neighbors(x, k).**

KNN was a simple and effective method for spam classification, but it was computationally expensive, as the distance between the new email and all the training emails must be calculated. Furthermore, the effectiveness of KNN may depend heavily on both the selection of the distance metric and the K value.

4.4 Gaussian NB

GNB work on the principle of Bayes theorem, which in our case was classifying email as “spam” and “not spam.” GNB correspond to an email as a feature vector, which holds the values of the predictor variables (also known as features). These characteristics comprised the sender, subject line and text of the email. The likelihood of every feature assumed the “spam” and “not spam” labels was judged via the training data. For instance, the chances that the sender of an email was in the spam directory were considered as the number of emails in the training group with that sender and a label of “spam” separated by the entire amount of emails in the training group with that sender. The possibility of the email being “spam” or “not spam” was estimated using Bayes’ theorem (Eq. 4) [36].

$$P(y|x) = \frac{P(x|y)}{P(x)} \quad (4)$$

In the above equation, x represents the feature vector of email, y refers to the label either spam or not spam, **p(y|x)** are probability values calculated on the basis of $\boldsymbol{y}$ given $\boldsymbol{x}$ features that identify the email belong to spam or not spam, **p(x|y)** is the possibility of $\boldsymbol{x}$ given y to identify the email class and **p(x)** is the likelihood of $\boldsymbol{x}$, which was the likelihood of the email having the specified features in the training set. The label with the higher probability was chosen as the prediction for the email. Given a new email **x** and a training set $\{(\mathbf{x_1}, \mathbf{y_1}), (\mathbf{x_2}, \mathbf{y_2}), ..., (\mathbf{x_n}, \mathbf{y_n})\}$, where $\mathbf{x_i}$ is the feature vector for the $\mathbf{i^{th}}$ training email and $\mathbf{y_i}$ was the label (“spam” or “not spam”) for that email, the prediction for **x** using GNB is (Equation 5):

$$y = \sum \left(P(Y_i) * P\left(x|y_j\right) \text{ for} Y_i in\{\text{spam, not spam}\} \right) \quad (5)$$

In this equation, $\boldsymbol{x}$ is the feature vector for the new email, $\boldsymbol{y_i}$ is the label for the ***ith*** training email, $\boldsymbol{p(y_i)}$ is the prior probability of $\boldsymbol{y_i}$, which was the probability of an email being classified as "spam" or "not spam" in the training set, $\boldsymbol{p(x|y_j)}$ is the probability of $\boldsymbol{x}$ given $\boldsymbol{y_j}$, which was the probability of the email having the given features given that it was "spam" or "not spam" and argmax returns the label with the highest probability. GNB is a fast and simple method for spam classification, but it assumes that the features are independent, which may not always hold true in practice.

4.5 AdaBoost

AdaBoost is an ensemble scheme that chains the calculations of several "weak" classifiers to build a robust "final" classifier. To classify an email as spam or not spam using AdaBoost, the email was represented as a feature vector, which consists of the values of the sender, the subject line, the content of the email and further features of the email. For this purpose, decision tree and random forest classifiers were trained on the training data. The estimation of these two classifiers was combined to form a final prediction using a weighted majority vote. The weight of each training example was updated based on the performance of each of these classifiers. Specifically, the weight of each example was increased if it is misclassified by the existing weak learner and decreased if it is correctly classified. This prediction was represented by the following (Eq. 6):

Given a new email $\boldsymbol{x}$ and a training set $\{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)\}$, where $\boldsymbol{x_i}$ is the feature vector for the ***ith*** training email and $\boldsymbol{y_i}$ is the label ("spam" or "not spam") for that email, the prediction for $\boldsymbol{x}$ using AdaBoost is [37]:

$$Y = \text{sign}\left(\text{sum}\left(\text{alpha}_i \times h_i(x)\right)\right) + b \quad (6)$$

In this equation, $\mathbf{alpha_i}$ is the weight assigned to the ***ith*** decision tree, $\boldsymbol{h_i(x)}$ is the prediction of the ***ith*** random forest for the email $\boldsymbol{x}$ and $\mathbf{sign(x)}$ is the sign of $\boldsymbol{x}$ (i.e., 1 if $x > 0$, -1 if $x < 0$ and 0 if $\boldsymbol{x} = 0$). If the prediction was positive, the email x was classified as "spam." If the prediction was negative or zero, the email x was classified as "not spam."

4.6 Stacking classifier

We proposed the use of a stacking ensemble method for spam classification. In this method, the base classifiers (LR, KNN, DT, Gaussian NB and AdaBoost) make predictions on the training data. The predicted class probabilities from the basis classifiers are joint to create the meta-training dataset ($\text{Meta}_{\text{Train}}$). The meta-classifier (logistic regression) is then trained on the meta-training dataset. For prediction, the base classifiers deliver predicted class probabilities for a new sample, which are then stacked to create the meta-testing data ($\text{Meta}_{\text{Test}}$). The meta-classifier used these meta-testing data for the final prediction (Eqs. 7 & 8).

$$\text{META}_{\text{TRAIN}} = [P_1(X_{\text{TRAIN}}),\ P_2(X_{\text{TRAIN}}),\ \ldots,\ P_M(X_{\text{TRAIN}})] \quad (7)$$

$$\text{METACLASSIFIER}_{\text{TRAIN}}(\text{META}_{\text{TRAIN}},\ Y_{\text{TRAIN}}) \quad (8)$$

In this equation, $p_i(X_{\text{train}})$ represents the predicted class probabilities for the ith base classifier on the training data. $\text{Meta}_{\text{Train}}$ is the stacked predicted class probabilities from all the base classifiers for each training data point. y_{train} represents the true class labels of the training data. $\text{MetaClassifier}_{\text{Train}}()$ trains the meta-classifier (logistic regression) using $\text{Meta}_{\text{Train}}$ and Y_{train} as input (Eqs. 9 & 10).

$$\text{META}_{\text{TEST}} = [P_1(X_{\text{NEW}}),\ P_2(X_{\text{NEW}}),\ \ldots,\ P_M(X_{\text{NEW}})] \quad (9)$$

$$\text{FINAL}_{\text{PREDICTION}} = \text{METACLASSIFIER}_{\text{PREDICT}}(\text{META}_{\text{TEST}}) \quad (10)$$

Here, $\text{Meta}_{\text{Test}}$ is the stacked predicted class probabilities from the base classifiers for a new test. $\text{Final}_{\text{Prediction}}$ is the concluding estimate made by the meta-classifier using $\text{Meta}_{\text{Test}}$ as input. $\text{MetaClassifier}_{\text{Predict}}()$ represents the prediction function of the trained meta-classifier (logistic regression).

5 Results

In our study, we aimed to improve the accuracy of spam email classification by applying various machine learning techniques and using a stacking method. The results of our initial experimentation on base classifiers showed that logistic regression, decision tree and AdaBoost perform well in spam classification task, and accuracy of our base classifiers is plotted in Figure 6. In our study, the precision, recall and F1 score assessment metrics were utilized to evaluate the performance of base and stacked classifiers (Eq. 11) [38].

$$f1 = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \quad \text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}$$
$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}$$
$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{FN} + \text{TN})} \quad (11)$$

Our results exhibited that the stacking technique outperformed all of the base classifiers and achieved the highest

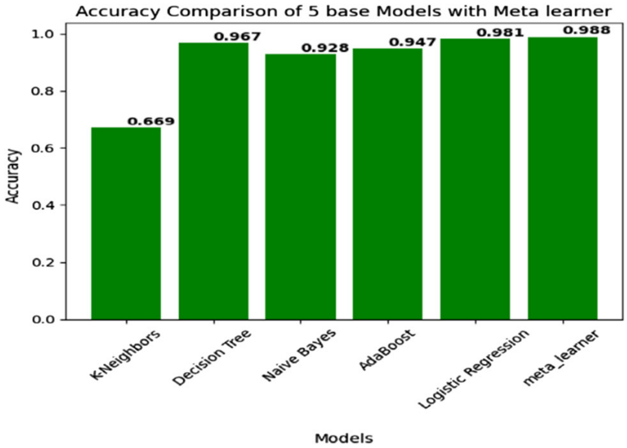


Fig. 5 (Accuracy comparison with meta-classifier)

precision, recall and F1 score. When we contrasted the performance of the individual classifiers and the stacking method, we found that the stacking method consistently outperformed the decision tree, logistic regression and KNN classifiers. The Gaussian naive Bayes and AdaBoost classifiers also performed well, but had slightly lower precision and F1 score than the stacking method. The accuracy comparison as compared with meta-classifier is shown in Figure 5.

Overall, our results suggest that the use of a stacking method can effectively fuse the predictions of multiple basis classifiers to improve the accuracy of spam email classification.

5.1 Performance Comparison

Initially, we perform five base classifiers and compare them with proposed stacking classifier for spam classification through confusion matrix in Table 1.

The results, summarized in Table 2 and Figure 6, show that the logistic regression and the meta-learner models have the highest accuracy of 0.981 and 0.988, respectively. The precision metric measures the proportion of correctly predicted positive instances, and the logistic regression and the meta-learner models have the highest precision scores of 0.972 and 0.988, respectively. The recall metric measures the quantity of genuine positive occurrences that are properly forecast, and the logistic regression and the meta-learner models have the highest recall scores of 0.992 and 0.989, respectively.

The F1-score, which is the harmonic mean of precision and recall, demonstrates that the logistic regression and meta-learner models achieved the highest scores, at 0.982 and 0.989, respectively. On the other hand, the *k*-nearest neighbors' model had low performance across all metrics, with a precision of 0.669, precision of 1.0, recall of 0.359 and F1-score of 0.528. These results suggest that the logistic regression and meta-learner models are well suited for this dataset, whereas the k-nearest neighbors' model is not appropriate for this classification task.

In addition to the performance evaluation of the classifiers on our diversified spam email dataset, additional experiments were also conducted to assess the generalizability and robustness of achieved results. These further experiments included changing the size of the training and test datasets, as well as using multiple and different combinations of the base classifiers in the stacking approach. Our results consistently demonstrated that proposed method outperformed the individual base classifiers and achieved comparatively high classification performance.

5.2 Discussion

To validate our findings, additional experiments were performed to ensure consistency of results. We experimented with three different training set sizes, 50%, 70% and 90% of our dataset. In current case, data are divided randomly into training and testing sets and reevaluated the precision and f1 scores of both individual classifiers and the stacking method. We used two base classifiers in this experiment: decision tree and random forest. Figure 7 Experimentation with training set sizes

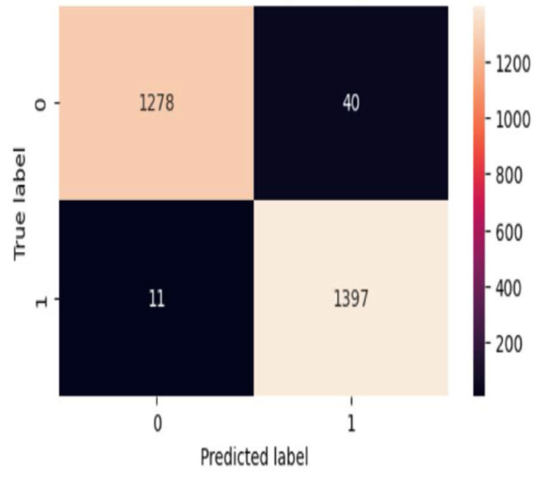
The results in Figure 7 further validated our initial results. Stacking method with different combinations of base classifiers consistently performed better than the individual classifiers on all training set sizes. Stacking method achieved F1 score of 0.92 for 50% training set size, 0.94 for 70% and 0.95 for 90% training set size. Furthermore, F1 score of random forest which performed better than decision tree was 0.84 for the 50% training set size, 0.88 for the 70% training set size and 0.91 for the 90% training set size.

In another experiment, we changed the combinations of base classifiers utilized in the stacking technique. Specifically, we tested with these four combinations: logistic regression with DT, LR with random forest, decision tree with random forest and all three classifiers combined. We used the full dataset for this experimentation.

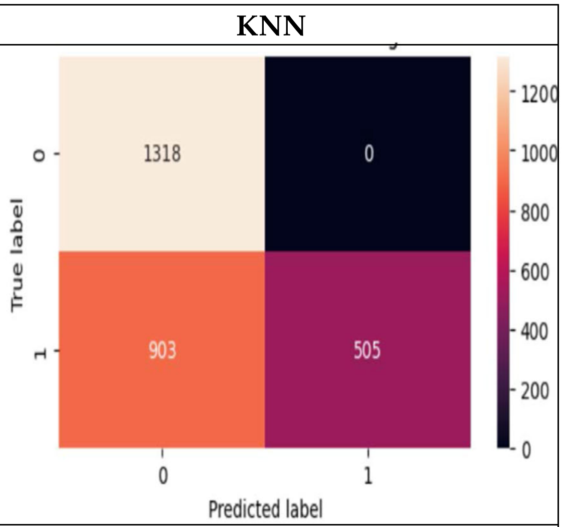
The outcomes indicated that the stacking of all three classifiers combined accomplished the excellent performance, achieving F1 score of 0.95. The other sequences performed somewhat fewer, with F1 scores varying from 0.91 to 0.94. These outcomes indicated that combining multiple base classifiers in the stacking technique can escort to the peak performance gains with additional classification time.

Table 1 Confusion matrixes of used models

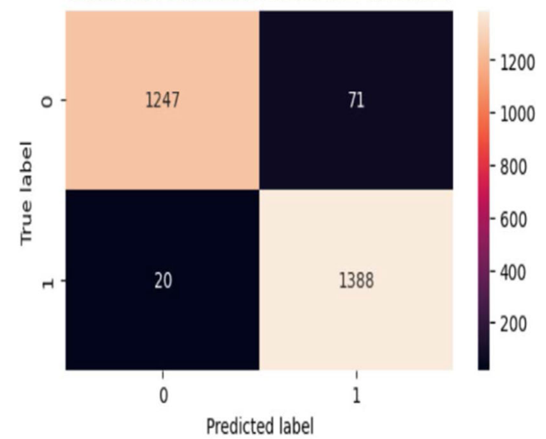
Logistic Regression



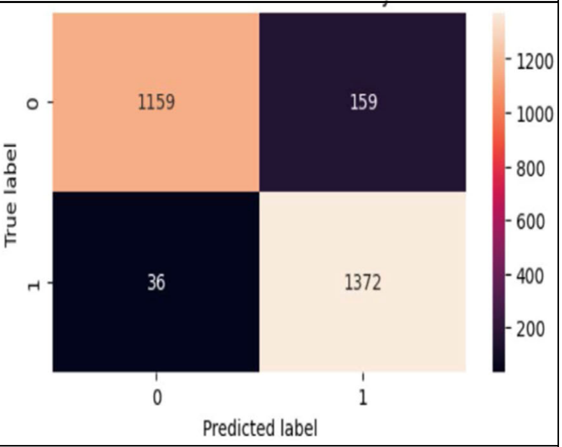
KNN



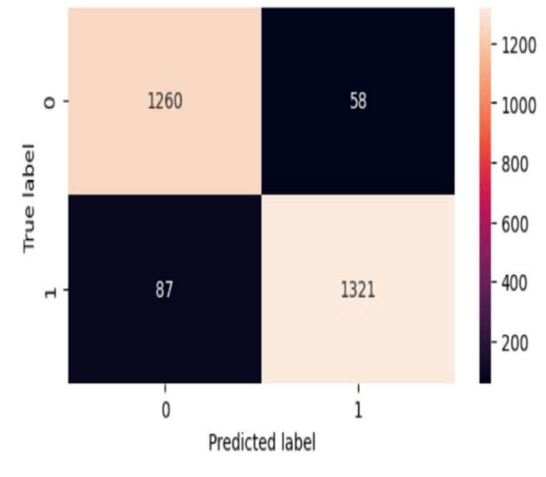
Decision Tree



GNB



Ada Boost



Stacking Classifier

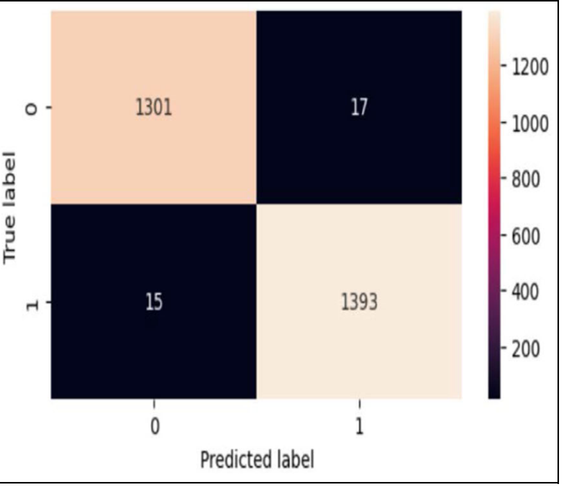


Table 2 Comparison of classifiers

Classifier	Accuracy	Precision	Recall	F1-score	CPU time (total)	Wall time
Naive Bayes	0.928	0.896	0.974	0.933	375 ms	560 ms
Decision tree	0.967	0.951	0.986	0.968	12 s	12.6 s
Logistic regression	0.981	0.972	0.992	0.982	1.45 s	1.5 s
AdaBoost	0.947	0.958	0.938	0.948	1 m 10 s	1 m 14 s
K-Nearest neighbors	0.669	1.000	0.359	0.528	3 m 50 s	1 m 9 s
Meta-learner	0.988	0.988	0.989	0.989	656 ms	654 ms

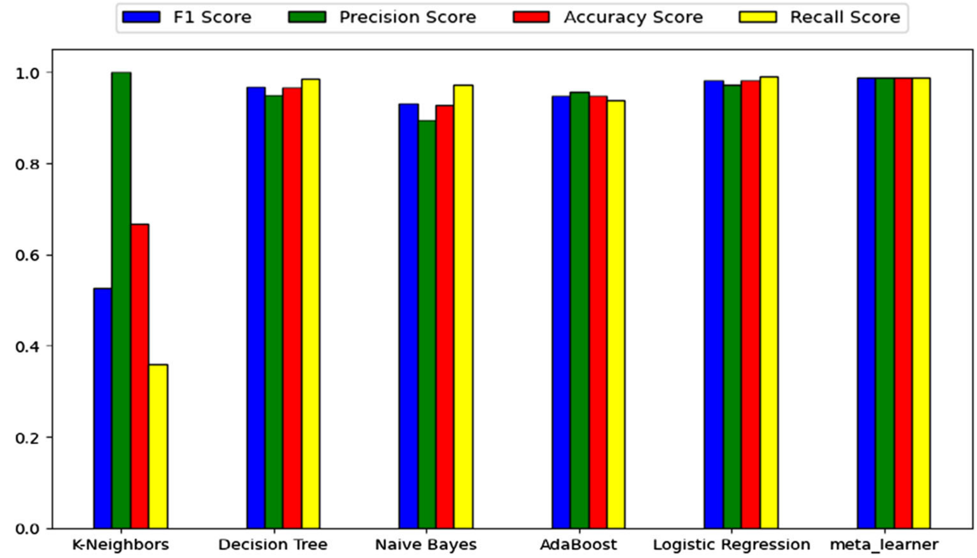


Fig. 6 (Overall performance of base classifiers and meta-learner)

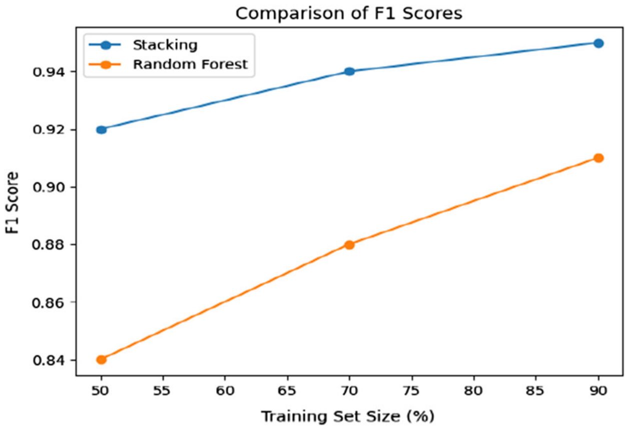


Fig. 7 F1 scores of additional experimentation

6 Conclusion

The current proposed scheme improved the spam email classification precision substantially as shown in the result section using the ensemble ML algorithms. Based on the performed experiment, it is easily judged that by merging the output of multiple base classifiers can lead to better precision, recall and F1 scores. Generally, our results indicate that stacking technique could be a favorable technique for increasing the correctness of spam email classification in real applications. Additional research and development are needed to strengthen these outcomes and discover additional benefits of using stacking method in numerous other classification applications.

Stacking approach enhanced the performance by letting them to concentrate on several attributes of the dataset. For example, one base classifier alone was not good at identifying spam emails, while another base classifier was not good at identifying spam emails with certain keywords in the subject line. By combining their predictions, the stacking ensemble potentially captured a wider range of features that are relevant for spam classification and offer much more accurate classification.

In future, datasets that include both images and personalized email content may offer promising results. First of all, we need to collect and compile relevant data to solve this challenge. Another future work we consider is classification of spam emails after being passed through email warmer tools. These tools tend to fool the email server or algorithm by sending a series of fake emails to email addresses in order to establish a positive sending reputation with the email providers. The tool can then increase the likelihood that future emails by the sender will not be marked as spam. Such emails may also affect the accuracy of spam email classifiers. Dataset of such emails is required for further analysis and scientific study.

Author contributions When writing your statement, please specify who did what. All people work together wrote the main manuscript text and E.F. prepared figures All authors reviewed the manuscript."

Funding Open access funding provided by UiT The Arctic University of Norway (incl University Hospital of North Norway).

Data availability Not applicable.

Declarations

Cnflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Human Participants and/or Animals Not applicable.

References

1. Pfleeger, S.L., Bloom, G.: Canning spam: proposed solutions to unwanted email. IEEE Secur. Priv. **3**(2), 40–47 (2005)
2. Grier, C., Thomas, K., Paxson, V., & Zhang, M. (2010, October). @ spam: the underground on 140 characters or less. in Proceedings of the 17th ACM conference on Computer and communications security (pp. 27–37)
3. Agarwal, D.K., Kumar, R.: Spam filtering using SVM with different kernel functions. Int. J. Comput. Appl. **136**(5), 16–23 (2016)
4. Heartfield, R., Loukas, G.: A taxonomy of attacks and a survey of defence mechanisms for semantic social engineering attacks. ACM Comput. Surv. (CSUR) **48**(3), 1–39 (2015)
5. John, J. P., Moshchuk, A., Gribble, S. D., & Krishnamurthy, A.: Studying spamming botnets using botlab. in NSDI (Vol. 9, No. 2009) (2009, April)
6. Kumar, N., & Sonowal, S.: Email spam detection using machine learning algorithms. in 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 108–113). IEEE. (2020)
7. Junnarkar, A., Adhikari, S., Fagania, J., Chimurkar, P., & Karia, D.: 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) (pp. 693–699). IEEE. (2021, February)
8. Awad, W.A., ELseuofi, S.M.: Machine learning methods for spam e-mail classification. Int. J. Comput. Sci. Inf. Technol. (IJCSIT) **3**(1), 173–184 (2011)
9. Zhang, F., Chan, P.P., Biggio, B., Yeung, D.S., Roli, F.: Adversarial feature selection against evasion attacks. IEEE Trans. Cybern. **46**(3), 766–777 (2015)
10. Shaukat, K., Luo, S., Chen, S., & Liu, D.: Cyber threat detection using machine learning techniques: A performance evaluation perspective. in 2020 international conference on cyber warfare and security (ICCWS) (pp. 1–6). IEEE. (2020, October)
11. Garavand, A., Salehnasab, C., Behmanesh, A., Aslani, N., Zadeh, A.H., Ghaderzadeh, M.: Efficient model for coronary artery disease diagnosis: a comparative study of several machine learning algorithms. J. Healthc. Eng. (2022). https://doi.org/10.1155/2022/5359540
12. Ghaderzadeh, M., Aria, M., Asadi, F.: X-ray equipped with artificial intelligence: changing the COVID-19 diagnostic paradigm during the pandemic. BioMed Res. Int. (2021). https://doi.org/10.1155/2021/9942873
13. Hajek, P., Barushka, A., Munk, M.: Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining. Neural Comput. Appl. **32**, 17259–17274 (2020)
14. Ramanathan, V., Wechsler, H.: Phishing detection and impersonated entity discovery using conditional random field and latent Dirichlet allocation. Comput. Secur. **34**, 123–139 (2013)
15. Ghourabi, A., Mahmood, M.A., Alzubi, Q.M.: A hybrid CNN-LSTM model for SMS spam detection in arabic and english messages. Future Internet **12**(9), 156 (2020)
16. Madhavan, M. V., Pande, S., Umekar, P., Mahore, T., & Kalyankar, D.: Comparative analysis of detection of email spam with the aid of machine learning approaches. in IOP conference series: materials science and engineering (Vol. 1022, No. 1, p. 012113). IOP Publishing. (2021)
17. Rayan, A.: Analysis of e-mail spam detection using a novel machine learning-based hybrid bagging technique. Comput. Intell. Neurosci. (2022). https://doi.org/10.1155/2022/2500772
18. Suborna, A.K., Saha, S., Roy, C., Sarkar, S., & Siddique, M.T.H.: An approach to improve the accuracy of detecting spam in online reviews. in 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD) (pp. 296–299). IEEE. (2021, February)
19. Frías-Blanco, I., Verdecia-Cabrera, A., Ortiz-Díaz, A., & Carvalho, A.: Fast adaptive stacking of ensembles. in Proceedings of the 31st Annual ACM Symposium on Applied Computing (pp. 929–934). (2016, April)
20. El-Kareem, A., Elshenawy, A., Elrfaey, F.: Mail spam detection using stacking classification. J. Al-Azhar Univ. Eng. Sector **12**(45), 1242–1255 (2017)
21. Madichetty, S.: A stacked convolutional neural network for detecting the resource tweets during a disaster. Multimed. Tools Appl. **80**, 3927–3949 (2021)
22. Oh, H.: A YouTube spam comments detection scheme using cascaded ensemble machine learning model. IEEE Access **9**, 144121–144128 (2021)
23. Zhao, C., Xin, Y., Li, X., Yang, Y., Chen, Y.: A heterogeneous ensemble learning framework for spam detection in social networks with imbalanced data. Appl. Sci. **10**(3), 936 (2020)
24. Liu, S., Wang, Y., Zhang, J., Chen, C., Xiang, Y.: Addressing the class imbalance problem in twitter spam detection using ensemble learning. Comput. Secur. **69**, 35–49 (2017)
25. Omotehinwa, T.O., Oyewola, D.O.: Hyperparameter optimization of ensemble models for spam email detection. Appl. Sci. **13**(3), 1971 (2023)

26. Sahu, K., Alzahrani, F.A., Srivastava, R.K., Kumar, R.: Evaluating the impact of prediction techniques: software reliability perspective. Comput., Mater. Contin. (2021). https://doi.org/10.32604/cmc.2021.014868
27. Sahu, K., Srivastava, R.K.: Needs and importance of reliability prediction: an industrial perspective. Inf. Sci. Lett. **9**(1), 33–37 (2020)
28. Sahu, K., Srivastava, R.K.: Soft computing approach for prediction of software reliability. Neural Netw. **17**, 19 (2018)
29. Apache Spam Assassin. (2022, November 22) https://spamassassin.apache.org/old/publiccorpus/
30. Enron Corp & Cohen, W. W. (2015) *Enron Email Dataset*. United States Federal Energy Regulatory Commissioniler, comp [Philadelphia, PA: William W. Cohen, MLD, CMU] [Software, E-Resource] Retrieved from the Library of Congress, https://www.loc.gov/item/2018487913/.
31. Scikit-Learn (2022, November 23) https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html#sklearn.feature_extraction.text.TfidfTransformer.
32. Dedeturk, Bilge & Akay, Bahriye. (2020). Spam filtering using a logistic regression model trained by an artificial bee colony algorithm. Applied Soft Computing. 91. 106229. https://doi.org/10.1016/j.asoc.2020.106229.
33. Kumar, P., Biswas, M.: SVM based image spam detection using kernels: linear, polynomial, RBF, and sigmoid. Int. J. Comput. Sci. Appl. **14**(2), 79–96 (2017)
34. Dedeturk, B.K., Akay, B.: Spam filtering using a logistic regression model trained by an artificial bee colony algorithm. Appl. Soft Comput. **91**, 106229 (2020)
35. Herrera, V.M., Khoshgoftaar, T.M., Villanustre, F., Furht, B.: Random forest implementation and optimization for Big Data analytics on LexisNexis's high performance computing cluster platform. J. Big Data **6**(1), 1–36 (2019)
36. Murphy, K.P.: Machine learning: a probabilistic perspective. MIT press, London (2012)
37. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
38. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Inf. Process. Manage. **45**(4), 427–437 (2009)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.