




# Detecting ham and spam emails using feature union and supervised machine learning models

Furqan Rustam<sup>1</sup> · Najia Saher<sup>2</sup> · Arif Mehmood<sup>2</sup> · Ernesto Lee<sup>3</sup> · Sandrilla Washington<sup>4</sup> · Imran Ashraf<sup>5</sup> 

Received: 16 April 2022 / Revised: 15 June 2022 / Accepted: 5 February 2023 /

Published online: 8 March 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Spam emails are cyber nuisances that cause serious security threats including personal and financial information. Although several spam detection approaches exist, detecting new strains of spam messages is challenging that requires a reliable and efficient intelligent spam email detection approach. This study utilizes features from the text of emails to determine whether it is spam or normal. Multiple features are combined to obtain a higher accuracy for spam email detection. Experiments involve machine learning and deep learning models and the influence of data resampling is also investigated. Performance analysis is done using F1 score, recall, precision, and accuracy, as well as comparison with state-of-the-art approaches. Random forest and logistic regression achieve the highest accuracy scores 0.991 and 0.990, respectively which is much better than existing models.

**Keywords** Spam detection · Features extraction · Machine learning classifiers · Term frequency · Sampling

## 1 Introduction

Internet users are exposed to several threats including personal and financial information theft, damage to sensitive information stored on a computer, ransom demands, unauthorized online purchases, etc. The users are prone to these and similar other threats where the

---

✉ Imran Ashraf  
imranashraf@ynu.ac.kr

<sup>1</sup> School of Computer Science, University College Dublin, D04 V1W8 Dublin, Ireland

<sup>2</sup> Department of CS and IT, The Islamia University of Bahawalpur, Bahawalpur, 63100, Pakistan

<sup>3</sup> College of Engineering and Technology, Miami Dade College, Miami, FL 33132, USA

<sup>4</sup> Department of Computer and Information Sciences, Spelman College, Atlanta, GA, USA

<sup>5</sup> Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, 38544, South Korea

attacker uses computer viruses, spam messages to obtain the user's private information, ransomware, and similar other tools. Spam messages often contained in e-mails have become a frequent tool for stealing users' information. Aiming at stealing financial information, such e-mails contain malware files, invitations, and uniform resource locator (URL) links that lead to various malware-hosting and phishing websites. Over the past few years, spam emails have been increased substantially, as reported in [2] which indicates that the phishing e-mails for 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> quarters of 2020 are 118260, 132553, and 139685, respectively.

Spamming indicates any fraudulent activity that targets the financial and personal information of internet users and involves social engineering and similar other concepts. Spam emails are designed to show that they are from genuine and registered companies which the user may be using. The idea is to lure the user to click the provided link for further information or verification. Once the user clicks the provided link, the information can be gathered by the attacker. Such attacks can be detected using available programs and models to a certain extent, yet, change in the design and strategy of such attacks makes the detection more difficult and complex for the available whitelist or blacklist-based techniques. The classification accuracy of such techniques is reduced over time if they are not updated [13] as the strategy and structure of such attacks have evolved. Similarly, a large number of auto-generated emails makes it a time-consuming process and further increases the complexity of spam email detection. Research indicates that of the 205 billion emails sent every day, approximately 22.8% are unnecessary and 18.5% are irrelevant [19].

Automated systems are developed by the researchers for different purposes such as spam email detection [17, 30], health care systems [6–8], anomaly detection [1, 14], etc. This study also contributes to spam email detection using machine learning techniques. Electronic mail (e-mail) has become the most common source for spammers to steal sensitive information [10] and developing an automatic system to detect spam email is very important to safeguard individuals and companies alike. Despite the availability of several spam detection techniques, the provided accuracy is not up to the standard. Furthermore, predominantly these techniques require longer training time and the false positive rate is high. Devising an approach that can detect spam emails before they are opened is critical. The available solutions do not possess this capability despite being sophisticated and adaptive. This research aims to solve this problem by employing machine learning algorithms and various feature extraction techniques. This study introduces an approach for spam e-mail detection and makes the following contributions

- This study proposes an approach for spam e-mail detection using features from textual data. Two important feature extraction techniques are investigated in this regard.
- Besides using term frequency-inverse document frequency (TF-IDF) and bag of words (BoW), an intuitive feature extraction approach, feature union, is introduced that combines TF-IDF and BoW to make an effective feature set.
- To resolve data imbalance, experiments are performed with under-sampling, and results are analyzed to investigate the impact of undersampling on the performance of the machine and deep learning models.
- Several machine learning models are employed for this purpose including random forest (RF), gradient boosting machine (GBM), support vector machines (SVM), Gaussian naïve Bayes (GNB), and logistic regression (LR). The performance of machine learning models is enhanced by optimizing several hyperparameters. Deep learning models such

as long short term memory (LSTM) and gated recurrent unit (GRU) are also adopted for spam email detection.

- Extensive experiments are carried out and performance is evaluated using accuracy, precision, recall, F1 score, and micro average. In addition, the performance is compared with several state-of-the-art models.

The rest of the paper is organized as follows. Section 2 discusses important research works related to the current study. The proposed approach, dataset used for experiments, machine learning models, and sampling approaches are given in Section 3. Section 4 provides results and discussions while the conclusion is given in Section 5.

## 2 Related work

Spam emails contain advertising messages, as well as, URLs and file attachments for stealing the personal and financial information of the users. The advertising emails are considered to be legal as long as the content is not fraudulent; these can be considered spam only if the emails contain any unsolicited content [15]. Spammers frequently work to discover techniques to make a spam email look legitimate to dodge email filters. One of the major problems is that spam has different forms that can be considered a legitimate message [10]. Due to the importance of spam detection, a large number of research works can be found in the literature. Both machine learning and deep learning approaches have been adopted for spam classification. To entertain the required need, some spam-related studies have been discussed in this section.

Various machine learning techniques based on spam detection have been used by researchers. Specific keyword pattern in emails for spam detection is used in most of the existing statistical models. For example, [9] explored the major characteristics of spam by reviewing the content-based spam detection techniques. Both statistical and non-statistical methods are used for spam detection, however, the statistical approaches appear to be more effective. At first, the SMS spam collection dataset is collected for training and classification. Later, classification is done using the decision tree (DT), LR, and k-nearest neighbor (KNN). Results show that LR outperforms with the highest accuracy of 99%. Francisco et al. [17] proposed hierarchical clustering and a combination of supervised learning for spam detection. The clustering algorithm is used to generate SPEMC-11K (Spam Email Classification) and emails are categorized into multiple classes. The obtained dataset consists of three distinct classes including health and technology, sexual content, and personal scams. Moreover, various combinations of TF-IDF and bag of words (BoW) feature embedding are applied. Spam emails are classified through SVM, LR, and Naïve Bayes. Results indicate that the NB with TF-IDF has the best classification speed and SVM combined with TF-IDF outperforms all the other combinations with the highest accuracy of 95.39%.

The study [5] used the spam base UCI dataset for spam classification using ten state-of-the-art classifiers. Similarly, infinite latent feature selection (ILFS) is employed to select the most relevant features from the dataset. 10-fold cross-validation is used for SVM, radial bases function (RBF), decision table (DT), Bayes net (BN), KNN, NB, random tree (RT), LR, ANN, and RF. RF tends to show superior performance by achieving 95.45% accuracy. The authors propose a framework that uses S-Cuckoo and hybrid kernel-based SVM for email spam classification in [24]. Both text and image features are extracted from emails

where TF features are used for text data, and Correlograms and wavelet moments for image data. The HKSVM model is designed by combining three different kernel functions to form a hybrid function that achieves an accuracy score of 95%. A comparative study based on data mining techniques used Fisher filtering (FF), Relief-F, stepwise discriminant analysis (StepDisc), and runs filtering techniques for feature selection [23]. The classifiers including random Tree, LDA, MLP, NB, KNN, SVM, and LR-Trials are applied for spam classification. The combination that outperformed all the employed methods is RF Tree which achieved 99% accuracy when applied with the FF technique.

An NB approach for spam classification is performed in [30] where NB has been applied on two different datasets UCI spam base dataset and Spam data. The UCI spam base dataset is used to train the model while the performance is tested on the Spam data. Results show that the number of instances of the dataset and the type of email has an impact on the performance of NB and the classifier achieves an accuracy score of 91.13%. The study [36] pursued various machine learning methods to make a hybrid model for enhancing spam classification accuracy. Feature selection has been performed by information gain, Chi-square, and gain ratio methods. The hybrid classifier uses a stacking method and builds a Meta learner to make the prediction-based Meta classifier. The applied hybrid classifier involves various combinations of sequential minimal optimization, SVM, NB, and J48 from decision tree algorithms. The best accuracy score of 93.22% is achieved by using J48 and NB with J48 as the Meta classifier.

The use of artificial neural networks (ANN) is reported to show better performance than traditional machine learning models in [12]. ANN is used with backpropagation (BP) and the combination of backpropagation with momentum (BP+M) on the UCI spam base dataset [33]. The BP+M optimized ANN shows better performance with an accuracy score of 95.38% with less training time. The authors utilize a feature-centric spam email detection model (FSEDM) with novel and existing features in [35]. Several sets of features are used including user-based, content, semantic, sentiment, and spam lexicons. Sentiment features are used along with the proposed features to perform the classification. The feature selection is performed through information gain, Relief-F, and gain ratio methods. For classification, SVM, bagging, RF, AdaBoost, DNN, J48, and MLP have been used where DNN shows the best performance with an accuracy of 97.2% when applied with sentiment features.

Similarly, the study [32] focuses on using a convolutional neural network (CNN) approach for spam classification. For email classification containing both text and image data, a hybrid multimodal architecture is proposed containing one CNN each for text and image. GloVe word embedding is used for multi-modal feature fusion, and a multi-modal learned rule is proposed for spam detection. The achieved accuracy is 98.11% using the Enron spam dataset. An ANN is used with radial basis function neural networks (RBFNN) to classify spam e-mails in [4]. The approach combines particle swarm optimization (PSO) algorithm with RBFNN for spam detection. The PSO algorithm is used to optimize the appropriate position  $c$  for the applied model, the singular value decomposition algorithm is used to optimize weights  $w$  and the radii  $r$  is optimized by using KNN. Experiments conducted on the UCI spam base dataset show 91.4% accuracy.

Despite the availability of sophisticated spam detection approaches, the provided accuracy is not up to the standard. In addition, existing approaches are not adaptable and robust. A comprehensive summary of the discussed research work is presented in Table 1. This research aims to fill this gap by introducing an effective approach for spam classification with high accuracy.

**Table 1** Review of the discussed research works

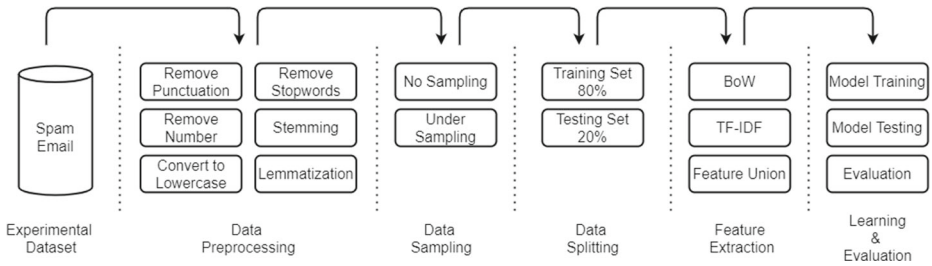
Ref.	Dataset	Models	Accuracy
Jánez-Martino et al. [17]	English spam emails INCIBE	BoW and TF-IDF with LR, SVM and NB	95.39
Bassiouni et al. [5]	UCI spam base dataset	ILFS with SVM, Radial Bases Function, Decision Table, Bayes Net, KNN, NB, Random Tree, LR, RF, and ANN	95.45
Kumaresan et al. [24]	Ling-spam and Spam-archive dataset	A hybrid approach HKSVM using text and visual features	95
Kumar et al. [23]	UCI spam base dataset	Fisher Filtering, Relief-F, StepDisc, 99 and Runs Filtering along with KNN, SVM, LR-Trirls, Rnd Tree, LDA, MLP, NB	99
Rusland et al. [30]	Usenet Spam Data and UCI spam base dataset	NB	91.13
ZhiWei et al. [36]	UCI spam base dataset	Metaclassifiers stacking through feature selection methods; GR, IG, and Chi-square and SVM, NB, and J48	93.22
GuangJun et al. [12]	SMS spam collection dataset	DT, LR, KNN	99.0
Sinha et al. [33]	UCI spam base dataset	ANN with BP and BP+M	95.38
Zamir et al. [35]	CSDMC2010 spam email dataset	User-based, lexicon-based, and sentiment-based features with SVM, bagging, RF, AdaBoost, DNN, J48, and MLP	97.2
Seth and Biswas [32]	Enron Spam Dataset	CNN based hybrid multimodal architecture	98.11

### 3 Materials and methods

The proposed methodology and its working mechanism are discussed here comprising dataset description, preprocessing steps followed for noise removal, feature extraction approaches, and a brief description of machine learning models used in this study.

#### 3.1 Proposed methodology

Figure 1 shows the flow of the adopted methodology for ham and spam email classification. Machine learning techniques are used to solve the e-mail classification problem. The proposed approach involves data collection, data preprocessing, feature extraction, model training, and model evaluation techniques. Following this approach, data is first collected and cleaned using a sequence of preprocessing steps. First, numbers and punctuation is removed, followed by case conversion and stemming. In the end, stop words are removed to clean the data. This process ensures feature space reduction and improves the learning process of the machine learning models. Later feature extraction techniques are applied to extract the features from the cleaned data. Finally, the machine learning models are trained on these extracted features, and the test data is used to evaluate the trained models. New data is fed to the trained models to classify as spam and ham e-mails.



**Fig. 1** Flow of the proposed methodology

### 3.1.1 Dataset description

This study considers two datasets to conduct experiments for spam email detection. Owing to the need for a large dataset for experiments, both datasets are combined into a single dataset. Both datasets are obtained from the Kaggle, although the sources are different; Dataset 1 ‘Spam or Ham - EMP Week 2 ML HW Dataset is acquired from Kaggle [22] and Dataset 2 ‘Spam filter is also acquired from Kaggle [34]. Both datasets contain two classes, one for ‘Spam’ and the other for ‘Ham’ emails. The distribution of the number of records of each dataset is provided in Table 2 and a few samples from both datasets are shown in Table 3.

### 3.1.2 Preprocessing

This study uses several preprocessing steps such as number removal, punctuation, and stop-words removal, conversion to lower case, stemming, and lemmatization. Preprocessing is important in this study because emails contain a lot of unnecessary raw text that can influence the models’ performance, so the removal of raw data will help to reduce the complexity of the feature set.

- **Punctuation & number removal:** Emails contain punctuation and numbers which are not useful features for model training; so removing them helps to reduce complexity in features. Regular expressions are used in this step.
- **Convert to lowercase:** This step is very important to reduce redundancy in the feature set as the email contains words in upper and lower cases, such as ‘hello’, and ‘Hello’. Such words are written following the language rules and they are the same for a human reader. However, feature extraction methods consider them two different words and treat them separately which increases the size of the feature space. So converting all text to lowercase will reduce the complexity of feature space and help to improve the performance of machine learning models. Case conversion is performed using the natural language tool kit (NLTK) library of Python.

**Table 2** Number of records in datasets

Dataset	Source	Spam	Ham	Total
Dataset 1	Kaggle [22]	1,368	4,358	5,726
Dataset 2	Kaggle [34]	747	4,825	5,572
Total	-	2115	9183	11298

**Table 3** Sample from both datasets

Text	Target
	Dataset 1
Subject: naturally irresistible your corporate identity It is really hard to recollect a company : ...	1
Subject: the stock trading gunslinger fanny is merrill but muzo not colza attainer and penultimate...	1
Subject: fw : california electricity crisis : what to do for your reading enjoyment . we should sh...	0
Last chance 2 claim ur £150 worth of discount vouchers-Text YES to 85023 now!SavaMob-member offers ...	1
You still around? Looking to pick up later	0
Hi the way I was with u 2day, is the normal way&this is the real me. UR unique&I hope I know u 4 the...	0

- **Stop words removal:** Text contains several stop words such as ‘a’, ‘the’, ‘an’, ‘is’, and ‘are’, etc., which are used to clarify the meaning of a sentence. However, stop words are not important for the training of the machine learning models. Instead, they increase the complexity of the feature vector, and the models’ performance is affected. So, stop words are removed to elevate the performance.
- **Stemming and lemmatization:** Both techniques are used to get the basic/root form of words as several variations of a word may be used in sentences, such as ‘gone’, ‘going’, and ‘goes’. Although, these are the extended form of the word, ‘go’, during the feature extraction, such words are treated as unique words, and their features are extracted separately which increases the feature vector complexity. Stemming and lemmatization are used to transform the extended form of the words to their root form. Stemming simply removes the ‘s’ or ‘es’ at the end of words and causes spelling mistakes or wrong words. Lemmatization is more appropriate as it considers the context in which a word is used and changes it into the proper base form. This study uses the NLTK Porter Stemmer and Word Net Lemmatizer libraries for experiments.

### 3.1.3 Feature union

The dataset which is used to train machine learning algorithms is small. That is the reason the feature set is also small which makes model training inefficient. To resolve this problem, we propose a feature union approach in which we combine two features to generate a large feature set which helps to improve the model performance. Features union is the combination of TF-IDF features. The BoW is a simple term count technique that often produces good results when the dataset is large and complex. TF-IDF is a weighted feature extraction technique that computes the weight of each term in the corpus. TF-IDF can be a good choice for models that require a large feature set. Feature union combines both TF-IDF and

BoW features and gives a large feature set which can be good for machine learning models, especially when only a small dataset is available.

$$TF - IDF = tf_{t,d} * \log \left( \frac{N}{D_t} \right) \tag{1}$$

where  $tf_{t,d}$  is the frequency of term  $t$  in document  $d$  and  $N$  is number of documents while  $D_t$  is number documents that contain term  $t$ .

TF counts the number of occurrences of each unique term in a given document, resulting in higher values of more common terms. IDF, on the other hand, considers rare terms more important and assigns higher weights to those terms which appear less often. For feature union, TF-IDF and BoW features are considered as follows

$$TF - IDF_{features} = \begin{pmatrix} TFIDF_{11} & TFIDF_{12} & \dots & TFIDF_{1q} \\ TFIDF_{21} & TFIDF_{22} & \dots & TFIDF_{2q} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ TFIDF_{p1} & TFIDF_{p2} & \dots & TFIDF_{pxq} \end{pmatrix} \tag{2}$$

$$BoW_{features} = \begin{pmatrix} BoW_{11} & BoW_{12} & \dots & BoW_{1n} \\ BoW_{21} & BoW_{22} & \dots & BoW_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ BoW_{m1} & BoW_{m2} & \dots & BoW_{mxn} \end{pmatrix} \tag{3}$$

The combination of weighted and simple term count features can improve the performance of learning models. The mathematical representation of feature union is shown in (4).

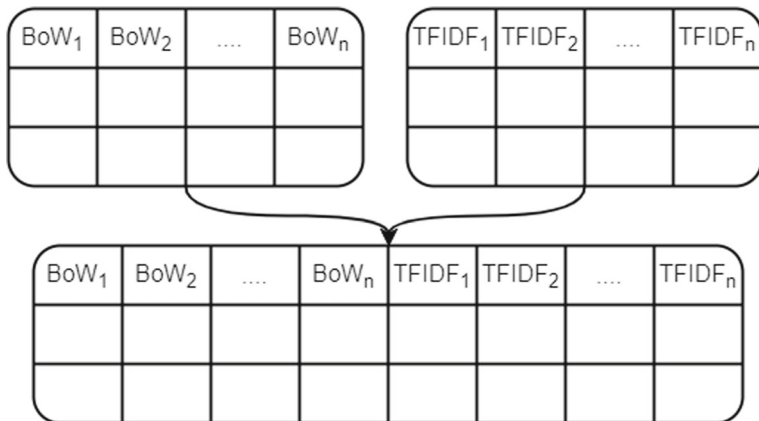
$$Feature\ Union = \begin{pmatrix} BoW_{11} & BoW_{12} & \dots & BoW_{1n} & TFIDF_{11} & TFIDF_{12} & \dots & TFIDF_{1q} \\ BoW_{21} & BoW_{22} & \dots & BoW_{2n} & TFIDF_{21} & TFIDF_{22} & \dots & TFIDF_{2q} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ BoW_{m1} & BoW_{m2} & \dots & BoW_{mxn} & TFIDF_{p1} & TFIDF_{p2} & \dots & TFIDF_{pxq} \end{pmatrix}_{ixj} \tag{4}$$

where, (3) and (2) show TF-IDF and BoW matrix, and (4) shows the feature union which is combination of BoW and TF-IDF. In (4),  $m = p = i$  and  $n + q = j$ . The illustration of feature union shown in Fig. 2.

### 3.1.4 Under-sampling approach

This study performs under-sampling to mitigate the influence of model over-fitting. For under-sampling, random under-sampling is used where the extracted data from the majority class is made almost equal to the minority class. The majority and minority classes for this study are ‘ham’ and ‘spam’, respectively. For obtaining a more balanced data distribution,





**Fig. 2** Schematic diagram for feature fusion

data re-sampling has been a useful strategy and is adopted by many researchers. The random under-sampling approach randomly discards the samples in the training data from the majority class, ‘am’, until a balanced distribution of majority and minority class is reached. The ratio of data after re-sampling is shown in Table 4.

### 3.1.5 Supervised machine learning models

Several machine learning models are used in this study for the classification of ‘spam’ and ‘ham’ emails such as GBM, SVM, GNB, RF, and LR. Each model is implemented with BoW, TF-IDF, and the derived fused feature approach. Models are optimized to obtain better performance using a set of hyperparameters as given in Table 5. For clarification and completeness, a short description of machine learning models is provided in Table 6.

For a better performance appraisal, this study also adopts several deep learning models to classify emails into ham and spam. For this purpose, LSTM and GRU models are used with the best architectures. Both are recurrent neural networks application and work better on text data [27].

The architecture of both models is shown in Table 7. Both models take inputs through the embedding layer which consists of three parameters. One vocabulary size which is 5000 in our case, the second output dimension is 100, and the third is the input length. Vocabulary size defines how much bigger value can be the input for learning models [26]. Both models consist of dropout layers which help to reduce the complexity of models by drooping the neurons from models randomly [25]. LSTM and GRU both are used with 100 units. In the end, models are compiled with the ‘binary\_crossentropy’ loss function because of the binary classification problem and ‘Adam’ optimizer [29]. The batch size is set to 32, while the models are fitted using 100 epochs.

**Table 4** Number of records in datasets

Dataset	Spam	Ham	Total
Original	2115	9183	11298
Under-sampling	2115	2115	4230

**Table 5** List of hyperparameters and their used values for experiments

Model	Hyperparameters	Tuning Range
RF	n_estimators= 200; max_depth=200	n_estimators= 50 to 500; max_depth=50 to 500;
GBM	n_estimators= 200; max_depth=200; learning_rate=0.2	n_estimators= 50 to 500; max_depth=50 to 500; learning_rate=0.1 to 0.8
SVM	kernel= linear; C=3.0	kernel= {linear, poly}; C=1.0 to 5.0
GNB	var_smoothing= 1e-9	var_smoothing= 1e-9
LR	solver= liblinear; C=3.0	solver= {liblinear, saga, sag}; C=1.0 to 5.0

## 4 Results and discussions

This section contains the results of machine learning and deep learning models for spam email classification. The performance is evaluated in terms of accuracy, precision, recall, and F1 score.

### 4.1 Results of machine learning models without re-sampling

Machine learning models are implemented using BoW, TF-IDF, and feature union techniques separately on the original dataset. The original dataset is imbalanced so the models may experience overfitting with respect to the majority class. In that case, accuracy is not a preferable metric for performance evaluation, instead, an F1 score is used.

The results of machine learning models using TF-IDF features are shown in Table 8. The performance of SVM and LR is significant in terms of F1 scores as they achieve 0.95 and 0.94 F1 scores, respectively. In terms of accuracy, SVM is best with a 0.983 accuracy score. The significant performance of SVM and LR is because of the sparse feature set because text data generates a large feature set which is good for SVM and LR. The accuracy score and F1 score have high variation as RF achieves 0.973 accuracy but the F1 score is 0.92 which shows the impact of the data imbalance. RF and GBM show poor performance with respect to SVM and LR when the F1 score is considered while GNB has the worst performance with a 0.61 F1 score.

Experimental results using BoW features are provided in Table 9. Models show almost similar performance with BoW features. SVM is still the best performer with a 0.94 F1 score which is 1% low as compared to its score with TF-IDF features. The primary rationale for that is the difference in feature vector; TF-IDF provides weighted features as compared to BoW which gives only term counts. The BoW can be more suitable for tree-based models and probability-based models because of rule-based prediction. In the same fashion, linear models perform better when used with weighted features. GNB is still the worst performer with a 0.62 F1 score.

Table 10 shows the results of machine learning models when trained on feature union. The performance of all models has been improved with feature union as compared to BoW and TF-IDF alone. SVM, LR, and GBM achieve a 0.95 F1 score while RF improves its F1 score to 0.94. GNB achieves the highest F1 score on the original dataset with feature union which is 0.63. This significant performance of machine learning with feature union is because of the large feature set. Feature union generates a feature combination of weighted and simple terms which can be good for both linear and tree-based models. Thus, feature union leads to improved performance.

**Table 6** Brief description of machine learning models used in this study

Model	Description
RF	RF is a tree-based ensemble model used for classification and regression tasks [31]. RF combines a number of decision trees under majority voting criteria. Each individual tree makes its prediction and then RF performs the voting between these predictions to make the final prediction. RF uses bootstrapping in bagging which is more effective on the imbalanced dataset.
GBM	GBM is also an ensemble model used for both classification and regression tasks. GBM uses boosting algorithm which combines several base learners to reduce the prediction error. Boosting algorithm can be a good choice for small datasets and this study primarily uses it on that grounds. Learning rate in GBM can help to get the best optimization to enhance the performance even on small datasets.
SVM	SVM is a linear model that uses the concept of ‘hyperplanes’ to perform classification. The accuracy of the model depends on the hyperplanes’ accuracy [18]. Hyperplanes separate the data with the best margin between the samples of different classes. There can be several hyperplanes and optimization aims at choosing the hyperplanes that maximize the margin between the class boundaries. This study uses a linear kernel with SVM, as the dataset is linear separable.
GNB	GNB is a variant of Naive Bayes which is a probability-based model [28]. GNB utilizes the Bayesian Theorem and predicts the probability of each case. It is a simple model and can work on both continuous and discrete data. It does not require large training data and often performs well with small datasets. GNB is good for the dataset used in this study with respect to its size and distribution, as under-sampling leads to even smaller datasets.
LR	LR is a statistics-based model that uses a logistic function for the classification of data [3]. It can perform better when the feature set is large as in the oversampling case. This study uses LR with ‘lib-linear’ solver which is the best optimizer for small datasets. LR can be good for binary classification as is the case with the current study. LR is used to predict a data value based on prior observations of a dataset. Because of its best performance on binary and linear data, this study adopts it for the task at hand.

## 4.2 Performance of machine learning models with data under-sampling

Data under-sampling is performed to obtain a more balanced distribution of the training data for both classes so that the influence of the model over-fitting can be alleviated. Under-sampling is carried out until the number of samples of the majority class ‘ham’ become almost equal to the number of samples of the minority class ‘spam’.

**Table 7** Architectures of LSTM and GRU models

Sequential()	Sequential()
Embedding(5000,100, input_length=X.shape[1])	Embedding(5000,100, input_length=X.shape[1])
LSTM(100)	GRU(100, return_sequences=True)
Dense(16)	SimpleRNN(32)
Dense(2, activation='softmax')	Dense(16)
	Dense(2, activation='softmax')
loss='binary_crossentropy', optimizer='adam', batch_size=32, epochs =100	

Table 11 contains the results of machine learning models using TF-IDF features from the under-sampled data. Results suggest that the performance of the models is improved significantly. SVM achieves the highest accuracy score of 0.989 with the highest 0.99 F1 score. LR and RF follow this performance with 0.983 accuracy each and 0.98 F1 score, respectively. Results also indicate a high degree of agreement between the accuracy and F1 score and large deviations are not found between the accuracy and F1 score. GNB also improves its performance and achieves the highest accuracy and F1 scores when used with the under-sampled data.

Performance of models using the BoW features after under-sampling is performed and results are given in Table 12. LR is significantly better with 0.989 accuracy and an F1 score of 0.99. RF is just behind the LR with a 0.98 accuracy score while the F1 score is 0.98. Similarly, the performance of tree-based models and probability-based models are good with simple term count features than linear models, as the accuracy of SVM is reduced using BoW features as compared to TF-IDF features.

Models' results using feature union are shown in Table 13. The performance of machine learning models improved after under-sampling and feature union. LR and RF achieve the highest accuracy and F1 scores of 0.99 each. Here, both linear and tree-based models perform well based on the feature set that contains both weighted and simple term counts. These results show the impact of data balancing and feature union to improve the models' performance.

### 4.3 Classification using deep learning models LSTM and GRU

Deep learning models are also implemented using the original data, as well as, the balanced data using the random under-sampling. Results of deep learning models are shown in Table 14 which indicates that deep learning models also show good performance for spam email classification. LSTM and GRU perform well on the imbalanced dataset as GRU and

**Table 8** Results of machine learning models using TF-IDF features

Model	Accuracy	Precision	Recall	F1 Score
RF	0.973	0.99	0.86	0.92
GBM	0.978	0.99	0.86	0.92
SVM	0.983	0.96	0.94	0.95
GNB	0.811	0.46	0.88	0.61
LR	0.973	0.94	0.94	0.94

**Table 9** Results of machine learning models using BoW features

Model	Accuracy	Precision	Recall	F1 Score
RF	0.977	0.99	0.88	0.93
GBM	0.976	0.94	0.92	0.93
SVM	0.980	0.94	0.94	0.94
GNB	0.791	0.49	0.85	0.62
LR	0.979	0.99	0.86	0.92

**Table 10** Results of machine learning models using Feature Union

Model	Accuracy	Precision	Recall	F1 Score
RF	0.980	0.99	0.89	0.94
GBM	0.981	0.96	0.94	0.95
SVM	0.980	0.94	0.95	0.95
GNB	0.820	0.50	0.86	0.63
LR	0.980	0.95	0.94	0.95

**Table 11** Results of machine learning models using TF-IDF features and under-sampling approach

Model	Accuracy	Precision	Recall	F1 Score
RF	0.983	0.99	0.97	0.98
GBM	0.951	0.96	0.94	0.95
SVC	0.989	0.99	0.99	0.99
GNB	0.964	0.98	0.95	0.96
LR	0.983	0.98	0.99	0.98

**Table 12** Results of machine learning models using BoW features and under-sampling approach

Model	Accuracy	Precision	Recall	F1 Score
RF	0.983	0.99	0.97	0.98
GBM	0.963	0.96	0.96	0.96
SVC	0.972	0.98	0.96	0.97
GNB	0.972	0.98	0.96	0.97
LR	0.989	0.99	0.99	0.99

**Table 13** Results of machine learning models using features union and under-sampling approach

Model	Accuracy	Precision	Recall	F1 Score
RF	0.991	1.00	0.98	0.99
GBM	0.952	0.97	0.94	0.95
SVM	0.982	0.99	0.98	0.98
GNB	0.976	0.99	0.95	0.97
LR	0.990	0.99	0.99	0.99

**Table 14** LSTM and GRU results using each re-sampling technique

Sampling	Model	Accuracy	Class	Precision	Recall	F1 Score
Without	LSTM	0.98	0	0.98	0.99	0.98
			1	0.94	0.92	0.93
			Macro Avg.	0.96	0.95	0.96
	GRU	0.98	0	0.99	0.99	0.99
			1	0.95	0.95	0.95
			Macro Avg.	0.97	0.97	0.97
Under-Sampling	LSTM	0.98	0	0.98	0.98	0.98
			1	0.98	0.98	0.98
			Macro Avg.	0.98	0.98	0.98
	GRU	0.98	0	0.98	0.90	0.98
			1	0.98	0.98	0.98
			Macro Avg.	0.98	0.98	0.98

LSTM achieve the highest F1 scores of the study on the imbalanced dataset which are 0.97 and 0.96, respectively. The highest F1 score on the imbalanced dataset by machine learning models is 0.95 which is achieved by LR, RF, and SVM. Overall the performance of machine learning models is good as compared to deep learning models. Deep learning models are data-intensive and require large datasets to show better performance. Given the size of the dataset for the current study, machine learning models tend to show better performance. The highest accuracy of 0.991 is achieved by machine learning model RF using feature union from the under-sampled data, while for deep learning models LSTM and GRU both achieve an accuracy of 0.98.

#### 4.4 Computational complexity of models

Table 15 shows the computational complexity of each model using the hybrid feature set and other individual features. Models require low execution time using BoW and TF-IDF, however, do not provide higher classification accuracy. Execution time is higher when models are trained using BoW and TF-IDF features combined, as the size of the feature set increases when both features are combined. The computation time of best performer RF and LR is increased from 21.95 seconds to 96.28 seconds and 0.423 to 2.157 seconds, respectively. This increase in computation time is a limitation of this study. The proposed system is more accurate but also has high computation cost.

**Table 15** Computational time (seconds) of machine learning models

Model	BoW	TF-IDF	Feature Union
RF	21.95	21.10	96.28
GBM	36.58	40.77	105.8
SVC	6.05	18.12	16.45
GNB	0.512	0.781	2.087
LR	0.423	0.311	2.157

**Table 16** Performance analysis with respect to state-of-the-art studies on spam email classification

Reference	Year	Model	Accuracy	F1 Score
Gaurav et al. [11]	2020	RF	0.927	0.929
Khamis et al. [20]	2020	SVC	0.888	-
Kontsewaya et al. [21]	2021	LR; NB	0.990	0.970
Iqbal and Khan [16]	2022	ANN	0.981	0.978
This study	2022	RF; Feature Union; Under-sampling	0.991	0.990
	2022	LR; Feature Union; Under-sampling	0.990	0.990

#### 4.5 Performance comparison with state-of-the-art studies

For analyzing the efficiency of the proposed feature union approach, performance analysis with other studies is also carried out. Some recent studies based on spam email classification are used for comparison. For example, [11] used RF for spam email classification. Similarly, [20] used SVM for header-based spam email classification to achieve significant results. Another study dealing with the same task is [21] that performed experiments for spam email classification using natural language processing techniques. They used several models for spam email classification and achieved the highest results using LR and Naive Bayes (NB). Similarly, study [16] used machine learning approach for spam email classification. They used Artificial Neural Network (ANN) model to achieve significant accuracy. For a fair analysis, accuracy and F1 score are used to make the performance comparison of the current study with the discussed studies, and results are given in Table 16.

## 5 Conclusion

Internet users are exposed to several threats and spam emails present a potential tool for spammers to steal the financial and personal information of users. This study proposes a machine learning-based approach for spam email detection with high accuracy. For experiments, a hybrid dataset is made by combining two spam email datasets. For reducing the impact of data imbalance on models' overfitting, random under-sampling is used on the majority class. Similarly, feature fusion is proposed by combining BoW and TF-IDF features to elevate models' performance. Results indicate that RF achieves the highest accuracy of 0.991 and outperforms all other models. The significant performance of RF is due to its ensemble architecture and use of the proposed feature union approach. The small size of the dataset is complemented with feature union to increase the feature vector which helps to improve the performance. Besides RF, LR and SVM also perform better and obtain an accuracy of 0.99 each when used with feature union. Experiments using LSTM and GRU deep learning models show relatively low performance as compared to machine learning models. Furthermore, data under-sampling tends to improve the performance of deep learning models. This study has several limitations; the first is the high computational time of machine learning models with a feature union approach, and the second is the small size of the dataset with an imbalanced target class ratio. We will consider these limitations in our future work. We also intend to perform further experiments using over-sampling techniques to analyze its influence on deep learning models.

**Funding** “This research was supported by the Florida Center for Advanced Analytics and Data Science funded by Ernesto.Net (under the Algorithms for Good Grant).”

**Data Availability** The datasets used in this study are publicly available at the following links  
<https://www.kaggle.com/datasets/karthickveerakumar/spam-filter>  
<https://www.kaggle.com/washingtongold/spam-or-ham-emp-week-2-ml-hw-dataset>

## Declarations

**Conflict of Interests** The authors declare that they have no conflict of interest.

## References

1. A Chen YFU, Zheng X, Lu G (2022) An efficient network behavior anomaly detection using a hybrid dbn-lstm network. *Comput Secur* 114:102600
2. APWG (2021) Fishing activity trend reports. <https://apwg.org/trendsreports/>, Accessed 19 2021
3. Ahmed Arafa AH, Radad M, Badawy MM, El-Fishawy N (2022) Logistic regression hyperparameter optimization for cancer classification. *Menoufia J Electron Eng Res*
4. Awad M, Foqaha M (2016) Email spam classification using hybrid approach of rbf neural network and particle swarm optimization. *Int J Netw Secur Appl* 8(4):17–28
5. Bassiouni M, Ali M, El-Dahshan E (2018) Ham and spam e-mails classification using machine learning techniques. *J Appl Secur Res* 13(3):315–331
6. Bhatti UA, Huang M, Wang H, Zhang Y, Mehmood A, Di W (2018) Recommendation system for immunization coverage and monitoring. *Hum Vaccines Immunotherapeutics* 14(1):165–171
7. Bhatti UA, Huang M, Wu D, Zhang Y, Mehmood A, Han H (2019) Recommendation system using feature extraction and pattern recognition in clinical care systems. *Enterp Inf Syst* 13(3):329–351
8. Bhatti UA, Zeeshan Z, Nizamani MM, S Bazai ZYU, Yuan L (2022) Assessing the change of ambient air quality patterns in jiangsu province of China pre-to post-covid-19. *Chemosphere* 288:132569
9. Bhowmick A, Hazarika SM (2018) E-mail spam filtering: a review of techniques and trends. *Advances in electronics, communication and computing*, pp 583–590
10. Dada EG, Bassi JS, Chiroma H, Adetunmbi AO, Ajibuwa OE et al (2019) Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon* 5(6):e01802
11. Gaurav D, Tiwari SM, Goyal A, Gandhi N, Abraham A (2020) Machine intelligence-based algorithms for spam filtering on document labeling. *Soft Comput* 24(13):9625–9638
12. GuangJun L, Nazir S, Khan HU, Haq AU (2020) Spam detection approach for secure mobile message communication using machine learning algorithms. *Secur Commun Netw*, vol 2020
13. Hamid IRA, Abawajy J, Kim T (2013) Using feature selection and classification scheme for automating phishing email detection. *Studies in informatics and control* 22(1):61–70
14. Hilal W, Gadsden SA, Yawney J, Gadsden SA, Yawney J (2022) Financial fraud: a review of anomaly detection techniques and recent advances
15. Hulten G, Goodman J, Rounthwaite R (2004) Filtering spam e-mail on a global scale. In: *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pp 366–367
16. Iqbal K, Khan MS (2022) Email classification analysis using machine learning techniques. *Appl Comput Inform no. ahead-of-print*
17. Jénez-Martino F, Fidalgo E, González-Martínez S, Velasco-Mata J (2020) Classification of spam emails through hierarchical clustering and supervised learning. [arXiv:2005.08773](https://arxiv.org/abs/2005.08773)
18. Javaid A, Siddique MA, Reshi AA, Rustam F, Lee E, Rupapara V et al (2022) Coal mining accident causes classification using voting-based hybrid classifier (vhc). *J Ambient Intell Humanized Comput*, pp 1–11
19. Keivani FS, Jouzbarkand M, Khodadadi M, Sourkouhi ZK (2012) A general view on the e-banking. *Int Proc Econ Dev Res* 43:p62
20. Khamis SA, Foozy CFM, Ab Aziz MF, Rahim N (2020) Header based email spam detection framework using support vector machine (svm) technique. In: *International conference on soft computing and data mining*. Springer, pp 57–65
21. Kontsewaya Y, Antonov E, Artamonov A (2021) Evaluating the effectiveness of machine learning methods for spam detection. *Procedia Comput Sci* 190:479–486
22. Kumar KV (2021) Spam filer - identifying spam using emails. <https://www.kaggle.com/karthickveerakumar/spam-filter/metadata>, Accessed 27 2017



23. Kumar RK, Poonkuzhali G, Sudhakar P (2012) Comparative study on email spam classifier using data mining techniques. *Proceedings of the international multicongference of engineers and computer scientists* 1:14–16
24. Kumaresan T, Saravanakumar S, Balamurugan R (2019) Visual and textual features based email spam classification using s-cuckoo search and hybrid kernel support vector machine. *Clust Comput* 22(1):33–46
25. Lee E, Rustam F, Ashraf I, Washington PB, Narra M, Shafique R (2022) Inquest of current situation in Afghanistan under taliban rule using sentiment analysis and volume analysis. *IEEE Access* 10:10333–10348
26. Mujahid M, Lee E, Rustam F, Washington PB, Ullah S, Reshi AA, Ashraf I (2021) Sentiment analysis and topic modeling on tweets about online education during covid-19. *Appl Sci* 11(18):8438
27. Reshi AA, Rustam F, Aljedaani W, Shafi S, Alhossan A, Alrabiah Z, Ahmad A, Alsuwailem H, Alman-gour TA, Alshammari MA et al (2022). In: Covid-19 vaccination-related sentiments analysis: a case study using worldwide twitter dataset *Healthcare*, vol 110(3). MDPI, pp 411
28. Rish I et al (2001) An empirical study of the naive bayes classifier. In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol 3. (22), pp 41–46
29. Rupapara V, Rustam F, Amaar A, Washington PB, Lee E, Ashraf I (2021) Deepfake tweets classification using stacked bi-lstm and words embedding. *PeerJ Comput Sci* 7:e745
30. Rusland NF, Wahid N, Kasim S, Hafit H, Analysis of naïve bayes algorithm for email spam filtering across multiple datasets (2017). In: *IOP conference series: materials science and engineering*, vol 226, no 1. IOP Publishing, p 012091
31. Rustam F, Imtiaz Z, Mehmood A, Rupapara V, Choi GS, Din S, Ashraf I (2022) Automated disease diagnosis and precaution recommender system using supervised machine learning. *Multimed Tools Appl*, pp 1–24
32. Seth S, Biswas S (2017) Multimodal spam classification using deep learning techniques. In: *2017 13th international conference on signal-image technology & internet-based systems (SITIS)*. IEEE, pp 346–349
33. Sinha S, Ghosh I, Satapathy SC (2021) A study for ann model for spam classification. In: *Intelligent data engineering and analytics*. Springer, pp 331–343
34. Ye A (2021) Spam of ham - emp week 2 hw dataset. <https://www.kaggle.com/washingtongold/spam-or-ham-emp-week-2-ml-hw-dataset>, Accessed 27 2019
35. Zamir A, Khan HU, Mehmood W, Iqbal T, Akram AU (2020) A feature-centric spam email detection model using diverse supervised machine learning algorithms. *Electron Libr*
36. ZhiWei M, Singh MM, Zaaba ZF (2017) Email spam detection: a method of meta-classifiers stacking. In: *The 6th international conference on computing and informatics*, pp 750–757

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.