Multimedia Tools and Applications (2022) 81:39853–39871
https://doi.org/10.1007/s11042-022-12991-0

# Spam SMS filtering based on text features and supervised machine learning techniques

**Muhammad Adeel Abid[1] · Saleem Ullah[1] · Muhammad Abubakar Siddique[1] · Muhammad Faheem Mushtaq[2] · Wajdi Aljedaani[3] · Furqan Rustam[4]**

Received: 20 May 2021 / Revised: 18 January 2022 / Accepted: 27 March 2022 /
Published online: 4 May 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

**Abstract**
The advancement in technology made a significant mark with time, which affects every field of life like medicine, music, office, traveling, and communication. Telephone lines are used as a communication medium in ancient times. Currently, wireless technology overrides telephone wire technology with much broader features. The advertisement agencies and spammers mostly use SMS as a medium of communication to convey their business brochures to the typical person. Due to this reason, more than 60% of spam SMS are received daily. These spam messages cause users' anger and sometimes scam with innocent users, but it creates large profits for the spammer and advertisement companies. This study proposed an approach for the classification of spam and ham SMS using supervised machine learning techniques. The feature extracting techniques such as Term Frequency-Inverse Document Frequency (TF-IDF) and bag-of-words are used to extract features from data. The SMS dataset used was imbalanced, and to solve this problem, we used oversampling and under-sampling techniques. The support vector classifier, gradient boosting machine, random forest, Gaussian Naive Bayes, and logistics regression are applied on the spam and ham SMS dataset to evaluate the performance using accuracy, precision, recall, and F1 score. The experiment result shows that the random forest classifies spam ham SMS more accurately with 99% accuracy. The proposed model is trained well to identify the SMS category in terms of Ham or Spam with TF-IDF features and oversampling technique. The performance of the proposed approach was also evaluated on the spam email dataset with significant 99% accuracy.

**Keywords** SMS · Spam · Supervised machine learning · TF-IDF · Bag of words · Classification

## 1 Introduction

Technology gradually makes improvements in our daily life, and its development technology is a continuous and ongoing process. As a result, humans depend on technology

✉ Furqan Rustam
furqan.rustam1@gmail.com

Extended author information available on the last page of the article.

to perform their tasks in every field of life [1], like the medical domain, information technology, and communication domain [14]. Improvements in technology [13] solved the problems efficiently as compared to the previous 30 years. The enhancement in technology helps to facilitate and improve the existing facilities in every field of life. Nowadays, people are much more reliant on computers, and they would be considered reliable resources for doing tasks. It also improves the communication field and helps to connect people by using the phone. In ancient times, telephones are used that are employed as a medium to communicate with people over long distances. Meanwhile, more problems in communication are faced because there are resistance and noise factors over telephone lines [7] that disturb the normal voice. In later times, cellular technology is introduced [21] such as wireless technology [49] made a significant contribution to minimizing the distance between people. The Short Message Service (SMS) is one of the prominent features of wireless communication used as the communication medium between different users such as doctors, engineers, musicians, students, teachers, bankers, officials, business people, etc. [34]. The usage of SMS service becomes regularly used so that nearly a person sends 5-10 SMS per day.

As time passes, the advertisement of anything is becoming the key feature to improve the business. Due to the importance of advertisement, different advertising agencies now use SMS as a medium of advertisement. This is the fastest way to communicate the business brochure to a typical person. That is why spammer also uses SMS technology to contact people, a source of income for the spammer. There are many tools available that prevent spam SMS, but according to an estimate, a person daily receives the bulk of SMS, and more than 50% are spam SMS [23]. There would be a system that can identify ham, and spam SMS accurately [28]. Many researchers propose their approaches to predict spam and ham SMS, but accuracy is still the point for the researcher to work in this domain.

This study proposed an approach to classify spam and ham SMS using supervised machine learning algorithms. For this, the SMS dataset is collected from [25]. The dataset contains both spam and ham SMS. Firstly, this research applies to preprocessing techniques to clean SMS that include stemming, stop words removal technique, convert to lowercase, punctuation removal, and numeric removal techniques. Oversampling and under-sampling are also performed to get more accurate results. Term Frequency-Inverse Document Frequency (TF-IDF) [48] and Bags of Words (BOW) [5] are applied on text data for the feature extraction. After that, the dataset is split into two sets, 75% of the volume is considered a training set, and 25% is considered the testing set. The training set was used to train machine learning models such as support vector classifier, gradient boosting machine, random forest, Gaussian Naïve Bayes, and logistics regression. Later, the model is tested by applying 25% test data to trained models. In the end, we evaluate the machine learning models in terms of accuracy, precision, recall, and F1 score.

The contribution of this research is as follows:

- The classification of spam and ham messages is performed using machine learning.
- To reduce the complexity in feature set and increase models efficiency, we have done data preprocessing such as punctuation removal, numerical removal, convert to lower case, stemming, and stop words removal is performed.
- The used dataset was imbalanced, and to solve this problem, we used the data re-sampling techniques to reduce model over-fitting
- To find the weighted features, we used TF-IDF features in our approach to training the machine learning models.
- To utilize various machine learning algorithms to provide the highest accuracy, we also compare our study with existing studies.

The rest of this research is organized as follows: Section 2 presents some of the related work that includes the previous work applied to ham and spam SMS for classification. Section 3 illustrates the material and method such as data collection source, feature extraction Techniques TF-IDF, Bag of Words, models used for training, and evaluation parameters are discussed. Section 4 shows the results and discussion of the proposed methodology. Section 5 represents the conclusion and future work of this research.

## 2 Related work

Email is the electronic way of communication and is categorized as spam and ham emails. In email filtering, content-based filtering is most effective [16, 32, 33, 47, 51]. The content-based filtering approach mainly depends on some machine learning algorithms based on some features to differentiate between ham and spam using legitimate SMS techniques. The complete dataset is divided into training and testing set on which machine learning algorithms are applied to already separate ham and spam SMS. The testing dataset is used to analyze the efficiency of the technique. Machine-based learning approaches have been tried by researchers in SMS filtering [2, 22]. It is challenging to apply machine-based approaches for SMS because of the short length of content as compared to email that has a greater length of information [10]. The statistical learning-based classifier that is trained with lexical features is explored by [22] for SMS spam filtering. They have tested the feasibility of applying a Bayesian-based classifier for SMS spam filtering and discussed the state of SMS spam from various sources. The study [10] also discussed the content-based approach on SMS spam which consists of blog comments, SMS, and weblogs.

There is a problem that content-based classifier application faces are finding the features in SMS because of its short length. This work focuses on expanding the features of mobile spam filters with additional features such as orthogonal mobile word bi-gram [10]. They were very effective in vector-based machine learning algorithms like SVM and Orthogonal Sparse Bigrams with (OSBF)-Lua (Orthogonal Sparse Bigrams with confidence Factor). OSBF-Lua makes a relationship between concatenate words by putting distance between them; character Bigram like hocky could be broken down into “ho”, “oc”, “ck”, “ky”; character trigram like hockey could be broken down into “hoc”, “ock” etc. [45] took a slightly different approach and argued that the content-based approach did not work quite better on common spam words like “offer”, “sale” etc. that may be present in messages. So, they choose a feature set containing average message length, function word frequencies, and count special characters to measure the effect of this information in SMS spam filtering. The evaluation used in this work is Area Under Curve (AUC) in the ROC curve, specifically 1-AUC(%). Due to the absence of a dataset, it is more difficult to measure the accuracy of two different works. However, the accuracy is further improved by lowering the 1-AUC(%) [32] due to the advent of different microblogging websites like Twitter, which can open further opportunities for spammers. In [6], authors identify spam by using a confusion matrix to detect spam in tweets. The proposed model has an accuracy of 87.2% for detection of spam and 87.6% for spammer detection that is enough to accept.

Lots of researchers have done natural language processing (NLP) applications such as spam SMS classification, sentiment analysis, toxic comment classification, etc. Such as the study [12] proposed an approach for automatic detection of toxic text from tweets. They used TF-IDF and achieved 95.6% accuracy. Another study [38], proposed an ensemble approach for toxic comment classification. They proposed a regression vector voting classifier (RVVC) and achieved a 0.97 accuracy score using TF-IDF features. The study [37]

proposed an approach for sentiment analysis on deepfake tweets. They proposed a stacked Bi-LSTM model and achieved a 0.92 accuracy score.

The study [50], used the hidden Markov model (HMM) as a based model to propose a method to perform spam SMS filtering. The proposed HMM model was not language-sensitive, and the author evaluates the proposed model on two datasets. One of them is the same dataset that we used in this study, and the second a Chinese SMS data. The study [50] shows that their proposed model outperforms all other approaches with a 0.959 accuracy score. The study [36] proposed a convolution neural networks (CNN) model to perform the spam and ham SMS filtering and achieved the highest 0.977 accuracy score on the UCI dataset. They also used long short-term memory (LSTM) for spam filtering and achieved 0.953 accuracy using the LSTM. Similarly, another study [19] used LSTM for spam and ham SMS filtering. They performed a deep comparison between machine learning and deep learning models on the UCI dataset and found LSTM with the highest accuracy score of 0.985. The study [44] used an ensemble approach for spam SMS filtering on the same dataset as we used. The approach deploys RF with the chi2 feature selection technique to achieve the best results. The ensemble approach gives the 97.5% accuracy score for spam ham filtering.

Concerning all the presented works, our study focus on the performance comparison of spam filtering using various machine learning algorithms. We solve the data imbalance problem to avoid the models over-fitting which is not focused on in previous studies.

## 3 Experimental setup

The SMS dataset is downloaded from Kaggle. The preprocessing is performed to clean the data. The feature extraction techniques such as TF-IDF and Bag of Words are used for tokenization. Different machine learning algorithms are used to train the proposed model. The evaluation parameters such as precision, recall, and f1-score are used to evaluate the performance of the proposed model.

### 3.1 Data collection

The quality of the dataset is of great value while performing any experiments in data mining. The dataset contains the context of SMS and the category of that SMS as ham or spam. First, choose a dataset [8] that contains SMS's context with category. The statistics of the dataset used in this research is given in Table 1 below:

Although the number of SMS of both categories varies in total count for training purposes, an equal number of SMS [42, 43] from both categories is chosen. An equal number of SMS is chosen to reduce the impact of biased results. Data resampling SMOTE and random under-sampling are also used for examining the performance accordingly. Tables 2 and 3 show the description of dataset variables and sample text of SMS.

**Table 1** Statistics of Dataset

| Sr No. | Category | Total SMS | Under-sampling | Over-sampling |
|---|---|---|---|---|
| 1 | Ham | 4,825 | 747 | 4,825 |
| 2 | Spam | 747 | 747 | 4,825 |

**Table 2** Description of dataset variables

| Variables | Description |
|---|---|
| Category | Category of SMS whether Ham or Spam |
| Text | Contents of SMS |

### 3.2 Data visualization

This section illustrates the statistics of the dataset graphically. Figure 1(a) shows the percentage of Ham and Spam SMS found in the complete dataset. Ham SMS is in Bulk as compared to spam SMS. This would undoubtedly disturb the training of the proposed model. So balanced SMS is chosen from each category so that training of the proposed model would not be affected, and the proposed model predicts the category of SMS with higher accuracy. Figure 1(b) shows the selected SMS percentage.

### 3.3 Methodology

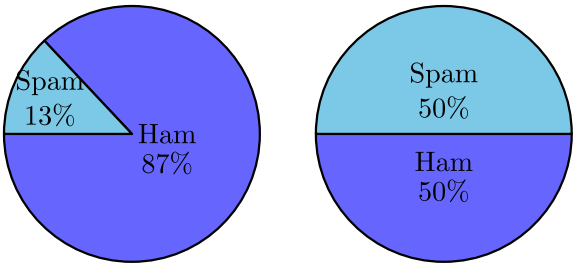
In this research, support vector classifier, gradient boosting machine, random forest, Gaussian Naive Bayes, and logistics regression are applied on keywords extracted from TF-IDF and BOW individually. The pre-processing is also performed to clean up the context of SMS from unnecessary material.

#### 3.3.1 Pre-Processing

In this step, pre-processing of data is performed to improve the training model's learning process [24]. SMS contains stop words, punctuation, and upper and lower case words that can affect and reduce the learning of the training model. The processing is applied after collecting the dataset with an equal number of SMS. Firstly, tokens of SMS are made because SMS is the string of words and is difficult to understand for the model's training. Each SMS splits into words so that pre-processing can be applied. At a later stage, stop words are removed as they have no weight-age. Afterward, stemming is performed because SMS words are sometimes not complete or characters are not typed. So, Stemming is necessary to correct the spellings of tokenized words. Furthermore, the numeric values are removed because digits make no impact in identifying ham or spam SMS and are considered ignored. Finally, the punctuation is removed, and the proposed model will be well trained. The pre-processing process that is applied on SMS is shown in Fig. 2. Table 4 shows four SMS taken as a sample to show pre-processing followed in this study.

**Table 3** Sample of dataset

| Category | SMS Content |
|---|---|
| Ham | I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today. |
| Spam | England v Macedonia - dont miss the goals/team news. Txt ur national team to 87077 eg ENGLAND to 87077 Try:WALES, SCOTLAND 4txt/Ì $\frac{1}{4}$1.20 POBOXox36504W45WQ 16+ |




(a) Complete dataset statistics. (b) Selected SMS statistics.

**Fig. 1** SMS Dataset statistics

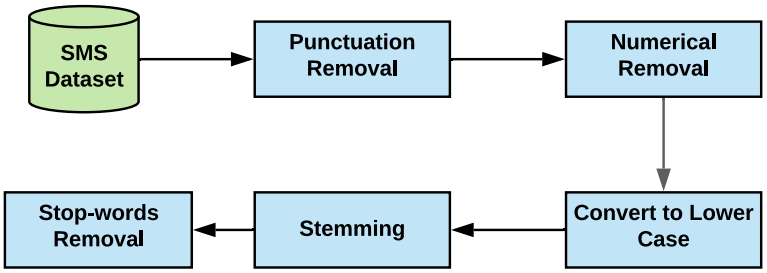



**Fig. 2** Sequence of steps followed in pre-processing of SMS dataset

**Table 4** Sample SMS taken for pre-processing

| Sr No. | Category | SMS |
|---|---|---|
| 1 | spam | WINNER!! As a valued network customer you have been selected to receivea å£ 900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only. Had your mobile 11 months or more? |
| 2 | spam | U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030 |
| 3 | ham | I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today. |
| 4 | ham | I've been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times. |

**Remove Punctuation:** Using punctuation in the text helps the reader to understand the message that is being conveyed clearly. But these marks have no meaning, so they are not helpful for model training, and we remove them in preprocessing.

**Remove Numbers:** SMS can contain numbers that are not useful in machine learning models training. We remove these numbers to reduce complexity in the features set. We remove these numbers using regular expressions.

**Convert to Lowercase:** This technique is vital to reduce complexity in features set such as 'Go', 'go', and 'GO' are the same in meaning, but each word will be considered a separate feature because of differences in cases. Convert to lower case technique will reduce complexity converting case to lower cases such as 'Go', 'go', 'GO' will be 'go'. We deploy this technique using the python tolower() function.

**Stemming:** To convert each word into its root form we used the stemming technique. We used the Porter stemmer technique to perform stemming [37].

**Remove Stopwords:** Stopwords are the parts of text but have no meaning, so to focus on the meaningful words during training, we remove stopwords. We remove stopwords using the natural language toolkit. Table 5, contains the results after preprocessing of sample data.

### 3.4 Implementation detail

All experiments are deployed on the Corei7 11th generation system with Windows 10 operating system. To implement the approach, we used Jupyter Notebook and python language.

Figure 3 shows the methodology that is deployed for SMS Spam filtering. First, an SMS dataset was acquired from the UCI data repository. To clean the dataset, different preprocessing techniques have been used. After data preprocessing, data re-sampling techniques were used to make the dataset balanced to reduce the over-fitting of models. Dataset splits for train and test purposes. The 75% dataset of the selected SMS of both categories is chosen for training the model. The remaining 25% of the selected SMS of both categories are used for testing purposes to check whether the proposed system is correctly trained or required results are obtained.

**Table 5** Sample SMS after preprocessing

| Sr No. | Category | SMS |
|---|---|---|
| 1 | | winner valu network custom select receivea prize |
| | spam | reward claim call claim code kl valid hour onli |
| | | mobil month u r entitl updat latest colour |
| 2 | spam | mobil camera free call mobil updat co free |
| | | im gon na home soon dont want talk thi |
| 3 | ham | stuff anymor tonight k ive cri enough today |
| | | ive search right word thank thi breather promis |
| 4 | ham | wont take help grant fulfil promis wonder bless time |

**Algorithm 1** Proposed approach steps algorithm.

**Input:** Input: SMS dataset
**Output:** SPAM or HAM
initialization;
PreprocessedData ⟵PreprocessingTechniques(Data)
Resampling ⟵ SMOTE(PreprocessedData)
TrainingSet, TestingSet ⟵ Splitting(Resampling)
TrainingFeatures ⟵ TFIDF(TrainingSet)
TestingFeatures ⟵ TFIDF(TestingSet)
TrainedModel⟵ ModelTraining(TrainingFeatures)
Prediction⟵ TrainingSet(TestingFeatures)
Scores ⟵ Evaluation (Prediction)

### 3.4.1 Feature engineering

TF-IDF and bag of words are used as the feature extraction technique discussed as follows:

**Term Frequency-Inverse Document Frequency (TF-IDF)** TF-IDF is the most common and famous technique for extraction of features of a given corpus [12, 35, 48]. The following equation is used for the computation of TF-IDF.

$$TF_{ij} = \frac{n_{ij}}{\sum_k nkj} \tag{1}$$

Equation (1) describes the calculation TF of terms $i$ in the domain of documents $j$. IDF is calculated via.

$$IDF_i = log\left[\frac{|D|}{|\{d : t_i \epsilon d\}|}\right] \tag{2}$$

Where $D$ is the total number of all documents and $t_i$ is the term. Once the $TF_{ij}$ and $IDF_i$ are calculated, $TF - IDF_{ij}$ is calculated by multiplying $TF_{ij}$ and $IDF_i$ and illustrated.

$$TF - IDF = TF_{ij} * IDF_i \tag{3}$$

**Bag of Words** Bag of words model [5, 29] is a simple representation of information retrieval and natural language processing. For training the model, the text is taken as a bag of its

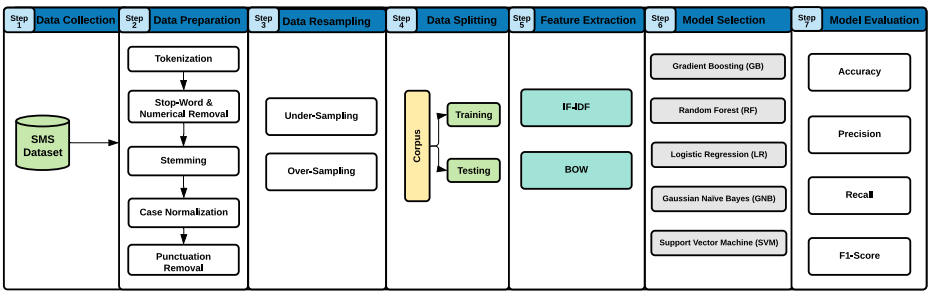



**Fig. 3** Methodology applied for SMS Spam filtering

words, not focusing on word order. This model is also used for computer vision. It is commonly used in document classification, where the frequency of each word is used as a keyword for the training of a system.

### 3.5 Oversampling and undersampling techniques

In this study, we used Oversampling (SMOTE) and Undersampling Techniques such (Random Undersampling). Over-sampling is a technique where the number of the minority class in the majority class ratio is raised. Oversampling increases the sample size, generating additional features for model training and enhancing the model's accuracy. The SMOTE approach employed in this study is the over-sampling of a synthetic minority. SMOTE is a state-of-the-art method proposed in [9] to tackle the unbalanced datasets over-fitting issue. SMOTE takes the smaller category randomly and discovers the K-close neighbors of all smaller classes. The selected samples are assessed utilizing the nearest K neighbor to produce a new minority class at that particular moment. In view of the findings described in [11, 17], SMOTE has been employed.

By excluding examples of the major class, undersampling decreases the dataset. In this research, a random undersampling technique is employed for undersampling. This technique works by rejecting randomly selected examples of the majority class and deleting them so that the distribution of target classes can be balanced. To put it simply, undersampling tries to equalize the distribution of classes by removing samples of classes that are in majority at random. This method of re-sampling is popularly used and was chosen for this study because it delivers performance [30]

#### 3.5.1 Machine learning algorithms

There is a large number of machine learning algorithms available. For the proposed approach, support vector classifier, gradient boosting machine, random forest, Gaussian naive Bayes, and logistics regression applied on the results that are obtained from TF-IDF and bag of words individually. The first model is trained by using 75% of selected data so that it is able to predict the SMS category. The hyperparameters setting for each model is shown in Table 6.

**Random Forest (RF)** is an ensemble tree-based model consisting of many weak decision trees [31]. The bagging technique is used in this model to train decision trees using a number of different bootstrap samples [46]. In Random Forest, subsampling of different training datasets with replacement is done to obtain a bootstrap sample where the size of the sample is similar to the size of the training dataset [39]. Bootstrap aggregating is a method

**Table 6** Machine Learning Models Parameters

| Algorithm | Hyper Parameters |
|---|---|
| RF | n_estimators=300, random_state=5, max_depth=300 |
| GBM | n_estimators=300, max_depth=300 |
| LR | solver='saga', C=3.0, max_iter=100, penalty='l2' |
| SVM | kernel='linear', C=2.0, random_state=500 |
| GNB | default setting |

in which based on bootstrapped samples. Random Forest can be calculated as in Eqs. 1 and 2:

$$p = mode\{T_1(y), T_2(y), T_3(y), ..., T_n(y)\} \tag{4}$$

$$p = mode\{\sum_{i=1}^{N} T_i(y)\} \tag{5}$$

Where $p$ is the final prediction calculated by the majority of decision trees. While $T_1(y), T_2(y), ..., T_m(y)$ are the number of decision trees taking part in prediction.

**Logistic Regression (LR)** is a method for examining data where one or more variables are used to produce output [38]. Logistic regression is used to calculate the probability of class members. That is why it is considered the best learning model when there is categorical target data [18]. It works on the relationship between independent and dependent variables. A logistic function is common "S" shaped as in (6):

$$f(x) = \frac{L}{1 + e^{-m(v - v_o)}} \tag{6}$$

Where,

- $e$ is Euler Number.
- $v_o$ is the sigmoid midpoint's x-value.
- $L$ shows the curve's maximum value.
- $m$ shows the steepness of the curve.

**Support Vector Machine (SVM)** is widely used in classification and pattern recognition problems [3, 47]. It works well on high dimensioned data by calculating a hyper-plane that maximizes the margin between the classes causes minimize the error rate in classification problems. Its performance regarding classification is compromised when we apply it to such data, which is overlapped because this algorithm cannot maximize the margin between two classes. It is a supervised machine learning model that is used to solve two-group classification problems. It is more convenient and gives better accuracy in most circumstances.

**Gradient Boosting (GB)** can be used for both classification and regression. The purpose of boosting is actually to increase the capability of any machine or algorithm in such a manner to catch the weakness of the model and replace it with a strong learner to find out the near to accurate solutions [27]. The gradient boosting machine does this task by training many model gradual, additive, and sequential manners.

**Gaussian Naïve Bayes (GNB)** It is a classifier based on Bayes theorem [20]. The classifier has some assumptions in implementing its algorithm, like all the features found in the model are independent. It is used in the classification of objects which have normally distributed data. Due to these properties, it is also called the Gaussians Naive Bayes classifier. It is calculated as follows:

$$P(c|x) = \frac{P(c|x)P(c)}{P(x)} \tag{7}$$

$$P(c|x) = P(x_1|x) * ...., P(x_1|x) * P(c) \tag{8}$$

- $P(c|x)$ is the said to be the posterior probability of target class
- $P(c)$ shows the prior probability of class.

– $P(x|c)$ is the probability of predictor class.
– $P(x)$ shows the prior probability of predictor.

### 3.5.2 Evaluation parameters

Testing data is applied to check whether it is correctly trained. For testing purposes, a large number of evaluation parameters are available. Precision, Recall, and F1 score techniques [40] are applied to the proposed model to check the validity of results.

**Accuracy** is used to calculate and measure the correctness for target classes. The highest value of this score is 1, and the lowest value is 0.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

For Binary Classification accuracy calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

Where,

– **TP (True Positive):** The proposed model predicted ham (SMS is ham), and the real value is also ham.
– **TN (True Negative):** The Proposed model predicted spam (SMS is spam), and the real value is also spam.
– **FP (False Positive):** The proposed model predicted spam, but the real value is ham.
– **FN (False Negative):** The proposed model predicted ham, but the real value is spam.

**Precision** Precision is used to calculate and measure the correctness of classifiers [15, 39]. Precision can be calculated as the number of true positives divided by the sum of the number of true positives and the number of false negatives.

**Recall** Recall is used to calculate the completeness of classifier [4, 16]. Recall can be calculated as the number of true positives divided by the sum of the number of true positives and the number of false negatives.

$$Recall = \frac{TP}{TP \ + \ FN} \tag{11}$$

**F1-Score** F1 score shows the balance between Recall and precision [41]. In another way, F1-score is the harmonic mean between precision and Recall. Its value ranges from 1 to 0.

$$F - score = 2 \ * \ \frac{Precision \ * \ Recall}{Precision \ + \ Recall} \tag{12}$$

## 4 Study results

This section provides a discussion about experiments and results of evaluation parameters. For feature extraction, the TF-IDF and BOW are applied to data obtained after pre-processing. Support vector classifier, gradient boosting machine, random forest, Gaussian naïve Bayes, and logistics regression are applied individually on the result obtained from TF-IDF and bag of words, respectively. The precision, recall, and F1-score are applied

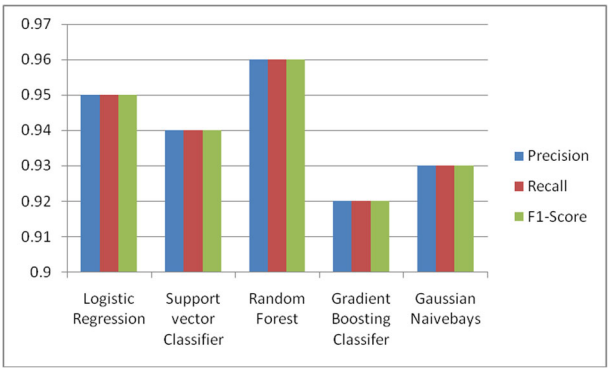



**Fig. 4** Machine learning models results using TF-IDF features

as evaluation parameters to check the accuracy of the proposed model. Figure 4 illustrates the result of the accuracy of all three testing parameters for TF-IDF.

Figure 4 shows the graph of testing results regarding each technique for TF-IDF. Random Forest shows the highest accuracy 96%. The lowest testing result is of Gradient Boosting Classifier that is 92%. Figure 5 shows the result of evaluation parameters on the results obtained from the BOW model.

Figure 5 shows the testing results regarding each technique for Bag of Words. This figure has clearly shown that Random Forest achieves the highest accuracy that is 96%. The lowest testing result is of Gaussian Naive Bayes. Table 7 shows the comparison of results of both keyword extraction techniques.

Table 7 shows the comparison of testing for TF-IDF and BOW. Overall, Random Forest shows the highest results in testing, and Gaussian Naive Bayes has shown lower results in evaluation parameters. Oversampling using SMOTE method is applied. Machine learning algorithms such as support vector classifier, gradient boosting machine, random forest, Gaussian naive Bayes, and logistics regression are applied individually on the keywords obtained from TF-IDF and Bag of Words.

Figure 6 shows the results of precision, recall, and F1-Score against machine learning algorithms. Support vector classifier shows better results as compared to others. Meanwhile,

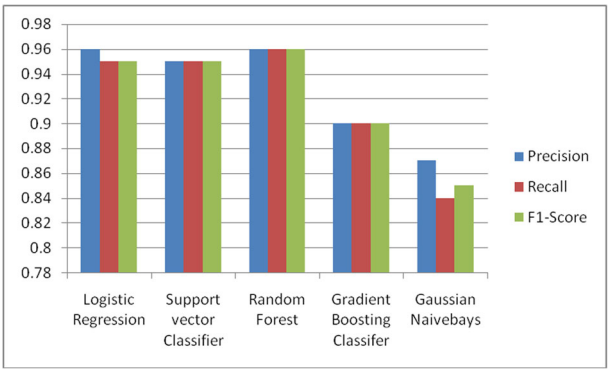



**Fig. 5** Machine learning models results using BOW features

**Table 7** Comparison of machine learning models results using TF-IDF and BOW features

| | TF-IDF | | | Bag of Words | | |
|---|---|---|---|---|---|---|
| Model | Precision | Recall | F1-Score | Precision | Recall | F1-Score |
| LR | 0.95 | 0.95 | 0.95 | 0.96 | 0.95 | 0.95 |
| SVM | 0.94 | 0.94 | 0.94 | 0.95 | 0.95 | 0.95 |
| RF | 0.96 | 0.96 | 0.96 | 0.96 | 0.96 | 0.96 |
| GB | 0.92 | 0.92 | 0.92 | 0.90 | 0.90 | 0.90 |
| GNB | 0.93 | 0.93 | 0.93 | 0.87 | 0.84 | 0.85 |

the Gaussian naïve bays show lower results among all applied machine learning algorithms. Figure 7 shows the results of machine learning algorithms applied to the keywords obtained from Bag of words.

Figure 7 shows the results of precision, recall, and F1-Score against machine learning algorithms applied on the keywords obtained from the bag of words. Support vector classifier and Gradient Boosting Classifier show better results as compared to others. On the other hand, Gaussian naïve bays show lower results among all applied machine learning algorithms. The comparison of TF-IDF and Bag of Words against each machine learning algorithm in oversampling is as follows:

Table 8 shows the results of precision, recall, and F1-Score against the machine learning algorithm for TF-IDF and Bag of Words. The random under-sampling method is applied, and machine learning algorithms used above are applied individually on the keywords obtained from TF-IDF and Bag of Words, respectively.

Figure 8 shows the results of precision, recall, and F1-Score against each machine learning algorithm applied to the keywords obtained from the TF-IDF. Random Forest shows better results than others, while Gaussian naïve bays shows lower performance among others. Afterward, machine learning algorithms are applied to the keywords obtained from Bag of words are shown as follows:

Figure 9 shows the results of precision, recall, and F1-Score against each machine learning algorithm applied to the keywords obtained from the bag of words. The comparison of TF-IDF and Bag of Words against each machine learning algorithm in under-sampling is as

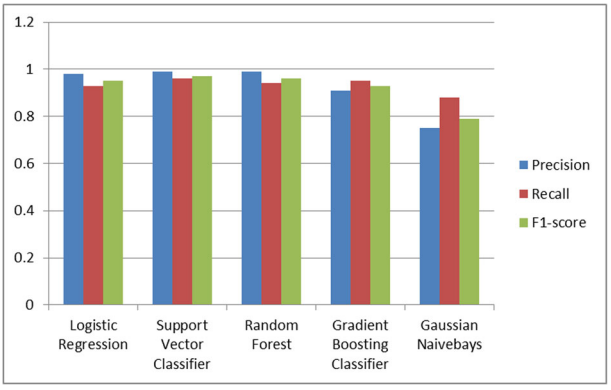



**Fig. 6** Machine learning models results using TF-IDF features (Over-sampling)

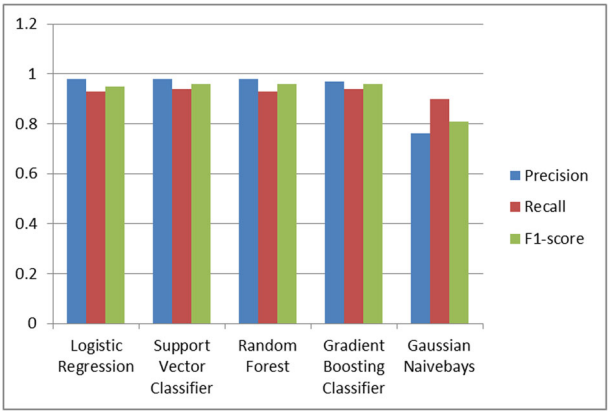



**Fig. 7** Machine learning models results using BOW features (Over-sampling)

follows: Table 9 Comparison of evaluation parameters of techniques for TF-IDF and Bag of Words (Under-sampling)

Table 9 shows the results of precision, recall, and F1-Score against machine learning algorithm for TF-IDF and Bag of Words in under-sampling.

The performance of machine learning models is evaluated by precision, recall, and F1-Score. TF-IDF shows better results as compared to the bag of words model in feature extraction techniques. Support Vector Machine, Random Forest shows better results in different circumstances of solving user classification problems.

### 4.1 Models performance results on another spam, ham dataset

To validate the performance of our proposed approach, we experiment on another spam dataset. The dataset contains the 5171 spam and ham emails obtain from [26]. The results on the Spam emails dataset using all approaches are shown in Table 10. According to the results of our proposed approach, RF with TF-IDF features and SMOTE oversampling technique outperform with 0.99 accuracy to all other stat of the art models. The significant performance of our proposed approach on another dataset shows the efficiency that this approach can be good on also other datasets.

**Table 8** Comparison of machine learning models results using TF-IDF and BOW features (Over sampling)

| | TF-IDF | | | Bag of Words | | |
|---|---|---|---|---|---|---|
| Model | Precision | Recall | F1-Score | Precision | Recall | F1-Score |
| LR | 0.98 | 0.93 | 0.95 | 0.98 | 0.93 | 0.95 |
| SVM | 0.99 | 0.96 | 0.97 | 0.98 | 0.94 | 0.96 |
| RF | 0.99 | 0.94 | 0.96 | 0.98 | 0.93 | 0.96 |
| GB | 0.91 | 0.95 | 0.93 | 0.97 | 0.94 | 0.96 |
| GNB | 0.75 | 0.88 | 0.79 | 0.76 | 0.90 | 0.81 |

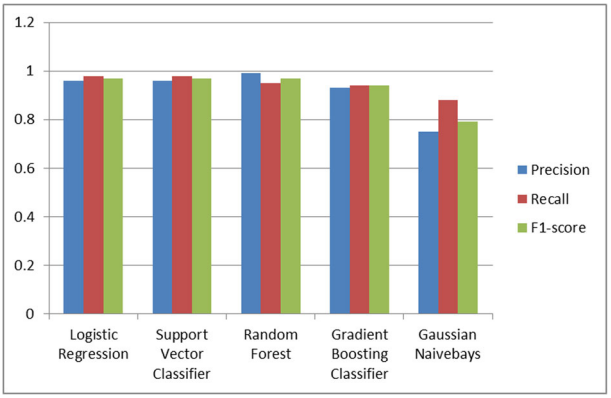



**Fig. 8** Machine learning models results using TF-IDF features (Under-sampling)

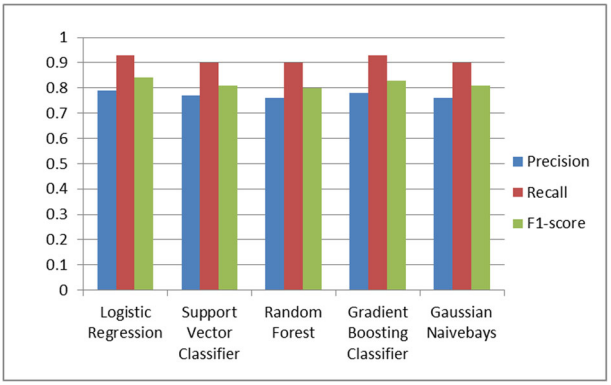



**Fig. 9** Machine learning models results using BOW features (Under-sampling)

**Table 9** Comparison of machine learning models results using TF-IDF and BOW features (Under-sampling)

| | TF-IDF | | | Bag of Words | | |
|---|---|---|---|---|---|---|
| Model | Precision | Recall | F1-Score | Precision | Recall | F1-Score |
| LR | 0.96 | 0.98 | 0.97 | 0.79 | 0.93 | 0.84 |
| SVM | 0.96 | 0.98 | 0.97 | 0.77 | 0.90 | 0.81 |
| RF | 0.99 | 0.95 | 0.97 | 0.76 | 0.90 | 0.80 |
| GB | 0.93 | 0.94 | 0.94 | 0.78 | 0.93 | 0.83 |
| GNB | 0.75 | 0.88 | 0.79 | 0.76 | 0.90 | 0.81 |

**Table 10** Performance results on spam email dataset

| | Orignial | | Over-sampling | | Under-sampling | |
|---|---|---|---|---|---|---|
| Model | TF-IDF | BOW | TF-IDF | BOW | TF-IDF | BOW |
| LR | 0.98 | 0.84 | 0.98 | 0.97 | 0.98 | 0.95 |
| SVC | 0.97 | 0.95 | 0.98 | 0.97 | 0.97 | 0.96 |
| RF | 0.97 | 0.97 | 0.99 | 0.98 | 0.97 | 0.97 |
| GBM | 0.95 | 0.95 | 0.96 | 0.96 | 0.95 | 0.95 |
| GNB | 0.96 | 0.94 | 0.94 | 0.92 | 0.93 | 0.87 |

### 4.2 Comparison with previous studies on spam filtering

In this section, we compare our proposed approach with previous studies that have used the same UCI dataset. For the comparison, we have filters latest studies such as study [50], which used the HMM and achieved a 0.959 accuracy score. The study [36] used the proposed CNN and LSTM models to achieve the highest accuracies 0.977, 0.953 respectively, on the UCI dataset. In another study [19], LSTM for spam and ham SMS filtering achieved 0.985 accuracy, and in [44], RF with the chi2 feature selection technique to achieve the best results. The ensemble approach gives the 97.5% accuracy score for spam ham filtering. Our approach provides the highest accuracy with the RF model, TF-IDF features, and SMOTE oversampling techniques compared to all these studies. The comparison with previous studies is shown in Table 11.

## 5 Conclusion and future work

SMS is the most common and widely used communication network now a day. Besides useful SMS, there are bulks of SMS that are spam sent from different companies and persons for the promotion of some offers. This spam SMS sometimes contains some malicious content which can be the cause for scams. The main goal of this research is to train the system so that it could be able to distinguish between ham and spam SMS. In the proposed approach, support vector classifier, gradient boosting machine, random forest, Gaussian Naive Bayes, and logistics regression are used with TF-IDF and BOW features. To avoid the models' over-fitting data re-sampling techniques such as SMOTE and random under-sampling have been used. Evaluation parameters Precision, Recall, and F1 score techniques are applied to check the validity of our proposed model. Random forest shows better accuracy of 99%

**Table 11** Proposed approach comparison with previous studies

| Ref. | Year | Model | Accuracy |
|---|---|---|---|
| [50] | 2020 | HMM | 0.959 |
| [36] | 2020 | CNN, LSTM | 0.977, 0.953 |
| [19] | 2021 | LSTM | 0.985 |
| [44] | 2020 | RF with Chi2 | 0.975 |
| This study | 2021 | RF with TF-IDF & SMOTE for SMS and Email | 0.990, 0.990 |

on balanced data using TF-IDF features, compared to other machine learning algorithms. The significant performance of random forest is because of its ensemble architecture, and oversampling also help the random forest by generating a balanced and large feature set for training. TF-IDF gives weighted features compared to BOW, which also impacts the random forest performance to achieve high accuracy. The proposed approach is helpful because it can automatically detect SMS categories. So, there is no need for human interaction for categorical purposes, and the proposed model will automatically detect the SMS category.

This research can be further explored by hybrid machine learning techniques to enhance the accuracy of results, which will be beneficial in categorizing SMS.

**Acknowledgements** The authors would like to thank the Department of Software Engineering, School of Systems and Technology, University of Management & Technology, for providing a research-oriented environment.

**Data Availability** The used dataset is publicly available on Kaggle. https://www.kaggle.com/uciml/sms-spam-collection-dataset/

## Declarations

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

## References

1. Abid MA, Mushtaq MF, Akram U, Mughal B, Ahmad M, Imran M (2020) Recommending domain specific keywords for twitter. In: International conference on soft computing and data mining, Springer, pp 253–263
2. Ahmed I, Guan D, Chung TC (2014) Sms classification based on naive bayes classifier and apriori algorithm frequent itemset. Int J Mach Learn Comput 4(2):183
3. Alkhazi B, DiStasi A, Aljedaani W, Alrubaye H, Ye X, Mkaouer MW (2020) Learning to rank developers for bug report assignment. Appl Soft Comput 106667:95
4. AlOmar EA, Aljedaani W, Tamjeed M, Mkaouer MW, El-Glaly YN (2021) Finding the needle in a haystack: On the automatic identification of accessibility user reviews. In: Proceedings of the 2021 CHI conference on human factors in computing systems, pp 1–15
5. Angeli A, Filliat D, Doncieux S, Meyer JA (2008) Fast and incremental method for loop-closure detection using bags of visual words. IEEE Trans Robot 24(5):1027–1037
6. Benevenuto F, Magno G, Rodrigues T, Almeida V (2010) Detecting spammers on twitter Collaboration, electronic messaging, anti-abuse and spam conference (CEAS), vol 6, p 12
7. Bo H, Xiao-Ling R, ZHANG CJ, Qin HQ, Chong-Hui G (2017) (2017) Telephone Traffic forecasting of electric system based on multi-factor decomposition. In: 3rd Annual International Conference on Electronics, Electrical Engineering and Information Science. Atlantis Press, EEEIS
8. Cernian A, Carstoiu D, Olteanu A, Sgarciu V (2016) Assessing the performance of compression based clustering for text mining. Econ Comput Econ Cybern Stud Res 50:2
9. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
10. Cormack GV, Hidalgo JMG, Sánz EP (2007) Feature engineering for mobile (sms) spam filtering. In: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval, pp 871–872
11. Dittman DJ, Khoshgoftaar TM, Wald R, Napolitano A (2014) Comparison of data sampling approaches for imbalanced bioinformatics data. In: The twenty-seventh international FLAIRS conference
12. Doma V, Kendre S, Bhagwat L (2018) Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach. arXiv:180908651
13. Duc GM, Manh L et al (2016) A novel method to improve the speed and the accuracy of location prediction algorithm of mobile users for cellular networks. Chuyê,n san Các công trình nghiên cu, phát trin và ng dng Công ngh thông tin và Truyn thông

14. Fallgren M, Abbas T, Allio S, Alonso-Zarate J, Fodor G, Gallo L, Kousaridas A, Li Y, Li Z, Li Z et al (2019) Multicast and broadcast enablers for high-performing cellular v2x systems. IEEE Trans Broadcast 65(2):454–463
15. Fang F, Wu J, Li Y, Ye X, Aljedaani W, Mkaouer MW (2021) On the classification of bug reports to improve bug localization. Soft Comput 25(11):7307–7323
16. Faris H, Ala'm AZ, Heidari AA, Aljarah I, Mafarja M, Hassonah MA, Fujita H (2019) An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks. Information Fusion 48:67–83
17. Fernández A, Garcia S, Herrera F, Chawla NV (2018) Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. J Artif Intell Res 61:863–905
18. Fraser JS, Wang WJ, He HS, Thompson FR (2019) Modeling post-fire tree mortality using a logistic regression method within a forest landscape model. Forests 10(1):25
19. Gadde S, Lakshmanarao A, Satyanarayana S (2021) Sms spam detection using machine learning and deep learning techniques 2021 7Th international conference on advanced computing and communication systems (ICACCS), vol 1, pp 358–362, 10.1109/ICACCS51430.2021.9441783
20. Gayathri B, Sumathi C (2016) An automated technique using gaussian naïve bayes classifier to classify breast cancer. Int J Comput Appl 148(6):16–21
21. Ghosh A, Maeder A, Baker M, Chandramouli D (2019). 5g evolution: A view on 5g cellular technology beyond 3gpp release 15. IEEE Access 7:127639–127651
22. Gómez Hidalgo JM, Bringas GC, Sánz EP, García FC (2006) Content based sms spam filtering. In: Proceedings of the 2006 ACM symposium on Document engineering, pp 107–114
23. Ishtiaq A, Islam MA, Iqbal MA, Aleem M, Ahmed U (2019) Graph centrality based spam sms detection. In: 2019 16Th international bhurban conference on applied sciences and technology. IEEE, IBCAST, pp 629–633
24. Jamil R, Ashraf I, Rustam F, Saad E, Mehmood A, Choi GS (2021) Detecting sarcasm in multi-domain datasets using convolutional neural networks and long short term memory network model. PeerJ Computer Science e645:7
25. Kaggle (2016) Sms spam collection dataset. https://www.kaggle.com/uciml/sms-spam-collection-dataset/. Accessed 20 Apr 2021
26. Kaggle (2021) Spam mails dataset. https://www.kaggle.com/venky73/spam-mails-dataset. Accessed 24 Apr 2021
27. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu TY (2017) Lightgbm: a highly efficient gradient boosting decision tree. Advances in neural information processing systems 30:3146–3154
28. Lee HY, Kang SS (2019) Word embedding method of sms messages for spam message filtering, IEEE, BigComp
29. Lee MC, Chang JW, Hsieh TC, Chen HH, Chen CH (2012) A sentence similarity metric based on semantic patterns. Adv Inf Sci Serv Sci 4:18
30. Lin WC, Tsai CF, Hu YH, Jhang JS (2017) Clustering-based undersampling in class-imbalanced data. Inf Sci 409:17–26
31. 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
32. Nagwani NK, Sharaff A (2017) Sms spam filtering and thread identification using bi-level text classification and clustering techniques. J Inf Sci 43(1):75–87
33. Nikam S, Chaudhari R (2017) A review paper on image spam filtering
34. Pavlopoulos S, Kyriacou E, Berler A, Dembeyiotis S, Koutsouris D (1998) A novel emergency telemedicine system based on wireless communication technology-ambulance. IEEE Trans Inf Technol Biomed 2(4):261–267
35. Ramsingh J, Bhuvaneswari V (2021) An efficient map reduce-based hybrid nbc-tfidf algorithm to mine the public sentiment on diabetes mellitus–a big data approach. J King Saud University Comput Inf Sci 33(8):1018–1029
36. Roy PK, Singh JP, Banerjee S (2020) Deep learning to filter sms spam. Futur Gener Comput Syst 102:524–533
37. Rupapara V, Rustam F, Amaar A, Washington PB, Lee E, Ashraf I (2021a) Deepfake tweets classification using stacked bi-lstm and words embedding. PeerJ Computer Science 7:e745
38. Rupapara V, Rustam F, Shahzad HF, Mehmood A, Ashraf I, Choi GS (2021b) Impact of smote on imbalanced text features for toxic comments classification using rvvc model. IEEE Access
39. Russo DP, Zorn KM, Clark AM, Zhu H, Ekins S (2018) Comparing multiple machine learning algorithms and metrics for estrogen receptor binding prediction. Mol Pharm 15(10):4361–4370
40. Rustam F, Ashraf I, Mehmood A, Ullah S, Choi GS (2019) Tweets classification on the base of sentiments for us airline companies. Entropy 21(11):1078

41. Safdari N, Alrubaye H, Aljedaani W, Baez BB, DiStasi A, Mkaouer MW (2019) Learning to rank faulty source files for dependent bug reports. In: Big Data: learning, analytics, and applications, international society for optics and photonics, vol 10989, p 109890B
42. Sajedi H, Parast GZ, Akbari F (2016) Sms spam filtering using machine learning techniques: a survey. Mach Learn Res 1(1):1
43. Shafi'I MA, Abd Latiff MS, Chiroma H, Osho O, Abdul-Salaam G, Abubakar AI, Herawan T (2017) A review on mobile sms spam filtering techniques. IEEE Access 5:15650–15666
44. Sisodia DS, Mahapatra S, Sharma A (2020) Automated sms classification and spam analysis using topic modeling. In: 2nd International Conference on Data, Engineering and Applications (IDEA), pp 1–6
45. Sohn DN, Lee JT, Rim HC (2009) The contribution of stylistic information to content-based mobile spam filtering. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pp 321–324
46. Speiser JL, Wolf BJ, Chung D, Karvellas CJ, Koch DG, Durkalski VL (2019) Bimm forest: a random forest method for modeling clustered and longitudinal binary outcomes. Chemometr Intell Lab Syst 185:122–134
47. Subramaniam T, Jalab HA, Taqa AY (2010) Overview of textual anti-spam filtering techniques. Int J Phys Sci 5(12):1869–1882
48. VRL N (2009) An unsupervised approach to domain-specific term extraction. In: Australasian language technology association workshop, vol 2009, p 94
49. Willig A, Matheus K, Wolisz A (2005) Wireless technology in industrial networks. Proc IEEE 93(6):1130–1151
50. Xia T, Chen X (2020) A discrete hidden markov model for sms spam detection. Appl Sci 10(14):5011
51. Zamel YK, Ali SA, Naser MA (2018) Analysis study of spam image-based emails filtering techniques. Int J Pur Appl Math 119(15):325–346

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

## Affiliations

**Muhammad Adeel Abid[1] · Saleem Ullah[1] · Muhammad Abubakar Siddique[1] · Muhammad Faheem Mushtaq[2] · Wajdi Aljedaani[3] · Furqan Rustam[4]**

Muhammad Adeel Abid
creativemind.adeel@gmail.com

Saleem Ullah
saleem.ullah@kfueit.edu.pk

Muhammad Abubakar Siddique
abubakar.ahmadani@kfueit.edu.pk

Muhammad Faheem Mushtaq
faheem.mushtaq88@gmail.com

Wajdi Aljedaani
wajdialjedaani@my.unt.edu

1 Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan, Pakistan

2 The Islmia University of Bahwalpur, Bahwalpur, Pakistan

3 University of North Texas, Denton, TX 76203, USA

4 Department of Software Engineering, School of Systems and Technology, University of Management and Technology, 54770, Lahore, Pakistan