Email spam detection using bagging and boosting of machine learning classifiers

Uma Bhardwaj* and Priti Sharma

Department of Computer Science and Applications,
Maharshi Dayanand University,
Rohtak, Haryana – 124001, India
Email: umabhardwaj90@gmail.com
Email: pritish80@yahoo.co.in
*Corresponding author

Abstract: The increase in the popularity, utility, and significance of electronic mails has also raised the exposure of spam emails. This paper endeavours to detect email spam by constructing an ensemble system using bagging and boosting of machine learning techniques. The dataset used for the experimentation is Ling-Spam Corpus. The system detects spam email by bagging the machine learning-based multinomial Naïve Bayes (MNB) and J48 decision tree classifiers followed by the boosting technique of converting weak classifiers into strong by implementing the Adaboost algorithm. The experimentation includes three different experiments and the results attained are compared with each other. Experiment 1 employs the individual classifiers, experiment 2 ensembles the classifiers with bagging approach, and experiment 3 ensembles the classifiers by implementing the boosting approach for the email spam detection. The effectiveness of the ensemble methods is manifested by comparing the evaluated results with individual classifiers in terms of evaluation metrics.

Keywords: email spam; text mining; Naïve Bayes; J48 algorithm; spam filtering; correlation based feature selection; bagging; boosting.

Reference to this paper should be made as follows: Bhardwaj, U. and Sharma, P. (2023) 'Email spam detection using bagging and boosting of machine learning classifiers', *I Int. J. Advanced Intelligence Paradigms*, Vol. 24, Nos. 1/2, pp.229–253.

Biographical notes: Uma Bhardwaj received her BCA in 2010 and MCA in 2013 from Maharshi Dayanand University, Rohtak. She is a Research Scholar at Maharshi Dayanand University Rohtak in the Department of Computer Science and Applications. Her area of research is "spam – ham mail classification". Her research interest includes data mining, text mining, character recognition, and natural language processing.

Priti Sharma is an Assistant Professor in the Department of Computer Science and Applications in Maharshi Dayanand University, Rohtak. She received her PhD in Computer Science from Kurukshetra University, Kurukshetra. Her research interest includes software engineering, software re-engineering, data mining, and software metrics. She has teaching experience of 10 years having approximately 40 publications.

1 Introduction

Email is considered as the lifeline for the modern era people as it is easy to use, user-friendly environment, cheap in cost and instant delivery of information (McIver and Birdsall, 2002). It has made communication flexible and convenient (Douzi et al., 2017). Email system holds a strong place in both the personal and business world. In the world of business, an email is a primary communication approach and can be considered as an official document too (Shen and Li, 2013). Despite several interpersonal communication systems (social networks, instant chat messengers, etc.) are available, the use of emails continues to grow. The daily email traffic (total number of emails sent and received daily) of both consumer and business emails has been estimated to continue its growth with an average rate of 4.4%, reaching daily email traffic of 319.6 billion emails by the end of the year 2021 (Email Statistics Report, 2017–2021). Table 1 presents the estimated annual growth in daily email traffic along with worldwide email users (Email Statistics Report, 2017–2021).

Table 1 Worldwide email forecast (daily traffic and users)

Year	*Daily email traffic* (in billions)	*Growth in daily email traffic* (%)	*Worldwide users* (in millions)	*Growth in worldwide users* (%)
2018	281.1	4.5	3823	3
2019	293.6	4.4	3930	3
2020	306.4	4.4	4037	3
2021	319.6	4.3	4147	3

Source: Email statistics report (2017–2021)

Unfortunately, the increase in dependency on emails has also increased the exposure of spam emails. Spam is inapposite information crafted typically to broadcast over the internet with a motive of phishing, advertising or especially spreading malware. Malware can be available in any form of spyware, Trojan horses, worms, and viruses that can badly harm the legitimate users (Gandotra et al., 2019). Thus, spam email can be defined as "Unwanted or unsolicited email, sent indiscriminately by a sender with no current relationship with the recipient" (Cormack and Lynam, 2005). The spam email senders are called as spammers (Fazil and Abulaish, 2018). The purpose of sending spam emails by spammers could be promoting fraud scheme, advertising product, or broadcasting computer malware with the motive to seize the recipient's computer (Mangalindan, 2002). Spam emails not only annoy the information users but also affect the user's storage space, time and communication channel bandwidth (Nizamani et al., 2014). In other words, spam emails financially influence business organisations. The rapid growth of email accounts generation also influenced the spammers to increase the number of spamming emails. The recent report discussed by Talos Intelligence (2019) for the spam and authorised emails is illustrated in Table 2. In Table 2, the email statistics for the total number of emails and spam email from August 2018 to July 2019 is presented. This indicates the rise in the total volume of emails and email spam as well.

Table 2 The statistics of spam emails among the total number of emails

Month, Year	*Total number of emails* (in Billion)	*Total number of spam emails* (in Billion)
August 2018	303.03	258.49
September 2018	354.5	301.95
October 2018	340.07	289.99
November 2018	302.2	257.75
December 2018	364.25	311.24
January 2019	339.27	289.71
February 2019	239.22	204.19
March 2019	346.64	295.67
April 2019	489.34	416.78
May 2019	430.96	366.51
June 2019	539.22	459.4
July 2019	496.11	422.49

Table 2 indicates that there is not even the availability of 25% of legitimate emails from the total number of emails sent and received. Further, sub-section presents the motivation to consider the research work of email spam.

1.1 Motivation

The email spam is not a novel concept as it was started in 1978 by Gary Thuerk (Bawm and Nath, 2014) who have manually sent the first spam message on ARPANET to 400 (ARPANET, 2006) people to seek the attention of people about the introduction of their DECSYSTEM computer products. But the increasing number of spamming is a thing to worry as Talos Intelligence has reported the spam email of 85.44% from the total worldwide email communication in their latest report of July 2019. Moreover, email spam is not limited to wastage of receiver's time, energy and bandwidth but it can also redirect the users to websites that contain phishing or malware content which can disrupt the computer system of the receiver. Email spam can also lead to financial frauds and terrorism activities by taking the personal information from the receiver with some advertisement notification. In the field of internet technology, email spam acts as the plague of networking technology. Spam emails not only include the bulk unwanted useless emails but email spam is also a type of spreading the various spyware, Trojans, worms, and viruses, etc. (Drake, 2005). Another category of email spam is the blocking network with huge traffic and denial of service attacks. These lead to reducing the internet speed and data interruption. The most affected users are the employee of an organisation who has to spend a lot of time to handle the email spam. This not only wastes the time and energy of employees but productivity of the organisation also affected. As per the research report of Ferris (Sampson, 2003), usual employee wastes around 4000 USD and time of 115 h per year to sort the useful emails. Spammers do not reveal their identity but send emails in bulk with different user address (Email Assessment, 2005). To reduce this, some users block the spammers or filter out the spam emails in a different folder.

The afore-mentioned facts, the continuous growth of email spam, and current statistics of email spam urge to improve the exiting concepts of autonomous email spam detection. There are various approaches for email spam detection such as greylisting, scanning message headings, detecting bulk messages, analysing the user behaviour and preferences, blacklisting, and content-based email spam detection. Greylisting method directly rejects the spam emails with an error message back to the sender. Scanning message heading methods scans the heading of the email and try to detect the spam content. Detecting bulk email method involves a number of receipts to detect the receiving of the same email with multiple users. The spam emails can also be extracted by analysis and extraction of features based on the user behaviour and preferences (Takashita et al., 2008). Blacklisting method uses the IP address method to track and detect the email spam. The content-based method uses the textual content in the form of training and testing of database and detects the email spam using different algorithms. In this research, the content-based method is used for email spam detection using ensemble methods of machine learning based classifiers. The methods of machine learning are ensembled as the machine learning classifiers performs better than the traditional rule based methods (Yahya and El-Bashir, 2014).

1.2 Contribution

This paper conducts three experiments for email spam detection. Initially, machine learning based individual concepts of Naïve Bayes and J48 (Decision Tree approach) are employed. Then, both the methods are ensembles using bagging approach in experiment 2. But the bagging of machine learning concepts may lack due to parallel processing of data in bagging. Final experiment 3 is performed with Adaboost using the boosting approach. Boosting concept boosts the weak leaner with the property of strong learner. In all three experiments, the database of Ling Spam database is used and results are observed.

The key contributions of the present work are:

- Machine learning based individual and ensemble-based approaches of bagging and boosting are used for the email spam detection.
- The implementation of the individual concepts, bagging, and boosting methods is conducted on Ling Spam Database.
- The evaluation of all the three experiments is performed in terms of evaluation parameters of precision, recall, accuracy, F-measure, true negative rate (TNR), false negative rate (FNR), and false positive rate (FPR).
- Further, evaluated results of ensemble methods are compared with individual methods.

1.3 Organisation of the paper

The research paper is organised into six sections. In the current section, the discussion of the concepts related to the basics of email spam detection, statistics of growing users and spam of email, motivation, and contribution for this research work is presented. Section 2 presents the work related to email spam classification with a detailed description of the

method, dataset, and key features. Section 3 discusses the considered database of Ling Spam database used for experimentation. Section 4 presents the experimentation with individual methods, bagging and boosting concepts. This section also shed some light on the basic of the considered concepts. All three experiments are discussed in this section. Section 5 presents the results and discussion with evaluation parameters of precision, recall, accuracy, F-measure, TNR, FNR, and FPR. The comparisons of all the experimental results are also discussed in this section. Section 6 concludes the research work based on the experimental simulation with some future references.

2 Related work

In today's world, although email is one of the efficient and convenient sources of conversation, the increasing user accounts also increase the number of spam emails in the daily routines. There are various available methods and techniques based on machine learning and computational intelligence concepts.

In 2011, Renuka et al. (2011) have used different machine learning algorithms like multilayer perceptron (MLP) classifier, Naïve Bayes approach, and J48 classifier for the classification of emails as spam or non-spam. Authors performed the experimentation on different annotated email dataset collected from different email ids over a period of two months. To increase the performance of Naïve Bayes approach, Filtered Bayesian Learning (FBL) is considered. Further, Prilepok et al. (2012) have used the data compression algorithm and particle swarm optimisation (PSO) for the email spam classification. Here, Bayesian filter is improved with a data compression algorithm for the email spam classification. From evaluated results, authors reported the efficient results for both the approaches but PSO performs lacks in practical performance. Behjat et al. (2012) have used MLP for the classification of email spam and Genetic Algorithm for the feature selection. Experimentation is performed on LingSpam dataset and results are evaluated in terms of a number of extracted and selected features and accuracy. Authors have reported the outperformed performance of MLP with genetic algorithm. Further, Trivedi and Dey (2013) have enhanced the concept of Genetic Programming and used it for email spam filtration. Greedy Step size search approach is used for feature selection and shows efficient results. Datasets of Enron Email (Version 5& 6) and SpamAssassin are used for the experimentation and evaluated in terms of Accuracy, F-value and FPR. Enhanced Genetic Programming also performs well in terms of accuracy and FPR in comparison with other considered concepts.

Further work on machine learning classifiers has been observed by Shams and Mercer (2013). Classifiers of Naïve Bayes (NB), Support Vector Machine (SVM), ADABOOSTM1, Bagging, and random forest (RF) are used for the experimentation on Enron-Spam, LingSpam, SpamAssassin, and CSDMC2010. From the evaluated results, bagging approach dominates over other considered concepts. Wijayantoa and Takdir (2014) have used fuzzy c-means approach for the email spam classification. The database of SpamBase is used for the experimentation and results are evaluated in terms of accuracy and shows better performance of fuzzy c-means clustering in terms of other considered concepts. Renuka and Visalakshi (2014) have used support vector machine (SVM) for the classification of Email Spam detection along with the use of latent

semantic indexing (LSI) for feature selection. Dataset of Ling Spam Email Corpus is used. In this process, initially data preprocessing is performed then feature extraction is performed with TF-IDF approach. Further, Harisinghaney et al. (2014) have used text and image based data for the email spam classification. Methods of Naïve Bayes, KNN algorithm and Reverse DBSCAN algorithm are used for the experimentation with Enron Corpus's dataset. Authors reported better results with Naïve Bayes approach and preprocessing steps as compared to without preprocessing. The disadvantage of this concept is huge time consumption for text filtration is mentioned by authors.

In 2015, Idris et al. (2015) have introduced the improved concept of email spam detection with PSO and negative selection algorithm (NSA). The local outlier factor (LOF) is adapted to analyse the fitness function of the proposed NSA-PSO approach. The satisfactory accuracy results were noted by authors for the proposed NSA-PSO approach. Mohamad and Selamat (2015) focused on the hybridised approach of Rough set theory and term frequency inverse document frequency (TF-IDF) for the feature selection. Authors used a hybrid concept for experimentation on Malay and English language. The unessential words of the dataset were removed using the rough set exploration system (RSES) tool. Faris et al. (2015) adapted the neural network to improve email spam detection outcomes. The authors used the Krill Herd algorithm to train the network. The results outcomes with proposed network indicate better efficacy as compared to other considered concepts of back-propagation and genetic algorithm. Faris et al. (2016) also exploited PSO and random forest algorithm for the detection of spam mails. Kaur and Sharma (2016) amalgamated the decision tree with PSO algorithm for the improvement of email spam detection. The results of the proposed integrated concept are compared to k-means and SVM approach with and without unsupervised filtration on the basis of evaluation parameters of correctly classified ratio, mean absolute error, and F-Measure. The authors have not discussed the feature extraction approach in their research work.

Feng et al. (2016) have improved the detection of email spam by exploiting the SVM-NB algorithm. As an individual algorithm of SVM and NB are not strong enough for optimum classification, so both the concepts integrated where SVM creates the hyperplane separations among the available feature sets and NB handles huge database. Kumaresan and Palanisamy (2017) have added the step size feature in the algorithm of Cuckoo Search and integrated with the SVM approach to improve the result outcomes for the detection of email spam. The results outcomes indicate the improved results for the proposed approach in comparison with CS-SVM approach. Olatunji et al. (2017) considered machine learning concepts of Extreme Learning Machines (ELM) and SVM for the classification of email spam. Authors reported better performance of SVM in terms of accuracy but ELM approach dominates in terms of time taken. Further, ELM and SVM concepts are also compared with Fuzzy logic, BART, NSA, PSO and NSA-PSO concept and shows better results than others for the classification of email spam.

In 2018, Chawathe (2018) have used the fuzzy rule (FURIA) to improve the security of the email system. Here, fuzzy rules-based system is designed to detect the email spam and database of SpamBase is used for the experimentation. The author observed the comparable performance of the proposed concepts with other concepts. Naem et al. (2018) have used the predictive model of ALO-Boosting which is the combinational method of antlion optimisation (ALO) and boosting approach. The authors have analysed the performance of the method by experimentation on the CSDMC2010 and

SpamAssassin dataset. The result values indicate the higher classification accuracy as compared to SVM, KNN, bagging, and combination of ALO with mentioned methods. Gupta et al. (2019) have considered the ensemble learning approach for the classification of email and SMS spam using the machine learning classifiers of the Decision Tree (DT), Bernoulli Naïve Bayes (BNB), multinomial Naïve Bayes (MNB), and Gaussian Naïve Bayes (GNB). The authors have used the voting ensemble for the classification with different combinations of mentioned classifiers. The experimentation was conducted on the dataset collected from the UCI website. The authors have achieved higher performance accuracy result with a combination of DT, BNB, and GNB classifiers in case of SMS dataset and a combination of all the classifiers in case of email dataset. Chikh and Chikhi (2019) have initially improved the NSA with k-means clustering and then combined with fruit fly optimisation (FFO) for the detection of email spam. The combined approach is termed as CNSA-FFO approach. The evaluated results indicate the performance efficacy of CNSA-FFO approach as compared to NSA and NSA-PSO algorithm. Faris et al. (2019) have hybridised the concept of genetic algorithm with random weight network. Authors performed the experimentation on the database of CSDMC2010 Corpus, LingSpam, and SpamAssassin. Moreover, the authors reported the important features for email spam detection are payload-body, header features, payload-readability, and payload-lexical.

From the existing work on email spam detection using machine learning techniques and computational intelligence, it has been analysed that the individual concept works less efficiently. The individual concepts lack due to the drawback of some characteristic in an individual approach, but the combinational approaches can perform better in comparison with individual concepts. Also, the evaluation using an individual method illustrates lesser performance accuracy as compared to ensemble methods. This research gap can be fulfilled by integrating the multiple methods. In the present scenario, researchers are focusing more on the integration of multiple methods instead of focusing on individual approach. This urges to consider the ensemble-based methods of bagging and boosting to work for email spam detection.

Next section discusses the considered database of Ling Spam database. The concept of email spam detection is performed by experimentation on the Ling Spam database.

3 Database

In this research work, Ling-Spam Corpus is used for the experimentation. This database consists of legitimate and spam emails collected from the different scientific and professional linguistics by Androutsopoulos et al. (2000). There are four sub-directories of database based on the stop-word list and lemmatiser. These categories are bare, lemm, lemm_stop, and stop. Further, each category contains 10 folders which contains legitimate and spam emails. Spam emails are available with the name containing ‘spmsg’ and all others are legitimate (ham) emails. The distribution of these categories based on availability of the stop-word list and lemmatiser is presented in Table 3. Each sub-directory consists of 2412 legitimate emails and 481 spam emails which make a total of 2893 emails.

Table 3 Distribution of ling-spam corpus categories

Corpus category	*Lemmatiser*	*Stop-word list*
Bare	✘	✘
Lemm	✔	✘
Lemm_stop	✔	✔
Stop	✘	✔

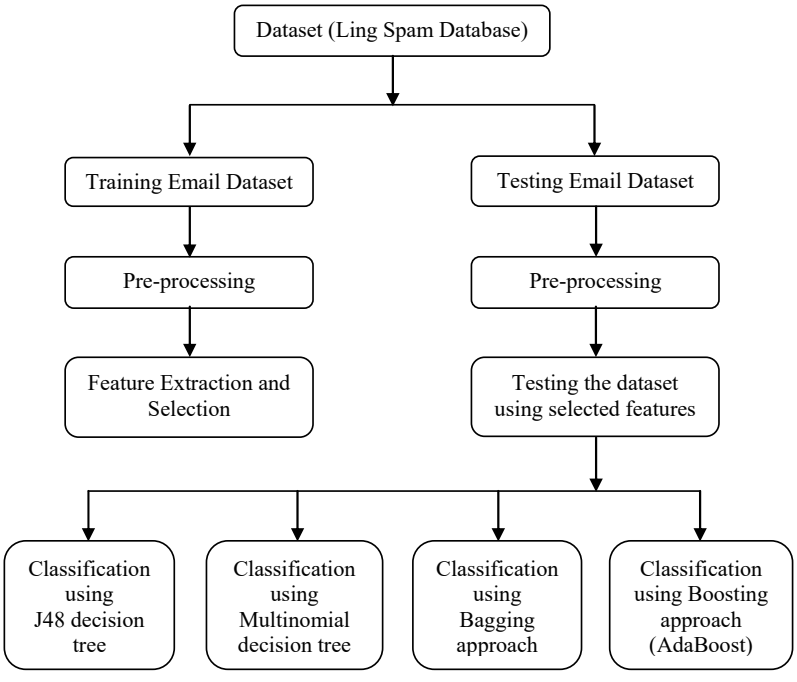
Here, the symbol ✔ indicates enabled and symbol ✘ indicates disabled.

Next section presents the working process of email spam detection using individual methods, bagging and boosting processes.

4 Research methodology

All the three experiments of individual classifiers, bagging and boosting for email spam detection are discussed in this section. The overall process of email spam detection is categorised into three modules of pre-processing, feature extraction and selection, and Classification. This overall process is presented in Figure 1.

Figure 1 Overall email spam classification experimentation



The experimentation is performed on each category of Ling-Spam Corpus. Among each category of the database, eight parts (folders) are used for the training and two parts are considered for testing. The modules of email spam detection are discussed here.

4.1 Pre-processing

The initial step of email spam detection is pre-processing of the database as the considered database is available in raw form. The first step is the filtration of text emails from the image-based emails. Then, the tokenisation of the corpus is performed in which each word of email corpus is considered as the individual token. These collections of n emails containing m tokens (words) after tokenisation are illustrated in Table 4. Each email is represented in the following manner with equation (1).

$$email_i = \left(t_{i1}, t_{i2}, \ldots, t_{im}\right) \tag{1}$$

where, t_{ij} describe the token frequency for the token t_j in $email_i$. Their values are evaluated in terms of binary frequency order.

Table 4 Representation of emails

	t_1	t_2		t_m
$email_1$	a_{11}	a_{12}	…	a_{1m}
$email_2$	a_{21}	a_{22}	…	a_{2m}
…	…	…	…	…
$email_n$	a_{n1}	a_{n2}	…	a_{nm}

Further, these token are considered to remove the numeric digits from the textual data to decrease the search space. Although the search space can be further reduced by removing the stops-words and applying the lemmatisation process, the database already consists of four categories based on the presence and absence of stop-words list and lemmatiser. There is a total of 258 features in terms of tokens are considered for the processing and frequency count of each feature is calculated. The pre-processing of both the training and testing is performed separately. For each category of Ling-Spam Corpus, eight parts are considered for training and two parts for the testing.

4.2 Feature extraction and selection

The efficiency of email spam detection system greatly depends on the features (Menghour and Souici-Meslati, 2014). The working of email spam detection depends on the assumed feature of differentiation in the content of the legitimate email from the spam email. The feature set consists of numerous features such as document length, inappropriate words, alphanumeric words, frequency count, spelling or grammatical errors, language, etc. There are a total of 258 features extracted to classify the spam and legitimate email. In this proposed system, the correlation feature selection (CFS) method is adapted to identify the superlative features from the set of available features that can help to improve the system efficacy. CFS approach considers the assumption that "Good feature subsets contain features highly correlated with the classification, yet uncorrelated to each other" (Chandrashekar and Sahin, 2014).

Initially, a bag of words as tokens are assessed as the feature set. The numbers of words per document are evaluated using the term frequency method. Then, the words with less than threshold frequency are removed to reduce the search space. Further, the concept of CFS is applied for the feature selection which selects the required feature set. The CFS approach selects only relevant features to specified class among the set of

features. If there are a number of classes c, number of features k, and feature sub-set S having features f, then evaluation of CFS can be elaborated as mentioned in equation (2).

$$CFS = max_{S_k}\left[\frac{r_{cf_1}, r_{cf_2}, r_{cf_3}, \ldots, r_{cf_k}}{\sqrt{k + 2\left(r_{f_1 f_2} + \ldots r_{f_i f_j} + \ldots r_{f_k f_1}\right)}}\right] \tag{2}$$

Here, r_{ff} and r_{cf} is the average of feature-feature correlation and feature-class correlation respectively.

4.3 Classification

All three experiments of email spam detection using individual machine learning approaches, bagging, and boosting concepts are discussed in this section. In all the experiments, the used methods are discussed along with some basic concept of the respective method. Experiment 1 is sub-categorised into two sections of Experiment 1.1 which include the email spam detection using MNB approach and Experiment 1.2 which includes the email spam detection using J48 decision tree algorithm. Further, Experiment 2 presents the bagging concept using the MNB approach and J48 decision tree algorithm. Experiment 3 presents the boosting concept using Adaboost approach.

4.3.1 Experiment 1.1 (Multinomial Naïve Bayes classifier)

Naïve Bayes classifier is multiclass probabilistic machine learning based classifier that considers Bayes theorem for the classification (Yadav et al., 2019). The attribute of strong independence is adapted by Bayes classification (Tang et al., 2016). The available instances are used to evaluate the class probability and the class probability closer to the rear end is exploited by the classifier. The Naïve Bayes classifier as a supervised learning approach is elaborated in equation (3).

$$P(y|x) = \frac{P(x|y)\ P(y)}{P(x)} \tag{3}$$

where x is the set of feature vectors (x_1, x_2, x_3, …, x_n) and y stands for the class variable with m possible outcomes (y_1, y_2, y_3, …, y_n). $P(y|x)$ is the posterior probability which depends on the likelihood of the feature set or attribute value belonging to particular class $P(x|y)$, $P(y)$ is the prior probability and $P(x)$ is the evidence depending on the known feature variables.

In this research work, the MNB classifier is used that represents the data in the format of word vectors (Kibriya et al., 2004). For each y class, the parameterised distribution by vectors $\theta_y = (\theta_{y1}, \ldots, \theta_{yn})$ where θ_{yl} is the probability $P(x_i|y)$ of feature i in any specified instance belongs to class y and n represents features. The mathematical representation of MNB classifier is elaborated in equation (4).

$$\hat{\theta} yi = \frac{N_{yi} + \alpha}{N_y + \alpha n} \tag{4}$$

The working of MNB classifier is rigorously explained by considering an instance of email that contains the word ‘lottery’. The users would know the possibility of the word

'lottery' is spam. The detection of email spam by considering MNB classifier calculates the email classification probability for the 'lottery' word as expressed in equation (5).

$$p(s|w) = \frac{p(w|s)\,p(s)}{p(w|s)\,p(s) + p(w|h)\,p(h)} \tag{5}$$

Here w stands for the 'lottery' word, s and h stands for email spam and ham respectively. The probability of email belongs to spam class that contains the word 'lottery' is $p(s|w)$. It depends on the overall probability of any email belonging to spam class $p(s)$, the probability of occurrence of word 'lottery' in ham emails $p(w|h)$, the overall probability of any email belonging to ham class $p(h)$, and the probability of occurrence of word 'lottery' in spam emails $p(w|s)$.

4.3.2 Experiment 1.2 (J48 decision tree)

In the recent years, the decision tree has becomes one of the frequently used machine learning method (Bresfelean, 2007). The hierarchy of decision-tree is tree based structure having terminal nodes indicate decision outcomes and non-terminal nodes present the test attributes. Although the decision tree approach is moderately vulnerable to noise and generate only single outcomes, it has an advantage that it can easily solve the classification problems with graphical representation. There are various decision tree algorithms such as C4.5, ID3, CHAID, and classification and regression tree. Among these versions, C4.5 (J48 decision tree) is the refined version of ID3 and possesses good classification accuracy.

The decision tree generated using J48 depends on the training data attribute values for the classification of the new data item. J48 follows the concept that after splitting the data into multiple sets, each feature attribute of data can be used to form a decision. The selected features of the email tokens of training data are considered as the leaf nodes of the decision tree. In the test case, if the near feature qualifies the label condition of feature node, then the level of that feature node is lifted up in the same decision tree branch. Gradually, it generates two branches of the tree with the available and lifted feature nodes. J48 uses the entropy function to generate rules of a decision tree with the help of target emails. From the available test dataset, J48 uses entropy function to test the classification of emails as described in equation (6)

$$Entropy(Email) = -\sum_{j=1}^{n} \frac{|Email_j|}{|Email|} \log_2 \frac{|Email_j|}{|Email|} \tag{6}$$

where $Email$ can be unigram, bigram, and trigram. Entropy evaluates the prediction of email as the spam or legitimate email with the concept of J48 decision tree.

The algorithm works recursively until each data attribute is processed and categorised i.e., the features extracted with the help of this algorithm are the best possible features belonging to the particular class data.

4.3.3 Experiment 2 (Bagging)

Bagging is derived from the term *bootstrap aggregating* and one amongst the easiest and earliest developed ensemble-based methods. In Bagging, multiple weak classifiers are

ensembles to improve the classification process by reducing the variance of classification error. In this research work, the bagging process is performed with machine learning based algorithms of MNB and J48 decision tree classifier. The bagging approach is further illustrated by considered total k number of samples. Among these k samples, n samples are selected randomly and different bags are created with the instructive iteration process. Further, classes are predicted based on the votes of classification with each bag. Bagging process consumes lesser computation time as it is a parallel process and training database distributed in different small sample sets. Using the base learner, the decision has been generated from each sub-sample set and aggregated to generate the overall result. In this approach, multiple models are generated by dividing the email dataset is randomly divided into separate sample email datasets. The considered database contains a total of ten subfolders of emails which are further subcategorised into eight folders for training and two folders for testing. The emails contain in these eight folders are distributed among the two classifiers of MNB and J48 decision tree classifiers. The overall system's result is the average of the result of the two classification algorithms. J48 decision tree algorithm and MNB are used for the multi-class learning and for the classification. The classification result considered is the average of the predicted values. The pseudo-code for the Bagging approach is mentioned with Algorithm 1.

Algorithm 1

Input: Classification model, training samples, number of iterations

Output: Results P_ϵ

Parameter Initialization: Consider prediction set as $P_1 \ldots P_q$, classification algorithm A, m number of learners, n number of training samples, and ensemble $\in = \emptyset$;

Training

for j = 1 to m;

generate the boostrap samples D_i from the n number of training samples

build the learner C_i to train the algorithm A based on boostrap samples D_i

ensemble the learner C_i with learning set as follows: $\in = \in \cup\ C_i$

return the ensemble $\in$;

end for

Testing

Evaluate the predictions $C_j\ (i)$ based on the novel instances i by applying $C_1 \ldots C_m$

Evaluate the prediction based on the ensemble as follows:

$$P_\in = \arg max_k \sum_{j=1}^{n} X(C_j(i) = P_k)$$

4.3.4 Experiment 3 (Boosting-AdaBoost)

Boosting can be defined as the ensemble method that has the capability to build a strong classifier using two or more weak classifiers. In Boosting, a series based process is followed with the first model to classify the training sample results, further introducing the second model to rectify the errors from the first model and continues until the perfect rectification of training samples. Boosting reduces both the bias and variance of the

classification and improves the classification results. Boosting processes the data samples with its weight values and weight of such samples are increased which are found to be misclassified samples so that the focus of base training algorithm can be diverted to such samples. The computation time of the boosting process is more as compared to bagging and boosting is sensitive to noise. In this research work, Adaboost is used for the boosting of classifiers Naïve Bayes and J48 decision tree classifiers. Adaboost was the first successful method developed to boost the binary classification. Originally, Adaboost was introduced with the name AdaBoost. M1. It is also referred with the name discrete AdaBoost as it is used for classification instead of regression. Sometimes, Naïve Bayes algorithm lacks for the classification of contextual emails. This drawback is recovered with the J48 based decision tree algorithm with the help of Adaboost. Moreover, individual J48 decision tree lacks in case of a noisy and long email. This increases the space complexity and makes the classification process slower. To overcome this drawback, Adaboost adds the property of MNB approach. Adaboost also uses the property to focus more on misclassified instances by increasing its weight value. Here, adaptive resampling technique is used to select the training sets. In each iteration, weights are assigned to the datasets so that misclassified datasets can be considered with higher priority in the next generation. The overall classification is the weighted sum of all the ensemble predictions. The pseudo-code for the Adaboost is presented in Algorithm 2.

Algorithm 2

Input: Classification model, training samples, number of iterations

Output: Results h_{fin}

Parameter Initialization: Consider the weak learning algorithm *WeakLearn*, the *m* number of sample sequences $((x_1 y_1) \ldots (x_m y_m)$, for $y_i \in Y = \{1, \ldots, k\}$, with number of iteration *T*, and $D_i = \frac{1}{m}$ for all *i*.

Training

for *t* = 1 to *T*;

apply WeakLearn for the database samples D_i

Obtain the hypothesis of result $h_t : X \to Y$

Evaluate the error of previous hypothesis results $h_t : \epsilon_t = \sum_{i=h_t(x_i)\neq y_i} D_t(i)$.

If $\epsilon_t > {}^1/_2$, then set $T = t - 1$ and end the loop.

Set $\beta_t = {}^{\epsilon_t}/_{(1-\epsilon_t)}$

Update the database sample distribution D_i: $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t, & if\ h_t(x_i) = y_i \\ 1, & Otherwise \end{cases}$

where, Z_t is the normalization constant

end for

Testing

The final hypothetical results can be evaluated as follows:

$$h_{fin}(x) = \arg max_{y \in Y} \sum_{t:h_t(x)=y} \log \frac{1}{\beta_t}$$

Next section discusses the evaluated results in all the three experiments of classification with individual methods, bagging and boosting processes. Moreover, the comparison of the results with existing concepts is also discussed.

5 Results and discussion

The Experimental results are evaluated for each category of Ling Spam Corpus. The evaluation parameters, evaluated results with all the three experiments, and four database categories, and comparison of results with the individual concept are discussed in this section.

5.1 Evaluation metrics

The efficacy of the proposed system is accessed by calculating the performance metrics. The evaluation metrics of precision, recall, accuracy, F-measure, TNR, FNR, and FPR are evaluated to analyse the efficacy of the proposed email spam detection system. The formulation of these parameters depends on the value of true positive (TP), false negative (FN), false positive (FP), and true negative (TN).

True positive (TP) indicates the number of emails predicted as spam by the system and the actual value of emails is also spam. True negative (TN) indicates the number of emails predicted as legitimate and the actual value of emails is also legitimate. False positive (FP) indicates the number of emails predicted as spam by the system and the actual value of emails is legitimate. False negative (FN) indicates the number of emails predicted as legitimate by the system and the actual value of emails is spam. Based on these prediction metrics, the formulation of precision, recall (sensitivity/true positive rate), accuracy, F-Measure, TNR/specificity, FPR, and FNR is discussed.

Precision defines the efficacy of classifier. It indicates the probability of spam email detection with true value using the classifier. The formulation of precision is presented in equation (7).

$$P = \frac{TP}{TP + FP} \tag{7}$$

Recall indicates the possibility of actual detection of email spam. It is the positive labelled data returned by the system classifier out of total class data. The recall is also popular with the names true positive rate and sensitivity. The formulation of the recall is presented in equation (8).

$$R = \frac{TP}{TP + FN} \tag{8}$$

F-Measure indicates the overall positive performance of the classifier. It is evaluated using the evaluated values of precision and recall. The formulation of the recall is presented in equation (9).

$$F = 2 \,.\, \frac{Precision \,.Recall}{Precision + Recall} \tag{9}$$

Accuracy indicates the ratio of the positive predicted values to total data values. The formulation of the accuracy is presented in equation 10.

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

True negative rate (*TNR*) indicates the ratio of the correctly identified legitimate emails to the total number of legitimate emails. TNR is also known as Specificity. The formulation of the TNR is presented in equation (11).

$$TNR = \frac{TN}{TN + FP} \tag{11}$$

False negative rate (*FNR*) indicates the number of miss of legitimate. The formulation of the FNR is presented in equation (12).

$$FNR = \frac{FN}{FN + TP} \tag{12}$$

False positive rate (*FPR*) indicates the ratio of the incorrect identification of legitimate email to the total number of legitimate emails. The formulation of the FNR is presented in equation 13.

$$FPR = \frac{FP}{FP + TN} \tag{13}$$

The above-mentioned parameters are evaluated for the performance assessment of all the experiments with respect to each category of Ling Spam Corpus.

5.2 Result evaluation

The results are evaluated for the four categories of the Ling Spam Corpus database: Bare, Lemm, Lemm-Stop, and Stop. There are total of ten folders of emails which contain both the legitimate and spam emails. Among these ten folders, eight folders are considered for training and two folders are used for testing. In these two folders, there are total of 580 emails which contain 483 legitimate emails and 97 spam emails. The testing results are evaluated based on these emails.

5.2.1 Bare category

In the bare category of ling spam corpus, both the stop-word list and lemmatiser are disabled. The evaluated results of bare category for experiment 1 (individual classifiers of J48 decision tree and MNB), experiment 2 (Bagging), and experiment 3 (Boosting-Adaboost) are illustrated in Table 5.

The results evaluated in Table 5 indicate the efficient results of boosting approach with an accuracy of 92.07%. Bagging approach has also achieved the accuracy of 89.83% which is better than the accuracy of MNB (82.07%) and J48 decision tree (85.17%) algorithms. This indicates the achievement of good accuracy results.

Table 5 Evaluated results of bare ling spam corpus

Parameter/method	*Multinomial Naïve Bayes*	*J48 decision tree*	*Bagging*	*Boosting*
TP	61	67	71	77
TN	415	427	450	457
FP	68	56	33	26
FN	36	30	26	20
Precision (%)	47.29	54.47	68.27	74.76
Recall (%)	62.89	69.07	73.20	79.38
F-Measure (%)	53.99	60.91	70.65	77.01
Accuracy (%)	82.07	85.17	89.83	92.07
TNR (%)	85.92	88.40	93.17	94.62
FNR (%)	37.11	30.93	26.80	20.62
FPR (%)	14.08	11.60	06.83	05.38

5.2.2 Lemm category

In lemm category of ling spam corpus, lemmatiser is enabled but stop-word list is disabled. Lemmatiser plays an important role in the detection of email spam by converting linguistic words (tokens) into their base words (tokens). Lemmatisation process considers the inflected forms of words and converts it into their intended meanings. The evaluated results of lemm category for all the experiments are illustrated in Table 6.

Table 6 Evaluated results of lemm ling spam corpus

Parameter/method	*Multinomial Naïve Bayes*	*J48 decision tree*	*Bagging*	*Boosting*
TP	68	70	79	86
TN	429	441	461	478
FP	54	42	22	05
FN	29	27	18	11
Precision (%)	55.74	62.50	78.22	94.51
Recall (%)	70.10	72.16	81.44	88.66
F-Measure (%)	61.48	66.98	79.80	91.49
Accuracy (%)	85.69	88.10	93.10	97.24
TNR (%)	88.82	91.30	95.45	98.96
FNR (%)	29.90	27.84	18.56	11.34
FPR (%)	11.18	08.70	04.55	01.04

The results obtained with lemm category (refer to Table 6) are the improved results as compared to bare category results. Boosting approach has attained the accuracy value of 97.24% which is superior to the accuracy obtained with bagging (93.10%), MNB

(85.69%), and J48 decision tree (88.10%). Further, the results with the lemm-stop category are evaluated.

5.2.3 Lemm-Stop category

Both the Lemmatiser and Stop-word list are enabled in the Lemm-Stop category of ling spam corpus. Lemm-Stop category further improves the results due to filtered corpus with lemmatisation process and applicability of the stop-word list. As earlier discussed in Section 5.2.2., lemmatisation process converts the inflected words into their respective base words. Stop words are common words (e.g., 'the') that can be ignored. The removal of stop words from corpus reduces the search space and makes the database filtered. The applicability of lemmatisation database and stop-word list improves the database and overall results. The evaluated results of lemm-stop category for all the experiments are illustrated in Table 7.

Table 7 Evaluated results of lemm-stop ling spam corpus

Parameter/method	*Multinomial Naïve Bayes*	*J48 decision tree*	*Bagging*	*Boosting*
TP	72	77	83	90
TN	443	448	474	483
FP	40	35	09	00
FN	25	20	14	07
Precision (%)	64.29	68.75	90.22	100
Recall (%)	74.23	79.38	85.57	92.78
F-Measure (%)	68.90	73.68	87.83	96.25
Accuracy (%)	88.79	90.52	96.03	98.79
TNR (%)	91.72	92.75	98.14	100
FNR (%)	25.77	20.62	14.43	07.22
FPR (%)	08.28	07.25	01.86	00

The evaluated results in Table 7 with the lemm-stop category indicate outperforms results as compared to bare and lemm category. Boosting approach has attained 98.79% accuracy with TNR value of 100% which indicates the nil error rate with legitimate emails. Email spam detection is also dominant in this case with recall value of 92.78%. Boosting concept has also achieved dominant results in comparison with individual concepts of J48 decision tree and MNB.

5.2.4 Stop category

The Stop category of ling spam corpus contains the enabled list of stop-words but lemmatiser is disabled. The consideration of stop-word list reduces the search space by removing the commonly known words as they are of no use for the knowledge extraction. The evaluated results with the stop category are illustrated in Table 8.

Table 8 Evaluated results of stop ling spam corpus

Parameter/method	*Multinomial Naïve Bayes*	*J48 decision tree*	*Bagging*	*Boosting*
TP	65	68	73	80
TN	425	434	455	467
FP	58	49	28	16
FN	32	29	24	17
Precision (%)	52.85	58.12	72.28	83.33
Recall (%)	67.01	70.10	75.26	82.47
F-Measure (%)	59.09	63.55	73.74	82.90
Accuracy (%)	84.48	86.55	91.03	94.31
TNR (%)	87.99	89.86	94.20	96.69
FNR (%)	32.99	29.90	24.74	17.53
FPR (%)	12.01	10.14	05.80	03.31

The results calculated in Table 8 with Stop category also obtained the dominant results of boosting category. Boosting approach has attained the accuracy value of 94.31% which is superior to the accuracy obtained with bagging (91.03%), MNB (84.48%), and J48 decision tree (86.55%).

The results outcomes illustrated in Table 8 with stop category indicates the better accuracy results as compared to bare category (refer to Table 5) but lacks from the lemm (refer to Table 6) and lemm-stop category (refer to Table 7). In this stop category as well, the accuracy of the boosting approach dominated over other concepts. Further, evaluated results in each category are compared based on the concept.

5.3 Performance analysis

The performance of the ensemble methods (bagging and boosting) is assessed by comparing the evaluated results illustrated in Tables 5–8. Results are evaluated with evaluation metrics of precision, recall, F-measure, accuracy, TNR, FNR, and FPR. The comparisons of all the three experiments including individual methods of MNB and J48 decision tree and ensemble methods of bagging and boosting are presented from Figures 2–8.

The comparison illustrated in Figure 2 is based on the precision parameters. The comparison of results among all the methods and database categories indicate the superiority of results for the boosting approach and lemm-stop category. The maximum precision value of 100% for the boosting approach in the lemm-stop category is noted. The concept of Multinomial Naive Bayes approach lacks with minimum precision values of 47.29% in the bare category.

Figure 3 illustrates the comparison based on the recall parameter. The comparison results illustrated in Figure 3 indicates the higher results value (92.78%) for the boosting ensemble method and lemm-stop category. Here, the category is bare with Multinomial Naive Bayes approach lacking.

Figure 2 Comparison based on evaluated precision values (see online version for colours)

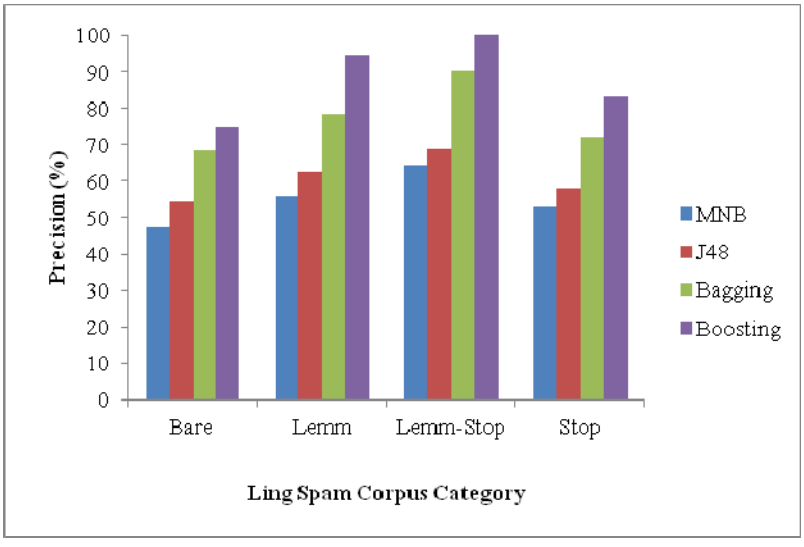
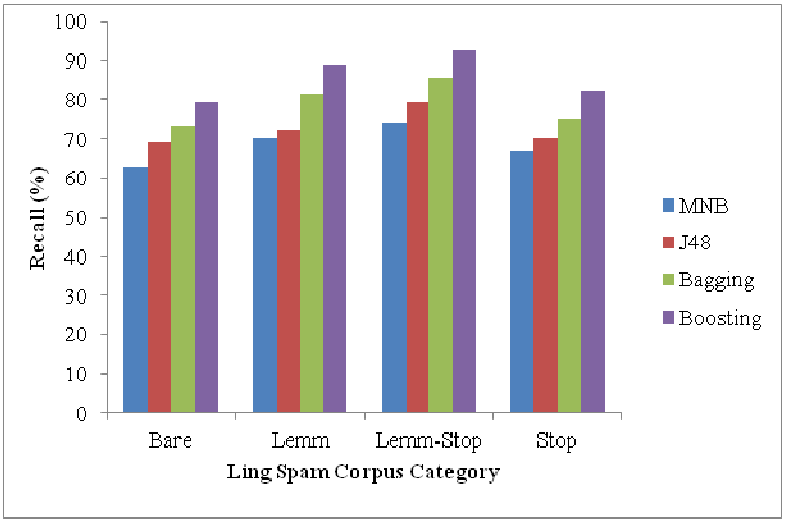


Figure 3 Comparison based on evaluated recall values (see online version for colours)



Further, Figure 4 illustrates the result comparison based on the f-measure parameter. This parameter also indicates the surpassing results for the boosting approach and lemm-stop category. As f-measure is the derived form of precision and recall. This also makes it clear the lower result values for the bare category of the database with Multinomial Naive Bayes approach.

Furthermore, Figures 5 and 6 illustrates the comparison based on accuracy and TNR parameters respectively. The case of Figures 5 and 6 are also clear with the indication of superior results of boosting approach and lemm-stop category. The lower result values of the bare category are noted.

Figure 4 Comparison based on evaluated F-measure values (see online version for colours)

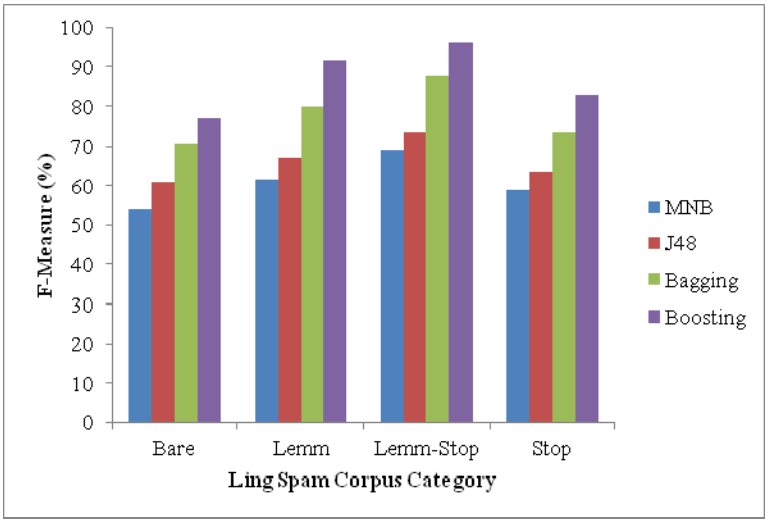
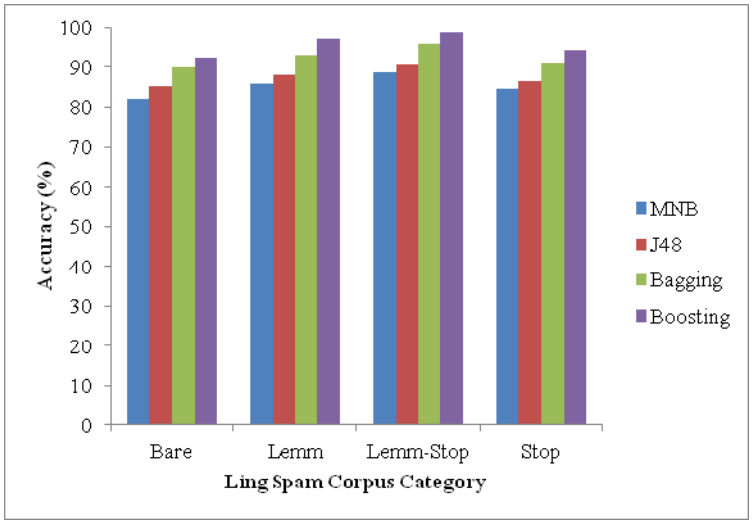


Figure 5 Comparison based on evaluated accuracy values (see online version for colours)



Figures 7 and 8 illustrate the results based on the FNR and FPR parameters respectively. The lower the result values of FNR and FPR parameters indicate the higher efficacy of the method. In this case, method of boosting and lemm-stop category has achieved the lesser result values which indicate the higher efficacy of the mentioned approach. Here, the bare category with Multinomial Naive Bayes category has achieved the higher result values which indicate the inferiority of method.

Figure 6 Comparison based on evaluated true negative rate (TNR) values (see online version for colours)

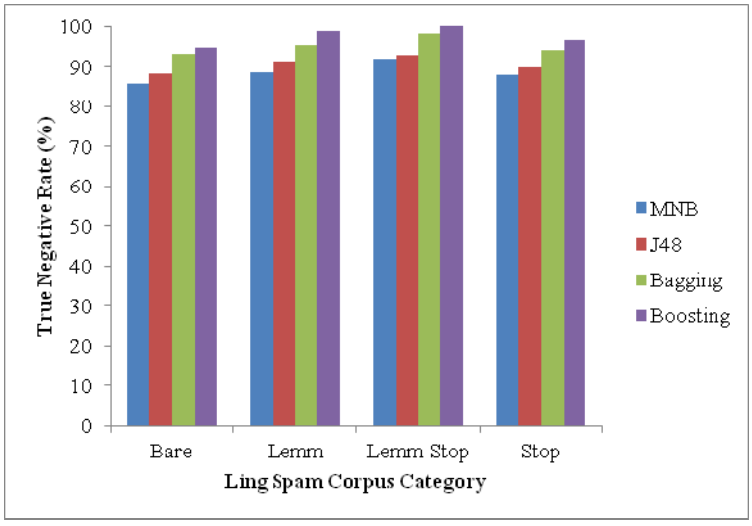
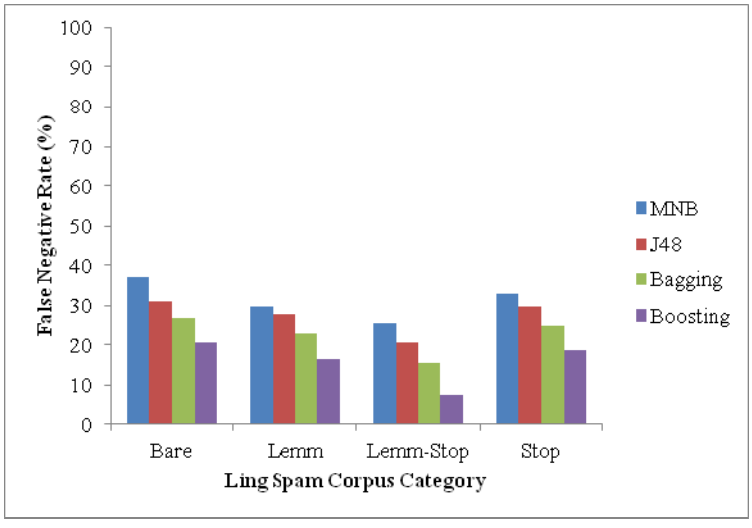
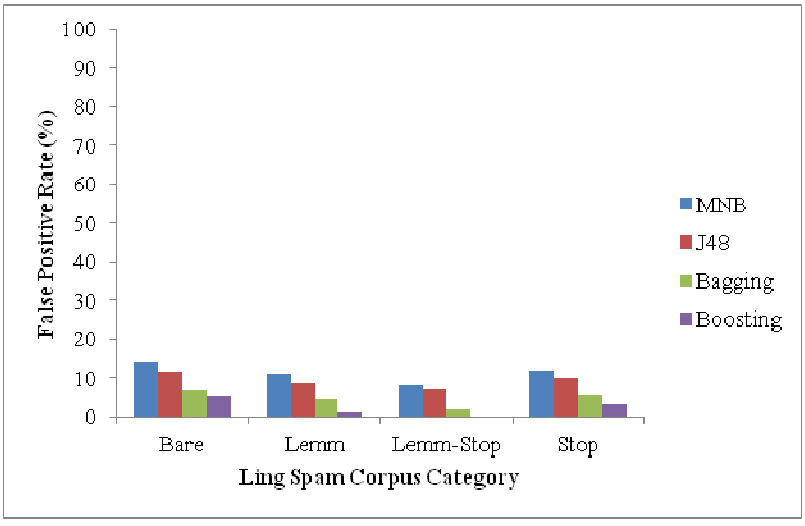


Figure 7 Comparison based on evaluated false negative rate (FNR) values (see online version for colours)



The performance analysis graphs illustrated in Figures 2–6 indicates the superiority of ensemble-based bagging and boosting approaches in comparison with individual concepts of MNB and J48 decision tree. Figures 7 and 8 indicates the lower error rates of ensemble methods (Bagging and Boosting) in terms of FPR and FNR which also illustrate the dominance of ensemble methods. It can also be noted from Figures 2–8 that the results with lemm-stop category of ling spam corpus are most efficient results obtained in each experiment. Then, the category of lemm, stop, and bare holds the rank in the descending order.

Figure 8 Comparison based on evaluated false positive rate (FPR) values (see online version for colours)



6 Conclusion

Spamming of email content has become one of the challenging issues in the field of information technology. Spamming includes the receiving and sending useless junk, spoofing, and phishing emails. The email spam has increased to the levels that threatening users send spamming content to hack someone's account details, to create more network traffic, and to waste someone's energy and time. Machine learning algorithms are playing a crucial role to tackle the vexed issue. In this research work, email spam filtration is adapted as a text mining approach for the classification of available email text content into legitimate and spam emails. The analysis and research gaps from the existing concepts urge to do consider the ensemble based bagging and boosting approaches for the email spam detection. The experimentation using individual classifiers of MNB and J48 decision tree classifiers is also discussed. The dataset of Ling spam Corpus with available categories is considered for the experimentation. The system is accessed with the performance metrics of precision, recall, f-measure, accuracy, TNR, FPR, and FNR. The results are evaluated with all the available categories (bare, lemm, lemm-stop, and stop) of ling spam corpus. The evaluated results indicate the superiority of ensemble methods (bagging and boosting) in comparison with individual classifiers of MNB and J48 decision tree approach. In all the experiments, results evaluated with lemm-stop category are more efficient as compared to other categories of ling spam corpus. Overall boosting approach dominates with the lemm-stop category of ling spam corpus in terms of evaluation accuracy of 98.79%, precision of 100%, and recall of 92.78%. This indicates that boosting concept have classified all the legitimate emails as true values and spam emails also have a lesser error rate of 7.22%. This also fulfils the theoretical aspects of the betterment of boosting approach as compared to bagging approach and individual concepts. In a theoretical manner, boosting approach has the attributes to overcome the drawbacks of one classifier by adding the strengths of another

classifier. Moreover, it also reduces the biasness and adds the weight values to the misclassified dataset sample to prioritise for classification. Bagging approach has also achieved efficient results superior to individual classifiers but lacks than boosting approach. For possible future scope, the ensemble methods of bagging and boosting can be applied to other applications such as fake news detection, suspicious activities detection on online social media data, rumour detection, etc. Furthermore, the ensemble methods can also be integrated with some computational intelligence approach such as deep neural network for further improvement of spam detection.

References

Androutsopoulos, I., Koutsias, J., Chandrinos, K.V., Paliouras, G. and Spyropoulos, C.D. (2000) *An Evaluation of Naive Bayesian Anti-Spam Filtering*, arXiv preprint cs/0006013.

Arpanet (2006) *Online Encyclopedia*, Available online: http://www.webopedia.com/TERM/A/ARPANET.html (Last accessed 1 December, 2017).

Bawm, Z.L. and Nath, R.P.D. (2014) 'A conceptual model for effective email marketing', *2014 17th International Conference on Computer and Information Technology (ICCIT)*, December, IEEE, pp.250–256.

Behjat, A.R., Mustapha, A., Nezamabadi-pour, H., Sulaiman, M.N. and Mustapha, N. (2012) 'GA-based feature subset selection in a spam/non-spam detection system', *2012 International Conference on Computer and Communication Engineering (ICCCE),* July, IEEE, pp.675–679.

Bresfelean, V.P. (2007) 'Analysis and predictions on students' behavior using decision trees in Weka environment', *Information Technology Interfaces*, pp.51–56.

Chandrashekar, G. and Sahin, F. (2014) 'A survey on feature selection methods', *Computers and Electrical Engineering*, Vol. 40, No. 1, pp.16–28.

Chawathe, S. (2018) 'Improving email security with fuzzy rules', *2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, IEEE, pp.1864–1869.

Chikh, R. and Chikhi, S. (2019) 'Clustered negative selection algorithm and fruit fly optimization for email spam detection', *Journal of Ambient Intelligence and Humanized Computing*, Vol. 10, No. 1, pp.143–152.

Cormack, G.V. and Lynam, T.R. (2005) 'TREC 2005 spam track overview', *TREC*, November, pp.500–274.

Douzi, S., Amar, M., El Ouahidi, B. and Laanaya, H. (2017) 'Towards a new spam filter based on PV-DM (paragraph vector-distributed memory approach)', *Procedia Computer Science*, Vol. 110, pp.486–491.

Drake, W.J. (Ed.): (2005) *Reforming Internet Governance: Perspectives from the Working Group on Internet Governance (WGIG) (No. 12),* United Nations Publications.

Email Assessment (2005) *Email Sifter, Email Effectiveness: Assessing the Threats of Spam and Viruses and Evaluating Strategies for Managing Them*, White Paper, Email Sifter, Available online at: http://www.emailsifter.com/whitepapers (Last accessed 1 December, 2017).

Email Statistics Report (2017-2021) *Executive Summary, Radicati Group, 2011*, https://www.radicati.com/wp/wp-content/uploads/2017/01/Email-statistics-report-2017-2021-executive-summarfy.pdf (Last accessed 1 December, 2018).

Faris, H. and Aljarah, I. (2015) 'Optimizing feedforward neural networks using krill herd algorithm for e-mail spam detection', *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, November, IEEE, pp.1–5.

Faris, H., Ala'M, A.Z., Heidari, A.A., Aljarah, I., Mafarja, M., Hassonah, M.A. and Fujita, H. (2019) 'An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks', *Information Fusion*, Vol. 48, pp.67–83.

Faris, H., Aljarah, I. and Al-shboul, B. (2016) 'A hybrid approach based on particle swarm optimization and random forests for e-mail spam filtering', *International Conference on Computational Collective Intelligence*, September, Springer, Cham, pp.498–508.

Fazil, M. and Abulaish, M. (2018) 'A hybrid approach for detecting automated spammers in twitter', *IEEE Transactions on Information Forensics and Security*, Vol. 13, No. 11, pp.2707–2719.

Feng, W., Sun, J., Zhang, L., Cao, C. and Yang, Q. (2016) 'A support vector machine based naive bayes algorithm for spam filtering', *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC),* December, IEEE, pp.1–8.

Gandotra, E., Bansal, D. and Sofat, S. (2019) 'Malware intelligence: beyond malware analysis', *International Journal of Advanced Intelligence Paradigms*, Vol. 13, Nos. 1–2, pp.80–100.

Gupta, V., Mehta, A., Goel, A., Dixit, U. and Pandey, A.C. (2019) 'Spam detection using ensemble learning', *Harmony Search and Nature Inspired Optimization Algorithms*, Springer, Singapore, pp.661–668.

Harisinghaney, A., Dixit, A., Gupta, S. and Arora, A. (2014) 'Text and image based spam email classification using KNN, Naïve Bayes and reverse DBSCAN algorithm', *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, IEEE, February, pp.153–155.

Idris, I., Selamat, A., Nguyen, N.T., Omatu, S., Krejcar, O., Kuca, K. and Penhaker, M. (2015) 'A combined negative selection algorithm–particle swarm optimization for an email spam detection system', *Engineering Applications of Artificial Intelligence*, Vol. 39, pp.33–44.

Kaur, H. and Sharma, A. (2016) 'Improved email spam classification method using integrated particle swarm optimization and decision tree', *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, October, IEEE, pp.516–521.

Kibriya, A.M., Frank, E., Pfahringer, B. and Holmes, G. (2004) 'December. multinomial naive bayes for text categorization revisited', *Australasian Joint Conference on Artificial Intelligence*, Springer, Berlin, Heidelberg, pp.488–499.

Kumaresan, T. and Palanisamy, C. (2017) 'E-mail spam classification using S-cuckoo search and support vector machine', *International Journal of Bio-Inspired Computation*, Vol. 9, No. 3, pp.142–156.

Mangalindan, M. (2002) 'For bulk E-mailer, pestering millions offers path to profit', *Wall Street Journal*, Vol. 13, pp.7–13.

McIver Jr., W.J. and Birdsall, W.F. (2002) 'Technological evolution and the right to communicate', *Euricomm Colloquium: Electronic Networks and Democracy*, Nijmegen, The Netherlands, October, pp.9–12.

Menghour, K. and Souici-Meslati, L. (2014) 'Classical and swarm-based approaches for feature selection in spam filtering', *International Journal of Advanced Intelligence Paradigms*, Vol. 6, No. 3, pp.214–234.

Mohamad, M. and Selamat, A. (2015) 'An evaluation on the efficiency of hybrid feature selection in spam email classification', *2015 International Conference on Computer, Communications, and Control Technology (I4CT)*, April, IEEE, pp.227–231.

Naem, A.A., Ghali, N.I. and Saleh, A.A. (2018) 'Antlion optimization and boosting classifier for spam email detection', *Future Computing and Informatics Journal*, Vol. 3, No. 2, pp.436–442.

Nizamani, S., Memon, N., Glasdam, M. and Nguyen, D.D. (2014) 'Detection of fraudulent emails by employing advanced feature abundance', *Egyptian Informatics Journal*, Vol. 15, No. 3, pp.169–174.

Olatunji, S.O. (2017) 'Extreme learning machines and support vector machines models for email spam detection', *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE),* April, IEEE, pp.1–6.

Prilepok, M., Jezowicz, T., Platos, J. and Snasel, V. (2012) 'Spam detection using compression and PSO', *2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN)*, November, IEEE, pp.263–270.

Renuka, D.K., Hamsapriya, T., Chakkaravarthi, M.R. and Surya, P.L. (2011) 'Spam classification based on supervised learning using machine learning techniques', *2011 International Conference on Process Automation, Control and Computing*, IEEE, July, pp.1–7.

Renuka, K.D. and Visalakshi, P. (2014) 'Latent semantic indexing based SVM model for email spam classification', *Journal of Scientific & Industrial Research*, Vol. 73, No. 07, pp.437–442.

Sampson, M. (2003) *Spam Control: Problems and Opportunities*, Ferris Research Publication, 2003, Available online at: tp://www.ferris.com/view_content.php?o=Spam+Control&id=105 (Last accessed 1 December, 2017).

Shams, R. and Mercer, R.E. (2013) 'Classifying spam emails using text and readability features', *2013 IEEE 13th International Conference on Data Mining*, December, IEEE, pp.657–666.

Shen, H. and Li, Z. (2013) 'Leveraging social networks for effective spam filtering', *IEEE Transactions on Computers*, Vol. 63, No. 11, pp.2743–2759.

Takashita, T., Itokawa, T., Kitasuka, T. and Aritsugi, M. (2008) 'Extracting user preference from web browsing behaviour for spam filtering', *International Journal of Advanced Intelligence Paradigms*, Vol. 1, No. 2, pp.126–138.

Talos Intelligence Report (2019) *Total Global Email and Spam Volume FOR July 2019*, Available online at: https://www.talosintelligence.com/reputation_center/email_rep (Last accessed 20 August, 2019).

Tang, B., Kay, S. and He, H. (2016) 'Toward optimal feature selection in Naive Bayes for text categorization', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No. 9, pp.2508–2521.

Trivedi, S.K. and Dey, S. (2013) 'An enhanced genetic programming approach for detecting unsolicited emails', *2013 IEEE 16th International Conference on Computational Science and Engineering,* December, IEEE, pp.1153–1160.

Wijayanto, A.W. (2014) 'Fighting cyber crime in email spamming: an evaluation of fuzzy clustering approach to classify spam messages', *2014 International Conference on Information Technology Systems and Innovation (ICITSI)*, November, IEEE, pp.19–24.

Yadav, S.K., Tayal, D.K. and Shivhare, S.N. (2019) 'Perplexed Bayes classifier-based secure and intelligent approach for aspect level sentiment analysis', *International Journal of Advanced Intelligence Paradigms*, Vol. 13, Nos. 1–2, pp.15–31.

Yahya, A.A. and El Bashir, M.S. (2014) 'Applying machine learning to analyse teachers' *Instructional Questions. International Journal of Advanced Intelligence Paradigms*, Vol. 6, No. 4, pp.312–327.