Journal of Multi Disciplinary Engineering Technologies
Volume 18, Issue 01

Performance Evaluation of Email Spam Detection Algorithms: A Case Study

Suhan Lakshakar[1], Vinay Kumar Patel[1*], Varsha Sharma[1], Rachna Jain[2]

[1]Department of Artificial Intelligence and Data Science, Bhagwan Parshuram Institute of Technology, New Delhi, India, 110089.
[2]Department of Information Technology, Bhagwan Parshuram Institute of Technology, New Delhi, India, 110089.

*Corresponding author(s). E-mail(s): vinayptl47@gmail.com;
Contributing authors: suhanlakshakar3222@gmail.com;
varshasharma@bpitindia.com; rachnajain@bpitindia.com;

Abstract

Electronic mail (E-mail) is one of the most frequently utilized medium of communication. Emails are used for personal and business-related purposes because it is simple to use and cheapest. Unwanted elements in society misuse email and they send spam messages that have malicious, phishing links, financial scams, and undesired advertisements. Users face problems such as excess usage of memory and a hard time differentiating spam emails from ham emails. To tackle this problem studies and research have been done where techniques and methodologies such as natural language processing(NLP) and machine-based learning approach have been implemented to classify emails. In this work, different algorithms were implemented such as Random Forest, Bagging, Support Vector Machine, and Boosting Algorithms, K Nearest Neighbors, Decision Tree, Logistic Regression and Naive Bayes classifiers such as MNB, BNB, GNB, and found that Multinomial Naive Bayes algorithm outperformed and was able to classify efficiently in terms of accuracy and precision. We were able to classify email as ham/spam based on features extracted from the training dataset.

Keywords: Naive Bayes, Tfidf Vectorizer, Spam, Ham , Multinomial Naive Bayes

1 Introduction

In today's time, communication has become faster due to the invention of Email which is used for information communication over the Internet. Some companies and individuals misuse this facility to create trouble for users by sending junk mail often referred to as spam emails. The spam emails are forwarded to multiple individuals in the type of advertising emails, lottery or free trips to some places, and inappropriate content. These spam emails contain malicious URLs known as phishing links and malware that contains viruses, worms, and spyware. hackers, phishers send these emails where target individuals can lose their money and also their social media privacy, personal

Published on: November, 2024

credentials such as debit card details, passwords, and some confidential data[1]. Spammers can bypass the existing filtering techniques of emails. Approaches adopted for spam filtering include analysis of text, white listing, and black listing of the domain names[2]. Text analysis basically uses Natural Language Processing to filter mail. In a whitelist trusted contacts or domains are included while in a blacklist unknown contacts and domains are included, this helps in delivering only genuine mail and filtering out blacklisted ones. Spam and Ham Emails: Spam emails are unsolicited bulk email messages that the receiver does not wish to receive those. These include financial scams, advertisements, and malware. Ham emails are legitimate or genuine emails that users wish to receive.

1.1 Background

The growth of internet users has led to the problem of email classifying. Traditional techniques are being outsmarted by scammers, therefore, there is a need to come up with some other filtering techniques that use machine learning and NLP.

1.2 Objective

1. The primary goal is to develop a classification model that uses machine learning algorithms. The algorithm would textual features for classifying emails as ham or spam.
2. Compare algorithms to understand how well they perform in classifying emails and select the best model based on its performance metrics.

This paper has Seven sections. The first section discusses about Introduction of email spam detection, second section discusses about Literature review/related work done in this area. The third section talks about the proposed work that we are going to follow, The fourth section is about implementation where first we have discussed data preprocessing techniques, Label encoding, Feature extraction, different algorithms, testing, and evaluation. In the fifth section, we have compared different algorithms according to their performance metrics and selected the best model. In the sixth section, we have concluded our results and future works. In the seventh Section, we have cited different references we have made.

2 Related Work

In this section, previous studies have been discussed. These studies provide insights into the limitations and challenges of traditional spam filtering methods and the advancements made to address these problems, such as the application of machine learning, ensemble techniques, and deep learning approaches.

Nikhil Kumar, Sanket Sonowal et al. (2020) [2] concluded in their study that the Naive Bayes algorithm achieves better accuracy than other classifiers, but its limitation of class conditional independence causes misclassification on certain tuples. Another applied method suggested that using multiple classifiers could yield better results. Sunday Olusanya Olatunji (2017) [3] discussed an email spam detection system based on an SVM classifier, trained and evaluated using a standard dataset. The study highlighted the need for greater precision in spam detection models and proposed an improved approach that outperforms existing systems, achieving a 3.11% improvement compared to the NSA–PSO hybrid solution. This result demonstrates SVM's effectiveness in classification across diverse areas. Kriti Agarwal, Tarun Kumar et al. (2018) [4] proposed the Naive Bayes algorithm for detecting spam emails. Leveraging its probability distribution property, Naive Bayes differentiates between spam and ham mail based on keyword analysis. PSO was used to optimize Naive Bayes parameters, enhancing accuracy and performance. Feature selection was performed using CFS, and the Ling spam dataset was used for experimentation. The results showed that Naive Bayes performed better than other methods. Future research integrating Naive Bayes with other classifiers and algorithms is suggested to improve efficiency and accuracy. Shrawan Kumar Trivedi (2016) [5] found SVM to be the most effective classifier in their study, achieving the lowest false

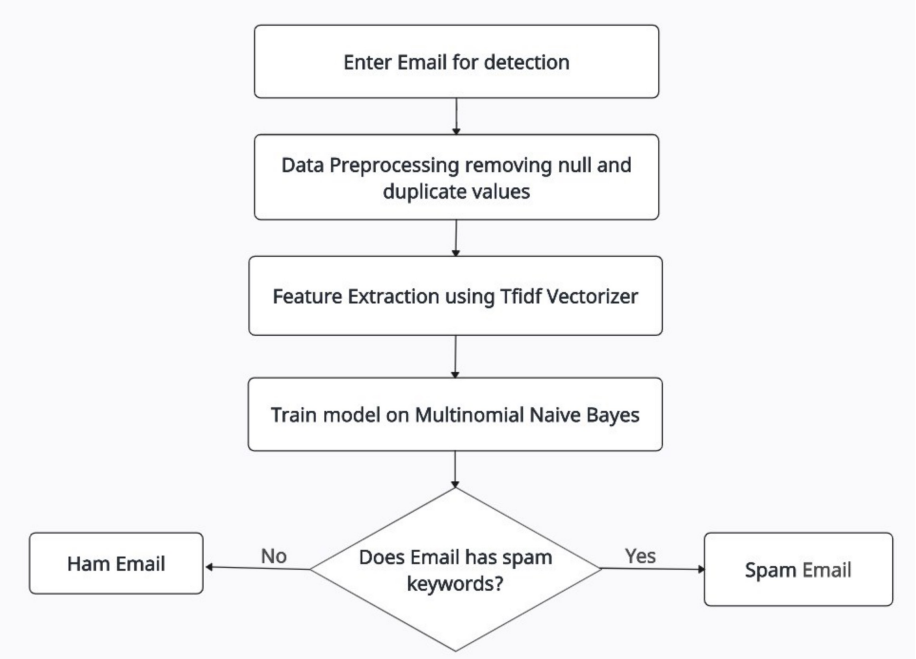

Fig. 1 Flowchart for Email Spam detection

positive rate and highest F-measure. Despite the relatively high model-building time, the effectiveness of SVM results was deemed to justify the trade-off between speed and accuracy, making SVM a recommended choice for spam detection. Sunil B. Rathod and Tareek M. Pattewar (2015) [1] employed the Bayesian method to categorize spam and ham emails through supervised learning. Their experiment achieved 96.46% accuracy when tested against real-world Gmail datasets, demonstrating the system's effectiveness in detecting spam emails. M. Bassiouni, M. Ali, et al. (2018) [6] achieved better accuracy in spam classification using the Spam-base UCI dataset. Their approach involved preprocessing data and utilizing multiple classifiers, including Random Tree, SVM, ANN, Decision Table, Bayes Net, Logistic Regression, and k-Nearest Neighbors. Random Forest outperformed other methods, achieving an accuracy of 95.45%. Yuliya Kontsewayaa, Evgeniy Antonova et al. (2021) [7] conducted a study on spam mail detection using machine learning techniques. They trained a dataset of 4,360 ham messages and 1,368 spam messages, employing classifiers like Random Forest, Logistic Regression, Decision Tree, Naive Bayes, and SVM. Logistic Regression and Naive Bayes were the most effective, achieving 99% accuracy. Gibson, S., Issac, B., Zhang, L., et al. (2020) [8] integrated bio-inspired algorithms with models for email classification. Testing on 50,000 emails showed that top-performing models included Multinomial Naive Bayes, Random Forest, SVM, and Decision Tree. Later experiments with Scikit-learn improved SVM performance with an SGD classifier. Naive Bayes optimized with the Spam Assassin dataset achieved 100% accuracy. Sethi, P., Bhandari, V., et al. (2017) [9] observed that Naive Bayes is more efficient than Logistic Regression for email spam detection, achieving a maximum accuracy of 98.445%. They noted that while Random Forest performed better, Naive Bayes remained superior in efficiency and accuracy. Gadde, S., Lakshmanarao, et al. (2021) [10] used a deep learning system for email spam detection, applying techniques such as Count Vectorizer, TF-IDF, and Hashing Vectorizer. Using the UCI dataset, they achieved 98.5% accuracy with the LSTM model, which outperformed previous systems. Vyas, T., Prajapati, P., et al. (2015) [11] observed that Naive Bayes provided faster and better accuracy than other methods, except for SVM and ID3, which offered higher accuracy but required more time. Their findings highlighted the potential for future improvements by analyzing the email header and initial content. Abdullahi, M. Mohammed, et al. (2021) [12] noted that Naive

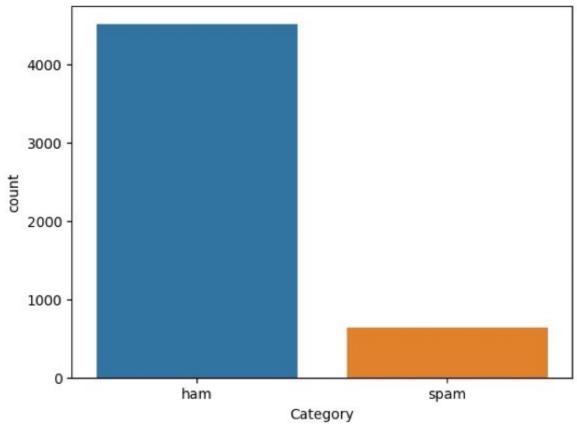

Fig. 2 Count-plot for ham and spam email/messages

Bayes excelled in F-measure with a score of 99.25 but emphasized that no method achieved 100% accuracy in spam detection. The study highlighted the importance of hybrid systems and labeled datasets for optimal filtering. Gupta, V., Mehta, A., et al. (2019) [13] explored a voting classifier approach based on ensemble learning. They found that the voting classifier outperformed supervised and unsupervised methods, achieving high accuracy with Naive Bayes and Decision Trees. Unnikrishnan, V., Kamath, et al. (2021) [14] concluded that Naive Bayes was the most effective algorithm for spam mail detection, achieving 98.86% accuracy compared to SVM's 97.68%. Agboola, O. S. (2020) [15] demonstrated that Naive Bayes achieved 100% efficiency in spam detection, while SVM showed fewer false positives, making both algorithms valuable for different scenarios.

3 Proposed Work

In the proposed work discussion was done related to how email is taken, and transformed for training of the model, and then the model is classifying emails. Emails are taken from the user and data preprocessing tasks such as the removal of null values and duplicate values. Label encoding is done to convert the category spam and ham as 1 or 0. Feature extraction technique such as TF-IDF Vectorizer was applied for the conversion of textual data into numerical binary vectors which machine learning algorithms understand. Then various classifiers such as Bernoulli, Gaussian Naive Bayes, Logistic Regression, Multinomial Naive Bayes, Bagging and Boosting Algorithms, K Nearest Neighbors, Support Vector Machine, and Decision Tree, were implemented. Each classifier's performance in terms of their accuracy and precision score was calculated and then compared. Best-performing algorithms were selected as models for categorizing spam or ham emails.

4 Implementation

4.1 Data Collection and Preprocessing

For this study, data was collected from Kaggle. Next data preprocessing was done where null values and duplicate values were checked and removed. Prepared data by removing stop words, HTML tags and URLs, punctuation, and special characters.

4.2 Label Encoding

This had been used for converting target attributes which are in categorical format to the numerical format which is necessary for calculations and predictions in Machine learning algorithms.

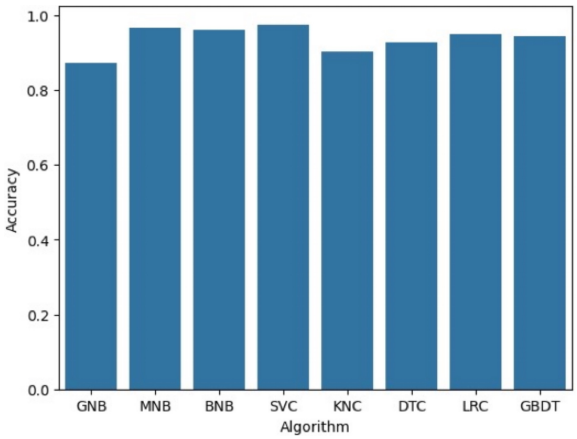

Fig. 3 Accuracy Score for Classifier

4.3 Feature Extraction

This was used for converting email text data to numerical vectors using various vectorizer techniques such as Count Vectorizer and TF-IDF Vectorizer. In the TF-IDF vectorizer, the word TF stands for Term Frequency which was used to measure the frequency of a term appearing in the text, and the word IDF stands for Inverse Document Frequency which was used to measure the words' relevance in the entire text. TF-IDF converts text data to numerical vectors by assigning weight for every word based on its occurrence frequency in the entire dataset.

4.4 Model Selection

Algorithms training is done on the training data and then the best model is selected for classifying tasks based on performance metrics like accuracy and precision. The algorithms used for this study are briefly explained here.

4.4.1 Naive Bayes

This is an algorithm based on Bayes theorem which finds the probability of each class which was applied to classification problems such as text classification. Algorithms assume that the presence of each word is independent of others which makes calculations efficient and accurate. When algorithms are trained with data it calculates probabilities for words in spam and ham emails. Once a new email is given for classification it calculates the probability, and if it exceeds certain values it is categorized as spam or ham. In this study, various algorithms such as Gaussian NB, Bernoulli NB, and Multinomial NB were implemented. The Bayesian theorem is:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

4.4.2 K-Nearest Neighbor(KNN)

It is a classification algorithm applied for text classification, pattern recognition, and data mining. In this algorithm, distance is calculated between new data points and each training point using Euclidean distance, K data points which had the smallest distance will be considered as nearest neighbors. Class is then determined by majority voting of K neighbors. In email spam detection first, it will calculate the distance between the test email and the training email using Euclidean distance, then select K nearest neighbors to test email. It then classifies email to be spam or ham by majority voting of the class label of K Nearest Neighbors.

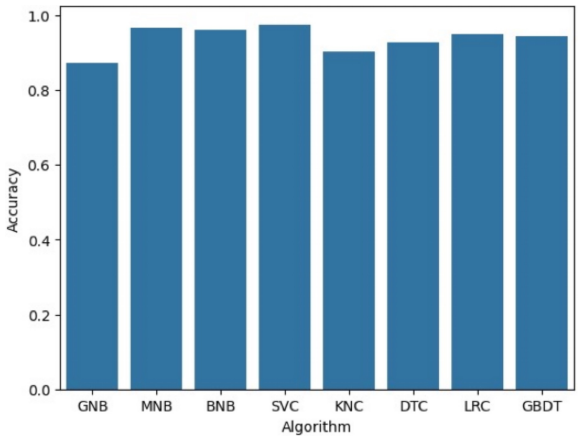

Fig. 4 Precision Score for classifier

4.4.3 Support Vector Classifier

This is a form of Support Vector Machine that was applied for data classification tasks, particularly in linearly separable data. SVM's goal is to locate a hyperplane in many-dimensional space that maximizes the margin between classes which effectively separates classes. In non-linear classification tasks, kernel functions such as sigmoid kernels, and Gaussian Kernel Radial Basis Function (RBF) are used. Kernel Functions are used to transform input non-linear separable data to a space in the higher dimension where separation of classes becomes easy using a hyperplane.

4.4.4 Decision Tree Algorithm

It is a supervised machine-learning algorithm that was applied for classification tasks and regression tasks. The decision tree is a tree-like structure where the root node represents complete datasets, internal nodes are features of the data, branches denote the rules, and the leaf node indicates the category label. The steps followed include Information gain being used to select the best feature that would split data at the root node and Gini Impurity, recursively splitting data into subsets until maximum depth is reached or minimum samples in a node are left, and assigning class labels to each leaf node. Entropy is a disorder or impurity in the dataset, its goal is to select an at- tribute that minimizes the entropy of a subset. The reduction of entropy is measured using Information gain which was achieved by splitting datasets based on specific properties.

4.4.5 Logistic Regression

Logistic regression was used for binary classification tasks, used in email spam detection, healthcare, and fraud detection. The goal of this algorithm was to make a prediction whether an instance belongs to a certain class or not such as spam or ham, yes or no, pass or fail. This differs from linear regression because in linear regression target attribute is a numerical value but in logistic regression target attribute is in categorical format. The sigmoid function is used which takes independent variables as input and produces a probability output between 0 and 1. 0 and 1 are class labels, and 0.5 is considered to be the threshold value if certain values lie below 0.5 then it belongs to class 0 and if values lie above 0.5 then they belong to class 1. The logistic function is :

$$f(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

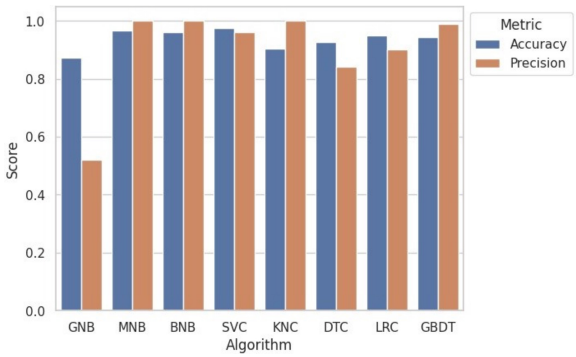

Fig. 5 Accuracy and Precision Score Comparison

4.4.6 Random Forest Classifier

A random forest classifier follows an ensemble learning method applied for classification tasks. In this classifier, multiple decision trees were built using row sampling which is randomly selecting a subset from the dataset, and feature sampling which is a random selection of features, and each tree votes on a given set of test data then the algorithm selects the class label with the highest votes. It employs a technique called bagging(Bootstrap Aggregating), in this dataset is recursively split and a random subset is selected then the feature is selected. This prevents the dominance of single features in prediction, making the model robust. Since it employs multiple decision trees, it has better accuracy and handles complex datasets.

4.4.7 Gradient Boosting Classifier

It is an algorithm that uses machine learning and follows ensemble techniques where multiple weak learning models are combined together so that they can perform better. It has been used since it can handle complex datasets and model accuracy can also be significantly increased. It has three components :

1. **loss function** It estimates how good a model is based on its predictions made by it on the given data. This varies with the type of dataset provided.
2. **Weak Learner** The model is said to be a weak learner when it has a high error rate in predicting output.
3. **Additive Model** This is an iterative approach where at each step a weak learner is added which results in reducing the value of the loss function.

4.4.8 Bagging Classifier

It is an ensemble learning technique for classification tasks where multiple weak models are combined to improve their performance. In this classifier,multiple weak base models such as decision trees, and SVM were trained on different subsets of training data and trained with the best feature of training data. Then predictions of the base model are aggregated either through majority voting or taking the average of predictions. It would be effective to use in unstable learning algorithms such as decision trees. This learning technique reduces overfitting, improves model stability, and handles noisy or imbalanced datasets.

4.5 Testing and Evaluation

The model was first trained with the training set of data provided by the user, and after that model was used for predicting some output or results. The testing set of data was employed for the performance evaluation of the model. Model performance was determined using various performance measures such as precision, F-score, confusion matrix, and accuracy. To improve the performance

of the model various steps are followed such as using high-quality datasets and optimizing model parameters. The best model is selected based on its performance score.

4.6 Model Deployment

The selected model is integrated into the live environment where it would used to make real-world predictions. First, a deployment environment is selected such as AWS, or Google Cloud. Then all required files are loaded into the deployment environment. Then it is hosted in the deployment environment.

5 Result and Discussion

In this section, results and findings have been discussed. Here the performance of multiple algorithms used in this study is compared on the basis of its precision score and accuracy score. After a comparison of results, the model which had the highest precision score and accuracy score was selected.

5.1 Performance Comparison of Algorithms

Different algorithms were implemented for email spam detection, The Support Vector Classifier had the highest accuracy at 97.5%. This shows SVC had effec- tively classified emails as spam or ham. After SVC, Multinomial Naive Bayes and Bernoulli Naive Bayes have accuracy scores of 96.7% and 96.1% respectively this shows these two models are also effective in distinguishing emails as spam or ham. Rest other algorithms such as Logistic Regression Classifier(LRC), Gradient Boosting Decision Tree(GBDT), Decision Tree Classifier(DTC), K Neighbors Classifier(KNC), Gaussian Naïve Bayes(GNB) had accuracy scores in the range of 86.7% to 95%. Other measures are taken for better performance analysis of models, therefore precision score was selected.

5.2 Precision Analysis

It provides accuracy of positive predictions made by the model. The aim is to minimize false positives which helps in classifying a ham email as a spam email. MNB, BNB, and KNC had a precision score of 100% which meant the model made no false positive predictions. Other algorithms such as GBDT, LRC, DTC, and SVC had precision scores in the range of 98.8 to 84.3%. GNB had the lowest precision score at 52.2% which shows it is making a lot of errors in classification.

5.3 Algorithm Selection

Support Vector Classifier(SVC)and Multinomial Naïve Bayes (MNB) had the highest accuracy while Bernoulli NB(BNB), Multinomial NB(MNB) and K Neighbors Classifier(KNC) had the highest precision score. The model with high accu- racy and high precision score indicates better performance and minimizes false positives. Since Multinomial Naive Bayes had the highest accuracy and precision it was selected for classifying emails.

5.4 Future Directions

Spammers can bypass the existing filtering methods so there is a need to develop advanced filtering techniques that use ensemble methodology, machine learning, and deep learning which learn patterns from the text and can be used to predict future emails. Natural Language Processing can understand language so it will be better to identify spam and ham emails and Collaborative Filtering where feedback and reports from a large user base can help to identify emails. Understanding spammers' tactics and the dynamics of email data help to improve email spam detection.

Table 1 Comparison of Algorithms

No.	Algorithm	Accuracy	Precision
0	GNB	0.873	0.522
1	MNB	0.967	1.000
2	BNB	0.961	1.000
3	SVC	0.975	0.960
4	KNC	0.903	1.000
5	DTC	0.927	0.843
6	LRC	0.950	0.901
7	GBDT	0.944	0.988

6 Conclusion

Email spam detection is necessary for users because it protects the privacy and data of users, easily classifies genuine emails for users and security risk is reduced. Many algorithms were implemented in this study, Multinomial NB(MNB) had an accuracy of 96.7% and a precision of 100% which made it the best classifier for detecting spam email tasks. It can accurately categorize spam or ham emails, and handle high-dimensional data such as text features which is present in emails. To improve email spam detection integrated approach is followed where algorithms are combined with behavioral analysis, collaborative filtering, and Natural Language Processing. Ongoing research and development in this domain can help to understand the spammers' tactics and emerging threats and devise methods and approaches that could help to enhance spam detection.

Declarations

- The authors received no specific funding for this study.
- The authors declare that they have no conflicts of interest to report regarding the present study.
- No Human subject or animals are involved in the research.
- All authors have mutually consented to participate.
- All the authors have consented the Journal to publish this paper.
- Authors declare that all the data being used in the design and production cum layout of the manuscript is declared in the manuscript.

References

[1] Rathod, S.B., Pattewar, T.M.: Content based spam detection in email using bayesian classifier. In: 2015 International Conference on Communications and Signal Processing (ICCSP), pp. 1257–1261 (2015). IEEE

[2] Kumar, N., Sonowal, S., *et al.*: Email spam detection using machine learning algorithms. In: 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 108–113 (2020). IEEE

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

[4] Agarwal, K., Kumar, T.: Email spam detection using integrated approach of naïve bayes and particle swarm optimization. In: 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 685–690 (2018). IEEE

[5] Trivedi, S.K.: A study of machine learning classifiers for spam detection. In: 2016 4th International Symposium on Computational and Business Intelligence (ISCBI), pp. 176–180 (2016). IEEE

[6] Bassiouni, M., Ali, M., El-Dahshan, E.A.: Ham and spam e-mails classification using machine learning techniques. Journal of Applied Security Research **13**(3), 315–331 (2018)

[7] Kontsewaya, Y., Antonov, E., Artamonov, A.: Evaluating the effectiveness of machine learning methods for spam detection. Procedia Computer Science **190**, 479–486 (2021)

[8] Gibson, S., Issac, B., Zhang, L., Jacob, S.M.: Detecting spam email with machine learning optimized with bio-inspired metaheuristic algorithms. IEEE Access **8**, 187914–187932 (2020)

[9] Sethi, P., Bhandari, V., Kohli, B.: Sms spam detection and comparison of various machine learning algorithms. In: 2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN), pp. 28–31 (2017). IEEE

[10] Gadde, S., Lakshmanarao, A., Satyanarayana, S.: Sms spam detection using machine learning and deep learning techniques. In: 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), vol. 1, pp. 358–362 (2021). IEEE

[11] Vyas, T., Prajapati, P., Gadhwal, S.: A survey and evaluation of supervised machine learning techniques for spam e-mail filtering. In: 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), pp. 1–7 (2015). IEEE

[12] Abdullahi, M., Mohammed, A.D., Bashir, S.A., Abisoye, O.O.: A review on machine learning techniques for image based spam emails detection. In: 2020 IEEE 2nd International Conference on Cyberspace (CYBER NIGERIA), pp. 59–65 (2021). IEEE

[13] Gupta, V., Mehta, A., Goel, A., Dixit, U., Pandey, A.C.: Spam detection using ensemble learning. In: Harmony Search and Nature Inspired Optimization Algorithms: Theory and Applications, ICHSA 2018, pp. 661–668 (2019). Springer Singapore

[14] Unnikrishnan, V., Kamath, P.: Analysis of email spam detection using machine learning. International Research Journal of Modernization in Engineering Technology and Science **3**(9), 409–416 (2021)

[15] Agboola, O.S.: Spam detection using machine learning. In: Conference on Intelligent Computing and Control Systems (ICICCS) (2020)