IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 27, No. 1, February, 2022, pp. 157-163
Journal homepage: http://iieta.org/journals/isi

Spam Mail Detection Using Optimization Techniques

Koneru Anupriya*, Kurakula Harini, Kethe Balaji, Karnati Geetha Sudha

Department of Information Technology, Lakireddy Bali Reddy College of Engineering, Mylavaram, Andhra Pradesh 521230, India

Corresponding Author Email: smartykoneru@gmail.com

https://doi.org/10.18280/isi.270119

Received: 4 December 2021
Accepted: 12 January 2022

Keywords:
bio-inspired algorithms, classification, machine learning, optimization, Tkinter

ABSTRACT

On account of the widespread availability of internet access, email correspondence is one among the most well-known cost-effective and convenient method for users in the office and in business. Many people abuse this convenient mode of communication by spamming others with conciseness bulk emails. They use emails to collect personal information of the users to benefit financially. A literature review is conducted to investigate the most effective strategies for achieving successful outcomes while working with various spam mail datasets. K-Nearest Neighbor, Support Vector Machine, Naive Bayes, Decision Tree, Random Forest, and Logistic Regression are all employed in the implementation of machine learning techniques. To make classifiers more efficient, bio-inspired algorithms such as BAT and PSO are used. The accuracy of every classification algorithm along with and without optimization is observed. Factors such as accuracy, f1-score, precision, and recall are used to compare the results. This work is implemented in Python along with GUI interface Tkinter.

1. INTRODUCTION

Machine Learning algorithms have been developed in the sphere of information technology for a variety of tasks, including fixing network traffic issues and detecting malware. Many people use email to communicate and socialize regularly. Spammers can send spam (illegitimate) emails as a result of security breaches that compromise client data [1]. According to Naem et al. [2] decrease in privacy, taking up space in the inbox, propagate viruses, and ruining email servers is because of spam emails. For cancelling the unwanted email and to refine imports in email, a lot of time gets wasted. When unwanted mails are detected, they are classified as spam mails or non-spam (ham) mails and this procedure is done by using a classification algorithm. Although classification algorithms are used to detect spam and datasets frequently include a lot of tiny or repetitive characteristics, which can reduce classification precision.

The classification is a supervised learning algorithm which is used to predict the class label of a new sample based on the model build by existing samples. Optimization is the area of discovering the best result for an explicit problem. Researchers make a lot of activities to improve doing the best decide. There are number of nature inspired algorithms to provide optimized solution to a problem. Companies provide a variety of tools and strategies for detecting a network of spam emails. Filtering techniques have been designed to detect unwanted emails by raising criteria and establishing firewall rules. One of the most well-known corporations is Google which detects such emails with a 99.9% success rate. Spam filters can be deployed in a variety of places, including the gateway (router), applications hosted on the cloud [3, 4], and the workstation of user. To solve the problem of detecting spam email, content-based filtering, and Bayesian filtering techniques have been applied. Feature selection for the models can be used to further evaluate the suggested spam detection to tackle the problem of classifying spam mails. Bio-inspired algorithms like Particle Swarm Optimization (PSO) and BAT algorithms are used in this study to test six alternative machine learning models. Every classification algorithm is combined with an optimization algorithm to determine which combination provides higher accuracy.

In this paper, the next section provides the state of the art on this problem; third section provides the proposed methodology; Fourth section discuss on Implementation aspects and the fifth section provides results and discussions. The last section discusses the implications of research.

2. RELATED WORK

Alurkar et al. [5], described classifying email as spam or ham by considering various parameters such as Carbon copy/Blind carbon copy, header. Each attribute would be regarded as a distinctive characteristic for the machine learning algorithm. To implement the algorithm the author had used decision theory and conditional probability.

Abd Razak and Mohamad [6], identified email as spam by considering various features in email header such as from field, received field and receiver address. Mainly discussed features in Yahoo mail, and Hotmail. The author had identified the sender SMTP by utilizing the HELLO command, and "The sender-SMTP's hostname" should be included in the HELLO command's argument field. Rathod and Pattewar [7], used a Bayesian classifier for detecting spam emails. In pre-processing step Html tag removal, stop word removal, tokenization, and word frequency are included. The performance for the Bayesian classifier is measured using

precision, recall, time taken, error rate, and accuracy.

Agarwal and Kumar [8], detected spam email using combined strategy of Naïve Bayes & PSO. The relevant characteristics from the bags of words on which premise classification is done are selected using CFS (correlation-based feature selection). Performance of both naïve bayes and PSO is measured in terms of precision, accuracy, f-score, and recall. Kaur and Sharma [9] discussed previously used techniques and spam detection methods, and also discovered a new effective technique based on approximate set theory, in addition to working better than existing methods.

Murugavel and Santhi [10] introduced various popular methods of filtering the spam and recognized the shortcomings of content-based filtering methods. The research builds on previous work by using a content-based Bayesian filter to detect misspell words. Compared with normal spam filtering methods, this work has higher extraction accuracy.

Chanda and Majumdar [11], applied supervised learning techniques to mine significant insights from the large amounts of data. Many researchers worked in the area of sentiment analysis [12] to extract the sentiments present in the reviews of different products in the e-commerce sites [13], reviews extracted from the social media on different public issues or political reviews and also to build the recommendation systems based on their interest or emotion. Classification algorithms can be used in fault tolerance systems [14] also. In all these cases, the classification algorithms play a key role to present results.

Nafis and Awang [13], proposed a hybrid feature selection technique with combination of Term Frequency and Inverse Document Frequency (TFIDF)with support vector machines (SVM). As per this study, this hybrid approach had given an acceptable level of accuracy.

To identify the polarity of comparative sentences in tweets, Alhashmi et al. [15] proposed a hybrid approach which is the combination of SVM and Bayes Factor Tree Augmentation technique. Applied this technique on the COVID-19 dataset to test the performance of this algorithm and proved as efficient. Bui et al. [16], proposed a model to identify the density of traffic in the real time using a classifier which is built with the combination of Convolutional Neural Networks (CNN) and other Machine Learning algorithms. Researchers worked towards the combination of optimization algorithms with machine learning classification algorithms. As per the Comparative study given in Khan and Sahai [17] the PSO and BAT algorithms have given high classification performance. Table 1 provides the existing spam mail detection methods.

Table 1. A survey on existing spam detection methods

S.No	Author(s)	Techniques	Dataset
1	Alurkar et al. [5]	Decision Theory, Conditional Probability	Enron
2	Rathod and Pattewar [7]	Bayesian Classifier	Gmail Dataset with different volumes of 1000,1500 and 2100 mails
3	Agarwal and Kumar [8]	Naïve Bayes, Particle Swarm Optimization, Correlation based Feature Selection	Ling Spam
4	Kaur and Sharma [9]	Particle Swarm Optimization, J48, SVM, K-means, Unsupervised Filter	Spam Base
5	Murugavel and Santhi [10]	Multi-Split Spam Corpus, MATLAB	Email

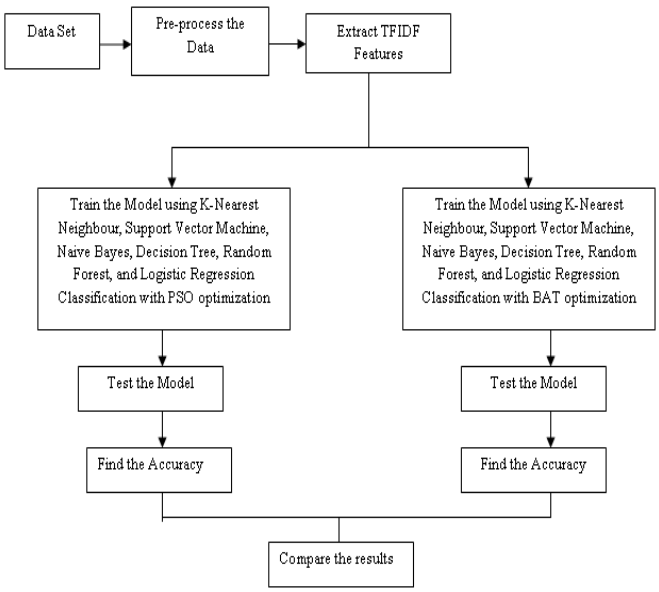

Figure 1. Workflow

3. METHODOLOGY

In the process of finding the most accurate algorithm for spam mail detection, a benchmark dataset is taken. The workflow of the proposed method is stated in the Figure 1.

3.1 Dataset

The dataset for train the model was collected from Kaggle Repository [18]. The SMS Spam Collection [19] is a collection of SMS Spam-tagged messages gathered for research purposes. It contains a single collection of 5,574 SMS messages in English that have been classified as ham (genuine) or spam. Each line in the files contains a single message. Two columns make up each line: v1 contains the label (ham or spam), and v2 has the subject matter.

3.2 Pre-process the data

There are no missing values in the dataset. The dataset consists of only two columns. If the dataset contains extra features like mail timestamp, author details, and recipient

details, these can be removed by considering as irrelevant. But in the real time, it gives more information while identifying spam mails. In this work, the dataset provides only message and the class label.

Consider the subject matter as an input and tokenize into words. Normalize the data by converting all the words into lowercase letters. Then remove the special characters and stop words which are not contributing anything towards the identification of class label (Spam or Ham). Next perform lemmatization on this data to bring the words into their base form.

3.3 Extract TFIDF features

Term Frequency and Inverse Document Frequency is a popular technique which is frequently used in NLP projects. It helps to make the vectors based on the frequency of the terms in the message. TFIDF states the importance of a particular word in the message. TFIDF of a document is a statistical value aimed at considerate the significance of a word for a document, as well as to describe links to remaining documents in the same corpus. This can be done by noting the number of times a word presents in the document, as well as noting how often it presents in remaining documents in the corpus. The Term Frequency of a document can be calculated using tf(w,d) = log(1+f(w,d)). In this the f(w,d) states the presence of word 'w' in a document 'd'. To calculate Inverse Term Frequency, idf(w,D) = log(N/f(w,D). In this N represents the Number of documents in the dataset, whereas f(w,D) represents the presence of word 'w' in the complete dataset. The TFIDF score is calculated using tfidf(w,d,D)=tf(w,d)*idf(w,D). The output of this step is a vector of v2.

3.4 Bio-inspired algorithms

The vectors of samples are portioned into training data and test data. Apply K-Nearest Neighbor, Support Vector Machine, Naive Bayes, Decision Tree, Random Forest, and Logistic Regression on the training data in the first phase to build the model. To improve the accuracy of the model, the following bio-inspired algorithms are adopted.

Particle Swarm Optimization: PSO is a swarm intelligence idea that was influenced by the human conduct of birds and fishes [8]. PSO uses the stochastic allocation property to identify a local search result first, then each item exchanges their response, and a global conclusion can be drawn. This technique is iterative, meaning it gets closer to the optimum route with each iteration. Because each item indicates a response, the PSO algorithm is based on two major variable factors: particle location and velocity, which change as per connections between the various items. Items start the procedure that consists of a population of N particle remedy, which is initialized. The location of the i^{th} item is considered as dot in the S-dimensional space, where S is the no. of factors involved. Items aim to discover the optimal global solution throughout the process.

BAT Algorithm (BA): Echolocation is a technique used by bats to detect and capture their prey. Bats create a continuous stream of piercing sounds that only heard by them during flying. Based on the species, their rhythms have different features and can be linked to their hunting techniques. When looking for prey and focusing on prey, the loudness fluctuates from the loudest to a lowest [20]. They transmit a signal having frequency that range from 20 kilo hertz to 200 kilo hertz. This signal is utilized to determine the distance S when it deflects back after contacting the object to bat as an echo signal. The bat's destination is the shortest distance between the bat and any entity. They fly towards entity that has minimum path and reduces its pulse rate when bat goes near to entity [21].

3.5 Test the model

The test data has to be given as an input to this model. The model predicts the class label of every SMS as Spam or Ham. Apply testing phase using all the models which are built in the previous section.

3.6 Accuracy

In this phase, compare the predicted class labels with the actual class labels of the test data. Draw a confusion matrix for each classification algorithm with the combination of Particle Swarm Optimization and BAT Algorithm. Factors such as accuracy, f1-score, precision, and recall are calculated from the confusion matrix.

3.7 Compare results

The model should able to predict the accurate class label when a new SMS is given as an input. For this purpose, a strong algorithm with high accuracy is required. To choose such an algorithm, the results of the previous section should be compared and visualized in a proper way by providing a GUI based support.

4. IMPLEMENTATION

These experiments are conducted on Windows 8 Operating System with 1.7GHZ Intel Corei5 and with 8GB RAM. This approach is implemented by using Python. The libraries [22] that are used to build the system are listed below.

Tkinter: The standard GUI library for Python is Tkinter. Creating GUI is fast and easy when Python is combined with Tkinter. Tkinter is a robust object-oriented interface included with the Tk GUI toolkit.

Scikit-learn: The most usable and reliable machine learning library in Python is Sklearn. It provides a suite of useful techniques for ML and statistics, including clustering and dimensionality reduction.

Nltk: Most tasks, including as punctuation, tokenization, stemming, character count, lemmatization, and word count, have been incorporated into NLTK (Natural Language Toolkit). It's really elegant and simple to use.

NumPy: NumPy (numerical Python) is a library that includes objects of array with multiple dimensions and a collection of functions for manipulating them. NumPy provides a way to conduct array operations such as arithmetic and logical operations. It also covers array functions, indexing kinds, and other issues.

Pandas: Pandas is a package which offers maximum accuracy, user-friendly data structures and analytics capabilities. It is BSD-licensed and open-source. It can be utilized in a number of areas, including academia or business.

PySwarms: PySwarms is a Python-based research tool for PSO. It is aimed for swarm intelligence researchers, practitioners, and students who want to use a declarative high-level interface to apply PSO in their issues. PySwarms offers interaction with swarm optimizations and basic optimization

with PSO.

SwarmPackagePy: SwarmPackagePy is a collection of swarm optimization algorithms written in Python. It contains 14 optimization algorithms, each of which can be used to solve a different type of optimization issue.

In the proposed work, PSO and BAT are used as optimization algorithms for various classification algorithms. Work is implemented using Tkinter module as shown in the following Figure 2.

Figure 2. GUI representation

4.1 Upload dataset

This step is used to upload the dataset to our application using filedialog module. For constructing file/directory selection windows, the tkinter filedialog module includes classes and factory functions. File dialogues assist you in opening, saving, and deleting files and directories. There is no need to manually create all of the code for this dialogue because it is generated by the module Filedialog. A spoken filename is used to create an open file dialog which asks for filename. When dataset is loaded, it displays few values from dataset and this dataset has two columns where first column contains message label as 'spam' or 'ham' and second column contains email message.

4.2 Pre-process dataset

A data mining strategy is pre-processing data which transforms unprocessed data into a readable format. Actual information is frequently inadequate, irregular, or missing in particular behaviors instead of trends, and also being riddled with errors. Data pre-processing is really a tried-and-true method of resolving such problems. In this step clean all messages by removing stop words and special symbols.

4.3 TFIDF feature generation

The Term Frequency & Inverse Document Frequency (TFIDF), would be a strong feature engineering methodology for identifying essential or, more specifically, unusual terms in text data [23-25]. Almost all text data applications, such as classification, information retrieval systems, and text data mining, are possible. The first row contains words from messages and remaining rows contains count of that word. If word is available, then that row will have count of word else 0 will be displayed.

4.4 Machine learning algorithms with PSO

PSO is a basic conceptual structure and the analogy of birds flocking, which helps to visualize the searching process. Location and velocity are the two most important properties of each particle. Using the velocity, each particle travels to a new location. Optimum place of each particle and optimum place of swarm are modified as required until a new position is reached. Decision Tree, Naïve Bayes, Logistic Regression, SVM, Random Forest, and KNN techniques are used along with PSO.

Step 1: Initialize the number of features to a variable

Step 2: Create PSO objects

Step 3: Optimize the features

Step 4: PSO selects important features where the value is 1.

4.5 Machine learning algorithms with BAT

The Bat approach is a global optimization meta heuristic algorithm. It was motivated by micro bat locomotion, which has changing pulse rates of emission and loudness. Decision Tree, Naïve Bayes, Logistic Regression, SVM, Random Forest, and KNN techniques are used along with BAT. To solve an optimization problem, the following assumptions are made to represent bat echolocation properties:

i. Echolocation is used by all bats to detect distance.

ii. Bats look for prey by flying at random speeds vj at location xj with a fixed wavelength or frequency fmin, fluctuating frequency, & volume A0.

iii. In reaction to the vicinity of the prey, bats adjust their rate of emission of pulses r [0 1] & frequency or wavelength.

iv. Bats' volume shifts from a high A0 to a low Amin value as they move towards prey.

5. RESULTS AND DISCUSSIONS

5.1 Without optimization

Several algorithms related to classification are trained without optimization and the values of Precision, Recall, F1score and accuracy are shown in Table 2.

5.2 With PSO

A number of classification techniques are trained with particle swarm optimization algorithm and the values of Precision, Recall, F1score and accuracy are shown in Table 3. PSO gives more accurate results when compared to classification algorithms without optimization.

Table 2. Classification algorithm without optimization

Classification Algorithms	Precision	Recall	F1-Score	Accuracy
SVM	94.09	85.34	89.55	93.89
Naïve Bayes	58.32	70.45	49.86	53.34
Decision Tree	87.56	87.23	87.14	92.91
Random Forest	94.25	89.64	91.48	95.18
K-Nearest Neighbor	92.19	75.65	81.53	91.03
Logistic Regression	90.28	87.54	89.87	94.01

Table 3. Classification algorithms with PSO

ML Algorithm with PSO	Precision	Recall	F1-Score	Accuracy
SVM	97.45	88.63	92.41	96.77
Naïve Bayes	60.68	72.42	51.35	55.69
Decision Tree	90.98	90.52	90.75	95.69
Random Forest	96.52	91.74	93.94	97.30
K-Nearest Neighbor	95.27	78.98	84.70	94.08
Logistic Regression	92.81	89.89	91.27	96.05

Table 4. Classification algorithms with BAT

ML Algorithm with BAT	Precision	Recall	F1-Score	Accuracy
SVM	99.94	99.66	99.80	99.91
Naïve Bayes	97.48	99.58	98.50	99.28
Decision Tree	99.94	99.66	99.80	99.91
Random Forest	99.94	99.66	99.80	99.91
K-Nearest Neighbor	99.94	99.66	99.80	99.91
Logistic Regression	99.94	99.66	99.80	99.91

5.3 With BAT

Many ML algorithms are trained with BAT optimization algorithm and the values of Precision, Recall, F1score and accuracy are shown in Table 4. BAT optimization gives more accurate results than PSO algorithm.

Here, the comparison of PSO and BAT using various classification algorithms can be clearly shown.

(1) SVM when applied with BAT gave higher accuracy when SVM applied with PSO as shown in Figure 3. SVM without optimization is less accurate than SVM with optimization. Overall, between PSO and BAT, BAT produces efficient results.
(2) Naïve Bayes has produced very lower accuracy when applied with PSO. But when it applied with BAT, the accuracy has increased that can be clearly shown in Figure 4.
(3) Decision Tree with BAT has given higher accuracy than Decision Tree with PSO in Figure 5.
(4) From Figure 6, there is slight change in accuracy between Random Forest with PSO and BAT. But among both optimizations, BAT has given better results.
(5) K-Nearest Neighbor (KNN) when applied with PSO is compared with BAT in Figure 7.
(6) Logistic Regression with BAT worked more efficient and gave maximum accuracy than Logistic Regression with PSO as shown in Figure 8.

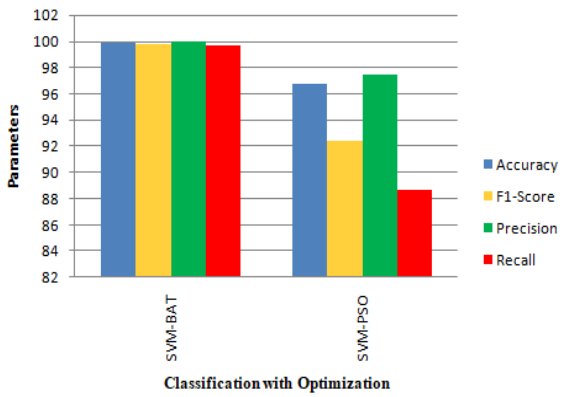

Figure 3. SVM with PSO and BAT

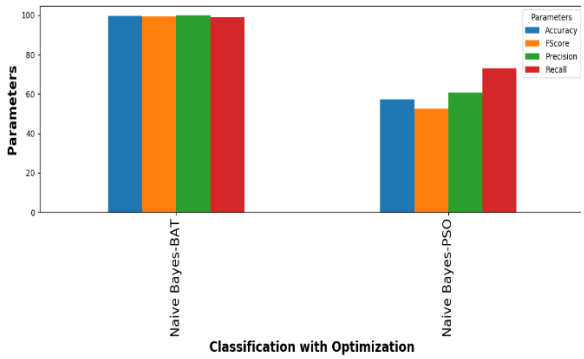

Figure 4. Naïve Bayes with PSO and BAT

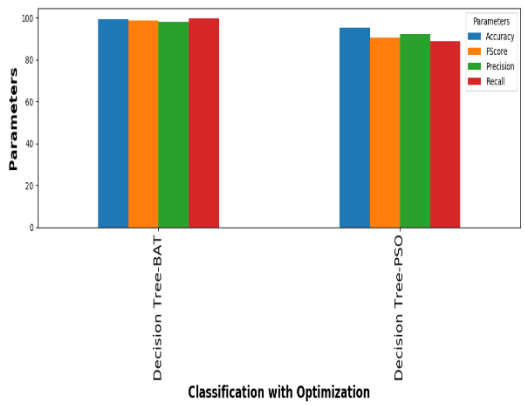

Figure 5. Decision Tree with PSO and BAT

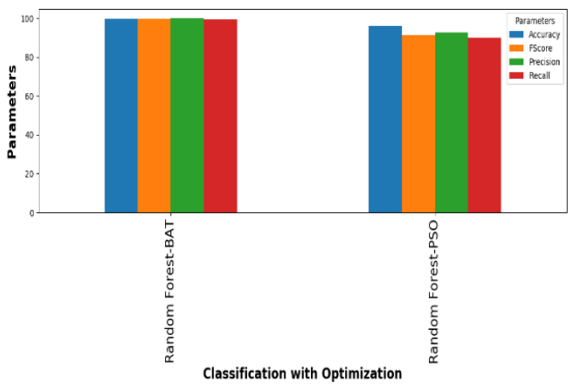

Figure 6. Random Forest with PSO and BAT

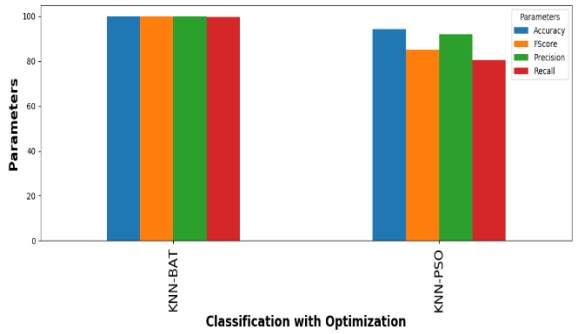

Figure 7. K-Nearest Neighbor with PSO and BAT

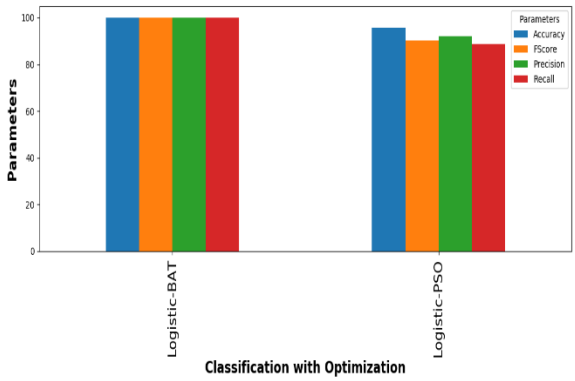

Figure 8. Logistic Regression with PSO and BAT

Among all classification algorithms, Naïve Bayes has given less accuracy. The dataset contains 500 features and after applying PSO, features have been reduced or optimized to 312 and after applying BAT, features have been reduced or optimized to 270. Finally, it is clear that every classification algorithm along with BAT provided higher accuracy when compared to PSO.

6. CONCLUSIONS

In this paper, spam emails detection model is build using classification and optimization algorithms. Based on comparison of spam mail detection without optimization and with optimization, it is clear that optimization gave more accuracy. Between two optimization algorithms such as particle swarm optimization (PSO) and BAT optimization, the PSO optimization gave around 95% accuracy for every classification algorithm except Naïve Bayes and BAT optimization gave 99% accuracy for every classification algorithm. BAT optimization gives higher metrics than PSO. In future, this work can be extended by considering the meta data of the mail to detect the spam mails.

REFERENCES

[1] Gibson, S., Issac, B., Zhang, L., Jacob, S.M. (2020). Detecting spam email with machine learning optimized with bio-inspired metaheuristic algorithms. IEEE Access, 8: 187914-187932. https://doi.org/10.1109/ACCESS.2020.3030751

[2] Naem, A.A., Ghali, N.I., Saleh, A.A. (2018). Antlion optimization and boosting classifier for spam email detection. Future Computing and Informatics Journal, 3(2): 436-442. https://doi.org/10.1016/j.fcij.2018.11.006

[3] Anupriya Koneru, S.M. (2017). Cloud service broker: Selection of providers using DTRFV evaluation. Journal of Theoretical and Applied Information Technology, 95(15): 3551-3559.

[4] Koneru, A., Sreelatha, M. (2018). Broker decision verification system using MR cloud tree. International Journal of Engineering & Technology, 7(4): 3306-3311. https://doi.org/10.14419/ijet.v7i4.21583

[5] Alurkar, A.A., Ranade, S.B., Joshi, S.V., Ranade, S.S., Sonewar, P.A., Mahalle, P.N., Deshpande, A.V. (2017). A proposed data science approach for email spam classification using machine learning techniques. In 2017 Internet of Things Business Models, Users, and Networks, pp. 1-5. https://doi.org/10.1109/CTTE.2017.8260935

[6] Abd Razak, S.B., Mohamad, A.F.B. (2013). Identification of spam email based on information from email header. In 2013 13th International Conference on Intellient Systems Design and Applications, pp. 347-353. https://doi.org/10.1109/ISDA.2013.6920762

[7] Rathod, S.B., Pattewar, T.M. (2015). Content based spam detection in email using Bayesian classifier. In 2015 International Conference on Communications and Signal Processing (ICCSP), pp. 1257-1261. https://doi.org/10.1109/ICCSP.2015.7322709

[8] Agarwal, K., Kumar, T. (2018). 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. https://doi.org/10.1109/ICCONS.2018.8662957

[9] Kaur, H., Sharma, A. (2016). Improved email spam classification method using integrated particle swarm optimization and decision tree. In 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), pp. 516-521. https://doi.org/10.1109/NGCT.2016.7877470

[10] Murugavel, U., Santhi, R. (2020). Detection of spam and threads identification in E-mail spam corpus using content based text analytics method. Materials Today: Proceedings, 33: 3319-3323. https://doi.org/10.1016/j.matpr.2020.04.742

[11] Chanda, B., Majumdar, S. (2021). A technique for extracting user specified information from streaming data. In 2021 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), pp. 1-8. https://doi.org/10.23919/SPECTS52716.2021.9639305

[12] Yan, R., Xia, Z., Xie, Y., Wang, X., Song, Z. (2020). Research on sentiment classification algorithms on online review. Complexity, pp. 1-6. https://doi.org/10.1155/2020/5093620

[13] Nafis, N.S.M., Awang, S. (2021). An enhanced hybrid feature selection technique using term frequency-inverse document frequency and support vector machine-recursive feature elimination for sentiment classification. IEEE Access, 9: 52177-52192. https://doi.org/10.1109/ACCESS.2021.3069001

[14] Eskandari, A., Milimonfared, J., Aghaei, M. (2021). Fault Detection and Classification for Photovoltaic Systems Based on Hierarchical Classification and Machine Learning Technique. IEEE Transactions on Industrial Electronics, 68(12): 12750-12759.

https://doi.org/10.1109/TIE.2020.3047066
[15] Alhashmi, S.M., Khedr, A.M., Arif, I., El Bannany, M. (2021). Using a hybrid-classification method to analyze twitter data during critical events. IEEE Access, 9: 141023-141035. https://doi.org/10.1109/ACCESS.2021.311906
[16] Bui, K.H.N., Oh, H., Yi, H. (2020). Traffic density classification using sound datasets: An empirical study on traffic flow at asymmetric roads. IEEE Access, 8: 125671-125679. https://doi.org/10.1109/ACCESS.2020.3007917
[17] Khan, K., Sahai, A. (2012). A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context. International Journal of Intelligent Systems and Applications, 4(7): 23. https://doi.org/10.9781/ijimai.2012.173
[18] UCI Machine Learning. (2016). SMS Spam Collection Dataset. Kaggle, India.
[19] Kaggle. (n.d). Retrieved from https://www.kaggle.com/datasets?search=spam+mail&datasetsOnly=true.
[20] Arulanand, N., Premalatha, K. (2014). Bin bloom filter using heuristic optimization techniques for spam detection. International Journal of Computer and Information Engineering, 8(8): 1472-1478. https://doi.org/10.5281/zenodo.1095923
[21] Mishra, S., Shaw, K., Mishra, D. (2012). A new meta-heuristic bat inspired classification approach for microarray data. Procedia Technology, 4: 802-806. https://doi.org/10.1016/j.protcy.2012.05.131
[22] Python.org. (n.d). Retrieved from https://pythonbasics.org/machine-learning-libraries/.
[23] Koneru, A., Yamuna, S., Pavan, G., Divya, B. (2020). FBP recommendation system through sentiment analysis. International Journal of Advanced Science and Technology, 29(5): 896-907.
[24] Chang, J.R., Chen, L.S., Chang, C.W. (2020). New term weighting methods for classifying textual sentiment data. International Journal of Applied Science and Engineering, 17(3): 257-268. https://doi.org/10.6703/IJASE.202009_17(3).257
[25] Koneru, A., Bhavani, N.B.N.S.R., Rao, K.P., Prakash, G.S., Kumar, I.P., Kumar, V.V. (2018). Sentiment analysis on top five cloud service providers in the market. In 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), pp. 293-297. https://doi.org/10.1109/ICOEI.2018.8553970