Grid Search Tuning of Hyperparameters in Random Forest Classifier for Customer Feedback Sentiment Prediction

Siji George C G[1]
Research Scholar
Department of Computer Science
CMS College of Science and Commerce
Coimbatore, Tamilnadu, India

B.Sumathi[2]
Associate Professor
Department of Computer Science
CMS College of Science and Commerce
Coimbatore, Tamilnadu, India

***Abstract*—Text classification is a common task in machine learning. One of the supervised classification algorithm called Random Forest has been generally used for this task. There is a group of parameters in Random Forest classifier which need to be tuned. If proper tuning is performed on these hyperparameters, the classifier will give a better result. This paper proposes a hybrid approach of Random Forest classifier and Grid Search method for customer feedback data analysis. The tuning approach of Grid Search is applied for tuning the hyperparameters of Random Forest classifier. The Random Forest classifier is used for customer feedback data analysis and then the result is compared with the results which get after applying Grid Search method. The proposed approach provided a promising result in customer feedback data analysis. The experiments in this work show that the accuracy of the proposed model to predict the sentiment on customer feedback data is greater than the performance accuracy obtained by the model without applying parameter tuning.**

***Keywords*—Classification; grid search; hyperparameters; parameter tuning; random forest classifier; sentiment analysis**

I. INTRODUCTION

The Classification is a text mining tasks in which class of a particular input is identified by using a given set of labelled data. Both supervised and unsupervised methods are used for classification. In the first method, learning is done through predefined labelled data. In this, a set of labelled input documents are given to the model by the end-user. The two main categories of supervised learning are parametric and non-parametric classification. The probability distribution of each class is the base of parametric classification. If the density function is known, it will be better to use non-parametric classification. Recently, people are using this classification process especially supervised classification to develop multiple interesting platforms for business. Sentiment analysis is the most attractive platforms which make use of the advantages of supervised classification methods.

Sentiment can be described as a person's feeling about a particular thing. It includes the task of binary classification in which documents are classified into two different classes such as positive sentiment or negative sentiment. Due to the fast popularity of social networks [1], people are using it for sharing their views, opinion and ideas. Social networks provide a platform for the people to create a virtual civilization [2]. Sentiment analysis is a mining process based on user-generated comments to identify positive or negative feelings. Opinions are always important to a business. Most of the business decision is performed based on customers' reviews. The analysis of customer or product review involves the extraction of sentiment from product document [3]. Business organizations are very conscious to know whether customers like their product or service, what customers feel about the product, which type of product or service customers like or dislike, etc.

Sentiment analysis is usually applied text input which help to identify the sentiment in a particular document and thus it is considered as the main part of text mining. Other than text classification, it requires more knowledge of the language. Generally, machine learning algorithms are considering the occurrence of the words in a document, so it tough to recognize the supreme attitude in that specific document. The sentiment analysis should be the process of identifying the polarity present in the given text or document i.e., positive or negative.

There are number of supervised machine learning algorithms are used for sentiment analysis. The performance of these classification algorithm is depending on its specific domain [4]. Random Forest classifier is largely used for this purpose. It is considered as an ensemble method [5] which generates many classifiers and finally aggregates their result for prediction. This will create a number of decision trees in the training phase [6]. The risk of noise and outliers will be high when having a single tree in classifier and it will definitely reduce the output of the processing. Due to the randomness property of Random Forest classifier, it is highly robust to outliers and noises. This classifier can handle missing values also.

One better approach to increase the outcome of any classifier is to tune the hyperparameters of that classifier [7]. The parameters that are set by the data analysts before the training process is called hyperparameters and it is independent of the training process. For example, in a random forest, a hyperparameter would be how many trees have to be

included in the forest or how many nodes each tree can have. Optimizing these hyperparameters for the classifier is the key to the perfect prediction of unlabeled data. These can only be achieved through trial and error methods. Different values of hyperparameters are used, then compare their result and finally find the best combination of them. The tuning process of hyperparameters is mainly depended on experimental results and not the theoretical result.

In this work, the Grid Search approach is applied for tuning Random Forest classifier and tried to identify the best hyperparameters. The implementation of Grid Search is simple [8]. A set of hyperparameters and their values are feed to it first and then run an exhaustive search overall all possible combination of given values then training the model for each set of values. Then Grid Search algorithm will compare the score of each model it trains and keeps the best one. A common extension of Grid Search is to use cross-validation i.e., training the model on several different folds with different hyperparameter combinations to find more accurate results.

The rest of the paper is organized as follows. In Section II, previous work in these research topics are discussed. Section III explains the proposed system model and architecture. The experimental results are discussed in Section IV and it is followed by a conclusion in Section V.

II. Related Work

Rafael G. Mantovani et al. [9] made an investigation on random search and grid search methods. They aimed to tune the hyperparameters of the classifier called Support Vector Machine (SVM). Their experiment was performed by using a huge dataset, finally, they compared the performance of Random Search with four methods such as Particle Swarm Optimization, Genetic Algorithm, Grid Search method and Estimation of Distributed Algorithm. The result of this work reveals that the predictive power of SVM classifier with Random Search is same as the other four techniques used and the advantage of this combination was the lowest computational cost of the model.

Xingzhi Zhang et al. [10] effort was to propose an optimized novel of Random Forest Classifier for credit score analysis. For optimizing Random Forest Classifier, the authors developed a system called NCSM which uses grid search and feature selection. The developed model has the capability to overcome the problem of irrelevant and redundant features and got good performance accuracy. The model used the information entropy to select the optimal features. From the UCI database, two sets of data are selected as input to examine the performance of developed model. Their experiments show that proposed system has dominating the performance of some other methods.

A hybrid approach based on Random Forest and Support Vector Machine is proposed by Yassine Al Ambrani et al. in 2018 [11] for identifying Amazon product reviews. Cross-validation method with fold value 10 has been used for this work. Both Support Vector Machine and Random Forest Classifier are used by authors to do classification of product reviews. The classification result of both classifiers is with the hybrid method. The result shows that the hybrid method of random Forest and SVM outperforms the individual methods.

An ensemble-based customer review sentiment analysis is done in 2019 [12] by Ahlam Alrehili and Kholood Albalawi. The proposed method used a voting system which combines five classifiers Random Forest, Naive Bayes, SVM, bagging and boosting. Six different scenarios are performed by authors to measure the result of the proposed model against five used classifiers. They are using unigram (with/without) stop words removal, bigram (with/without) stop words removal and using trigram stop word removal. Among this, the highest accuracy of 89.87% is given by the Random Forest classifier.

Sentiment analysis on the blogs are carried by Prem Melville et al. in 2009 [13]. They combined classification of text with lexical knowledge. A unified framework is proposed by the authors and the framework used lexical information to filter information for a specific domain. The combination of training examples using Linear Pooling with background knowledge is performed well and had an accuracy of 91.21%.

The neural network has more hyperparameters which have to be set by hand. Nauria Rodriguez-Barroso et al. worked on these neural network parameters in 2019 [14]. They used SHADE evolutionary algorithm to perform optimization of different deep learning hyperparameters to perform twitter sentiment analysis. The Spanish tweets are selected as dataset for their work. The findings reveal that hyperparameters selected by SHADE algorithm help to improve the proposed model's performance.

Airline data sentiment analysis is performed by Bahrawi in 2019 [15]. Six airline tweet data from Kaggle is used for this study and Random Forest classifier is used for sentiment prediction. Classifier predicted 63% of tweets as negative, 21% as neutral and 16% as positive. The accuracy achieved by the Random Forest algorithm was only 75%. The author suggested to build model by using some other machine learning algorithms to get a better result.

A new credit scoring model called NCSM is proposed by Xingzhi et al. [16] in 2018. Grid search method and feature selection are applied for this model in order to optimize the Random Forest classifier's performance. This proposed model achieved high prediction accuracy as compared with some other commonly used methods.

III. Proposed Model

The architectural diagram of the proposed model is depicted in Fig. 1. The collected customer feedback data go through several processing stages and feature extraction is performed. After extracting the necessary features, it is given as input to the Random Forest classifier. Finally, parameter tuning by Grid Search method is applied to increase the classifier's performance. This section gives the detailed description of the proposed model.

A. Pre-Processing of Data

As an initial step, the original input data is examined in pre-processing stage and make the raw data convenient for using in classification process. It is the first and crucial step in creating a model. While creating a machine learning model, it

is not always possible to get clean and formatted data. For this, the data pre-processing task is used. The real-world data may be in an unusable format and contains missing values, noises, etc. This type of data is impossible to use directly for machine learning model. Data pre-processing is an important task to clean the original data for the machine learning model and thereby increase model accuracy and efficiency. The following steps are used for pre-processing:

- Tokenization- Divided the customer feedback input into a number of individual words called tokens.
- Removal of special characters, numbers, stop words and punctuations since it does not any sentiment.
- Stemming- It involves normalizing the input data. For example, reducing words like loves, loving and lovable into its root word i.e., love is often used in the same context.

B. Feature Extraction

In this step, new features are extracted from existing dataset and thereby reduce the count of features used for processing task. The new reduced feature set will be capable to represent majority information in the initial feature set. This text feature extraction will directly influence the accuracy of the classification. The two techniques used in this work for feature extraction are.

- CountVectorizer
- Term Frequency-Inverse Document Frequency (TF-IDF)

Count Vectorizer: The text data need special preparation before using it for predictive modelling. The number of occurrences of every word in a given document can be identified by using Count Vectorizer. It will provide a vector with frequencies of each token in the given document. Term Frequency-Inverse Document Frequency: TF represent the result after dividing the occurrence of a word in a particular document by the total count of words present in that document. IDF is used to find out the weight of rare words across the entire document in the corpus. When TF is multiplied by IDF it will result in TF-IDF.

C. Algorithms Applied

1) Sentiment classification-random forest classifier: Random Forest classifier is a flexible supervised algorithm which can be used for text classification. The working of this algorithm is based on tree collection in which every tree depends on different random variables [17]. It uses Divide-and-conquer approach. Forest represents a collection of many trees. From random subsets of input data, this algorithm will generate several small decision trees. Consider a random vector of dimension n, where A= $(A_1, A_2, \ldots\ldots, A_n)^T$ is a set of real-valued input variables and a random variable *B* which represent the real value response, then we assume an unknown joint distribution $P_{AB}(A, B)$. The goal of this algorithm is to find a prediction function *f(A)* for predicting B. A loss function L*(B, f(A))* is used to find the prediction function and it should minimize the expected value of the loss.

$$E_{AB}\left(L\left(B, f(A)\right)\right) \quad (1)$$

L(B, f(A)) is used to represent how prediction function *f(A)* is close to *B*. Zero-one loss is the choice of *L* for classification. Minimizing $E_{A,B}(L(B, f(A)))$ for zero-one loss gives

$$L\left(B, f(A)\right) = I\left(B \neq f(A)\right) = \begin{cases} 0 & if\ B = f(A) \\ 1 & otherwise \end{cases} \quad (2)$$

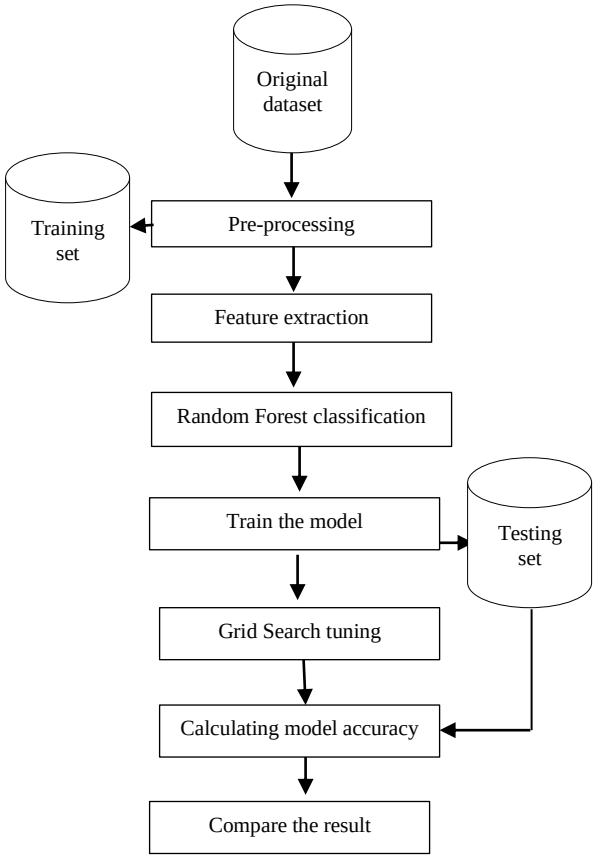

Fig. 1. System Architecture.

The procedure for the Random Forest classifier is given as follows:

RandomForestProc( )

Input: Training Data D, Number of Trees T, Total Features TF, Subset of Features tf

Output: labelled classes for the input dataset

- Repeat the followings for each trees in the forest T:
- Consider and choose a bootstrap sample S with size D from training data
- Recursively repeat the followings for generating the tree t
- Randomly select tf from total features TF
- Choose the better feature among TF
- Node to be split
- After generating trees, test instance should be given to every tree then based on majority votes, class label will be assigned.

2) Hyperparameter tuning- grid search method: Machine learning model has many parameters [18] to tune and by tweaking these parameters, the performance of the model can improve. Hyperparameter tuning is the best method to execute a different number of parameter combinations to assess a classifier's performance. Assessing a classifier by using training data will cause a fundamental machine learning problem called overfitting. The overfitting is the situation in which a model performs poorly on test data and highly on raining data. Therefore, cross-validation is used with the grid search method for hyperparameter optimization.

The grid search method is an approach used to identify the optimum parameters of a classifier so that a model can accurately predict some unlabeled data. The Grid Search method is used to tune some hyperparameters which cannot directly learn from the training process. The classification model has many hyperparameters and finding the best combination of these parameters is a challenging process. One of the best methods used for this purpose is the Grid Search method. Suppose, a machine learning model X has hyperparameters h1, h2 and h3. The Grid Search method defines a range of values for each hyperparameter h1, h2 and h3. It will construct many versions of X with all possible combinations of h1, h2 and h3. This range of hyperparameter values is known as a grid.

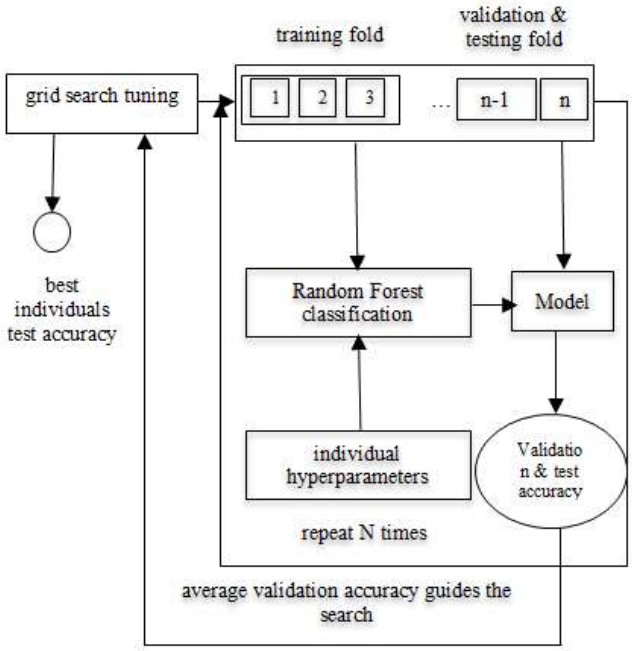

Fig. 2. Hyperparameter Tuning Architecture.

The hyperparameter tuning architecture is depicted in Fig. 2. The input data is divided into a training set, testing set and validation set. The tuning process is executed by separating the data set into n different portions. Then, the Random Forest Classifier trained in n-2 portions for each candidate solution selected by the tuning technique. The validation set is used to validate the developed model and the last portion is used to test the model. The test accuracy and validation accuracy are evaluated by using the model. Then the model is instigated by the training set and the hyperparameter value determined by the tuning technique. These steps are repeated for N times. To guide the search process, the average validation accuracy is used as the fitness value. Finally, it will return the individual with the highest accuracy and the performance of the method is the average test accuracy of that individual. The procedure for the proposed hybrid model of Random Forest classifier and Grid Search method for sentiment prediction is as follows.

RandomForest+GtridSearchProc( )

- Consider binary classification dataset of product reviews N samples and split it in a train and test set
- Define a pipeline with Random Forest Classifier
- Setup a grid for total number of features to be used and total number of trees to be constructed in the forest
- Define a function to run Grid Search method which takes the input such as defined pipeline, parameter grid and train and test set
- Define the objective function which takes set of hyper-parameters and output the accuracy score

$$accuracy = f(hyperparameters)$$

- Select a random combination
- Define the number of search iteration
- Iterate through all possible combination of values specified in the grid one at a time
- Pass these values to the objective function
- Repeatedly execute the objective functions for each and every combination of hyper-parameter values
- Evaluate the best hyper-parameter which maximize the accuracy

$$hyperparams *= argmax\ f(hyperparams)$$

IV. Experiments and Results

Labelled customer feedback data on electronic items collected from UCI database and it is used as input for this work. It includes 1500 reviews (750 positive and 750 negative reviews). This work aims to classify these customer feedbacks into two different categories such as positive feedback and negative feedback. 7-fold cross-validation is used for calculating the model's accuracy. First, customer feedback data analysis is performed by using Random Forest classifier with default hyperparameters and achieved 84.53% of accuracy. Table I gives the result of customer feedback analysis using Random Forest classifier.

From the Table I, it is clear that 1268 customer reviews are classified correctly among 1500 and 232 are wrongly classified by the model.

TABLE I. Random Forest Cross Validation Result

	Positive	Negative	Total
Positive	682	68	750
Negative	164	586	750
Total	846	654	1500

To increase the accuracy of the classifier, parameter tuning using Grid Search method is used in this work. The Random Forest classifier has several parameters, which can be adjusted to get optimal performance. Two of those parameters are the number of trees constructed for classifying new data and the maximum number of variables used in individual trees. The class GridSearchCV available in Scikit Learn is used for this study. The GridSearchCV evaluates, all possible combinations of parameter values and finally, the best parameter combination is retained. This work mainly concentrates on two parameters of Random Forest classifier.

The GridSearchCV uses max_features for denoting the maximum number of variables used in independent trees and n_estimators for denoting the total number of trees to be constructed in the forest. The Table II provides the result of parameter tuning of Random Classifier on customer feedback data. The score in Table II represents the accuracy of the classifier using the 7-fold cross-validation method. sqrt and log2 are the two options tried for max_features.

According to Table II, the highest accuracy of the Random Forest Classifier is 90.02% at the parameters 'max_features'='sqrt' and 'n_estimators'=400.

The Table III shows the result of the proposed method (with best parameters) which uses the Grid Search approach for hyperparameter tuning. By using the proposed method, among 1500 reviews, 1353 reviews of customers are classified correctly and 147 are not. The accuracy comparison of two used methods is given in Table IV.

Fig. 3 depicts the total number of instances classified by Random Forest Classifier and proposed system.

Fig. 4 depicts the most frequent words identified by the proposed system. It shows the top 15 words appeared in the customer feedback data and from the figure it is clear that the most frequent word is product.

A detailed comparison of Random Forest classifier and proposed system which uses the Grid Search method for parameter tuning is depicted in Table V.

TABLE II. BEST PARAMETERS IDENTIFIED BY GRID SEARCH

No	Tuning Parameters	Score(7-foldcross validation)	Best parameter
1	n_estimators= [10,100,1000,1500] max_feature=[sqrt , log2]	**87.04**	{n_estimators=1000 max_features=sqrt }
2	n_estimators= [600,1000,1300] max_features= [sqrt]	**89.56**	{n_estimators=600 max_features= sqrt}
3	n_estimators= [400,600,800,900] max_features= [sqrt]	**90.02**	{n_estimators= 400 max_features=sqrt,}
4	n_estimators= [300,400,500,550] max_features= [sqrt]	**90.02**	{n_estimators= 400 max_features=sqrt,}
5	n_estimators= [325,350,400,450] max_features= [sqrt]	**90.02**	{n_estimators= 400 max_features=sqrt,}
6	n_estimators= [340,380,400,425] max_features= [sqrt]	**90.02**	{n_estimators= 400 max_features=sqrt}

TABLE III. CROSS VALIDATION RESULT OF PROPOSED METHOD

	Positive	Negative	Total
Positive	721	29	750
Negative	118	632	750
Total	839	661	1500

TABLE IV. COMPARISON BASED ON CLASSIFIED INSTANCES

	True classification	Wrong classification	Accuracy (%)	Time taken (Seconds)
Random Forest	1268	68	84.53	6.70
Proposed Method	1353	147	90.02	8.02

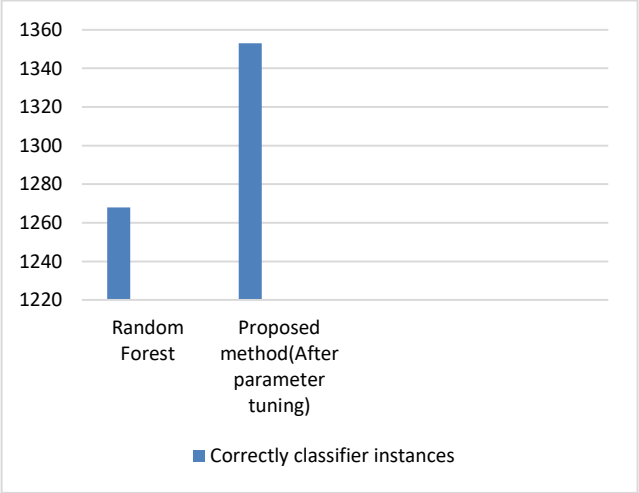

Fig. 3. Number of Instances Classified Correctly.

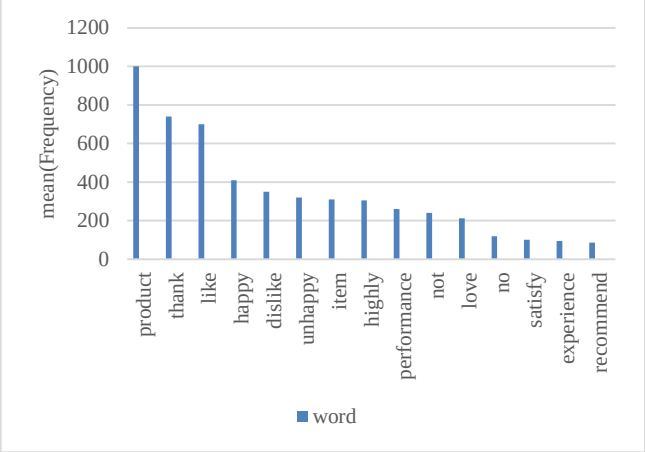

Fig. 4. Most Frequent Words.

TABLE V. COMPARISON

	Random Forest	Proposed Method(Random Forest + Grid Search)
Sensitivity	90.93	96.13
Specificity	78.13	84.26
Recall	90.93	96.13
Precision	80.61	85.93
F-Measure	85.45	90.74

V. CONCLUSION

Sentiment analysis is essential for a business organization to perform decision making. It can be used for different tasks such as calculating or expressing sentiment on any product or service. In this work, the best parameters are tuned by Grid Search method for Random Forest classifier. Experimental results on customer feedback data show that Random Forest provides the best result with an accuracy of 84.53%. But, by tuning number of maximum trees in the forest and depth of trees, the accuracy of the developed model increases to 90.02%. The result shows that parameter tuning has successfully helped to generate the best model to classify new data. At the same time, the Random Forest classifier take more execution time when the number of trees in the forest is increased. In the future work, the proposed model can use for multi-class sentiment prediction since it concentrated binary classification only.

REFERENCES

[1] M. Ahmad, S. Aftab, S.S Muhammad, and S. Ahmad, 2017. Machine learning techniques for sentiment analysis- A review, International Journal of Multidisciplinary Science and Engineering, vol. 8, no. 3, pp. 27-32.

[2] S. H. Yadav,and P. M. Manwatkar, 2015. An approach for offensive text detection and prevention in social network, 2015 IEEE International Conference Innovations in. Information, Embedded and Communication Systems (ICIIECS), pp. 3-6.

[3] Bagus Setya Rintyarna, Riyanarto Sarno, and Chastine Fatichah, 2020. Enhancing the Performance of Sentiment Analysis Task on Product Reviews by Handling Both Local and Global Context, International Journal of Information and Decision Sciences.

[4] R. Xia, C. Zonga, and S. Li, 2011. Ensemble of feature sets and classification algorithms for sentiment classification, Information Sciences, Elsevier, vol. 181, pp.1138-1152.

[5] Yashaswini Hegde, and S.K. Padma, 2017. Sentiment Analysis Using Random Forest Ensemble for Mobile Product Reviews in Kannada, 7th International Advance Computing Conference(IACC), IEEE.

[6] Shahnoor C. Eshan, and Mohammad S Hasan, 2017. An Application of Machine learning to Detect Abusive Bengali Text, 20th International Conference of Computer and Information Technology(ICCIT).

[7] Muhammad Murtadha Ramadhan, Imas Sukaesih Sitanggaang, Fahredi Rizky Nasution and Abdullah Ghifari, 2017. Parameter Tuning in random Forest Based on Grid Search Method for Gender Classification Based on Voice Frequency, International Conference on Computer, Electronics and Communication Engineering.

[8] J. Bergstra, and Y. Bengio, 2012. Random search for hyper-parameter optimization, Journal of Machine Learning Research, vol. 13, pp. 281-305.

[9] Rafael G. Mantovani, Andre L. D. Rossi, Joaquin Vanschoren, Bernd Bischl and Andre C. P. L. F., 2015. Effectiveness of Random Search in SVM hyper-parameter Tuning. IEEE Proceedings of the 2015 International Joint Conference on Neural Networks, July 2015.

[10] Xingzhi Zhang, Yan Yang, and Zhurong Zhou, 2018. A Novel Credit Scoring Model based on Optimized Random Forest. 8th Annual Computing and Communication Workshop and Conference (CCWC).

[11] Yassine Al Ambrani, Mohamed Lazaar, and Kamal Eddine El Kadiri, 2018. Random Forest and Support Vector Machine Based Hybrid Approach to Sentiment Analysis. The First International Conference on Intelligent Computing in Data sciences, vol. 127, pp. 511-520.

[12] Ahlam Alrehili and Kholood Albalawi, 2019. Sentiment Analysis of Customer Reviews using Ensemble Method, International Conference on Computer and Information Sciences (ICCIS), IEEE.

[13] Prem Melville, Wojciech Gryc, and Richard D. Lawrence, 2019. Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification, Knowledge Discovery and Datamining (KDD'09), Paris France.

[14] Nuria Rodriguez-Barroso, Antonio R. Moya, Jose A. Fernandez, Elena Romero, Eugenio Martinez-Camara, and Francisco Herrera, 2019. Deep Learning Hyper-Parameter Tuning for Sentiment Analysis in Twitter Based on Evolutionary Algorithms, Proceedings of federated Conference on Computer Science and Information Systems, pp. 255-264.

[15] Bahrawi, 2019. Sentiment Analysis using Random Forest Algorithm Online Social Media Based, Journal of Information Technology and Its Utilization, vol. 2, issue 2.

[16] Xingzhi Zhang, Yan Yang, and Zhurong Zhou, 2018. A Novel Credit Scoring Model based on Optimized Random Forest, 8th Annual Computing and Communication Workshop and Conference (CCWC).

[17] Adele Cutler, D. Richard Cutler, and John R. Stevens, 2012. Random Forests, Ensemble Machine Learning, pp. 157-175.

[18] Hitesh H Parmar, Sanjay Bhanderi, and Glory Shah, 2014. Sentiment Mining of Movie Reviews using Random Forest with Tuned Hyper-parameters, International Conference on Information Science.