International Journal of Statistics and Applications 2024, 14(1): 7-12
DOI: 10.5923/j.statistics.20241401.02

Comparing Regular Random Forest Model with Weighted Random Forest Model for Classification Problem

Sanjib Ghosh

Assistant Professor, Department of Statistics, University of Chittagong, Chittagong, Bangladesh

Abstract Several studies have demonstrated that effectively combining machine learning models can improve the individual predictions made by the base models. Random forests allow for the selection of a random number of features while bagging increases diversity by sampling with replacement and generating multiple training data sets. As a result, random forest has become a strong contender for various machine learning applications. Assuming equal weights for each base decision tree, however, seems unreasonable because different base decision trees may have varying decision-making abilities due to randomization in sampling and input feature selection. As a result, we offer several methods to enhance the regular random forest's weighting approach and prediction quality. The developed weighting frameworks include multiple stacking-based weighted random forest models, optimal weighted random forest based on area under the curve (AUC), and ideal weighted random forest based on accuracy. The numerical result shows that the stacking-based random forest with binary prediction can introduce significant improvements compared to regular random forest.

Keywords Optimization, Stacking, Weighted random forest, Out-of-bag prediction, Ensemble

1. Introduction

Several studies have shown that creating ensembles of base learners can significantly improve learning performance. Boosting [1], random forests [2], bagging [3], and their variations are among the most commonly utilized examples of this approach. When it comes to classification and regression, boosting and random forests are comparable and sometimes even outperform, state-of-the-art techniques [4]. The margin and correlation of base classifiers are typically used to describe the effectiveness of ensemble approaches [5]. Base classifiers must be accurate and diversified, meaning they should predict differently, to have a decent ensemble. The ensemble's extremely accurate predictions are then guaranteed by the voting mechanism that runs on the top of the base learners. It is important to have a variety of decision-makers in the "committee" of basic models to make better decisions. This diversity, referred to as the base learners' "diversity," is essential as there will be no progress from a collection of similar models. Ensemble models that perform well individually and collectively have been shown to exhibit diversity in base learners. Techniques like bagging, random forests, and boosting algorithms have been utilized to add variety to ensemble models.

In the bagging method described in [3], N samples are considered, replacing the training data to produce N training data sets. Each of these sets is used to build a learning algorithm, usually a decision tree. The final prediction is made by averaging or voting on the class label. Bagging introduces random discrepancies between the input data sets to add variety to the ensemble model. The random forest learning method [6] adds further variability to the bagging process. To reduce the interconnection among the constructed trees, the random forest method chooses a random set of features each time, while also using replacement to generate N training datasets. The result is again an average or a collective vote based on all predictions made by the forest's-built trees (Fig. 1).

Random forests are commonly used in various applications and have shown impressive performance. However, making minor adjustments in how the base learners are combined could enhance the predictions even more. Using a simple average for the final predictions, assuming that all base learners have equal weights, may seem illogical. The reason for randomizing input feature selection and sampling is to ensure that every constructed tree is not capable of making the same decisions [7]. Therefore, implementing a weighting process to weigh the trees based on their performance seems fair.

* Corresponding author:
sanjib.stat@cu.ac.bd (Sanjib Ghosh)
Received: Feb. 20, 2024; Accepted: Mar. 5, 2024; Published: Mar. 22, 2024
Published online at http://journal.sapub.org/statistics

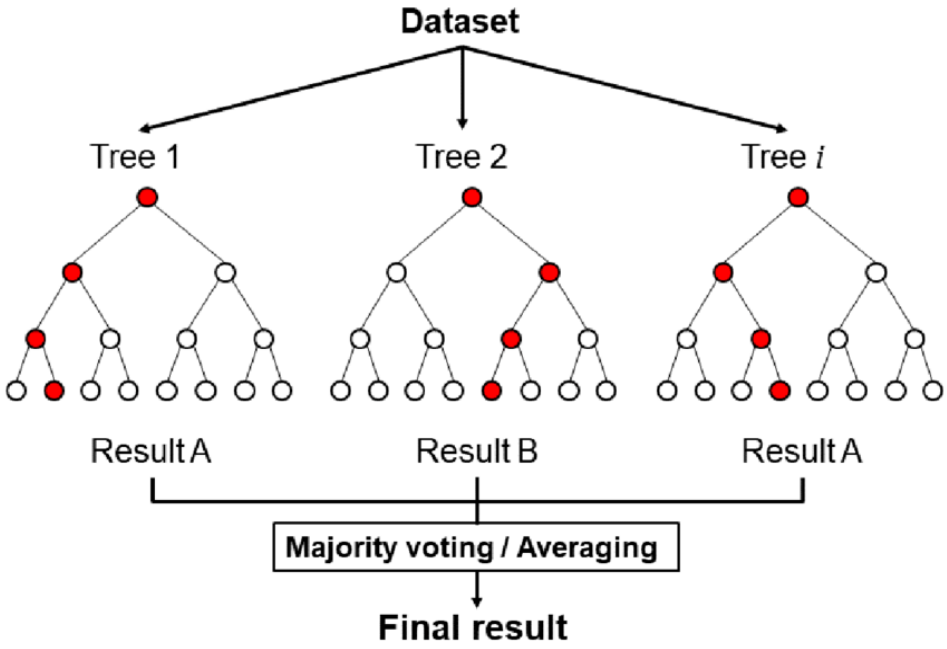

Figure 1. Random forest classifier uses majority voting of the predictions made by randomly created decision trees to make the final predictions

[7] proposed tree-weighted random forest (TWRF) method for classifying high dimensional noisy data. They contended that a new approach for ranking the trees based on their classification capability could serve as a solution for random forests, which are affected by noisy data and susceptible to making inaccurate decisions. By assessing the trees in the forest using an out-of-bag (OOB) subset of the training data, they calculated the tree weights based on the OOB accuracies. The results demonstrated that TWRF outperformed the regular random forest.

Various studies have proposed changes to the weighted random forest, and have demonstrated slight improvements in their results compared to traditional random forest predictions [8]. For instance, [9] examined some adjustments to enhance the performance of the random forest algorithm. Instead of using the Gini index for evaluation, the author combined five attributes evaluation measures. Additionally, the article identified the most similar examples to the target instance and suggested a weighted random forest with weights based on the vote margins of these similar instances. Numerical results on various datasets showed the effectiveness of the proposed method in improving performance.

In a separate study, [10] proposed a probabilistic weighted system for combining forest trees, considering four combination methods: recall combiner, majority vote, weighted majority vote, and naive Bayes combiner. The weighted majority vote is a preferable weighting approach for small unbalanced data sets, while the naïve Bayes combiner is somewhat superior to other options, particularly for large balanced data sets based on experimental results with 73 data sets.

In this study, we propose optimization and stacking-based weighting mechanisms to combine the trees of the forest more effectively. For integrating the benefits of the model combinations having diverse input models is a key. To achieve this, we aim to keep the trees of the forest shallow to prevent near-identical, non-diverse trees as input features. The designed models include optimal weighted random forest based on accuracy, optimal weighted random forest based on area under the curve (AUC), and several stacking-based weighted random forest models.

2. Material and Methods

A diverse set of initial base learners is necessary to improve the performance of the ensemble model. This means that there should be little association between the base models. Thus, the ensemble random forest models are designed with the assumption that the trees in the forest should be constructed shallow, meaning they should not have large depths. This results in a reasonable degree of variation amongst the base decision trees. The improved designed weighted random forest models are explained below.

2.1. Random Forest

Breiman [2] employed random forests, which are made up of an ensemble of K classifiers, $h_1(x)$, $h_2(x)$,..., $h_k(x)$. A winning class is assigned to an instance that is being classed, with each classifier casting a vote for one of the classes. The combined classifier is represented by h(x). A replacement is chosen at random from the training set of n instances for each training set of n instances. By using a sampling technique known as bootstrap replication, each tree is constructed with an average of 36.8% fewer training instances. These "out-of-bag" examples are useful when estimating the strength and correlation of the forest internally.

For classifier h_k, denote the set of out-of-bag instances as O_k. Let Q(x, y_j) represent the out-of-bag percentage of votes for class y_j at input x and P(h(x) = y_j) be an estimation.

$$Q(x, y_j) = \frac{\sum_{k=1}^{K} I(h_k(x) = y_j; (x, y) \in O_k)}{\sum_{k=1}^{K} I(h_k(x); (x, y) \in O_k)}$$

The indicator function is denoted by I(·). The margin function calculates the difference between the average vote in class y and the average vote in any other class:

$$mr(x, y) = P(h(x) = y) - \max_{\substack{j=1 \\ j \neq y}} P(h(x) = y_j)$$

It is estimated with Q(x,y) and Q(x,y_j). Strength is defined as the expected margin, and is computed as the average over the training set:

$$S = \frac{1}{n} \sum_{i=1}^{n} (Q(x_i, y) - \max_{\substack{j=1 \\ j \neq y}} Q(x_i, y_j))$$

The average correlation is computed as the variance of the margin over the square of standard deviation of the forest:

$$\bar{\rho} = \frac{\frac{1}{n} \sum_{i=1}^{n} (Q(x_i, y) - \max_{\substack{j=1 \\ j \neq y}} Q(x_i, y_j))^2 - S^2}{(\frac{1}{K} \sum_{k=1}^{K} \sqrt{(p_k + \hat{p}_k + (p_k - \hat{p}_k)^2)})^2}$$

$$where, p_k = \frac{\sum_{(x_{i,y \in O_k})} I(h_k(x) = y)}{\sum_{(x_{i,y \in O_k})} I(h_k(x))}$$

is an out-of-bag estimate of P(h_k(x) = y) and

Where $\hat{p}_k = \frac{\sum_{(x_{i,y \in O_k})} I(h_k(x) = \hat{y})}{\sum_{(x_{i,y \in O_k})} I(h_k(x))}$

is an out-of-bag estimate of $p(h_k(x) = \hat{y}_j$ and

$$\hat{y}_j = arg \max_{\substack{j=1 \\ j \neq y}} Q(x, y_j)$$

is estimated for every instance x in the training set with Q(x, y_j).

2.2. Accuracy-Based Optimal Weighted Random Forest

The motivation for creating an optimal weighted random forest is based on the optimization model suggested in [11], which aimed to minimize the mean squared error (MSE) of a linear combination of multiple base regressors. In this context, we present an optimization model to minimize the prediction accuracy of a weighted random forest ensemble model for binary classification, with the weights serving as decision variables. The out-of-bag predictions produced by k-fold cross-validation are treated as substitutes for unseen test observations and are utilized as inputs for the optimization problem. The following is the mathematical model.

$$\max accuracy(Y, [\sum_{j=1}^{k} w_j \hat{Y}_j + 0.5]) \quad (1)$$

$$such\ that, \sum_{j=1}^{k} w_j = 1$$

$$w_j \geq 0, \quad \forall = 1,2, \ldots \ldots k.$$

Where Y represents the vector of actual response values, Y_j is the out-of-bag prediction of decision tree j, and w_j are the weights corresponding to decision tree j (j = 1,.., k). The accuracy() function measures the percentage of accurate predictions (true positives and true negatives) among all examples investigated. In addition, for the ensemble model, $[\sum_{j=1}^{k} w_j \hat{Y}_j + 0.5]$ finds the closest integer between class labels (0 and 1). (see Fig. 2).

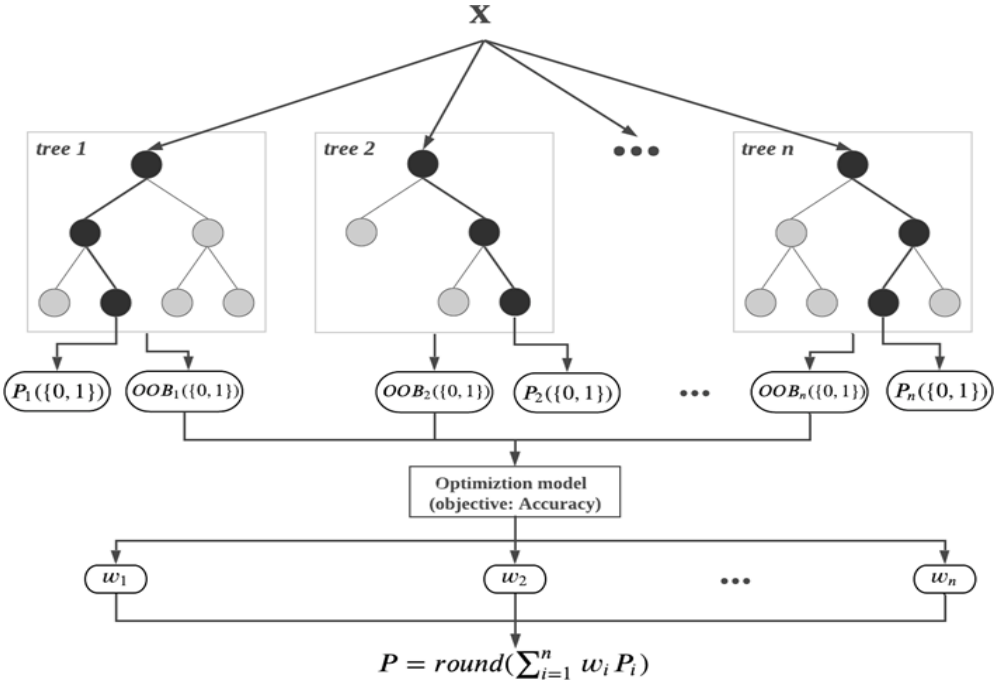

Figure 2. The optimal weighted random forest classifier utilizes out-of-bag (OOB) binary predictions from the randomly generated decision trees to enhance prediction accuracy

2.3. Area under the Curve (AUC)-Based Optimal Weighted Random Forest

The AUC (Area under the ROC curve) is a metric primarily used to compare various classifiers. The ROC curve is a commonly used graph that illustrates the balance between true positive and false positive rates at different thresholds for classification. The AUC, which represents the area under this curve, is valuable for comparing binary classifiers as it considers all potential thresholds. Furthermore, accuracy has a built-in limitation of reporting excessively high accuracy when classifying highly imbalanced data sets [12,21]. The optimization model below aims to determine the best weights for combining trees in a random forest model by optimizing the ensemble's AUC.

$$\max AUC\left(Y, \left[\sum_{j=1}^{k} w_j \widehat{P}_j\right]\right) \quad (2)$$

$$such\ that\ , \sum_{j=1}^{k} w_j = 1$$

$$w_j \geq 0, \quad \forall = 1,2, \ldots \ldots k$$

The out-of-bag probability vector of each base classifier is referred to as $\widehat{P}_j$ in the previous formulation, and the area under the ROC curve for the created ensemble is calculated by AUC().

2.4. Random Forest Model Based on Stacking

Random forest models with stacking involve combining multiple base learners to complete at least one additional level of the learning activity. The independent and dependent variables of the second-level learning problem are the actual response values of the training data and the out-of-bag predictions of the base learners [13]. Here, we used these steps to utilize the out-of-bag predictions from the forest's trees and train another machine learning model on top of them to create an enhanced random forest.

i. Build a random forest model using the training data.
ii. Using k-fold cross-validation to obtain the forest's decision trees and generate out-of-bag predictions for each tree.
iii. Create a new dataset with the response variable as the actual response values of the training data points and the input variables as the out-of-bag predictions.
iv. Train a second-level machine learning model using the generated dataset to predict test observations that have not been seen before.

As the second-level classifier, we have selected three machine learning models: logistic regression, K-nearest neighbors, and random forest. Additionally, for each second-level classifier, two scenarios are considered: either using out-of-bag predictions of the probability that an observation belongs to the majority class or using binary classifications of those predictions. In the second scenario, the probability of the actual class (class 1) is used as the input variable instead of binary predictions.

3. Experiments and Results

Ten public binary classification datasets from the UCI machine learning repository [14] were utilized to evaluate the effectiveness of the proposed enhanced weighted random forest classifiers. Minimal pre-processing work, such as handling missing values and one hot encoding, was conducted to prepare each dataset for training classification models. Twenty percent of each dataset was reserved as the test set to evaluate the actual performance of the models created, while the remaining eighty percent was used to build and optimize the ensemble.

The number of trees (n) is set at 100 to train the regular random forest and generate n randomly created decision trees. The maximum depth of the trees is set at half of the common choice for the maximum depth of random forest trees, which is the square root of the number of features ($\sqrt{p}/2$). This keeps the trees shallow and uncorrelated with one another. Ten-fold cross-validation is used to create decision tree out-of-bag predictions. The Sequential Least Squares Programming technique (SLSQP) from Python's SciPy optimization module was used to solve the optimization problems [15].

Table 1. Details of example data sets downloaded from UCI machine learning repository

Data set	Size	Features	Class 0 (%)	Class 1 (%)
1) Heart Disease	1025	14	49%	51%
2) Diabetes	768	9	65%	35%
3) Breast cancer	569	31	37%	63%
4) Indian liver patient dataset (ILPD)	583	10	28%	72%
5) Divorce predictors	170	54	49%	51%
6) Hepatitis	155	19	20%	80%
7) Bank	41188	21	89%	12%
8) Apple Quality	4000	9	49.9%	50.1%
9) Water Quality	2011	10	59.67%	40.33%
10) Smoking	55692	26	63.27%	36.73%

Table 1 displays the dimensions, number of features, and percentage of class labels for 10 sample data sets that were taken from the UCI machine learning repository. This table shows that varying sizes and percentage of class labels are covered by the selected data sets.

Table 2 shows the complete experimental results of all ensemble models created and used on the sample datasets. Two methods have been developed for stacking-based ensembles: binary OOB predictions and OOB probability forecasts of the true class label. The results indicate that at least one of the proposed models outperforms the standard random forest in 8 out of the 10 data sets considered (first column of the table). Moreover, it seems that the stacking-based random forest, with a second random forest model as the second-level classifier, outperforms other models more frequently based on binary OOB predictions (based on 10 data sets).

Table 2. Results of experiments comparing regular random forest classifiers to build improved random forest classifiers. For every data set, the top-performing classifier is highlighted. The last row displays the average accuracy of every model while taking into account every data set

SL No.	RF	Optimal WRF (Acc.)	Optimal WRF (AUC)	Log. stacked RF (binary)	Log. stacked RF (prob.)	KNN stacked RF (binary)	KNN stacked RF (prob.)	RF stacked RF (binary)	RF stacked RF (prob.)
1	85.36%	78.04%	79.02%	82.92%	86.34%	86.34%	86.82%	**90.73%**	85.36%
2	64.28%	61.68%	66.23%	**70.12%**	63.63%	67.53%	63.63%	69.48%	61.68%
3	94.69%	94.73%	94.22%	94.71%	94.76%	**94.83%**	94.69%	94.77%	93.93%
4	71.23%	71.23%	71.32%	**71.91%**	28.76%	64.38%	63.01%	63.01%	71.23%
5	97.26%	97.26%	97.26%	**97.35%**	94.40%	97.26%	97.26%	97.02%	97.12%
6	**83.22%**	83.05%	80.50%	80.10%%	80.05%	80.40%	79.65%	79.30%	79.50%
7	89.37%	90.39%	90.33%	91.09%	90.77%	90.75%	90.71%	**91.10%**	89.39%
8	**73.50%**	66.00%	56.80%	67.50%	60.12%	67.12%	70.00%	71.25%	71.30%
9	69.62%	66.12%	59.00%	67.50%	67.25%	70.75%	67.75%	**71.87%**	69.00%
10	73.82%	73.81%	67.82%	75.34%	73.50%	73.32%	74.13%	**75.40%**	71.84%
Average	**80.26%**	**78.23%**	**76.25%**	**79.83%**	**73.96%**	**79.27%**	**78.77%**	**80.39%**	**79.04%**

[1]Regular random forest classifier. [2]Optimal weighted random forest based on accuracy. [2]Optimal weighted random forest based on AUC. [4]Stacking-based random forest with logistic regression as the 2nd level classifier using binary OOB predictions. [5]Stacking-based random forest with logistic regression as the 2nd level classifier using probability OOB predictions. [6]Stacking-based random forest with KNN as the 2nd level classifier using binary OOB predictions. [7]Stacking-based random forest with KNN as the 2nd level classifier using probability OOB predictions. [8]Stacking-based random forest with random forest as the 2nd level classifier using binary OOB predictions. [9]Stacking-based random forest with random forest as the 2nd level classifier using probability OOB predictions.

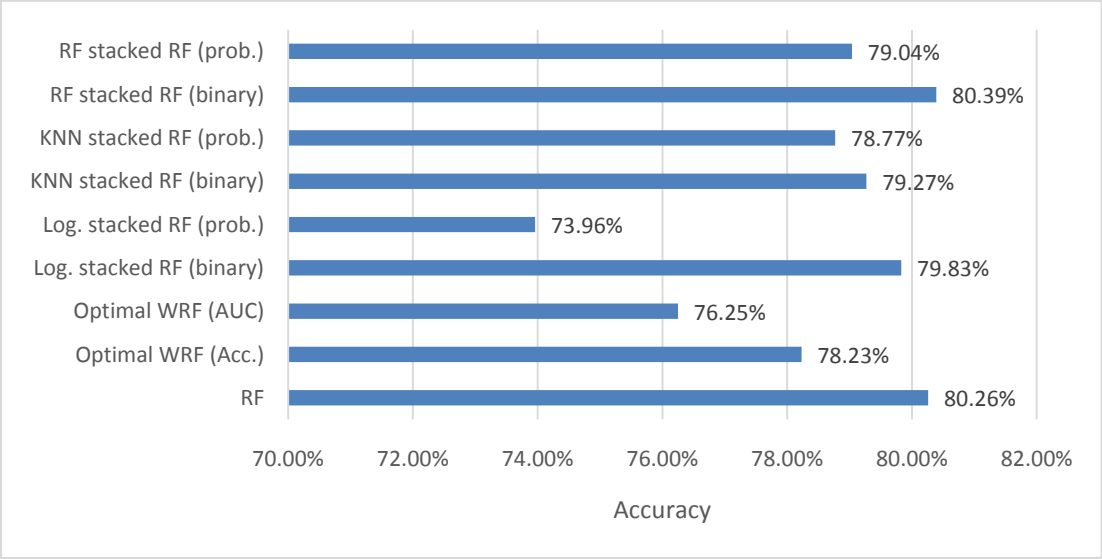

Figure 3. Comparing weighted random forest classifier with regular random forest

Figure 3 depicts the mean accuracy of the models created across all datasets in an effort to compare the performance of the suggested upgraded random forest classifiers with conventional random forests more effectively. This figure represents a comparison of the average accuracy scores of all models using a standard random forest classifier. The graph clearly indicates that RF stacked RF using binary OOB predictions outperforms regular random forest. This classifier has the potential to enhance predictions made by ordinary random forests by 0.13%.

4. Conclusions

The aim of this research was to improve the random forest, a popular machine learning model, as a classifier. Several models based on ensemble learning were developed for this purpose. The suggested models include stacking-based random forest and optimal weighted random forest using out-of-bag accuracy and AUC. The models were tested on 10 public datasets, and the findings showed that only the stacking-based random forest model was superior to the regular random forest classifier. The stacking-based random forest model, which trains a 2nd level of random forest on inner randomly created decision trees, outperformed all other generated models. After this study, future research may be explored the directions: finding an optimal weight solution through additional optimization techniques [16,17], and combining bagged and boosted trees to enhance prediction accuracy while reducing bias and variance. Developing a comparable framework to enhance random forest regressor;

and applying the same concept to other fields or research, such as data envelopment analysis (DEA) [18,19,20].

REFERENCES

[1] Yoav Freund and Robert E. Shapire. Experiments with a new boosting algorithm. In Lorenza Saitta, editor, Machine Learning: Proceedings of the Thirteenth International Conference (ICML' 96). Morgan Kaufmann, 1996.

[2] Leo Breiman. Random forests. Machine Learning Journal, 45:5–32, 2001.

[3] Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140.

[4] David Meyer, Friedrich Leisch, and Kurt Hornik. The support vector machine under test. Neurocomputing, 55:169–186, 2003.

[5] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. In Douglas H. Fisher, editor, Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97), pages 322–330. Morgan Kaufmann, 1997.

[6] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[7] Li, H. B., Wang, W., Ding, H. W., & Dong, J. (2010, 10-12 Nov. 2010). Trees Weighting Random Forest Method for Classifying High-Dimensional Noisy Data. Paper presented at the 2010 IEEE 7th International Conference on E-Business Engineering.

[8] Pham, H., & Olafsson, S. (2019). Bagged ensembles with tunable parameters. Computational Intelligence, 35(1), 184-203.

[9] Robnik-Šikonja, M. (2004, 2004//). Improving Random Forests. Paper presented at the Machine Learning: ECML 2004, Berlin, Heidelberg.

[10] Kuncheva, L. I., & Rodríguez, J. J. (2014). A weighted voting framework for classifiers ensembles. Knowledge and Information Systems, 38(2), 259-275.

[11] Shahhosseini, M., Hu, G., & Pham, H. (2019). Optimizing Ensemble Weights and Hyperparameters of Machine Learning Models for Regression Problems. arXiv preprint arXiv: 1908.05287.

[12] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112): Springer.

[13] Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2), 241-259.

[14] Dua, D. & Graff, C. (2019). UCI Machine Learning Repository. [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[15] Jones, E., Oliphant, T., & Peterson, P. (2001). SciPy: Open source scientific tools for Python.

[16] Peykani, P., Mohammadi, E., Saen, R. F., Sadjadi, S. J., & Rostamy-Malkhalifeh, M. (2020). Data envelopment analysis and robust optimization: A review. Expert Systems, e12534.

[17] Donate, J. P., Cortez, P., Sánchez, G. G., & De Miguel, A. S. (2013). Time series forecasting using a weighted cross-validation evolutionary artificial neural network ensemble. Neurocomputing, 109, 27-32.

[18] Zheng, Z., & Padmanabhan, B. (2007). Constructing ensembles from data envelopment analysis. INFORMS Journal on Computing, 19(4), 486-496.

[19] Peykani, P., & Mohammadi, E. (2020). Window network data envelopment analysis: an application to investment companies. International Journal of Industrial Mathematics, 12(1), 89-99.

[20] Hong, H. K., Ha, S. H., Shin, C. K., Park, S. C., & Kim, S. H. (1999). Evaluating the efficiency of system integration projects using data envelopment analysis (DEA) and machine learning. Expert Systems with Applications, 16(3), 283-296.

[21] Jerrold H. Zar. Biostatistical Analysis (4th Edition). Prentice Hall, Englewood Clifs, New Jersey, 1998.