An Optimized Bagging Ensemble Learning Approach Using BESTrees for Predicting Students' Performance

https://doi.org/10.3991/ijet.v18i10.38115

Edmund Evangelista(✉)
Zayed University, Abu Dhabi Campus, United Arab Emirates
edmund.evangelista@zu.ac.ae

Abstract—Every academic institution's goal is to identify students who require additional assistance and to take appropriate actions to improve their performance. As such, various research studies have focused on developing prediction models that can detect correlated patterns influencing students' performance, dropout, collaboration, and engagement. Among the influential predictive models available, the bagging ensemble has captured the interest of researchers seeking to improve prediction accuracy over single classifiers. However, prior work in this area has focused mainly on selecting single classifiers as the base classifier of the bagging ensemble, with little to no further optimization of the proposed framework. This study aimed to fill this gap by providing a bagging ensemble framework to optimize its hyperparameters and achieve improved prediction accuracy. The proposed model used the Weka BESTrees data mining tool and Math language course student dataset from UCI Machine Learning Repository. Based on the experiments performed, the proposed bagging optimization technique can effectively increase the accuracy of a traditional bagging ensemble method. It reveals further that the proposed BESTrees framework can achieve an optimized performance when trained with the appropriate hyperparameters and hill climb metrics.

Keywords—machine learning, Weka, ensemble, student prediction, bagging, optimization techniques, hyperparameters, BESTrees, decision tree

1 Introduction

Every educational institution aims to attain an exceptional record of student accomplishments and deliver the best knowledge, tools, and skills. Recognizing students who necessitate additional assistance and taking suitable actions to improve their performance is critical to reaching that goal [1]. It may include predicting and examining student performance which can support educators in finding weaknesses and improving academic records [2]. Recent research studies have focused on developing Machine Learning (ML) prediction models that can detect patterns that expressively impact students' performance, dropout, collaboration, and engagement [3].

ML algorithms train a model from past data to generate predictions or decisions without being overtly programmed. With minimal human intervention, it can detect

patterns and generate predictions built on historical data and accumulated experiences. It is generally subdivided into individual machine learning algorithms or commonly known as single classifiers (such as Naïve Bayes and Decision Trees), traditional ensemble learning methods, and combined boosting and multi-boosting ensembles [4]. However, individual classifiers are often overwhelmed by over-fitting and get stuck on a small learning rate; and these issues persist as motivating reasons for performance enhancement among researchers. As a result, ensemble learning became the learning method that can improve the performance of a collection of individual classifiers to solve similar learning tasks [5].

Ensemble learning frequently integrates numerous ML strategies into an ML framework to decrease inconsistency and systematic error while improving results. Its leading success directed several studies to explore the dominance of ensemble learning due to its capability to accomplish improved prediction performance than a single algorithm [6]. Specifically, bagging is the utmost frequently used variant of ensemble learning for performance improvement in classification tasks [7]. Bagging or bootstrap aggregation is an approach for accumulating multiple versions of an unbalanced estimator, each generated from a bootstrap sample. It causes several learner patterns to develop a combined predictor whose output is obtained by blending the results of each created subspace using majority voting. Its goal is to improve the base classifier's performance, to reduce variation, and to circumvent overfitting.

In recent years, a notable amount of research introduced new bagging variants to balance the missing properties of the prior ones. For example, in the study of Sahoo et al. [8], they proposed a roadmap to unravel common issues in the cargo shipping sector by forecasting the delay and by giving freight forwarders an advantage over their competitors. Furthermore, they bared that merging predictions attained by single classifiers into bagging ensemble improves overall accuracy while reducing error. Sumana et al. [9] proposed a Hybrid Ensemble Classifier Model (HECM) based on Bagging and Boosting to assess three UCI Repository benchmark datasets. The results verified that their suggested framework performed better than any prevailing framework. It also enhanced the performance of the classifiers involved spanning from 2% to 30.14% compared to standard ensemble models. Similarly, Saad [10] suggested a bagging ensemble framework to predict the appraisal of employees in determining their salary and incentive. The framework assisted the top management in deciding on a suitable performance that matches the correct wage. Moreover, it was utilized in identifying underachieving employees who necessitate management action. Experimental results reveal that a single classifier whose accuracy reached 94.21% improved further up to 99.16% when utilized in a bagging ensemble.

Most of these researches have focused only on choosing a base classifier and utilizing it in a bagging ensemble model to improve its predictive accuracy. On the contrary, this study proposed an optimized framework by analyzing various hyperparameters of the bagging ensemble algorithm, eventually maximizing its predictive accuracy. Furthermore, it aimed to investigate ways to optimize the bagging ensemble to achieve improved accuracy over a traditional bagging ensemble. The main objectives of the present study were (i) to optimize the bagging ensembles' hyperparameters and (ii) to utilize them in predicting students' academic performance.

The other parts of this paper cover four (4) sections. Section 2 covers the review of some related studies. Section 3 discusses the proposed methodology. Section 4 deliberates on the outcome of the experiments. Lastly, Section 5 discusses the conclusion and future plans.

2 Background and related works

Student's academic performance is vital in defining educational success at all levels. With this, several studies focused on finding factors and strategies to predict their performance [11]. To accomplish this goal with minimum human involvement, researchers employ ML prediction models to recognize students who struggle to learn and take preventive actions to aid them. As a result, it has become a critical need among educational institutions to offer such models for various objectives, such as distinguishing students on the verge of failing, improving passing rate, examining study trends, and many others [5]. Similarly, several researchers attempted to find the best classifier for predicting academic performance. Experimental results reveal that the ensemble learning model dominates in this area because it combines multiple classifiers' predictions to create the finest classification model [12].

The ultimate aim of the ensemble method is to lessen the possibility of choosing underperforming single classifier while advancing the outcome of a single algorithm by combining several individual algorithms into an ensemble model [13]. Building ensemble classifiers for high-dimensional and large-dataset problems is tremendously beneficial; finding an individual classifier in a single step is impossible due to the problem's scale and intricacy [7]. Moreover, ensembles yield improved results when the classifiers involved are diverse, which means that the ensemble classifiers should generate dissimilar errors on different subsets of data [9] [14].

Ensemble models are categorized into three types: bagging, stacking, and boosting. Bagging is concerned with making many decisions on different samples of the same dataset and calculating the average prediction; stacking is concerned with fitting various models on the same data while using another model to learn the combined predictions [15]. Similarly, boosting entails adding ensemble members successively to correct prior predictions made by other models, yielding the average of the predictions. However, comparative studies on the performance of various ensemble learning models have discovered that bagging outperforms boosting and stacking [16–17]. Bagging can significantly decrease the ML model's prediction error and variance when utilized with a base learner generation [17–18].

Bagging ensemble or bootstrap aggregation advances the precision of the base classifier by increasing the stability and by decreasing the variance of the utilized framework. It is an exceptional ensemble approach for unbalanced learning algorithms like Decision Trees and Neural Networks, where slight fluctuations in the training data set can result into big predictions [19]. It consists of two major components: aggregating the model and bootstrap sampling. Bootstrap sampling entails utilizing n samples based on a dataset through selection with replacement, ensuring independence among various testing datasets. Moreover, the model aggregation uses whichever achieved the

maximum instance as the final classification result among the outcome of multiple base learners [20].

For more than a decade, researchers seeking to improve prediction accuracy over single classifiers focused on bagging ensembles. Wang et al. [17] explored the ensemble method's effectivity on credit scoring problems using varied single classifiers trained on company credit datasets. They revealed that bagging outdoes boosting and stacking. Likewise, Aydogmus et al. [21] investigated the performance of bagging ensembles using various single classifiers and revealed that bagging outperformed their base classifiers. Furthermore, the experimental results of their predictive models demonstrated that bagging can improve the model's performance while decreasing the error rate.

Similarly, Olalekan et al. [22] investigated an ensemble bagging model with J48 and a Simple Cart model on the hypothyroid dataset. Compared to single classifiers, experimental results showed that the models achieved effective and accurate predictions. Another study [23] proposed an improved bagging ensemble selection that uses the unstable component to decrease the ensemble's variance. Their study's experimental results based on ten company datasets demonstrated that using the out-of-bag sample as the hill climb set generates a more effective ensemble model than the standard ensemble model. Equally, Ngo et al. [24] proposed evolutionary bagging ensemble learning which iteratively improves the ensemble by growing bag diversity. The outcome of testing on several benchmark datasets showed that the evolutionary bagging ensemble outperforms the ordinary bagging ensemble.

In the same way, Alan et al. [25] employed bagging ensemble classifiers for predicting birth modes such as cesarean section or normal delivery, an innovative strategy for predicting birth modes. Experimental results revealed that bagging ensembles outclass the single classifiers in this classification task. Likewise, Mosavi et al. [26] explored the predictive accuracy of various ensemble models for predicting potential groundwater zones, consisting of two boosting models and two bagging models. Their study discovered that the bagging models outperformed the boosting models.

The research results in the literature verified the superiority of bagging ensemble methods over standard individual algorithms and other ensemble counterparts. However, prior work in this area focused mainly on selecting single classifiers as the base classifier of the bagging ensemble, with little to no further optimization of the proposed framework. This study aimed to fill this gap by providing a bagging ensemble framework that would optimize its hyperparameters and ensure that the model can lessen bias and variance and improve its prediction accuracy better than a non-optimized one.

3 Materials and methods

The subsections below provide a high-level discussion of the proposed methodology utilized in this study.

3.1 The dataset

This study utilized the Math language course student dataset from the University of California Irvine's (UCI) repository, freely available in CSV format on their dataset

repository [27]. If interested, the dataset may also be retrieved here [28]. It includes 395 academic records and 32 columns from secondary schools in Portugal, containing student demographics, marks, social, and other educational information. The dataset is frequently used to unravel classification tasks in which the target class is students' final grades containing numeric values from 0 to 20.

3.2 Data pre-processing

Data pre-processing is a mining method that translates raw data into a functioning and efficient format. It defines the steps essential to convert or encode data so that any algorithm implemented can effortlessly construe the data's features. For example, in this dataset, the numeric value (1–20) of the Final Grade attribute, the target class, was converted to 'P' (>=10) and 'F' (<10) nominal values. Then, the nominal values ('Yes' or 'No') of the other attributes were transformed into binary values (0 and 1). This process is required because machine learning algorithms, at their core, tend to function well on numerical data [29].

3.3 Feature selection techniques

The role of the feature selection technique is to improve classifying data by eliminating properties that slightly impact prediction results. In addition, the decrease in properties reduces the data size; thereby, lessening the execution time [30]. This study used Chi-Squared Attribute Evaluator to identify the attributes that highly correlate with the predicted class, the Final Grade. It determined whether the correlation between two categorical variables in the sample reproduces their natural association in a dataset for categorical features (also known as nominal variables).

In addition, this study also utilized the Information Gain Attribute Evaluator, which helps measure the attribute's value by gauging the information gain about the target class. It is the consequence of the interaction of two values. The equality of the numerator and denominator values shows their independence, resulting in a 0 result. A higher information gain attribute suggests higher diversity [31].

These two feature selection techniques will rank the attributes based on their contribution to the target variable. Figures 1–2 show the top attributes chosen by Chi-Squared Attribute Evaluators and Information Gain Attribute Evaluator, respectively, using Ranker as the search method implemented using 10-fold cross-validation. Both evaluators agree that the top five (5) attributes based on merit are G2, G1, failures, GoOut, Mjob, and guardian. These five selected attributes will then be used to predict students' performance in the Portuguese course dataset.

Attribute selection output

```
=== Attribute selection 10 fold cross-validation (stratified), seed: 1 ===

average merit      average rank  attribute
257.787 +- 4.507      1   +- 0       31 G2
185.474 +- 4.856      2   +- 0       30 G1
38.001 +- 4.257       3   +- 0       14 failures
13.666 +- 3.301       4.3 +- 0.46    25 goout
 5.21  +- 1.17        6.8 +- 0.98     8 Mjob
 5.036 +- 1.568       7   +- 1.18    11 guardian
 4.378 +- 1.814       7.7 +- 1.19    10 reason
 2.158 +- 0.625       9.2 +- 0.6      9 Fjob
 1.881 +- 0.994       9.6 +- 1.28     1 sex
 7.225 +- 4.95       10.1 +- 8.25    20 higher
 1.07  +- 0.586      11.2 +- 0.98     3 address
 0.711 +- 0.248      12.2 +- 1.17     5 Pstatus
 0.718 +- 0.439      12.3 +- 0.78     4 famsize
 3.301 +- 5.045      14.8 +- 6.46     2 age
 0     +- 0          15.2 +- 1.17     7 Fedu
 0     +- 0          15.4 +- 1.69     6 Medu
 0     +- 0          16.9 +- 2.34    29 absences
```

Fig. 1. Ranking of top attributes using Chi-Squared Attribute Evaluator

Attribute selection output

```
=== Attribute selection 10 fold cross-validation (stratified), seed: 1 ===

average merit      average rank  attribute
 0.64  +- 0.013      1   +- 0       31 G2
 0.449 +- 0.014      2   +- 0       30 G1
 0.073 +- 0.008      3   +- 0       14 failures
 0.027 +- 0.007      4.3 +- 0.46    25 goout
 0.011 +- 0.003      6.7 +- 0.78     8 Mjob
 0.01  +- 0.003      7   +- 1.18    11 guardian
 0.009 +- 0.004      7.7 +- 1.19    10 reason
 0.004 +- 0.001      9.3 +- 0.46     9 Fjob
 0.004 +- 0.002      9.6 +- 1.28     1 sex
 0.014 +- 0.009     10.1 +- 8.25    20 higher
 0.002 +- 0.001     11.2 +- 0.98     3 address
 0.001 +- 0.001     12.1 +- 1.14     5 Pstatus
 0.001 +- 0.001     12.4 +- 0.8      4 famsize
 0.007 +- 0.01      14.8 +- 6.46     2 age
 0     +- 0         15.2 +- 1.17     7 Fedu
 0     +- 0         15.4 +- 1.69     6 Medu
```

Fig. 2. Ranking of top attributes using Information Gain Attribute Evaluator

As a result, Table 1 describes the dataset attributes based on the selected top five features of the evaluators and the target class.

Table 1. Dataset selected attributes

Attribute	Description	Values
G1	Grade of first period	0–20 (numeric)
G2	Grade of second period	0–20 (numeric)
Failures	Number of class failures	0–3 (numeric)
GoOut	Meet with friends and go out	1–5 (numeric)
MJob	Job of the mother	Teacher, Health, At home, Services Other (nominal)
FG	Final grade (target class)	P, F (nominal)

3.4 Model implementation

Before applying a machine learning algorithm to a dataset, one must explicitly define the hyperparameters to control the learning process. Hyperparameters are used to specify the model's learning capacity. An ML model's hyperparameters must be calibrated to fit into various problems. Consequently, selecting the best hyperparameter configuration of a model can yield supreme performance [32]. For hyperparameter optimization, the strategies commonly used are Grid Search, Random Search, and Bayesian Optimization [33–34].

This study utilized the BESTrees algorithm, a bagging ensemble selection strategy introduced in Weka. Weka is a free, open-source machine learning tool created by the New Zealand's University of Waikato. BESTrees is an ensemble learning algorithm that supports regression and classification problems and that employs the bagging ensemble selection strategy [35]. The base learner of the BESTrees algorithm is a CART-like decision tree algorithm. In addition, the user of this algorithm can select through the interface of Weka some various target evaluation metrics used to optimize any ML framework. The pseudocode of BESTrees [36] is given in Figure 3. It starts with dataset S trained on classifier E using T number of bootstrap samples. Then, it produces the model using the entire S_b bootstrap sample and chooses the hill climb set from the respective S_{oob} out-of-bag instances [37]. In addition, each E_i performs an ensemble model selection M based on base classifiers' performance on S_{oob}.

```
Procedure BESTree_Training
(input: Training set S; Ensemble classifier E; Integer T (number of bootstrap samples)
output: Trained Ensemble Classifier E)
1. for i = 1 to T
2. Sb = bootstrap sample from S // sample with replacement
3. Soob =out of bag sample
4. train base classifiers in E on Sb
5. Ei =BESTree_Selection(M , S)
6. Return E
```

Fig. 3. BESTrees training pseudocode

Figure 4 shows the flow of methodology used in this study. First, the dataset was subjected to various data pre-processing techniques to smoothen the noise and to transform it into a well-formed dataset. Then, based on the feature selection techniques implemented in the previous step, students' final grades would be predicted using the five selected attributes (G1, G2, Failures, GoOut, and MJob). In addition, these attributes would be trained using the Weka BESTrees algorithm which implements a bagging ensemble method using a decision tree as its base classifier.

As seen in Figure 5, this study would test eight (8) hill climb metrics of the BESTrees algorithm and select one that optimizes the performance of the given dataset. The hill climbing metrics is a local exhaustive search algorithm that moves toward increasing elevation/value to find the mountain's peak or the best solution to the problem [38]. It ends when it reaches a maximum value for which no neighbor has a higher value [39].

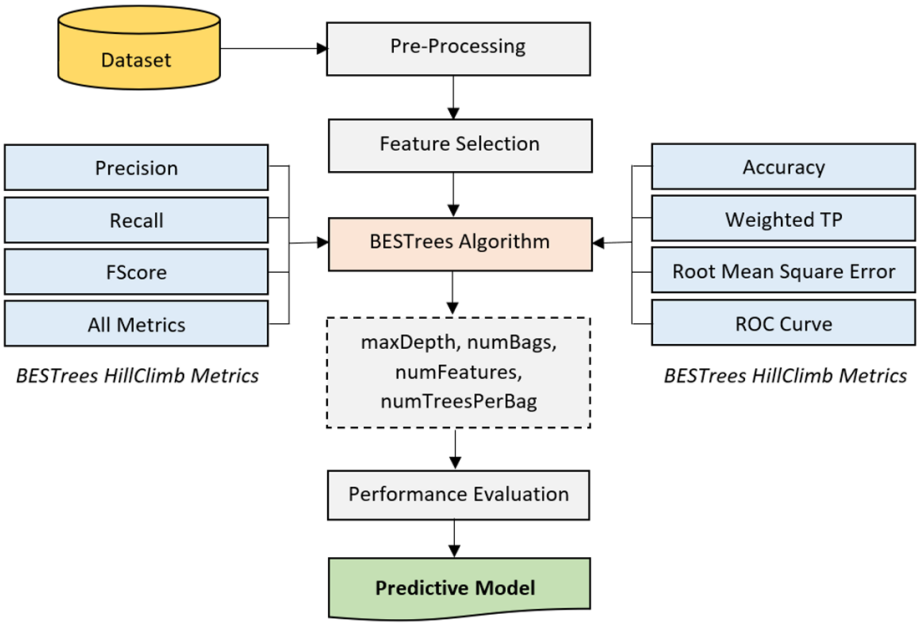

Fig. 4. The proposed framework

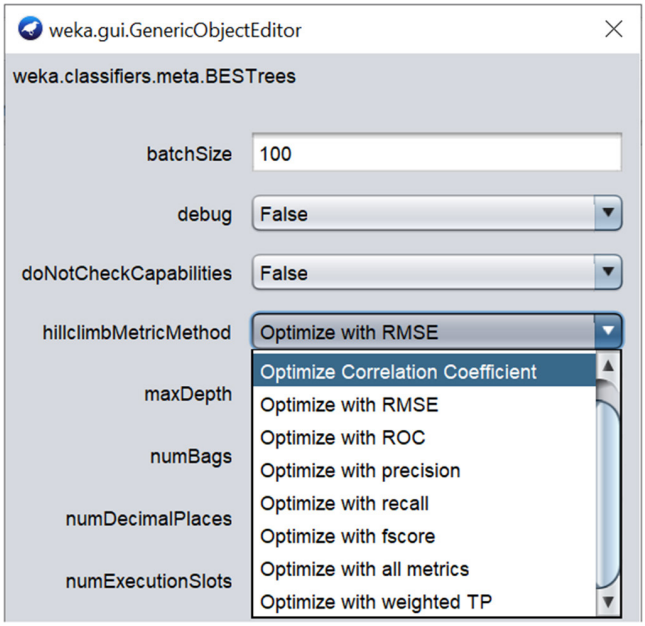

Fig. 5. BESTrees hill climb metrics

The hill climb metrics of BESTrees are precision, recall, fscore, accuracy, root mean square error, roc curve, or all metrics. Using one of these metrics alternatingly, it can perform a hill climb procedure by creating a bagging ensemble that maximizes a given performance metric on out-of-fold predictions. Then apply the same mapping to test predictions and make an ensemble-averaged/weighted set of test predictions. This set of test predictions should outperform any single model's predictions.

As presented in Figure 6, the BESTrees algorithm would train the dataset alongside hill climb metrics and various hyperparameters such as maxDepth, numBags, numFeatures, and numTreesPerBag to optimize the predictive accuracy of the proposed framework. Note that the values of the hyperparameters used in this study would be based on a manual search. Then the performance of the model would be tested using ten folds cross-validation. Eventually, the selected final predictive model is the combined hill climb metrics and hyperparameters that gain the highest outcome based on F-measure and accuracy.

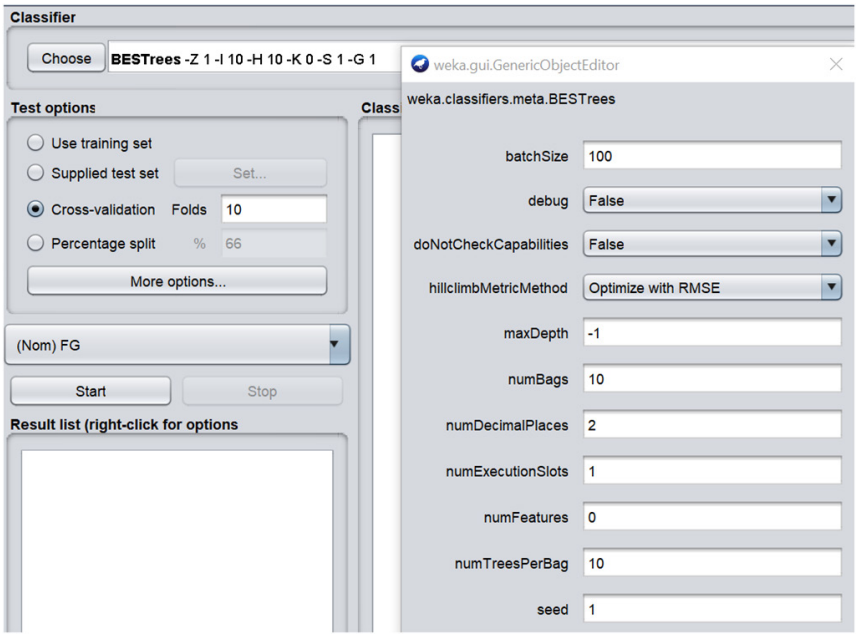

Fig. 6. BESTrees hyperparameters

4 Results and discussion

The proposed framework used the free and open-source Weka software developed by New Zealand's University of Waikato. Using Weka's BESTrees package, the model predicted students' final grades. It uses a bagging ensemble selection strategy alongside various pre-defined hill climb metrics and hyperparameters to optimize the model's performance.

4.1 Performance accuracy of BESTrees algorithm and single classifiers

To choose the appropriate model for predicting students' performance, various models tested were evaluated based on accuracy and F-measure performance metrics. Accuracy is a performance metric that calculates the percentage of correct predictions out of all predictions made [41]. Similarly, the harmonic mean of the model's precision and recall is the F-measure, as illustrated in Equations (1) and (2), respectively. If interested, view the details of the confusion matrix here [5] involving variables such as True Positive (TP), False Negative (FN), True Negative (TN), and False Positive (FP).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$F - Measure = \frac{2 \times (\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}} \quad (2)$$

Table 2 displays the results of the trained models using BESTrees and individual algorithms using ten-fold cross-validation. This study utilized single classifiers based on their dominance in the literature. As revealed in the table, the traditional bagging ensemble using RepTree (BET), commonly known as Decision Tree, gained an accuracy of 91.645% and an F-Measure of 0.918, which is way higher than the single classifiers. Furthermore, BET's accuracy increased to 91.899% when used as BESTrees containing default hyperparameters and metrics. It confirms the general observation that the bagging ensemble performs better when compared to single classifiers. In addition, it also demonstrates that the BESTrees optimization technique can be effective in increasing further the accuracy of a traditional bagging ensemble method.

Table 2. Performance accuracy of BESTrees and single classifiers

Algorithm	Accuracy	F-Measure
J48	90.633	0.907
Multilayer Perceptron (MP)	89.620	0.895
Naïve Bayes (NB)	87.089	0.870
Support Vector Machine (SVM)	90.633	0.907
K-Nearest Neighbor (KNN)	83.544	0.835
Logistic Regression (LR)	90.380	0.904
Bagging Ensemble using RepTree (BET)	91.645	0.918
BESTrees with default hyperparameters	**91.899**	**0.920**

4.2 Performance accuracy of BESTrees with hyperparameter optimization

Table 3 compares the accuracy of the BESTrees algorithm built on a consolidation of hill climb metrics and hyperparameters. The values of the hyperparameters assigned to maxDepth, numBags, numTreesPerBag, and numFeatures were based on a manual search. The experiment started with default values assigned to the hyperparameters, such as maxDepth = –1, numBags = 10, treesPerBag = 10, and numFeatures = 0, respectively. Then, in each iteration, these hyperparameters' values were gradually increased to determine the best configuration that responds well to the dataset. In addition, setting maxDepth = –1 and numberFeatures = 0 means that we allow the algorithm to determine the depth of the tree and auto-select correlated dataset attributes in training the model.

As shown in the experiments reflected in Table 3, the hill climb metrics such as precision, recall, F-Score, weighted TP, and all metrics gained the highest predictive accuracy of 91.90% executed at a total time of 0.07 seconds. However, results show that there is only a slight increase in accuracy compared to the performance of BESTrees with default hyperparameters in Table 2. We may have yet to find the best matching hyperparameters and metrics to maximize the trained model's predictive accuracy.

Table 3. Performance accuracy of BESTrees and single classifiers

Hill Climb Metric	Hyperparameters				Accuracy	F-Measure	Time
	max Depth	num Bags	Trees PerBag	num Features			
RMSE	−1	10	10	0	91.899	0.920	0.29
	2	20	20	2	91.899	0.920	0.27
	4	30	30	4	91.646	0.918	0.72
Accuracy	−1	10	10	0	91.899	0.920	0.06
	2	20	20	2	91.899	0.920	0.22
	4	30	30	4	91.899	0.920	0.71
ROC	−1	10	10	0	91.392	0.915	0.09
	2	20	20	2	91.650	0.917	0.25
	4	30	30	4	91.650	0.918	0.78
Precision	−1	10	10	0	91.900	0.920	0.07
	2	20	20	2	91.140	0.912	0.20
	4	30	30	4	91.392	0.915	0.75
Recall	−1	10	10	0	91.392	0.915	0.06
	2	20	20	2	91.140	0.912	0.23
	4	30	30	4	90.633	0.907	0.72
F-Score	−1	10	10	0	91.900	0.920	0.07
	2	20	20	2	91.900	0.920	0.21
	4	30	30	4	91.900	0.920	0.69
Weighted TP	−1	10	10	0	91.900	0.920	0.07
	2	20	20	2	91.900	0.920	0.23
	4	30	30	4	91.900	0.920	0.70
All Metrics	−1	10	10	0	91.900	0.920	0.07
	2	20	20	2	91.900	0.920	0.25
	4	30	30	4	91.650	0.918	0.76

In another experiment, as illustrated in Figure 7, the accuracy of various classifiers was compared with a decision tree (RepTree), bagging with a decision tree (BaggingRepTree) as a base classifier, and BESTrees. However, the experiment used all 32 attributes of the processed dataset. We let the BESTrees algorithm pick up the essential features of the processed dataset in training the model. Take note that the bagging ensemble is used in conjunction with random feature selection [40]. Every new subset created is derived with a replacement from the training dataset. Then, the tree grows using random feature selection based on the newly constructed training set. Each sampling with replacement contributes to forming an ensemble with reduced variance and bias.

As seen in Figure 7, BESTrees gained the highest predictive accuracy of 92.41% compared to the performance of RepTree and Bagging RepTree which yielded 91.9% and 92.15%, respectively. The improved accuracy obtained by BESTrees using auto-numFeatures hyperparameters and "Optimize with RMSE" as the hill climb metric is higher than the performance gained in the experiments performed in Tables 2 and 3. It shows that the BESTrees framework can achieve an optimized performance when trained with the appropriate hyperparameters and hill climb metrics.

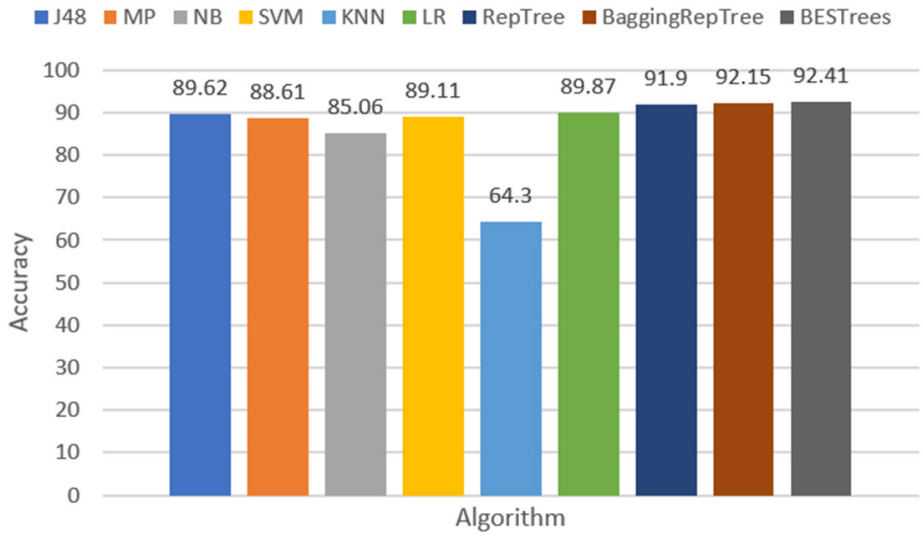

Fig. 7. Performance accuracy of BESTrees auto feature selection and single classifiers

5 Conclusion

This study proposed an optimized bagging ensemble framework by analyzing various hyperparameters and hill climb metrics, eventually maximizing its predictive accuracy compared to a traditional bagging ensemble. The primary goals of the study were (i) to optimize the bagging ensembles' hyperparameters and (ii) to utilize them in predicting students' academic performance. The proposed model utilized Math language course student dataset from the University of California Irvine's (UCI) repository.

In addition, the study used the BESTrees algorithm, a bagging ensemble selection strategy introduced in Weka. Weka is a free, open-source machine-learning tool created by the New Zealand's University of Waikato. It supports various hyperparameters and hill climb metrics that can be tweaked to optimize the performance accuracy of a bagging ensemble method. Comparing the model's accuracy reveals that the traditional bagging ensemble using RepTree (BET) is the most accurate, commonly known as Decision Tree. It gained an accuracy of 91.645% which is way higher than the single classifiers used in the study.

Furthermore, BET's accuracy increased to 91.899% when used in the BESTrees algorithm containing default hyperparameters and hill climb metrics. Likewise, the proposed framework increased further to 92.41% when it used auto-numFeatures hyperparameters and "Optimize with RMSE" as the hill climb metric. It shows that the BESTrees framework can achieve an optimized performance when trained with the appropriate hyperparameters and hill climb metrics.

Future work could involve evaluating the model using different datasets to study its predictive accuracy in a different domain. Moreover, the framework must train on more complex search algorithms, such as grids or random searches, and tweaking all the available BESTrees hyperparameters and hill climb metrics.

6 References

[1] Altabrawee, H., Ali, O., & Qaisar, S. (2019). Predicting Students' Performance Using Machine Learning Techniques. *Journal of University of Babylon for Pure and Applied Sciences*, vol. 27, pp. 194–205, 2019. https://doi.org/10.29196/jubpas.v27i1.2108

[2] Baashar, Y., Alkawsi, G., Ali, N., Alhussian, H., & Bahbouh, H. T. (2021). Predicting Student's Performance using Machine Learning Methods: A Systematic Literature Review. *2021 International Conference on Computer & Information Sciences (ICCOINS)*, pp. 357–362. https://doi.org/10.1109/ICCOINS49721.2021.9497185

[3] Evangelista, E., & Sy, B. (2022). An Approach for Improved Students' Performance Prediction Using Homogeneous and Heterogeneous Ensemble Methods. *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 5, pp. 5226–5235. https://doi.org/10.11591/ijece.v12i5

[4] Li, Y., & Chen, W. (2020). A Comparative Performance Assessment of Ensemble Learning for Credit Scoring. *Mathematics 2020*, vol. 8, no. 8, 1756. https://doi.org/10.3390/math8101756

[5] Evangelista, E. (2021). A Hybrid Machine Learning Framework for Predicting Students' Performance in Virtual Learning Environment. *International Journal of Emerging Technologies in Learning (iJET)*, vol. 16, no. 24, pp. 255–272. https://doi.org/10.3991/ijet.v16i24.26151

[6] Kausar, S., Oyelere, S., Salal, Y., Hussain, S., Cifci, M., Hilcenko, S., Iqbal, M., Wenhao, Z., & Huahu, X. (2020). *Mining Smart Learning Analytics Data Using Ensemble Classifiers. International Journal of Emerging Technologies in Learning (IJET)*, vol. 15, no. 12, pp. 81–102. https://doi.org/10.3991/ijet.v15i12.13455

[7] Tuysuzoglu, G., & Birant, D. (2020). Enhanced Bagging (eBagging): A Novel Approach for Ensemble Learning. *The International Arab Journal of Information Technology*, vol. 17, pp. 515–528. https://doi.org/10.34028/iajit%2F17%2F4%2F10

[8] Sahoo, R., Pasayat, A., Bhowmick, B., Fernandes, K., & Tiwari, M. (2022). A Hybrid Ensemble Learning-Based Prediction Model to Minimise Delay in Air Cargo Transport using Bagging and Stacking. *International Journal of Production Research*, vol. 60, no. 2, pp. 644–660. https://doi.org/10.1080/00207543.2021.2013563

[9] Sumana, B. V., & Santhanam, T. (2015). Optimizing the Prediction of Bagging and Boosting. *Indian Journal of Science and Technology*, vol. 8, pp. 1–13. https://doi.org/10.17485/IJST%2F2015%2FV8I35%2F78449

[10] Saad, H. (2018). Use Bagging Algorithm to Improve Prediction Accuracy for Evaluation of Worker Performances at a Production Company. *Industrial Engineering and Management*, vol. 7, pp. 1–7. https://doi.org/10.4172/2169-0316.1000257

[11] Bin Roslan, M. H., & Chen, C. J. (2022). Educational Data Mining for Student Performance Prediction: A Systematic Literature Review (2015–2021). *International Journal of Emerging Technologies in Learning (iJET)*, vol. 17, no. 05, pp. 147–179. https://doi.org/10.3991/ijet.v17i05.27685

[12] Arya, P., Bhagat, A., & Nair, R. (2019). Improved Performance of Machine Learning Algorithms via Ensemble Learning Methods of Sentiment Analysis. *International Journal on Emerging Technologies*, vol. 10, no. 2, pp. 110–116.

[13] Sahoo, R., Pasayat, A., Bhowmick, B., Fernandes, K., & Tiwari, M. (2021). A Hybrid Ensemble Learning-Based Prediction Model to Minimise Delay in Air Cargo Transport using Bagging and Stacking. *International Journal of Production Research*, vol. 60, pp. 1–17. https://doi.org/10.1080/00207543.2021.2013563

[14] Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. In: Multiple Classifier Systems. MCS 2000. Lecture Notes in Computer Science, vol 1857. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45014-9_1

[15] Akinbo, R. S., & Daramola, O. A. (2021). Ensemble Machine Learning Algorithms for Prediction and Classification of Medical Images. In (Ed.), *Machine Learning – Algorithms, Models and Applications*. IntechOpen. https://doi.org/10.5772/intechopen.100602

[16] Aydogmus, H. Y., Erdal, H. I., Karakurt, O., Namli, E., Turkan, Y. S., & Erdal, H. (2015). A Comparative Assessment of Bagging Ensemble Models for Modeling Concrete Slump Flow. *Computers and Concrete*, vol. 16, no. 5, pp. 741–757. https://doi.org/10.12989/cac.2015.16.5.741

[17] Wang, G., Hao, J., Mab. J., & Jiang, H. (2011). A Comparative Assessment of Ensemble Learning for Credit Scoring. *Exp. Syst. Appl*., vol. 38, pp. 223–230. https://doi.org/10.1016/j.eswa.2010.06.048

[18] Karakurt, O., Erdal, H., Namlı, E., Yumurtaci-Aydogmus, H., & Turkkan, Y. (2013). Comparing Ensembles of Decision Trees and Neural Networks for One-day-ahead Stream Flow Predict. *Science Park*, vol. 1, pp. 1–12. https://doi.org/10.9780/23218045/1172013/41

[19] Pandey, M., & Taruna, S. (2014). A Comparative Study of Ensemble Methods for Students' Performance Modeling. *International Journal of Computer Applications*, vol. 103, pp. 26–32. https://doi.org/10.5120/18095-9151

[20] Li, Y., & Chen, W. (2020). A Comparative Performance Assessment of Ensemble Learning for Credit Scoring. *Mathematics*, vol. 8, no. 10, pp. 1756. https://doi.org/10.3390/math8101756

[21] Aydogmus, H., Erdal, H., Karakurt, O., Namli, E., Turkan, Y., & Erdal, H. (2015). A Comparative Assessment of Bagging Ensemble Models for Modeling Concrete Slump Flow. *Computers and Concrete*, vol. 16, no. 5, pp. 741–757. https://doi.org/10.12989/cac.2015.16.5.741

[22] Olalekan, J., Ogwueleka, F. N., & Odion, P. O. (2020). Effective and Accurate Bootstrap Aggregating (Bagging) Ensemble Algorithm Model for Prediction and Classification of Hypothyroid Disease. *International Journal of Computer Applications*, vol. 176, pp. 40–48. https://doi.org/10.5120/ijca2020920542

[23] Sun, Q., & Pfahringer, B. (2011). Bagging Ensemble Selection. *Australasian Conference on Artificial Intelligence*, vol. 7106, pp. 251–260. https://doi.org/10.1007/978-3-642-25832-9_26

[24] Ngo, G., Beard, R., & Chandra, R. (2022). Evolutionary Bagging for Ensemble Learning. *Neurocomputing*, vol. 510, pp. 1–14. https://doi.org/10.1016/j.neucom.2022.08.055

[25] Alam, M., Patwary, M., & Hassan, M. (2021). Birth Mode Prediction Using Bagging Ensemble Classifier: A Case Study of Bangladesh. *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, pp. 95–99. https://doi.org/10.1109/ICICT4SD50815.2021.9396909

[26] Mosavi, A., Hosseini, F. S., Choubin, B., Goodarzi, M., Dineva, A. A., & Sardooi, E. R. (2020). Ensemble Boosting and Bagging Based Machine Learning Models for Groundwater Potential Prediction. *Water Resources Management*, vol. 35, no. 1, pp. 23–37. https://doi.org/10.1007/s11269-020-02704-3

[27] Cortez, P., & Silva, A. (2008). Using Data Mining to Predict Secondary School Student Performance. *In A. Brito and J. Teixeira Eds., Proceedings of 5th Future Business Technology Conference (FUBUTEC 2008)*, pp. 5–12, ISBN 978-9077381-39-7.

[28] UCI Machine Learning Repository. (2014). Student Performance Dataset. Retrieved November 22, 2022, from https://archive.ics.uci.edu/ml/datasets/student%2Bperformance

[29] Brink, H., Richards, J. W., & Fetherolf, M. (2017). Real-World Machine Learning. Manning Publications.

[30] Ouatik, F., Erritali, M., Ouatik, F., & Jourhmane, M. (2022). Predicting Student Success Using Big Data and Machine Learning Algorithms. *International Journal of Emerging Technologies in Learning (iJET)*, vol. 17, no. 12, pp. 236–251. https://doi.org/10.3991/ijet.v17i12.30259

[31] Naheed, N., Shaheen, M., Khan, S., Alawairdhi, M., & Khan, M. (2020). Importance of Features Selection, Attributes Selection, Challenges and Future Directions for Medical Imaging Data. *Computer Modeling in Engineering and Sciences*, vol. 125. https://doi.org/10.32604/cmes.2020.011380

[32] Yang, L., & Shami, A. (2020). On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice. *arXiv*. abs/2007.15745. https://doi.org/10.1016/j.neucom.2020.07.061

[33] Tso, W. W., Burnak, B., & Pistikopoulos, E. N. (2020). HY-POP: Hyperparameter Optimization of Machine Learning Models through Parametric Programming. *Comput. Chem. Eng.*, vol. 139, 106902. https://doi.org/10.1016/j.compchemeng.2020.106902

[34] Badriyah, T., Santoso, D., & Syarif, I. (2019). Deep Learning Algorithm for Data Classification with Hyperparameter Optimization Method. *Journal of Physics: Conference Series*, vol. 1193, 012033. https://doi.org/10.1088/1742-6596/1193/1/012033

[35] Sun, Q., & Pfahringer, B. (2012). Bagging ensemble selection for regression. *In Proceedings of the 25th Australasian joint conference on Advances in Artificial Intelligence (AI'12)*, Springer-Verlag, Berlin, Heidelberg, pp. 695–706. https://doi.org/10.1007/978-3-642-35101-3_59

[36] Alam, M., Pathak, N., & Roy, N. (2015). Mobeacon: An iBeacon-Assisted Smartphone-Based Real Time Activity Recognition Framework. *EAI Endorsed Transactions on Future Internet*, vol. 2. https://doi.org/10.4108/eai.22-7-2015.2260073

[37] Sun, Q., & Pfahringer, B. (2011). Bagging Ensemble Selection. In: Wang, D., Reynolds, M. (eds) AI 2011: Advances in Artificial Intelligence. AI 2011. *Lecture Notes in Computer Science*, vol. 7106. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-25832-9_26

[38] Janssen, F., & Fürnkranz, J. (2008). An Empirical Comparison of Hill Climbing and Exhaustive Search in Inductive Rule Learning.

[39] Ghahramani, A., Karvigh, S. A., & Becerik-Gerber, B. (2017). HVAC System Energy Optimization using an Adaptive Hybrid Metaheuristic. Energy and Buildings, vol. 152, pp. 149–161. https://doi.org/10.1016/j.enbuild.2017.07.053

[40] Breiman, L. (2001). Random Forests. *Machine Learning*, vol. 45, pp. 5–32. https://doi.org/10.1023/A:1010933404324

[41] Chen, Y., Zhang, X., Lu, L., Wang, Y., Liu, J., Qin, L., Ye, L., Zhu, J., Shia, B.-C., & Chen, M.-C. (2022). Machine Learning Methods to Identify Predictors of Psychological Distress. *Processes*, vol. 10, pp. 1030. https://doi.org/10.3390/pr10051030

7 Author

Edmund Evangelista is an Assistant Professor at the College of Technological Innovation at Zayed University, Abu Dhabi Campus, United Arab Emirates. His research interests include machine learning, data mining, and software engineering. He has over twenty years (20) of experience working in the IT Industry and IT Academia. He has also held positions as Team Leader, Software Engineer, and Web/Moodle Developer within the IT industry (Oman, Kuwait, and the Philippines).

Article submitted 2023-01-15. Resubmitted 2023-03-04. Final acceptance 2023-03-04. Final version published as submitted by the authors.