Jurnal Teknik Informatika (JUTIF) Vol. 7, No. 3, June 2026, Page. 3012-3024
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5627

Optimization of Machine Learning Model using Grid and Random Search Algorithms for Predicting Student Dropout

M. Faris Al Hakim*[1], Siti Wahyuni[2], Kholiq Budiman[3], Aditya Marianti[4], Bambang Eko Susilo[5], Nuni Widiarti[6], Sri Sukaesih[7], Rifaatunnisa[8]

[1,3]Department of Computer Science, Universitas Negeri Semarang, Indonesia
[2]Department of Physics, Universitas Negeri Semarang, Indonesia
[4]Department of Biology, Universitas Negeri Semarang, Indonesia
[5]Department of Mathematics, Universitas Negeri Semarang, Indonesia
[6]Department of Chemistry, Universitas Negeri Semarang, Indonesia
[7,8]Department of Integrated Science, Universitas Negeri Semarang, Indonesia

Email: [1]farisalhakim@mail.unnes.ac.id

Received : Jan 12, 2026; Revised : Mar 9, 2026; Accepted : Mar 18, 2026; Published : Jun 15, 2026

Abstract

Student dropout is a serious problem that can affect the quality of education and operational efficiency of higher education institutions. Early prediction of potential students who will dropout is essential to develop appropriate intervention strategies, so as to increase graduation rates and reduce the negative impact on academic continuity. A better model for student dropout prediction becomes an objective of this research. The method used in this research is to improve the performance of machine learning models through the selection of optimal hyperparameters. The research methodology consists of several stages, including data preprocessing, handling imbalanced data, model training, and performance evaluation. There are three machine learning models used in this research, namely XGBoost, AdaBoost, and Random Forest. The selection of optimal hyperparameter values is carried out using the Random Search and Grid Search methods. Model evaluation is conducted using k-fold cross-validation and multiple evaluation metrics, including accuracy, precision, recall, and F1-score. As part of the important results, the combination of XGBoost and Random Search produced the best performance with 91.18% accuracy, indicating that hyperparameter optimization significantly improves predictive performance. The findings of this research explicitly contribute to the field of informatics, particularly educational data mining, and provide insights for educational institutions to identify high-risk dropout students more accurately.

Keywords : *Ada Boost, Grid Search, Machine Learning, Student Dropout, Student Performance, XGBoost*

1. INTRODUCTION

Academic success is one of the keys and dropout has become a major concern in the field of higher education [1]. Predicting academic success and dropout can be the first step for administrators to map the condition of students. Early identification of students at risk of dropping out can help educational institutions to implement timely and effective interventions to improve student retention [2]-[4]. Through this identification, the treatment of student conditions can be more focused on the group that needs it most. In addition, proper handling of student conditions, especially those with a high potential to drop out, can help to reduce dropout rates. In addition, understanding the factors that influence dropout and academic success allows for the development of better policies to support student achievement [2], [5], [6]. In the long run, improving academic success through proper prediction can contribute to higher education quality and increased graduation rates [7] Therefore, research supporting the development of predictive models in academic contexts is a strategic step needed to support student success and the sustainability of higher education institutions [7]-[9].

Various studies have been carried out regarding drop out and academic success from various perspectives, including predictions using machine learning. Research done by [2] aimed to predict programming students at risk of dropping out using various predictive models within the educational data mining process. It evaluated models like k-nearest neighbors (kNN), decision tree, logistic regression, and neural network (NN). The result showed that the proposed models could generate accuracies of approximately 90%. Another research proposed by [10] explored three machine learning predictive models to early detect students at risk. The proposed predictive models were KNN, SVM, and (NN) and revealed that SVM outperformed the others with 86.7% of accuracy. The next research published by [11] leveraged machine learning in the issue of student dropout in higher education. The research used nine different machine learning models and produced FSC4RBF as the best predictor with 88.42% of accuracy.

Various studies have successfully integrated machine learning algorithms like Naive Bayes, Logistic Regression, Support Vector Machine, Decision Tree, Random Forest, K-Nearest Neighbor, LGBM, and Artificial Neural Networks to predict student success and dropouts with high accuracy [11]-[17]. Although most machine learning algorithms have been successfully applied, many previous studies focus primarily on algorithm performance without systematically analyzing the impact of hyperparameter optimization strategies across multiple ensemble classifiers using a consistent evaluation scheme. Several researches were explored various methods to improve the results. Research done by [18]-[21] was utilized SMOTE to address the imbalanced nature of the dataset. This technique helped to create a more balanced dataset by oversampling the minority class, which in this case is the dropout instances. Intelligent system for predicting early drop out also implemented using machine learning by [22]. The proposed model applied hybrid feature selection and dimensionality reduction techniques to extract relevant features from raw student log activity records.

Genetic algorithm was also applied as a strategy to improve early drop-out prediction [10], [23]. The study successfully demonstrated the effectiveness of Genetic Algorithms (GA) to enhance model performance. Other strategy used by [11], [22] to optimize hyperparameters of machine learning models. They combined GridSearchCV and the ensemble method to predict students' risk of dropping out of learning courses. Ensemble stacking became a method selected by [24] to increase model performance to predict student's drop-out. The researchers stacked 4 machine learning models, those were Random Forest, XGBoost, Gradient Boosting, and Feed-forward Neural Networks. Ref [25] proposed a new strategy to improve student performance prediction by utilizing adaptive feature selection. The method was combined with 5 machine learning models. Hyperparameter tuning was utilized by [3] to optimize the performance of logistic regression, random forest, and gradient boosting. The model was by systematically searching for the best parameters. Furthermore, limited studies explicitly compare Grid Search and Random Search as hyperparameter optimization techniques on multiple ensemble-based classifiers under the same evaluation framework, making it difficult to assess their relative effectiveness and robustness in student dropout prediction tasks.

Machine learning technique that is optimized by using other algorithms also had huge potential for predicting student dropout [24], [25]. This research aims to systematically analyze and compare the effectiveness of Grid Search and Random Search in optimizing hyperparameters of ensemble-based machine learning models for student dropout prediction. The study employed several machine learning algorithms optimized by RandomSearch and GridSearchCV to obtain the right hyperparameters for predicting student status into three classes. The proposed models were evaluated using accuracy, precision, recall, and f1-score. The main contributions of this study include a systematic comparative

Vol. 7, No. 3, June 2026, Page. 3012-3024
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5627

analysis of Grid Search and Random Search for hyperparameter optimization on multiple ensemble classifiers, as well as an evaluation of model robustness using consistent performance metrics.

2. METHOD

The proposed method begins with data acquisition, followed by data preprocessing. After the model selection stage, the selected model is optimized using Grid Search and Random Search. Finally, the model performance is evaluated using accuracy, precision, recall, and F1-score. The details of each stage are presented in the next section. Figure 1 visually illustrates the proposed model.

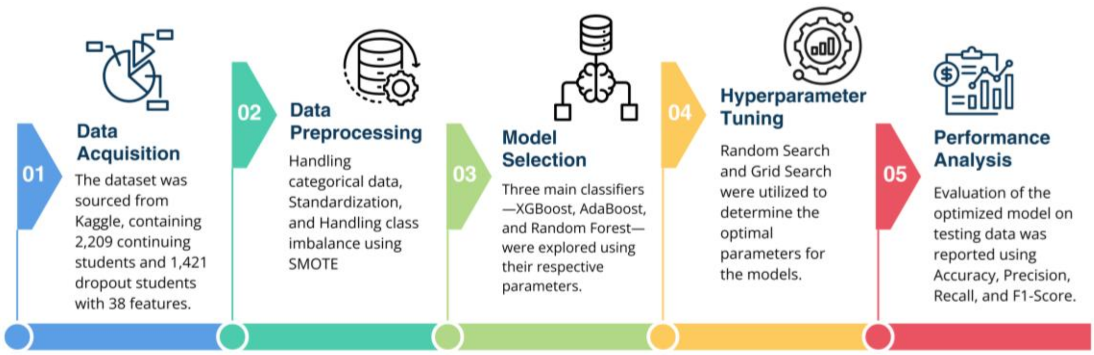

Figure 2. Performance of XGB-RandomSearchCV in 5-fold cross-validation

2.1. Data Description

This study explored 3360 dataset sourced from Kaggle platform in 2021. The dataset consists of 3630 rows, out of which 2209 are labelled as continuing students and 1421 as dropout students. There are 38 features in this dataset, including demographic features like gender, marital status, nationality, academic qualification, academic performance, and graduate, enrolled, or dropout as the target feature. In this study, only success and dropout targets were utilized. The dataset was also used by [26] for a similar task.

2.2. Data Preprocessing

The data preprocessing stage in this research involves a series of careful steps aimed at making the dataset feasible to increase its suitability for building a machine learning model.

2.2.1. Handling Categorical Data

A key component of data preprocessing was handling categorical data. Categorical variables were converted using Label Encoders to ensure smooth integration of the dataset with machine learning algorithms [27]. This conversion transformed categorical data into a numerical format, enabling its efficient use in predictive models. The Target feature in this study was one of the features that carries out label encoding.

2.2.2. Standardization

Standardization is a normalization technique widely used in data preprocessing. It is implemented to align all features to a consistent scale, reducing the effects of varying measurement units. The method standardizes features by removing the mean and scaling to unit variance, transforming the data into a standard normal distribution. This process is essential for algorithms that are sensitive to feature scaling, ensuring each feature contributes equally to the model's learning. By standardizing the data, it enhances the model's convergence and stability during the training phase of machine learning algorithms. In this research, StandardScaler was used.

Vol. 7, No. 3, June 2026, Page. 3012-3024
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5627

2.2.3. Handling Class Imbalance

Oversampling methods aim to generate new data by replicating important samples from minority classes [28]. One of the most popular Oversampling methods is SMOTE which was proposed by [29]. In general, the main idea of SMOTE is to create new minority data by interpolating multiple minority examples [30]. The process involves finding the nearest neighbor using the K-Nearest Neighbor (KNN) algorithm and creating new data along the line between the original sample and its neighbor. The feature values of those neighbors are multiplied by a random number between 0 and 1 to generate new samples using Euclidean distance.

2.3. XGBoost

The XGBoost algorithm proposed by [31] is currently the fastest and best integrated Decision Tree algorithm. To achieve better results, this approach incorporates various learning models [32]. A set of trees is built based on features and summed sequentially to predict the sample class. Each tree tries to recover the difference between the target and the prediction predicted by the previous set of trees as shown in the Equation (1) below [33].

$$p_i = \sum_{i=1}^{m} f_m(x_i), f_m \in F \tag{1}$$

Where p_i is the prediction result, x_i is the vector for input variables, m is the number of decision trees, f_m is the function for the function space *F*, and *F* itself is the space for all decision trees.

XGBoost is more efficient in handling large-scale data sets and complex models, while performing well in preventing overfitting and improving generalization ability. XGBoost is popular in data regression and classification due to its advantages in terms of computational efficiency, feature importance analysis, and handling of missed values [31].

2.4. AdaBoost

The AdaBoost method is built on combining several weak learners to form a strong learner [34]. The essence of AdaBoost is to adjust the weights of the training samples, where the weight of misclassified samples increases and the weight of correctly classified samples decreases in each iteration. This makes the model focus on samples that are difficult to classify. AdaBoost then combines the results of multiple base classifiers with voting weights determined based on each model's weighted error ratio [35].

AdaBoost has a simple construction, high classification accuracy, and easy-to-understand results. Unlike bagging and RF algorithms, AdaBoost fully considers the weight of each classifier. AdaBoost is particularly unaffected by overfitting [36]. AdaBoost is also effective in overcoming the problems of base classifier creation and merging, which are two important components of ensemble learning algorithms [37].

2.5. Random Forest

Random Forest [38], [39] is an ensemble classifier model formed from the combination of many decision trees. The equation that states it is shown in Equation (2) below.

$$\{DT(x, \theta k)\}Tk \tag{2}$$

The value of x is the input vector and θk denotes an independently drawn random vector with the same distribution value as the previous $\theta k, \ldots, \theta k - 1$. Bootstrap samples are taken as many as *T* from the training data and the pruned CART is taken from each bootstrap β sample where only one of the *M* features is randomly selected for splitting each CART node [40].

Vol. 7, No. 3, June 2026, Page. 3012-3024
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5627

Random Forest calculates results using the majority voting method. Random Forest also has an advantage over the Decision Tree model in dealing with overfitting problems [41]. This is because Random Forest usually uses different parts of the same training dataset in each tree and helps in averaging the results of that many Decision Trees. Consequently, this helps avoid overfitting and improves the overall performance of the model.

2.6. Random Search

Random Search proposed by [38] is a method designed to identify optimal hyperparameter combinations [42]. This method shows superior efficiency as it explores the hyperparameter space in more depth, resulting in highly accurate model performance, both empirically and theoretically with fewer computational resources [43].

Random search uses a random generator to make parameter choices. In this approach ranges for different hyperparameters are specified, then values are randomly selected within these ranges, the resulting architecture is trained and tested with the best one being taken [44]. Compared to local approximations made through gradient-based optimization, this technique allows to explore more important aspects of the search space [45].

2.7. Grid Search

Grid Search is a systematic way to search for a hyperparameter in the search space [46]. The way Grid Search works is by making all possible combinations regardless of the influence of elements in the optimization process. Each parameter has an equal chance to influence the process [47].

The principle of the grid search algorithm starts from the Cartesian product applied to the set of values of each hyperparameter to obtain the hyperparameter configuration space, which contains all possible hyperparameter combinations. Next, the grid search algorithm trains a model for each hyperparameter combination in the configuration space [48]. Since the model combinations for each parameter are stored, grid search tends to be computationally expensive. However, this algorithm will provide the best combination of parameters [49].

3. RESULT AND ANALYSIS

The experimental scheme was conducted on 3630 data and had 38 features with 2 classes using k-fold cross-validation testing with values 5 and 10. The XGBoost, Random Forest, and AdaBoost algorithms were used as base models and then optimized using Random Search and Grid Search to produce better results. Model performance in the first experiment is presented in Table 1.

Table 1. Model performance after training in 5-fold cross-validation

Model	Training Accuracy	Testing Accuracy	Precision	Recall	F1-Score
XGBoost	91.45	89.67	89.73	89.67	89.56
AdaBoost	90.00	88.84	88.80	88.84	88.79
Random Forest	84.35	82.92	83.19	82.92	83.00
XGBoost+Random Search	91.74	90.91	91.10	90.91	90.79
AdaBoost+Random Search	90.35	89.53	89.51	89.53	89.47
Random Forest+Random Search	90.89	90.22	90.32	90.22	90.11
XGBoost+Grid Search	92.02	90.77	90.85	90.77	90.68
AdaBoost+Grid Search	90.89	89.81	89.81	89.81	89.73
Random Forest+Grid Search	91.23	90.77	90.98	90.77	90.64

Vol. 7, No. 3, June 2026, Page. 3012-3024
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5627

Based on experiments done, Table 1 shows the result displayed as a comparison of model performance to predict the case using 5-fold cross validation. Overall it can clearly be seen that XGBoost algorithms dominated the highest score of every evaluation metric. Random Search as the optimization algorithm also showed effective working by improving classifiers and so did Grid Search. While XGBoost and its combination did the best performance, AdaBoost and Random Forest algorithms also yielded scores above 82.90 for all evaluation metrics.

As shown in Table 1, the improvement after hyperparameter optimization indicates that default configurations are not sufficient to capture the full predictive capability of ensemble-based models. The consistent increase in testing accuracy across all optimized models suggests that systematic hyperparameter tuning plays a crucial role in enhancing model generalization rather than merely improving training performance.

In terms of the three basic models, XGBoost generated the highest accuracy both in the training and testing phases. The score produced by XGBoost in the training and testing were 91.45% and 89.67% accuracy values, respectively. AdaBoost became the second and Random Forest was the last with accuracy scores in the training phase of 88.84% and 82.92%, respectively. In the precision metric, XGBoost also showed the best performance by resulting precision of 89.73%. XGBoost also generated score just above 89% in recall and f1-score, while AdaBoost got just below 89% and Random Forest bore approximately 82.95%. The performance of optimized XGB using RandomSearchCV run in 5-fold cross-validation is shown in Figure 2.

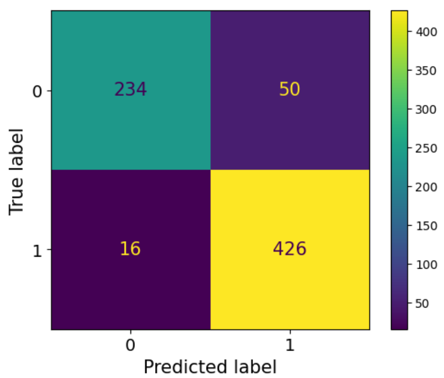

Figure 2. Performance of XGB-RandomSearchCV in 5-fold cross-validation

The superior performance of XGBoost can be attributed to its gradient boosting mechanism, which sequentially minimizes residual errors and allows fine-grained control over important parameters such as learning rate and maximum tree depth. These characteristics make XGBoost more responsive to hyperparameter optimization compared to AdaBoost and Random Forest, which rely on adaptive reweighting and bagging strategies, respectively. As a result, XGBoost is able to better capture complex patterns in the dataset after optimization.

The discussion can be made in several sub-chapters. Regarding the combination models, the best combination in the training stage was XGBoost and Grid Search and the best combination in the testing phase was XGBoost and Random Search with accuracy of 92.02% and 90.91%, respectively. In optimization mode, Random Forest performed better than AdaBoost both in the training and testing stages. In the testing stage, Random Forest contributed an accuracy of 90.22% optimized with Random Search and an accuracy of 90.77% optimized with Grid Search. Accuracy scores provided by AdaBoost in the testing are 89.53% when combined with Random Search and 89.81% when combined with Grid Search. In the precision score, Random Search succeeded in optimizing Random Forest by resulting at 90.73% as the highest.

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5627

Table 2. Model performance after training in 10-fold cross-validation

Model	Training Accuracy	Testing Accuracy	Precision	Recall	F1-Score
XGBoost	91.28	90.50	90.69	90.50	90.36
AdaBoost	90.46	88.15	88.11	88.15	88.09
Random Forest	85.51	84.57	84.47	84.87	84.66
XGBoost+Random Search	91.97	91.18	91.26	91.18	91.10
AdaBoost+Random Search	90.83	89.94	89.95	89.94	89.87
Random Forest+Random Search	90.81	90.63	90.73	90.63	90.53
XGBoost+Grid Search	92.19	90.36	90.44	90.36	90.26
AdaBoost+Grid Search	90.35	89.67	89.67	89.67	89.59
Random Forest+Grid Search	90.86	90.08	90.22	90.08	89.96

Table 2 presents the second experiment containing model comparisons after training using 10-fold cross-validation. The results showed that XGBoost kept its performance stable and became the best in almost all metrics used. The optimization techniques used worked effectively as the model performance could be improved especially for Random Forest, which increased significantly. Besides that, the lowest score provided by the combined models was 89.59% in the precision metric. Figure 3 shows the performance of optimized XGB using RandomSearchCV run in 10-fold cross-validation.

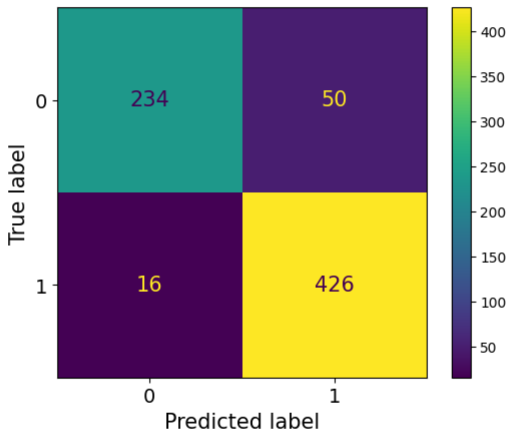

Figure 3. Performance of XGB-RandomSearchCV in 10-fold cross-validation

Among the three standard models, XGBoost achieved the highest accuracy in both the training and testing phases. The conditions that occurred in AdaBoost and Random Forest tended to be the same as in previous experiments. The difference value between AdaBoost and Random Forest in the testing stage was around 4%. In recall, precision, and f1-score metrics, the condition was almost similar with an order started by XGBoost, AdaBoost, and Random Forest.

Concerning the optimized models, the combination of XGBoost and Random Search became the best model generating an accuracy of 91.18% in the testing phase. In the training phase, the combination of XGBoost and Grid Search provided the highest score in this study, namely 92.19%. AdaBoost algorithms became the third in resulting accuracy score with the highest score at 89.94% in the testing stage. Moreover, the combination of XGBoost and Random Search also yielded the highest score in precision, recall, and f1-score metrics. The combination produced 91.26% in precision, 91.18% in recall, and 91.10% in f1-score. The Random Forest combinations experienced a significant rise to around 90.5% in the precision, to around 90.30% in the recall, and to around 90.10% in the f1-score. The AdaBoost combinations got almost similar conditions. The combinations witnessed grow slightly to about 89% in the other metrics. The comparison between 5-fold and 10-fold cross-validation demonstrates consistent performance patterns across different validation schemes. The relatively small difference in accuracy values indicates that the proposed models are stable and not highly sensitive to the choice of fold configuration.

Hyperparameter tuning performed using Random Search and Grid Search significantly improved the performance of all models. The Random Forest algorithm, which initially had a testing accuracy of 82.92% on 5-fold cross-validation and 84.57% on 10-fold cross-validation, increased drastically to around 90% after tuning with Random Search and Grid Search. Likewise, the AdaBoost algorithm, which had an accuracy of around 88%, then increased to around 89.5% after tuning. In addition, XGBoost provided the best performance compared to other models in terms of accuracy and other evaluation metrics, both before and after hyperparameter tuning. Without tuning, XGBoost has achieved a fairly high testing accuracy of 90.50%, and after tuning there was an increase or remained stable. Furthermore, Grid Search produced a more stable performance than Random Search. Although in some cases, testing accuracy with Random Search was higher (such as in 5-fold cross-validation), Grid Search generally provided a better balance between training and testing accuracy. This shows that Grid Search is better at avoiding overfitting than Random Search. By performing a systematic search on a predetermined parameter space, Grid Search is more likely to find the best combination [55, 56].

4. DISCUSSIONS

Referring to optimization algorithms, Random Search performed better, as in the first experiment. The condition may be caused by Random Search being more efficient in exploring a wider space by randomly selecting more varied combinations. Moreover, Random search can also serendipitously find better hyperparameter combinations over a wider range, while Grid Search can get stuck in suboptimal combinations because it only explores a small subset of a larger hyperparameter space. This finding indicates that Random Search is more suitable for optimizing complex ensemble-based models such as XGBoost, where interactions between hyperparameters are non-linear and difficult to exhaustively enumerate.

When compared with other results, the proposed method also contributes more to predicting student dropout. Several researches were done for predicting student dropout with various models and giving specific contributions. However, most prior works primarily focus on reporting performance metrics, particularly accuracy, without critically discussing the influence of hyperparameter optimization strategies and evaluation consistency across models. The detailed comparison of model performances is presented in Table 3.

Table 3. Result Comparison of Model Performances

Model	Accuracy
Logistic Regression [52]	75.63%
Gradient Boosting [3]	90.00%
FSC4RBF [11]	90.90%
Random Forest [53]	77.65%
Random Search [54]	80.56%
Proposed Method	91.18%

The results show that the proposed method achieves the highest accuracy of 91.18%, outperforming both traditional machine learning approaches and several optimized models reported in previous studies. This improvement highlights the significance of systematic hyperparameter optimization in enhancing model robustness and predictive reliability. Furthermore, these findings

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5627

suggest that the integration of ensemble learning with efficient optimization techniques can provide practical benefits for developing early warning systems to support academic decision-making in higher education institutions. From a broader perspective, this study contributes to the fields of machine learning optimization and educational data mining by offering empirical evidence on the effectiveness of Random Search compared to Grid Search in student dropout prediction tasks.

5. CONCLUSION

The optimized models of machine learning were proposed in this research for accurate prediction of student dropout. The model consists of XGB, AdaBoost, and Random Forest, and is optimized through Random Search and Grid Search techniques. Various evaluation metrics were used to validate the proposed model. Random Search and Grid Search have good performance in finding the best hyperparameter combination in the machine learning model. Objectively, XGBoost optimized using Random Search reaches the best accuracy of 91.18%, followed by precision of 91.26%, recall of 91.18%, and f1-score of 91.10%. This study provides a methodological contribution by systematically comparing the effectiveness of Grid Search and Random Search across multiple ensemble classifiers, demonstrating that Random Search offers better optimization efficiency and model robustness in student dropout prediction.

The proposed method holds substantial potential for early prediction of student dropout, which contributes to enhanced policy development in academics. The drawback of this study is that the dataset used may not fully cover all the important factors that influence dropout and may introduce potential bias due to reliance on a single dataset. Future research can be expanded by exploring other methods both classifier and optimization algorithms. Further research may also integrate explainable artificial intelligence techniques to improve model interpretability, apply deep learning approaches for broader comparison, and utilize datasets from multiple institutions to enhance model generalization.

REFERENCES

[1] D. Baneres, M. E. Rodriguez-Gonzalez, and A. E. Guerrero-Roldan, “A Real-Time Predictive Model for Identifying Course Dropout in Online Higher Education,” *IEEE Transactions on Learning Technologies*, vol. 16, no. 4, pp. 484–499, Aug. 2023, doi: 10.1109/TLT.2023.3267275.

[2] N. F. B. Casillano and K. W. Cantilang, “Employing educational data mining techniques to predict programming students at-risk of dropping out,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 35, no. 2, pp. 1219–1226, Aug. 2024, doi: 10.11591/ijeecs.v35.i2.pp1219-1226.

[3] Z. Alj and M. O. C. Malki, “Predicting Students Drop-out in MOOC from Learning Behavior Using Machine Learning,” *Journal of Uncertain Systems*, vol. 16, no. 3, Sep. 2023, doi: 10.1142/S1752890922500118.

[4] C. G. De Jesus and M. K. C. Ledda, “Intervention Support Program for Students at Risk of Dropping out Using Fuzzy Logic-Based Prescriptive Analytics,” in *Proceeding - 2021 IEEE 17th International Colloquium on Signal Processing and Its Applications, CSPA 2021*, 2021. doi: 10.1109/CSPA52141.2021.9377304.

[5] Nurmalitasari, Z. Awang Long, and M. Faizuddin Mohd Noor, “Factors Influencing Dropout Students in Higher Education,” *Educ Res Int*, vol. 2023, 2023, doi: 10.1155/2023/7704142.

[6] F. F. Ananna, R. Nowreen, S. S. R. Al Jahwari, E. A. Costa, L. Angeline, and S. R. Sindiramutty, “Analysing Influential Factors in Student Academic Achievement: Prediction Modelling and Insight,” *International Journal of Emerging Multidisciplinaries: Computer Science & Artificial Intelligence*, vol. 2, no. 1, 2023, doi: 10.54938/ijemdcsai.2023.02.1.254.

Vol. 7, No. 3, June 2026, Page. 3012-3024
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5627

[7] A. T. Rahman and A. Raihan, "Prediction of Student Performance in Higher Academy Through Different Data Mining Techniques," in *4th International Conference on Electrical, Computer and Telecommunication Engineering, ICECTE 2022*, 2022. doi: 10.1109/ICECTE57896.2022.10114531.

[8] E. E. Estandarte, D. P. Agbon, A. C. Azares, M. C. Elape, J. C. M. Bustillo, and J. R. Delos Arcos, "Prediction-based techniques on Academic Performance of Graduating Students through Machine Learning," in *7th International Conference on Inventive Computation Technologies, ICICT 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 281–287. doi: 10.1109/ICICT60155.2024.10544915.

[9] S. Tosun and D. Bakan Kalaycıoğlu, "Data mining approach for prediction of academic success in open and distance education," *Journal of Educational Technology and Online Learning*, vol. 7, no. 2, pp. 168–176, May 2024, doi: 10.31681/jetol.1334687.

[10] A. Wahdan, S. Hantoobi, M. Al-Emran, and K. Shaalan, "Early Detecting Students at Risk Using Machine Learning Predictive Models," in *Lecture Notes in Networks and Systems*, 2022. doi: 10.1007/978-3-030-85990-9_27.

[11] V. Christou *et al.*, "Performance and early drop prediction for higher education students using machine learning," *Expert Syst Appl*, vol. 225, Sep. 2023, doi: 10.1016/j.eswa.2023.120079.

[12] S. A. Alwarthan, N. Aslam, and I. U. Khan, "Predicting Student Academic Performance at Higher Education Using Data Mining: A Systematic Review," 2022, *Hindawi Limited*. doi: 10.1155/2022/8924028.

[13] D. Andrade-Girón *et al.*, "Predicting Student Dropout based on Machine Learning and Deep Learning: A Systematic Review," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 10, no. 5, 2023, doi: 10.4108/eetsis.3586.

[14] I. Issah, O. Appiah, P. Appiahene, and F. Inusah, "A systematic review of the literature on machine learning application of determining the attributes influencing academic performance," 2023. doi: 10.1016/j.dajour.2023.100204.

[15] L. Kemper, G. Vorhoff, and B. U. Wigger, "Predicting student dropout: A machine learning approach," *European Journal of Higher Education*, vol. 10, no. 1, 2020, doi: 10.1080/21568235.2020.1718520.

[16] J. Kabathova and M. Drlik, "Towards predicting student's dropout in university courses using different machine learning techniques," *Applied Sciences (Switzerland)*, vol. 11, no. 7, Apr. 2021, doi: 10.3390/app11073130.

[17] N. Mduma, K. Kalegele, and D. Machuve, "A survey of machine learning approaches and techniques for student dropout prediction," 2019. doi: 10.5334/dsj-2019-014.

[18] S. S. Rezk and K. S. Selim, "Comparing nine machine learning classifiers for school-dropouts using a revised performance measure," *J Comput Soc Sci*, Oct. 2024, doi: 10.1007/s42001-024-00281-8.

[19] E. S. Karabacak and Y. Yaslan, "Comparison of Machine Learning Methods for Early Detection of Student Dropouts," in *UBMK 2023 - Proceedings: 8th International Conference on Computer Science and Engineering*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 376–381. doi: 10.1109/UBMK59864.2023.10286747.

[20] S. W. Masood, M. Gogoi, and S. A. Begum, "Optimised SMOTE-based Imbalanced Learning for Student Dropout Prediction," *Arab J Sci Eng*, 2024, doi: 10.1007/s13369-024-09287-w.

[21] A. López-García, O. Blasco-Blasco, M. Liern-García, and S. E. Parada-Rico, "Early detection of students' failure using Machine Learning techniques," *Operations Research Perspectives*, vol. 11, 2023, doi: 10.1016/j.orp.2023.100292.

[22] Z. Alj, A. Bouayad, and O. Cherkaoui Malki Mohammed, "Intelligent System for Personalised Interventions and Early Drop-out Prediction in MOOCs," 2022. [Online]. Available: www.ijacsa.thesai.org

[23] W. Sabbir, Abdullah-Al-Kafi, A. S. M Afridi, M. S. Rahman, and M. Karmakar, "Improving Predictive Analytics for Student Dropout: A Comprehensive Analysis and Model Evaluation," *11th International Conference on Computing for Sustainable Global Development (INDIACom)*, Feb. 2024.

[24] J. Niyogisubizo, L. Liao, E. Nziyumva, E. Murwanashyaka, and P. C. Nshimyumukiza, "Predicting student's dropout in university classes using two-layer ensemble machine learning approach: A novel stacked generalization," *Computers and Education: Artificial Intelligence*, vol. 3, Jan. 2022, doi: 10.1016/j.caeai.2022.100066.

[25] K. Roy and D. M. Farid, "An Adaptive Feature Selection Algorithm for Student Performance Prediction," *IEEE Access*, vol. 12, pp. 75577–75598, 2024, doi: 10.1109/ACCESS.2024.3406252.

[26] P. Berka and L. Marek, "Bachelor's degree student dropouts: Who tend to stay and who tend to leave?," *Studies in Educational Evaluation*, vol. 70, 2021, doi: 10.1016/j.stueduc.2021.100999.

[27] Alamsyah, B. Prasetiyo, M. F. Al Hakim, and F. D. Pradana, "The improvement of COVID-19 prediction accuracy using optimal parameters in reccurent neural network model," in *AIP Conference Proceedings*, Semarang, 2023, p. 040019. doi: 10.1063/5.0125767.

[28] J. Li, Q. Zhu, Q. Wu, and Z. Fan, "A novel oversampling technique for class-imbalanced learning based on SMOTE and natural neighbors," *Inf Sci (N Y)*, vol. 565, no. 1, pp. 438–455, 2021, doi: 10.1016/j.ins.2021.03.041.

[29] B. Prasetiyo, Alamsyah, M. A. Muslim, and N. Baroroh, "Evaluation performance recall and F2 score of credit card fraud detection unbalanced dataset using SMOTE oversampling technique," *J Phys Conf Ser*, vol. 1918, no. 4, p. 42002, Jun. 2021, doi: 10.1088/1742-6596/1918/4/042002.

[30] G. A. Pradipta, R. Wardoyo, A. Musdholifah, I. N. H. Sanjaya, and M. Ismail, "SMOTE for Handling Imbalanced Data Problem : A Review," in *2021 6th International Conference on Informatics and Computing, ICIC 2021*, 2021. doi: 10.1109/ICIC54025.2021.9632912.

[31] Q. Wang *et al.*, "XGBoost algorithm assisted multi-component quantitative analysis with Raman spectroscopy," *Spectrochim Acta A Mol Biomol Spectrosc*, vol. 323, p. 124917, Dec. 2024, doi: 10.1016/j.saa.2024.124917.

[32] M. Ma *et al.*, "XGBoost-based method for flash flood risk assessment," *J Hydrol (Amst)*, vol. 598, p. 126382, 2021, doi: 10.1016/j.jhydrol.2021.126382.

[33] J. Wu, Y. Li, and Y. Ma, "Comparison of XGBoost and the Neural Network model on the class-balanced datasets," in *2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer, ICFTIC 2021*, 2021, pp. 457–461. doi: 10.1109/ICFTIC54370.2021.9647373.

[34] C. Tan, F. Yin, and T. Jiang, "Classification method of seabed sonar image substrate based on ELM-AdaBoost," *J Radiat Res Appl Sci*, p. 101126, Sep. 2024, doi: 10.1016/j.jrras.2024.101126.

[35] J. Bian, J. Wang, and Q. Yece, "A novel study on power consumption of an HVAC system using CatBoost and AdaBoost algorithms combined with the metaheuristic algorithms," *Energy*, vol. 302, p. 131841, Dec. 2024, doi: 10.1016/j.energy.2024.131841.

[36] C. Rao, M. Li, T. Huang, and F. Li, "Stroke Risk Assessment Decision-Making Using a Machine Learning Model: Logistic-AdaBoost," *CMES - Computer Modeling in Engineering and Sciences*, vol. 139, no. 1, pp. 699–724, 2023, doi: 10.32604/cmes.2023.044898.

[37] L. Hao and G. Huang, "An improved AdaBoost algorithm for identification of lung cancer based on electronic nose," *Heliyon*, vol. 9, no. 3, pp. 1–14, 2023, doi: 10.1016/j.heliyon.2023.e13633.

[38] M. F. Al Hakim, S. Subhan, and P. Listiaji, "SOCIAL MEDIA COMMENTS FOR GOVERNMENT INSTITUTION VIDEO CLASSIFICATION USING MACHINE LEARNING," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 10, no. 2, pp. 433–440, Nov. 2024, doi: 10.33480/jitk.v10i2.5187.

[39] M. F. Al Hakim, M. N. Tiadah, and S. Subhan, "Diabetes Diagnosis Using Machine Learning Model with Feature Selection," in *2025 International Conference on Smart Computing, IoT and*

Machine Learning (SIML), Surakarta: IEEE Access, Jun. 2025, pp. 1–6. doi: 10.1109/SIML65326.2025.11080810.

[40] J. Magidi, L. Nhamo, S. Mpandeli, and T. Mabhaudhi, "Application of the random forest classifier to map irrigated areas using google earth engine," *Remote Sens (Basel)*, vol. 13, no. 5, p. 876, 2021, doi: 10.3390/rs13050876.

[41] G. Garg and R. Garg, "Brain Tumor Detection and Classification based on Hybrid Ensemble Classifier," pp. 1–18, 2021, doi: 10.48550/arXiv.2101.00216.

[42] X. Chai, S. Li, and F. Liang, "A novel battery SOC estimation method based on random search optimized LSTM neural network," *Energy*, vol. 306, p. 132583, Dec. 2024, doi: 10.1016/j.energy.2024.132583.

[43] F. Hosseini, C. Prieto, and C. Álvarez, "Hyperparameter optimization of regional hydrological LSTMs by random search: A case study from Basque Country, Spain," *J Hydrol (Amst)*, vol. 643, p. 132003, Nov. 2024, doi: 10.1016/j.jhydrol.2024.132003.

[44] S. A. Kawa and M. Arifwani, "Randomized search on a grid of CNN networks with simplified search space," in *Proceedings of the 2021 8th International Conference on Computing for Sustainable Global Development, INDIACom 2021*, 2021, pp. 68–73. doi: 10.1109/INDIACom51348.2021.00014.

[45] F. S. Gharehchopogh, M. Namazi, L. Ebrahimi, and B. Abdollahzadeh, "Advances in Sparrow Search Algorithm: A Comprehensive Survey," *Archives of Computational Methods in Engineering*, vol. 30, no. 1, pp. 427–455, 2023, doi: 10.1007/s11831-022-09804-w.

[46] D. Marinov and D. Karapetyan, "Hyperparameter optimisation with early termination of poor performers," in *2019 11th Computer Science and Electronic Engineering Conference, CEEC 2019 - Proceedings*, 2019, pp. 160–163. doi: 10.1109/CEEC47804.2019.8974317.

[47] H. Alibrahim and S. A. Ludwig, "Hyperparameter Optimization: Comparing Genetic Algorithm against Grid Search and Bayesian Optimization," in *2021 IEEE Congress on Evolutionary Computation, CEC 2021 - Proceedings*, 2021, pp. 1551–1559. doi: 10.1109/CEC45853.2021.9504761.

[48] N. Zhu, C. Zhu, L. Zhou, Y. Zhu, and X. Zhang, "Optimization of the Random Forest Hyperparameters for Power Industrial Control Systems Intrusion Detection Using an Improved Grid Search Algorithm," *Applied Sciences (Switzerland)*, vol. 12, no. 20, p. 10456, 2022, doi: 10.3390/app122010456.

[49] I. Priyadarshini and C. Cotton, "A novel LSTM–CNN–grid search-based deep neural network for sentiment analysis," *Journal of Supercomputing*, vol. 77, no. 12, pp. 13911–13932, 2021, doi: 10.1007/s11227-021-03838-w.

[50] S. Gowriswari and S. Brindha, "Hyperparameters Optimization using Gridsearch Cross Validation Method for machine learning models in Predicting Diabetes Mellitus Risk," in *2022 International Conference on Communication, Computing and Internet of Things, IC3IoT 2022 - Proceedings*, 2022, pp. 1–4. doi: 10.1109/IC3IOT53935.2022.9768005.

[51] D. A. Anggoro and S. S. Mukti, "Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 6, pp. 198–207, 2021, doi: 10.22266/ijies2021.1231.19.

[52] M. K. Sharma and M. Yadav, "Predicting Students' Drop-Out Rate Using Machine Learning Models: A Comparative Study," in *Proceedings of the 2022 3rd International Conference on Intelligent Computing, Instrumentation and Control Technologies: Computational Intelligence for Smart Systems, ICICICT 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 1166–1171. doi: 10.1109/ICICICT54557.2022.9917841.

[53] S. Mohammad, I. A. Chowdhury, N. Roy, Md. N. Hasan, and D. Nandi, "Investigation of Student Dropout Problem by Using Data Mining Technique," *International Journal of Education and Management Engineering*, vol. 13, no. 5, pp. 43–61, Oct. 2023, doi: 10.5815/ijeme.2023.05.04.

Vol. 7, No. 3, June 2026, Page. 3012-3024
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5627

[54] P. Padmasiri and S. Kasthuriarachchi, “Interpretable Prediction of Student Dropout Using Explainable AI Models,” in *Proceedings - International Research Conference on Smart Computing and Systems Engineering, SCSE 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/SCSE61872.2024.10550525.