Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

A novel approach to mitigate academic underachievement in higher education: Feature selection, classifier performance, and interpretability in predicting student performance

Safira Begum *, M. V. Ashok

Department of Computer Applications, HKBKDC, Bangalore, India

ARTICLE INFO

Article history:
Received 24 August 2023
Received in revised form
10 February 2024
Accepted 1 May 2024

Keywords:
Academic underachievement
Machine learning
Slime mould algorithm
Feature selection
Predictive performance

ABSTRACT

The main goal of this study is to address the ongoing problem of low academic performance in higher education by using machine learning techniques. We use a dataset from a higher education institution that includes various information available at student enrollment, such as academic history, demographics, and socio-economic factors. To address this issue, we introduce a new method that combines the Slime Mould Algorithm (SMA) for efficient feature selection with a Forest-Optimized Neural Network (FO-NN) Classifier. Our method aims to identify students at risk of academic failure early. Using the SMA, we simplify the feature selection process, identifying important attributes for accurate predictions. The Forest Optimization technique improves the classification process by optimizing the neural network model. The experimental results of this study show that our proposed method is effective, with significant improvements in feature selection accuracy and notable enhancements in the predictive performance of the neural network classifier. By selecting a subset of relevant features, our approach deals with high-dimensional datasets and greatly improves the quality and interpretability of predictive models. The innovative combination of the SMA and the FO-NN classifier increases accuracy, interpretability, and the ability to generalize in predicting student performance. This work contributes to a more effective strategy for reducing academic underachievement in higher education.

1. Introduction

Predicting students' learning styles and academic performance is a significant challenge for higher education institutions. The availability of large-scale data provides an opportunity to use machine learning for accurate predictions and personalized student support (Romero and Ventura, 2020). By analyzing academic records, demographics, socio-economic factors, and student engagement data, machine learning models can identify patterns that influence student success and the risk of academic failure (Lampropoulos, 2023).

Machine learning, which includes classification algorithms, regression models, and clustering methods, helps create predictive models. These models can predict student needs and improve how resources are distributed (Batool et al., 2023; Andrade et al., 2021; Khan and Ghosh, 2021; Mangina and Psyrra, 2021; Tsiakmaki et al., 2020). Early warning signs of academic difficulties or dropouts can also be identified through historical data analysis, allowing proactive interventions (Hall et al., 2021; Xiao et al., 2022). Collaboration among educators, administrators, and data scientists is crucial for interpreting results and implementing interventions effectively. This research aims to address the following questions:

- Can machine learning classification models effectively predict the challenges students may face in completing their degrees at the Polytechnic Institute of Portalegre (IPP) in Portugal?
- How does combining the Slime Mould Algorithm (SMA) for feature selection with a Forest-Optimized Neural Network (FO-NN) Classifier affect the accuracy, interpretability, and generalization capability of student performance prediction models?

* Corresponding Author.
Email Address: safirabgm@gmail.com (S. Begum)

https://doi.org/10.21833/ijaas.2024.05.015
Corresponding author's ORCID profile:
https://orcid.org/0000-0003-2883-9994
2313-626X/© 2024 The Authors. Published by IASE.

Initially, we employed commonly used algorithms for student performance classification, including Logistic Regression (LR), Support Vector Machines (SVM), Decision Trees (DT), and Random Forests (RF) (Hamoud, 2016).

Our key contribution lies in proposing an innovative approach that combines the SMA for feature selection with an FO-NN classifier for student performance prediction. The SMA efficiently explores the feature space, identifying informative attributes, while Forest Optimization enhances classification by optimizing the neural network model, improving accuracy, interpretability, and generalization (Li et al., 2020; Orujpour et al., 2020). This novel approach addresses the challenges of high-dimensional datasets and complex relationships within student performance data.

By conducting experiments on a 36-attribute dataset, we aim to provide compelling evidence of the methodology's efficacy in addressing the stated research questions. It advances feature selection accuracy and classification performance compared to existing methods, offering educational institutions a dependable tool for data-informed decisions and interventions. Our approach navigates the complexities of student performance analysis, providing clarity and confidence in decision-making.

2. Literature review

Student performance analysis is a vital area of research that aims to understand and predict factors influencing student outcomes in educational settings. This literature review provides an overview of key studies and trends in student performance analysis, highlighting methodologies, influential factors, and the impact of interventions.

The research conducted by the authors of Phan et al. (2023) centers around the analysis of an extensive dataset sourced from a Canadian university, encompassing 38,842 students. The primary objective of the study is to utilize RF as a machine-learning model for predicting academic success. The accuracy achieved for program completion prediction is 79% overall. Notably, the accuracy for students who successfully completed their program reaches 91%, indicating a relatively high success rate in program completion prediction. However, a limitation of this approach is that it may not account for all individual variations and factors influencing student success, potentially leading to misclassifications in specific cases.

In a separate study, the authors of Smadi et al. (2023) aimed to identify profiles of freshmen who are likely to encounter significant challenges in completing their first academic year. They work with a dataset comprising information from 6,845 students and employ conventional classification methods, including RF, LR, and Artificial Neural Networks (ANN). While this approach offers valuable insights, it may not capture the complexities of students' academic journeys and may overlook some unique factors contributing to their challenges.

Another study conducted by Miguéis et al. (2018) focused on predicting overall academic performance based on available data at the end of students' first year of academic pursuit. The prediction models developed in this research employ a dataset comprising 2,459 students from a European Engineering School. While this approach provides early predictions, it may not consider the evolving dynamics of student performance over time and the potential for academic improvement beyond the first year.

In a comprehensive review (Pojon, 2017) of educational data mining for student performance prediction, various techniques are discussed along with their applications. These techniques incorporate a wide range of factors, including demographic and social aspects, as well as academic measures, such as assessments from first-year courses. Machine learning algorithms explored in the review include SVM, Naïve Bayes, DT, RF, Bagging DT, and Adaptive Boosting DT. The results indicate that RF and Adaptive Boosting DT demonstrate superior performance among the tested algorithms, achieving an overall accuracy of 96%. Nevertheless, it's essential to recognize that the choice of algorithm may not universally apply to all educational settings, and some situations may require a more tailored approach.

The authors of Mutrofin et al. (2019) conducted a study focusing on predicting dropout within a dataset comprising 21,654 students. The research investigates various class balancing strategies and conventional classification methods to enhance prediction accuracy. Class balancing becomes crucial when dealing with imbalanced datasets, where one class (in this case, dropout) is significantly underrepresented compared to the other class (non-dropout). The authors of Sha et al. (2022) compared different class balancing techniques, such as random under-sampling, random oversampling, and synthetic oversampling. Among the methods tested, the synthetic minority oversampling technique (SMOTE) (Alex et al., 2022) yields the most favorable outcomes. However, it's important to note that class balancing techniques can introduce certain biases and may not entirely eliminate the misclassification of minority class instances.

Our proposed research work addresses several of the limitations observed in the methods discussed in the literature review. While previous studies, such as those of Phan et al. (2023) and Smadi et al. (2023), achieved commendable accuracy in predicting student outcomes, they may not fully account for individual variations and unique factors influencing student success. In contrast, our innovative approach, which combines the SMA for feature selection with a FO-NN classifier, enhances feature selection accuracy, improving the precision of our predictive models. Moreover, earlier research, as exemplified by Miguéis et al. (2018), often focused on predicting student performance based on initial-year data, potentially overlooking the evolving dynamics of student success over time. In contrast, our approach employs a comprehensive dataset of 36 attributes to provide a holistic view of student performance, allowing for more accurate predictions and tailored interventions. Additionally, the class balancing strategies discussed by Mutrofin et al. (2019) and the various methods explored in Pojon (2017) demonstrated notable results but may introduce biases and limitations in handling imbalanced datasets. Our proposed approach effectively tackles the challenges associated with high-dimensional datasets and minimizes biases, offering improved accuracy, interpretability, and generalization capabilities for student performance prediction. Thus, our research work aims to rectify these limitations and provide educational institutions with a dependable tool for data-informed decisions and interventions, ultimately contributing to a more effective strategy for mitigating academic underachievement in higher education.

3. Materials and method

Fig. 1 shows a block diagram for the proposed student performance prediction. The proposed approach in this paper makes several contributions to the field of student performance prediction:

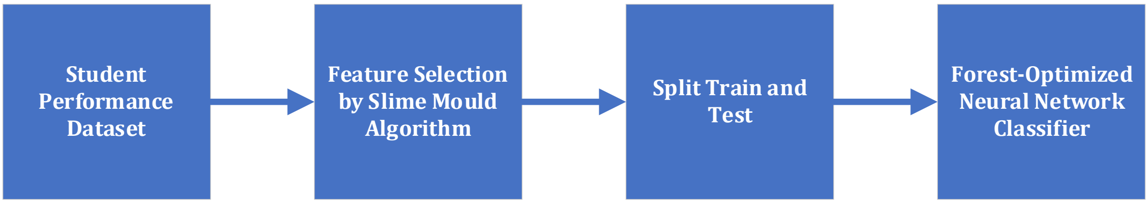

Student Performance Dataset
Feature Selection by Slime Mould Algorithm
Split Train and Test
Forest-Optimized Neural Network Classifier

Fig. 1: Proposed method for student performance classification

- Integration of the SMA for Feature Selection: The paper introduces the use of the SMA for feature selection in student performance prediction. The SMA is a nature-inspired optimization algorithm that efficiently explores the feature space to identify the most relevant attributes. By incorporating SMA, the proposed approach addresses the challenge of high-dimensional datasets, enabling the selection of a subset of features that have the most significant impact on student performance. This improves the accuracy and interpretability of the predictive model.
- FO-NN classifier: The paper combines the SMA with a FO-NN classifier for student performance prediction. This integration leverages the strengths of neural networks in capturing complex relationships within student performance data while also optimizing the classification model using Forest Optimization techniques. By optimizing the neural network model, the proposed approach enhances the predictive performance of the classifier and improves the accuracy of student performance prediction.
- Improved Accuracy and Interpretability: By utilizing the SMA for feature selection and optimizing the neural network classifier, the proposed approach aims to improve the accuracy of student performance prediction. Selecting the most relevant features enhances the model's ability to capture the key factors influencing student outcomes. Additionally, the FO-NN classifier ensures that the model is optimized to make accurate predictions. The combination of these techniques leads to a more accurate and reliable prediction of student performance. Furthermore, the feature selection process improves interpretability by identifying the most important factors contributing to student outcomes.
- Potential for Targeted Interventions: Accurate student performance prediction can facilitate targeted interventions to support students' academic progress. By identifying the key factors influencing student outcomes, educational institutions can develop tailored interventions and strategies to improve student performance. The proposed approach provides insights into the important features affecting student performance, enabling the design of effective interventions that address specific areas of improvement.

The contribution of this paper lies in proposing an innovative approach that combines the SMA for feature selection and the FO-NN classifier for student performance prediction. This integration improves the accuracy and interpretability of the predictive model, enabling educational institutions to identify factors influencing student outcomes and implement targeted interventions for academic success.

3.1. Dataset

The dataset used in this research paper is sourced from the institutional records of undergraduate students who enrolled at the Polytechnic Institute of Portalegre in Portugal (Martins et al., 2021). Data collection methods involved the aggregation of diverse databases spanning academic years from 2008/09 to 2018/2019. These databases encompassed students from various undergraduate disciplines, including agronomy, design, education, nursing, journalism, management, social service, and technologies. The integrity of the data was maintained through careful data preprocessing, which aimed to correct errors, address unexplained outliers, and manage missing values. During this phase, records that could not be accurately classified were carefully removed from the analysis. The criteria for removing records were based on the most recent 3 or 4 academic years, depending on the length of the respective courses. As a result of this thorough preprocessing, the final dataset consisted of 3,623 records with 25 independent variables.

The dataset encompasses a broad range of variables, incorporating demographic aspects such as age at enrollment, gender, marital status, nationality, address code, and special needs. It also considers socio-economic factors, including student-worker status, parental qualifications, parental professions, parental employment status, student grants, and student debt. Moreover, variables related to the students' academic trajectory, such as admission grade, retention years in high school, preference order for the chosen course, and the type of course pursued during high school, are included. It's important to note that the academic information is limited to observable factors preceding registration, excluding internal evaluations conducted subsequent to enrollment.

Each entry within the dataset is categorized into one of three groups: Success, Relative Success, or Failure, based on the duration taken by the student to attain their degree. Success denotes that the student accomplished their degree within the anticipated timeframe, while Relative Success indicates a delay of no more than three additional years. Conversely, Failure encompasses instances where students required more than three extra years to complete their degree or were unable to obtain it altogether. These classifications effectively represent three levels of risk: low-risk (Success), medium-risk (Relative Success), and high-risk (Failure). It's important to acknowledge that, despite the rigorous data preprocessing, the dataset may still carry certain biases and limitations inherent to institutional records and the data collection process.

3.2. Feature selection using SMA

The SMA serves as a nature-inspired optimization algorithm suitable for feature selection in student prediction datasets. SMA emulates the behavior of slime molds, which are single-celled organisms renowned for their aptitude to identify the shortest path between food sources. Here's a step-by-step guide on how to apply the SMA for feature selection in a dataset with 36 attributes: By employing the SMA on a student prediction dataset, a subset of features with substantial influence on student performance can be identified, thereby enhancing the accuracy and interpretability of predictive models. Hyperparameter Settings: The SMA lacks widely accepted standard hyperparameters, in contrast to machine learning algorithms with established parameter norms. This algorithm typically involves several essential parameters that can be adjusted according to a specific problem and dataset. These parameters encompass:

- Number of Slime Mould Individuals (Agents): This parameter defines the quantity of individual slime mould entities within the population. It should be determined based on the dataset's size, with the potential need for experimentation to identify an optimal value.
- Iteration Count: The number of iterations or generations the algorithm undergoes should be specified. A higher iteration count may result in a more comprehensive search but could potentially extend computational time.
- Chemotactic Step Size: This parameter governs the distance an individual can cover in a single step during the chemotactic phase. Smaller step sizes may yield more precise yet slower convergence.
- Evaporation Rate: Certain versions of SMA integrate an evaporation rate, which simulates the dissipation of pheromones or chemical signals left by the slime mould. The evaporation rate dictates the speed at which these signals dissipate.
- Stop Criteria: A stopping criterion must be established for the algorithm. This criterion could entail a maximum number of iterations, a designated fitness level, or other conditions based on the specific problem.
- Objective Function: The choice of a fitness function (objective function) should be aligned with the problem at hand. While the example herein employs accuracy as the objective function, alternative metrics may be chosen to suit the unique requirements of the application.

To devise a mathematical fitness function for the SMA within the context of student prediction utilizing a dataset containing 36 attributes, it is necessary to establish a quantitative measure that represents the performance of the predictive model based on the selected features. Here's a suggestion for a mathematical fitness function formula:

Assuming a binary classification task (e.g., predicting student success or failure) and selecting accuracy as the evaluation metric, the fitness function can be defined as the accuracy of the predictive model trained on the chosen subset of features. Let:

X_{train}: Training dataset containing the selected features
y_{train}: True labels for the training dataset
X_{val}: Validation dataset containing the selected features
y_{val}: True labels for the validation dataset
$model$: The chosen predictive model algorithm

The fitness function formula for accuracy can be written as:

$$Fitness\ function = Accuracy(X_{train}, y_{train}, X_{val}, y_{val}, model) \quad (1)$$

In this formula, Accuracy represents a function that calculates the accuracy of the predictive model. LR, DT, and SVM are among the diverse machine learning algorithms that can serve as the "model" parameter in the equation. The Accuracy function should be implemented to calculate the accuracy based on the true labels (y_{train}, y_{val}) and the predicted labels obtained from the model trained on the selected features (X_{train}, X_{val}). The specific implementation will depend on the programming language or framework being used.

By utilizing this fitness function formula, the SMA can optimize the selection of features by maximizing the accuracy of the predictive model. The fitness function can be customized by selecting different evaluation metrics or incorporating additional performance measures based on the specific requirements of the student prediction task.

```
Pseudo Code: function calculateFitness (features):
# Split the dataset into training and validation sets
$X_{train}$, $X_{val}$, $y_{train}$, $y_{val}$ = splitDataset(features)
# Train a predictive model using the selected features
model = trainModel($X_{train}$, $y_{train}$)
# Make predictions on the validation set
$y_{pred}$ = model.predict($X_{val}$)
# Calculate the accuracy of the model
accuracy = calculateAccuracy($y_{val}$, $y_{pred}$)
# Return the fitness value
return accuracy
```

This pseudocode provides a high-level representation of the SMA for feature selection. It starts by initializing a population of slime mould agents and runs a specified number of iterations. During each iteration, agents perform the chemotactic movement, evaluate their fitness, and update their positions based on the fitness value. Pheromone evaporation is also simulated, and the best features are selected based on the final agent positions.

3.3. Classification using FO-NN

FO-NN is a specific approach that combines the concepts of forest optimization and neural networks. It is a hybrid algorithm that leverages the strengths of both techniques to optimize the structure and parameters of a neural network. Forest optimization, as mentioned earlier, is a metaheuristic algorithm inspired by the behavior of a forest ecosystem. It utilizes the principles of collaboration and competition among different solution sets, referred to as "forests," to explore and exploit the search space. Conversely, neural networks represent a computational framework that draws inspiration from the intricate workings of the human brain. These networks are composed of interconnected nodes, referred to as neurons, which are structured in layers. Neural networks find extensive application in a range of machine-learning endeavors, encompassing tasks such as classification, regression, and pattern recognition.

In the context of a FO-NN, the algorithm applies the principles of forest optimization to optimize the architecture and parameters of the neural network. It may involve exploring different neural network topologies (e.g., number of layers, number of neurons per layer) and optimizing the weights and biases associated with the connections between neurons.

The algorithm typically starts by initializing multiple candidate solutions (neural network architectures) within different forests. These forests then undergo iterative processes of collaboration and competition, where successful architectures are preserved, shared, and improved upon. The algorithm gradually refines the architectures and optimizes the neural network parameters based on the evaluation of their performance on a specific task or problem.

The specific implementation details of FO-NNs can vary depending on the researchers or practitioners working on the approach. It is essential to consider the specific objectives, constraints, and problem domains when applying this hybrid algorithm to ensure its effectiveness and efficiency.

Algorithm:

```
Initialize the population of neural network architectures within different forests
Repeat until a stopping criterion is met:
Assess the efficacy of every neural network within the population
Select the best-performing neural networks from each forest
Share and exchange knowledge among the forests (cultural transmission)
Perform collaboration and competition within each forest:
For each forest:
Perform crossover and mutation operations to generate new neural network architectures
Evaluate the performance of the new architectures
Select the top-performing architectures to substitute the inferior ones
End for
End Repeat
```

The FO-NN is a hybrid model that integrates the adaptability of neural networks with the optimization capabilities of forest optimization algorithms. Its architecture comprises several key components:

1. Candidate architectures (Candidates): FO-NN maintains a "forest" of candidate neural network architectures. Each candidate represents a unique configuration of neural network layers, neurons, and connection weights. Mathematically, a candidate's architecture can be represented as C_i, where i denotes the specific candidate.
2. Neural network layers: A candidate architecture includes layers, typically comprising input, hidden, and output layers. The architecture optimizes the number of layers and the number of neurons in each layer. Mathematically, the layers can be represented as $L = [L_1, L_2, \ldots, L_n]$, where L_i represents the i^{th} layer.
3. Neuron configuration: Within each layer, the candidate architecture specifies the number of neurons and their activation functions. The configuration of neurons in a layer is represented

as $N = [N_1, N_2, \ldots, N_m]$, where N_j represents the j^{th} neuron in that layer.

4. Connection weights: Connection weights between neurons in the neural network are optimized. These connection weights determine the strength of connections between neurons in different layers. Mathematically, the connection weights can be represented as $W = [W_1, W_2, \ldots, W_p]$, where W_k represents the k^{th} connection weight.
5. Fitness function: Each candidate architecture is evaluated using a fitness function that measures its performance on the given classification task. The fitness function is typically based on a classification metric such as accuracy, precision, recall, or F1-score. Mathematically, the fitness function can be represented as $F(C_i)$, where C_i is the candidate architecture.
6. Integration with feature selection: The integration of FO-NN with feature selection is a two-step process:

- **Step 1:** Feature selection using the SMA: The SMA is applied to the dataset to select the most relevant features. SMA optimizes feature selection by considering the subset of features that maximizes the performance of the predictive model. Mathematically, this process can be represented as:

Let X represent the dataset with all features and $X_{selected}$ represent the dataset with the selected features. The feature selection process can be defined as:

$$X_{selected} = SMA(X, fitness\ function) \tag{2}$$

Here, SMASMA is the SMA, and the fitness function measures the performance of the predictive model.

- **Step 2:** FO-NN training:

  - FO-NN utilizes the dataset with the selected features ($X_{selected}$) to train the candidate architectures within the forest.
  - The optimization process involves iterating through different candidate architectures and evaluating their performance on the classification task.
  - The fitness function for a candidate architecture C_i is based on the performance of the neural network with that architecture using the selected features. Mathematically:

$$F(C_i) = Model\ Performance\ (X_{selected}, C_i) \tag{3}$$

where, $Model\ Performance$ measures the performance of the neural network with architecture C_i on the dataset $X_{selected}$.

  - FO-NN uses genetic algorithms to create new candidate architectures by performing crossover and mutation operations on the existing architectures. These operations introduce variations in the neural network structures. Mathematically, the crossover and mutation processes can be represented as:

$$C_{new} = Crossover(C_{parent1}, C_{parent2})$$
$$C_{mutated} = Mutation(C_{old}) \tag{4}$$

  - Collaboration and competition among candidates in the forest ensure that successful architectures are preserved, shared, and improved while less successful ones are gradually replaced.
  - The optimization process continues over multiple iterations until a termination criterion is met. Termination criteria may include a maximum number of iterations, achieving a satisfactory level of performance, or other conditions.
  - The FO-NN classifier selects the best-performing candidate architecture based on the fitness evaluations as the final model for the classification task.

By integrating feature selection with FO-NN, the classifier ensures that it works with the most relevant features, improving predictive accuracy and interpretability while optimizing the neural network architecture for the given classification task. Fig. 2 shows the flow diagram for the proposed FO-NN approach.

4. Results and discussion

4.1. Evaluation parameters

Table 1 presents the evaluation parameters used to assess the performance of a classification model. These parameters include TP (True Positive), which represents the number of records correctly classified by the model, TN (True Negative), indicating the number of records correctly classified as not belonging to a specific class, FP (False Positive), signifying the number of records incorrectly classified as belonging to the class, and FN (False Negative), which denotes the number of records incorrectly classified as not belonging to the class. These parameters are crucial for measuring the model's accuracy, precision, recall, and F1-score, enabling a comprehensive assessment of its effectiveness in classification tasks by capturing different aspects of its performance.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

$$Sensitivity = \frac{TP}{TP+FN} \tag{7}$$

$$Specificity = \frac{TN}{TN+FN} \tag{8}$$

$$ErrorRate = \frac{FP+FN}{TP+TN+FP+FN} \tag{9}$$

$$False\ Positive\ Rate(FPR) = \frac{FP}{FP+TN} \tag{10}$$

$$F - Score = \frac{2TP}{2TP+FP+FN} \tag{11}$$

$$Matthews\ Correlation\ Coefficient\ (MCC) = \frac{(TP \times TN)-(FP \times FN)}{\sqrt{(TP+FN)(TP+FP)(TN+FN)(TN+FP)}} \tag{12}$$

$$Kappa\ Statistics = \frac{2(TP \times TN - FN \times FP)}{(TP+FP) \times (FP+TN) + (TN+FN) \times (FN+TN)} \tag{13}$$

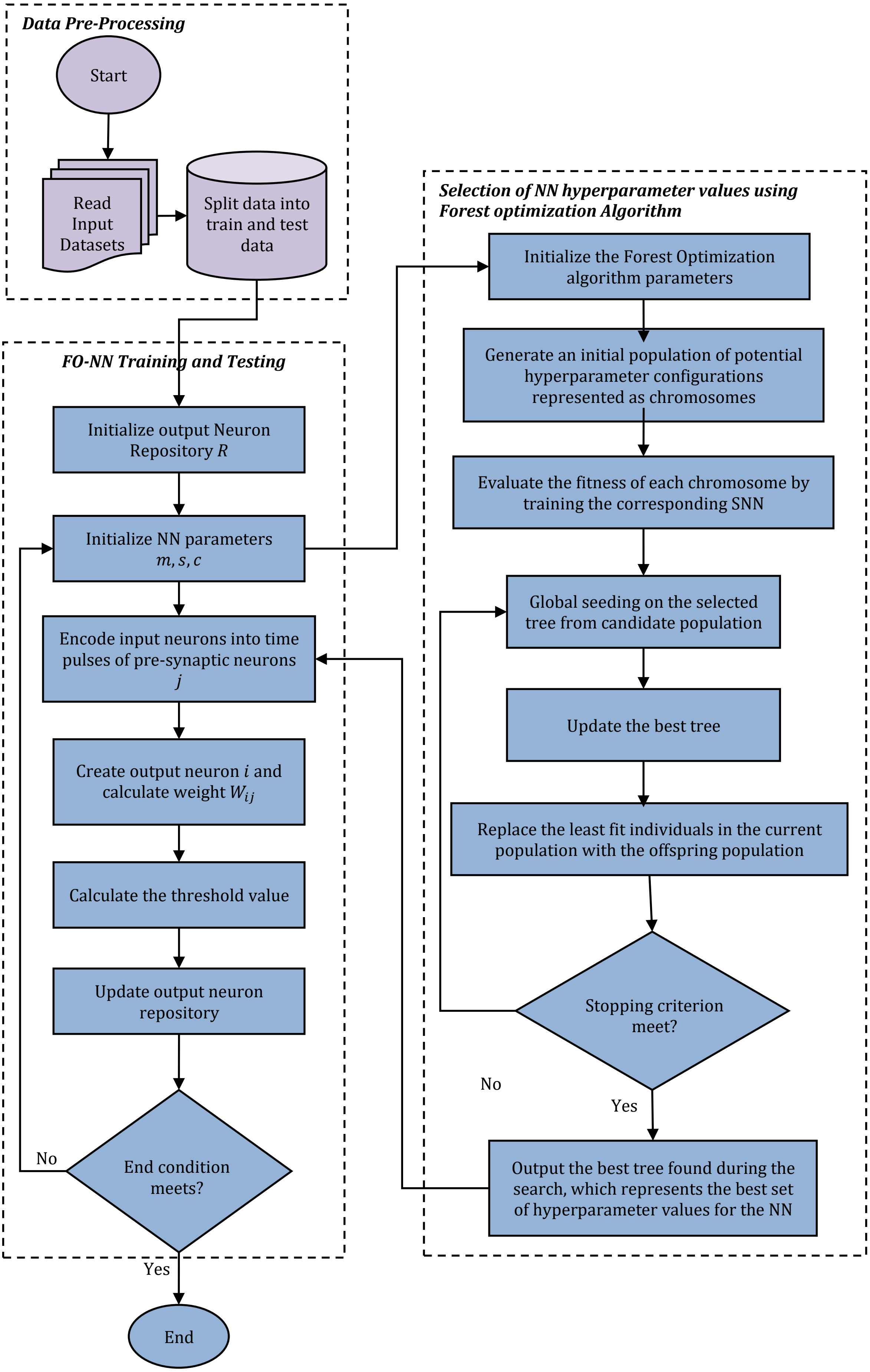

Data Pre-Processing
Start
Read Input Datasets
Split data into train and test data
FO-NN Training and Testing
Initialize output Neuron Repository R
Initialize NN parameters m, s, c
Encode input neurons into time pulses of pre-synaptic neurons j
Create output neuron i and calculate weight W_{ij}
Calculate the threshold value
Update output neuron repository
End condition meets?
No
Yes
End
Selection of NN hyperparameter values using Forest optimization Algorithm
Initialize the Forest Optimization algorithm parameters
Generate an initial population of potential hyperparameter configurations represented as chromosomes
Evaluate the fitness of each chromosome by training the corresponding SNN
Global seeding on the selected tree from candidate population
Update the best tree
Replace the least fit individuals in the current population with the offspring population
Stopping criterion meet?
No
Yes
Output the best tree found during the search, which represents the best set of hyperparameter values for the NN

Fig. 2: Flow diagram for the proposed approach

Table 1: Evaluation parameters

TP (true positive)	Indicated the number of records that were classified as correctly classified
TN (true negative)	Indicated the number of records that were classified as not classified correctly
FP (false positive)	Indicated the number of records that were classified as incorrectly classified
FN (false negative)	Indicated the number of records that were classified as not Classified incorrectly

4.2. Simulation parameters

Table 2 provides the simulation parameters for the SMA, which includes 50 agents, a maximum of 500 iterations, and a parameter Z set at 0.03. These parameters define the characteristics of the SMA-based optimization process. On the other hand, Table 3 outlines the simulation parameters for Forest Optimization. In this context, the problem domain dimension is 2, with a maximum of 500 iterations allowed. The forest in the optimization process is limited to 20 trees, and the maximum age for a tree is 15. Additionally, 15% of the candidate population is used for global seeding, indicating the proportion of candidate solutions shared in the optimization process. These parameters collectively guide the Forest Optimization algorithm's behavior and its exploration of the problem space.

Table 2: Simulation parameters for SMA

No of agents	50
Max iteration	500
Z	0.03

Table 3: Simulation parameters for forest optimization

The dimension of the problem domain	2
Maximum number of iterations	500
The limitation of the forest	20
The maximum allowed Age of a tree	15
The percentage of candidate population for global seeding	15

4.3. Results

Figs. 3, 4, and 5 show the graphical analysis for the graduate, dropout, and enrolled classes, respectively.

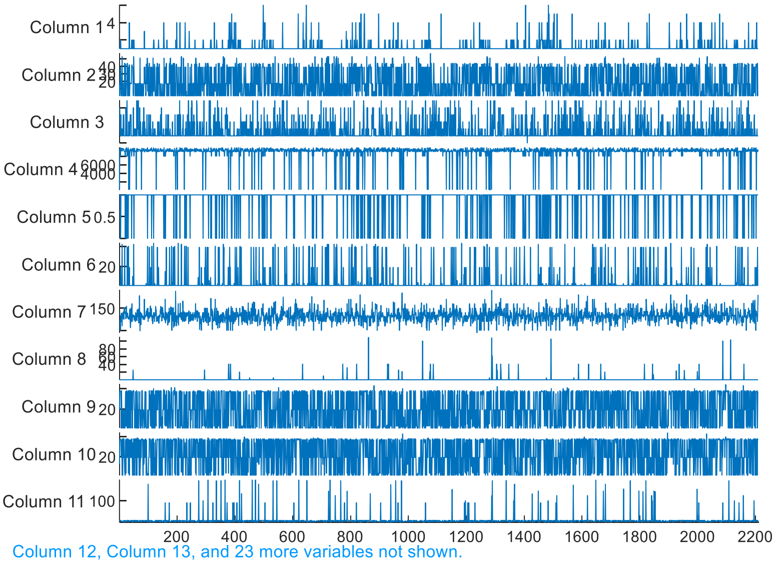


Fig. 3: Graphical analysis for graduate

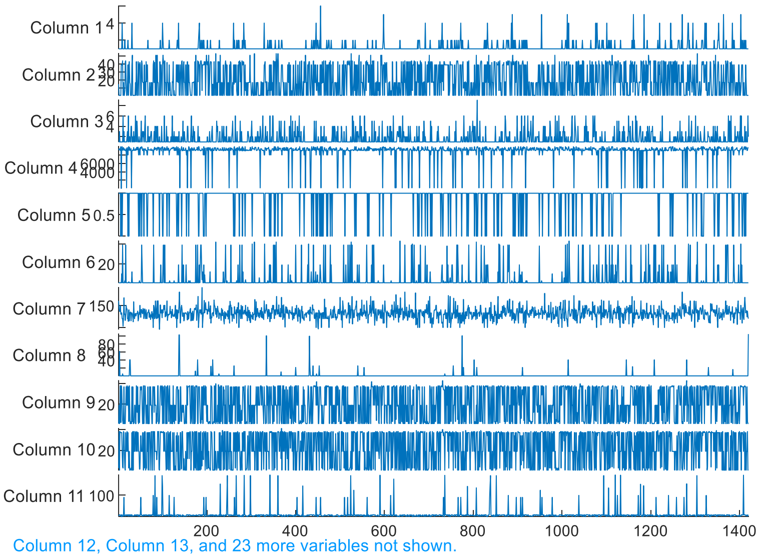


Fig. 4: Graphical analysis for dropout

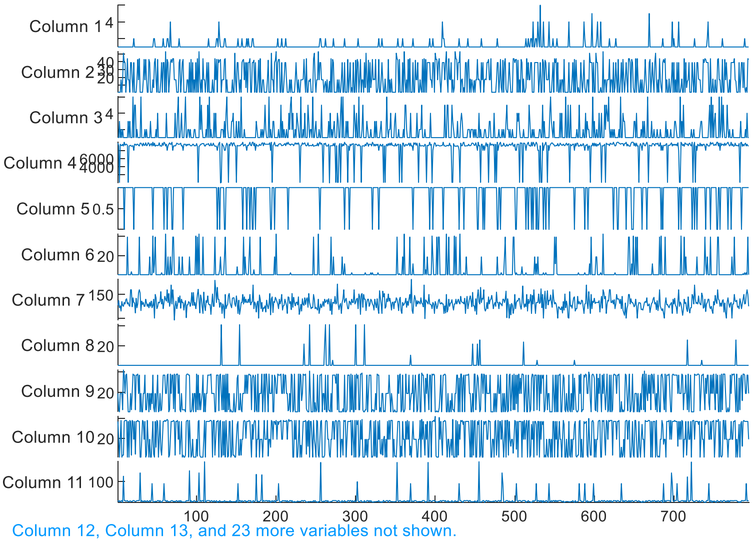


Fig. 5: Graphical analysis for enrolled

Figs. 6, 7, 8, and 9 display the output confusion matrix plots for the Neural Network, FO-NN, SVM, and KNN classifiers, respectively. These confusion matrices offer a comprehensive visual representation of how well our predictive models perform in categorizing students into different academic outcomes. Each matrix, set up as a 3×3 grid, illustrates the true positive, true negative, false positive, and false negative predictions for various student performance categories. These matrices are crucial in assessing the precision and recall of our system for each performance level, shedding light on its strengths and areas for improvement in predicting student success or potential academic challenges.

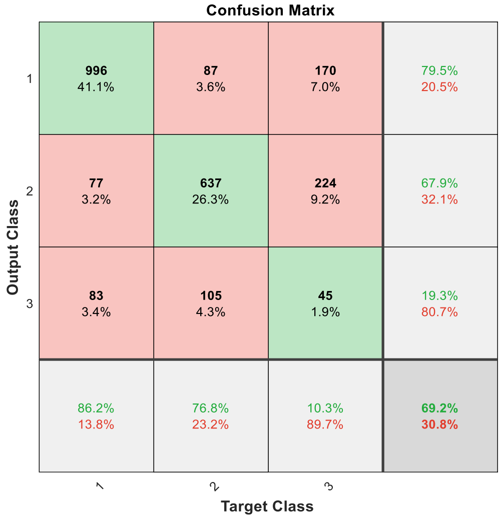


Fig. 6: Confusion matrix plot for unbalanced data using neural network classifier

In Table 4, the performance of different classifiers on balanced data was evaluated based on various parameters. Among the classifiers, the FO-NN classifierdemonstrated the highest accuracy (0.8611) and the lowest error rate (0.1389), indicating its effectiveness in predicting student performance. FO-NN classifier also showed excellent sensitivity (0.8611) and specificity (0.9537), suggesting its ability to correctly identify both positive and negative instances. It achieved a high precision (0.875) and a low false positive rate (0.0463), further highlighting its accuracy in classification.

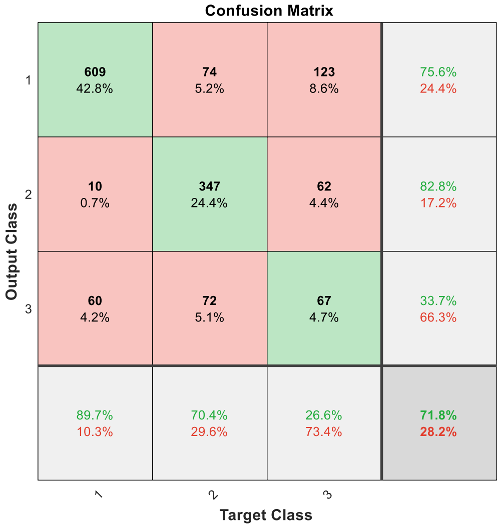


Fig. 7: Confusion matrix plot for unbalanced data using FO-NN classifier

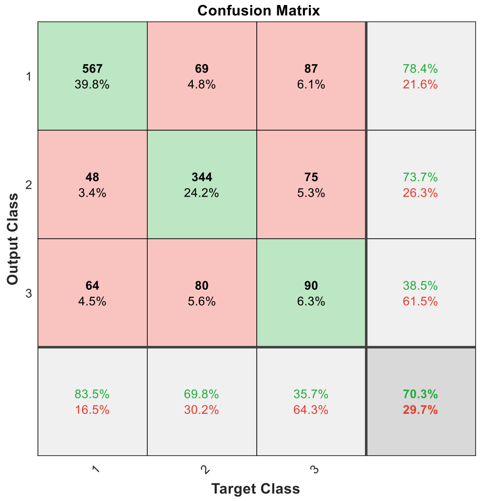
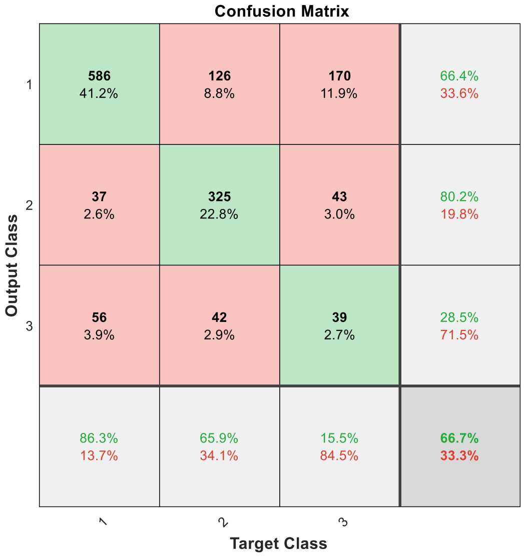


Fig. 8: Confusion matrix plot for unbalanced data using SVM classifier



Output Class \ Target Class	1	2	3	
1	586 41.2%	126 8.8%	170 11.9%	66.4% 33.6%
2	37 2.6%	325 22.8%	43 3.0%	80.2% 19.8%
3	56 3.9%	42 2.9%	39 2.7%	28.5% 71.5%
	86.3% 13.7%	65.9% 34.1%	15.5% 84.5%	66.7% 33.3%

Fig. 9: Confusion matrix plot for unbalanced data using KNN classifier

Additionally, FO-NN exhibited a competitive F1-score (0.8595) and Matthews Correlation Coefficient (0.8212), indicating its overall performance in balancing precision and recall. However, it is worth noting that the Kappa Statistics for FO-NN (0.6296) were lower than other classifiers, suggesting a moderate agreement beyond chance. Overall, the results emphasize the effectiveness of the FO-NN classifier in accurately predicting student performance on balanced data.

Fig. 10 shows a comparison between the number of features used in the SMA-selected features and normal SMOTE-balanced data. Specifically, 31 features are selected in the SMA approach and 36 features in the normal SMOTE balanced data approach. This comparison highlights the differences in feature selection between these two methods and provides insights into the dimensionality of the data used in the study.

Table 4: Comparative analysis of results for various classifier

Parameters	Classifier with balanced data			
	SVM (Mduma, 2023)	KNN (Mduma, 2023)	NN (Mduma, 2023)	FO-NN (Proposed)
Accuracy	0.8099	0.7547	0.7933	0.8611
Error	0.1901	0.2453	0.2067	0.1389
Sensitivity	0.8574	0.8160	0.8449	0.8611
Specificity	0.7149	0.6320	0.6901	0.9537
Precision	0.8574	0.8160	0.8452	0.875
False positive rate	0.2851	0.3680	0.3099	0.0463
F1-score	0.8574	0.8160	0.8450	0.8595
Matthews correlation coefficient	0.5723	0.4480	0.5349	0.8212
Kappa statistics	0.5723	0.4480	0.5349	0.6296

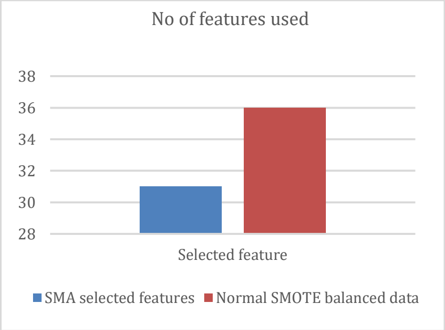


Fig. 10: Number of features used in training for classifier

Table 5 offers a comparative analysis of the performance of various classifiers, such as SVM, k-nearest neighbors (KNN), neural network (NN), and FO-NN, using selected features determined by the SMA. Table 5 presents multiple evaluation parameters, including accuracy, error rate, sensitivity, specificity, precision, false positive rate, F1-score, Matthews Correlation Coefficient, and Kappa Statistics. In the comparison, FO-NN stands out with the highest accuracy and precision (0.889 and 0.938, respectively), suggesting it excels in correctly classifying the dataset. SVM is closely followed by accuracy (0.8148) and Matthews Correlation Coefficient (0.5723). K-NN Classifier, based on a different study by Mutrofin et al. (2019), also exhibits competitive results. These parameter values provide insights into the relative strengths and weaknesses of each classifier for the specific task, aiding in the selection of the most appropriate classifier for the dataset. Overall, the results highlight the effectiveness of the FO-NN classifier when using selected features determined by the SMA for accurate prediction of student performance.

Table 5: Comparative analysis of results for various classifier

Parameters	Classifier with selected features by SMA				
	SVM (Mduma, 2023)	KNN (Mduma, 2023)	NN (Mduma, 2023)	K-NN classifier (Mutrofin et al., 2019)	FO-NN (Proposed)
Accuracy	0.8148	0.759	0.833	0.8455	0.889
Error	0.185	0.241	0.167	-	0.111
Sensitivity	0.852	0.778	0.722	-	0.833
Specificity	0.778	0.741	0.944	-	0.944
Precision	0.793	0.750	0.929	0.8619	0.938
False positive rate	0.821	0.764	0.813	-	0.882
F1-score	0.8574	0.759	0.833	-	0.893
Matthews correlation coefficient	0.5723	0.490	0.5349	-	0.5869
Kappa statistics	0.5856	0.490	0.5349	-	0.5959

5. Conclusion

This research addresses the critical issue of academic underachievement in higher education by introducing an innovative approach that combines the SMA for feature selection with the FO-NN for early identification of students at risk of academic failure. The results demonstrate the efficacy of this approach, particularly on balanced data, with FO-NN classifier achieving the highest accuracy of 88.9% and precision of 93.8% among the classifiers. Additionally, the SMA aids in feature selection, enhancing the interpretability of predictive models. These findings contribute to a more effective strategy for mitigating academic underachievement in higher education. For future research, exploring the scalability and robustness of this approach on larger and more diverse datasets, as well as investigating its applicability in real-world educational settings, could further enhance its practical utility in improving student outcomes. Additionally, incorporating more advanced machine learning techniques and considering the ethical implications of predictive modeling in education would be valuable avenues for future research.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Alex SA, Jhanjhi NZ, Humayun M, Ibrahim AO, and Abulfaraj AW (2022). Deep LSTM model for diabetes prediction with class balancing by SMOTE. Electronics, 11(17): 2737. https://doi.org/10.3390/electronics11172737

Andrade TLD, Rigo SJ, and Barbosa JLV (2021). Active methodology, educational data mining and learning analytics: A systematic mapping study. Informatics in Education, 20(2): 171-204.

Batool S, Rashid J, Nisar MW, Kim J, Kwon HY, and Hussain A (2023). Educational data mining to predict students' academic performance: A survey study. Education and Information Technologies, 28(1): 905-971. https://doi.org/10.1007/s10639-022-11152-y

Hall MM, Worsham RE, and Reavis G (2021). The effects of offering proactive student-success coaching on community college students' academic performance and persistence. Community College Review, 49(2): 202-237. https://doi.org/10.1177/0091552120982030

Hamoud A (2016). Selection of best decision tree algorithm for prediction and classification of students' action. American International Journal of Research in Science, Technology, Engineering and Mathematics, 16(1): 26-32.

Khan A and Ghosh SK (2021). Student performance analysis and prediction in classroom learning: A review of educational data mining studies. Education and Information Technologies, 26: 205-240. https://doi.org/10.1007/s10639-020-10230-3

Lampropoulos G (2023). Educational data mining and learning analytics in the 21st century. In: Wang J (Ed.), Encyclopedia of data science and machine learning: 1642-1651. IGI Global, Pennsylvania, USA. https://doi.org/10.4018/978-1-7998-9220-5.ch098

Li S, Chen H, Wang M, Heidari AA, and Mirjalili S (2020). Slime mould algorithm: A new method for stochastic optimization. Future Generation Computer Systems, 111: 300-323. https://doi.org/10.1016/j.future.2020.03.055

Mangina E and Psyrra G (2021). Review of learning analytics and educational data mining applications. In the 13th International Conference on Education and New Learning Technologies: 949-954. https://doi.org/10.21125/edulearn.2021.0250

Martins MV, Tolledo D, Machado J, Baptista LM, and Realinho V (2021). Early prediction of student's performance in higher education: A case study. In: Rocha Á, Adeli H, Dzemyda G, Moreira F, and Ramalho Correia AM (Eds.), Trends and applications in information systems and technologies: 166-175. Volume 19, Springer International Publishing, Cham, Switzerland. https://doi.org/10.1007/978-3-030-72657-7_16

Mduma N (2023). Data balancing techniques for predicting student dropout using machine learning. Data, 8(3): 49. https://doi.org/10.3390/data8030049

Miguéis VL, Freitas A, Garcia PJ, and Silva A (2018). Early segmentation of students according to their academic performance: A predictive modelling approach. Decision Support Systems, 115: 36-51. https://doi.org/10.1016/j.dss.2018.09.001

Mutrofin S, Mu'alif A, Ginardi RVH, and Fatichah C (2019). Solution of class imbalance of k-nearest neighbor for data of new student admission selection. International Journal of Artificial Intelligence Research, 3(2): 47-55. https://doi.org/10.29099/ijair.v3i2.92

Orujpour M, Feizi-Derakhshi MR, and Rahkar-Farshi T (2020). Multi-modal forest optimization algorithm. Neural Computing and Applications, 32(10): 6159-6173. https://doi.org/10.1007/s00521-019-04113-z

Phan M, De Caigny A, and Coussement K (2023). A decision support framework to incorporate textual data for early student dropout prediction in higher education. Decision Support Systems, 168: 113940. https://doi.org/10.1016/j.dss.2023.113940

Pojon M (2017). Using machine learning to predict student performance. M.Sc. Thesis, University of Tampere, Tampere, Finland.

Romero C and Ventura S (2020). Educational data mining and learning analytics: An updated survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10(3): e1355. https://doi.org/10.1002/widm.1355

Sha L, Raković M, Das A, Gašević D, and Chen G (2022). Leveraging class balancing techniques to alleviate algorithmic bias for predictive tasks in education. IEEE Transactions on Learning Technologies, 15(4): 481-492. https://doi.org/10.1109/TLT.2022.3196278

Smadi A, Al-Qerem A, Nabot A, Jebreen I, Aldweesh A, Alauthman M, and Alzghoul MB (2023). Unlocking the potential of competency exam data with machine learning: Improving higher education evaluation. Sustainability, 15(6): 5267. https://doi.org/10.3390/su15065267

Tsiakmaki M, Kostopoulos G, Kotsiantis S, and Ragos O (2020). Transfer learning from deep neural networks for predicting student performance. Applied Sciences, 10(6): 2145. https://doi.org/10.3390/app10062145

Xiao W, Ji P, and Hu J (2022). A survey on educational data mining methods used for predicting students' performance. Engineering Reports, 4(5): e12482. https://doi.org/10.1002/eng2.12482