IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 July 2024, accepted 16 August 2024, date of publication 20 August 2024, date of current version 10 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3446653

RESEARCH ARTICLE

Identifying and Understanding Student Dropouts Using Metaheuristic Optimized Classifiers and Explainable Artificial Intelligence Techniques

RADIC GORAN[1], LUKA JOVANOVIC[2], (Member, IEEE), NEBOJSA BACANIN[2,3,4], MILOŠ S. STANKOVIĆ[2], VLADIMIR SIMIC[5,6,7], MILOS ANTONIJEVIC[2], AND MIODRAG ZIVKOVIC[2]

[1]ITS—Information Technology School, Comtrade Technology Center, Belgrade, 11070 Zemun, Serbia
[2]Department of Informatics and Computing, Singidunum University, 11000 Belgrade, Serbia
[3]Department of Mathematics, Saveetha School of Engineering, SIMATS, Thandalam, Chennai, Tamil Nadu 602105, India
[4]MEU Research Unit, Middle East University, Amman 11831, Jordan
[5]Faculty of Transport and Traffic Engineering, University of Belgrade, 11010 Belgrade, Serbia
[6]College of Engineering, Department of Industrial Engineering and Management, Yuan Ze University, Taoyuan City 320315, Taiwan
[7]Department of Computer Science and Engineering, College of Informatics, Korea University, Seoul 02841, Republic of Korea

Corresponding author: Nebojsa Bacanin (nbacanin@singidunum.ac.rs)

This research was supported by the Science Fund of the Republic of Serbia, Grant No. 7373, Characterizing crises-caused air pollution alternations using an artificial intelligence-based framework—crAIRsis and Grant No. 7502, Intelligent Multi-Agent Control and Optimization applied to Green Buildings and Environmental Monitoring Drone Swarms—ECOSwarm.

ABSTRACT This study addresses the pressing issue of student dropout in higher education institutions and explores the potential of artificial intelligence (AI) to mitigate this challenge. Student dropout is a complex phenomenon influenced by diverse factors, including internal and external, student characteristics and skills. To enhance retention strategies, it is crucial to identify the nuanced reasons behind dropout decisions, which often go unnoticed by university staff. Therefore, this study investigates the integration of metaheuristic optimization techniques with Adaptive Boosting (AdaBoost) and eXtreme Gradient Boosting (XGBoost) machine learning (ML) models for student dropout identification. By leveraging these well-known ML techniques, the goal is to enhance the accuracy and reliability of dropout predictions in terms of standard classification metrics. Further, by harnessing the exploration and exploitation capabilities of metaheuristics, the study aims to fine-tune both models, thereby increasing their accuracy and robustness in identifying at-risk students. Additionally, to address limitations of existing metaheuristics, a modified version of recently proposed Sinh Cosh Optimizer (SCHO) was developed, that manages to generate well-performing XGBoost and AdaBoost models for students dropout prediction. The study demonstrates that both tuned models can effectively identify at-risk students, providing valuable insights for targeted educational support initiatives. Three experimental evaluations, two with binary and one with multi-class student dropout classification, are conducted on real-world datasets along with rigid comparative analysis and statistical validation with other cutting-edge metaheuristics. According to experimental outcomes, proposed methodology outscores significantly other approaches in terms of performance. Finally, a comprehensive analysis of influential factors was performed using SHapley Additive exPlanations (SHAP) and Shapley Additive Global importancE (SAGE) explainable AI techniques on the best generated models to identify the factors that most significantly influence dropout decisions. This work contributes to advancing AI applications in higher education, providing insights for policymakers and institutions to design targeted interventions for student retention, ultimately enhancing the overall success and effectiveness of higher education systems.

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

INDEX TERMS Higher education dropout, explainable artificial intelligence, metaheuristics, XGBoost, AdaBoost, optimization.

I. INTRODUCTION

Higher education institutions collect and store vast amounts of data related to students and the educational process itself. By analyzing this data and extracting valuable insights, various business segments can be improved, including addressing the issue of student dropout. Student dropout is one of the most challenging problems faced by higher education institutions. It is a complex process, with students having diverse motivations for leaving university. Previous research indicates that the decision to drop out is influenced by a variety of factors [1].

Students decide to drop out mainly due to four main reasons: internal reasons, external reasons, student characteristics, and student skills [2]. These reasons include many sub-factors such as academic integration, social integration, financial status, and personal reasons. University employees, instructors, and support staff are usually unaware of the reasons for the student dropout. The challenge for higher education institutions is to design and improve policies to increase student retention, especially within the first years [3]. Researchers have examined changes in student motivation and explored strategies universities can employ to maintain high levels of motivation among students [4]. Their findings suggest that perceived support from lecturers helps mitigate the decline in course interest.

Dropping out has severe consequences for individuals, educational institutions, and society as a whole [5]. Analyzing the phenomenon of dropping out of studies is complex, given the various social, family, individual, and institutional factors that come into play, and many are also related to socio-demographic and motivational reasons that influence students' decision to drop out of studies [6]. Several factors influencing decisions to leave higher education include having to leave work or other activities to complete studies, the challenge of returning to learning activities after education has been interrupted, and family obligations that are more difficult to accommodate compared to student days. Additional challenges arise from difficulties with re-enrollment and the recognition of previously passed exams, as well as the need to re-attend some courses. Personal problems and motivation to complete studies also contribute to the complexities of the decision.

Higher education institutions are also at a loss due to a part of students dropping out. One of the criteria for evaluating the success of higher education institutions is the number of graduated students, by generation and enrolled program [7]. Dropping out of studies primarily reduces the number of students enrolled in the next year of study and the success of studies itself. Administrative activities become more complex with students returning to studies after a break. This includes verifying eligibility, re-enrollment, recognition of exams, and other related tasks.

Given the importance and complexity of student dropout, this work aims to explore the potential of Artificial Intelligence (AI) in addressing the challenges associated with this issue. Due to the large amount of students' data available at universities and other higher education institutions (HEI) around the world, and considering that machine learning (ML) can identify patterns in historical data that humans cannot, the research proposed in this manuscript goals to improve the detection of students who are likely to drop out by using ML, and to identify the factors that significantly influence student dropout.

It is important to highlight that through a survey of current literature, provided in Section II, a noticeable gap in the application of ML for predicting student dropout exists. Specifically, there is a gap in the application of simpler, less computationally demanding ML models for student dropout prediction. Moreover, the optimization of hyper-parameters for these models in this context has not been thoroughly explored. This proposed work aims to address this research gap by evaluating contemporary optimizers for hyper-parameters' tuning using metaheuristics. Additionally, it seeks to provide insights into the potential benefits of incorporating these optimizers into dropout prediction models and introduce a modified algorithm tailored to the needs of this study.

To accomplish this, proposed study investigates the integration of metaheuristic optimization techniques with Adaptive Boosting (AdaBoost) and eXtreme Gradient Boosting (XGBoost) ML models for student dropout identification. By leveraging these well-known ML techniques, the goal is to enhance the accuracy and reliability of dropout predictions in terms of standard classification metrics. However, since the performance of ML algorithms is closely tied to hyper-parameter choices, metaheuristic algorithms are employed to enhance ML performance for this task. By leveraging the exploration and exploitation capabilities of metaheuristics, both models are fine-tuned, resulting in increased accuracy and robustness in identifying at-risk students.

Furthermore, to address the limitations of existing metaheuristics, a modified version of the recently proposed Sinh Cosh Optimizer (SCHO) [8] was developed. This improved version overcomes the shortcomings of the original approach and successfully designs well-performing AdaBoost and XGBoost models for predicting student dropout. The proposed methodology is integrated into a hybrid metaheuristics and ML framework for student dropout prediction and evaluated on a publicly available real-world dataset through a series of three experiments. The first two simulations focus on identifying which students are more likely to drop out and which ones are likely to continue their studies by applying metaheuristics-tuned AdaBoost and XGBoost algorithms as binary classification approach. The third experiment aims to distinguish between dropout, enrolled, and graduate students using only tuned XGBoost approach (multi-class classification).

To prove robustness of proposed methodology, rigid comparative analysis, as well as statistical validation with other contemporary metaheuristics were performed. According to experimental outcomes, proposed methodology outscores significantly other approaches in terms of standard classification metrics. Finally, a comprehensive analysis of influential factors was performed using SHapley Additive exPlanations (SHAP) [9] and Shapley Additive Global importancE (SAGE) [10] explainable AI techniques on the best generated models to identify the factors that most significantly influence dropout decisions.

Besides all mentioned, this work contributes to advancing AI applications in higher education, providing insights for policymakers and institutions to design targeted interventions for student retention, ultimately enhancing the overall success and effectiveness of higher education systems.

The main contributions of this work can be summarized as follows:

- An investigation into the potential of AdaBoost and XGBoost classifiers for detecting and identifying student dropout using a data-driven approach, filling the gap in the current literature in this area;
- Proposing an enhanced version of the recently introduced SCHO metaheuristic optimizer, specifically crafted to fine-tune hyper-parameters of AdaBoost and XGBoost models for students dropout detection tasks;
- The application of explainable AI techniques to better understand the reasons that drive students to leave education and
- Applying ML in higher education to provide insights for policymakers and institutions, enabling targeted interventions for student retention, and enhancing the overall success and effectiveness of higher education systems.

Rest of the manuscript is structured as follows: Section II presents essential preliminaries for readers to understand the employed methodology and includes a review of relevant literature. The proposed methodology is explained in detail in Section III. Section IV provides experimental setup insights regarding all three conducted simulations, so that the simulations could be reproduced by other researchers, experimental outcomes, comparative analysis and discussion are given in Section V, statistical validation of comparative analysis results, best models interpretation and proposed approach implementation strategy in real-world environment is shown in Section VI, while final remarks, as well as future research directions, are provided in Section VII.

II. RELATED WORKS AND PRELIMINARIES

Student dropout presents a prevalent open problem in academia. Many factors influence students' decisions to leave education, while some of them are social and economic pressure and cultural expectations. Studies suggest [11] that around 14.8% of students resign in the first year, while a further 21.6% resign by the third year of studies. The authors also suggest that swift action, within the first year of studies is needed to improve students outlook. Other factors such as bullying can include students abandoning education, with a study [12] suggesting that between 10.8% and 16.5% of dropouts can be associated with bullying. Further works [13], [14] advocate that academic challenges can be a leading cause of student dropouts. Nevertheless, many factors influence the decision of students to abandon academia, making dropout detection a challenging problem.

The applications of ML and the development of prediction models in the higher education sector have been an active field of research for many years. The topic of predicting dropout and determining the reasons why it happens has been extensively covered in the available literature. This section presents recent publications dealing with the prediction of dropping out of higher education. According to the type of problem addressed, two groups of papers can be observed, those dealing with students' performance [15] and those dealing with students' dropout [16]. The topic related to online education or e-learning systems was specially addressed [17], and prediction methods based on online attendance have also been proposed [18].

Research has been done that aims to evaluate the effectiveness of different ML algorithms in predicting student dropout and academic success [19]. A novel stacking ensemble model based on a hybrid between the random forest, XGBoost [20], gradient boosting, and feed-forward neural networks is also proposed to predict students dropout in university classes [21]. The AdaBoost [22] algorithm to combine regression analysis, neural networks, and decision trees is also employed to predict student dropout [23]. While these works show promising outcomes, the further potential of these methods could be unlocked through a process of hyper-parameter tuning by applying appropriate optimizers.

Research has been done to identify students at risk and predict student dropout of university programs based on the data available at the time of enrollment such as secondary school performance and personal data [11], as well as research where academic data from the first year of study has been processed. One of the indicators for assessing the quality of university careers is the dropout rate between the first and second years. Understanding learning patterns is crucial for predicting student dropout, offering insights into students' behaviors and motivations. Additionally, research has explored how AI and ML can aid in preventing dropout and improving student retention by identifying learning patterns [24].

Recent works have focused on challenges associated with data availability [25], exploring the potential of data augmentation to improve model training with neural networks demonstrating the best outcomes in comparative analysis. Factorization machines have also been effectively combined with deep neural networks for dropout forecasting [26] reporting good results. However, neural networks often suffer from slow convergence rates and demanding computational requirements and the potential of simpler models has yet to be explored in the literature.

While several algorithms have been applied to predict dropout, the application of emerging optimizers has yet to be explored to boost performance. Several promising contenders exist in the literature, with evolutionary algorithms such as differential evolution (DE) [27] and genetic algorithm (GA) [28] have shown impressive outcomes in practical applications, emerging optimizers such as the SCHO algorithm have yet to be fully explored. Promising reported outcomes on Congress on Evolutionary Computation (CEC) [8] testing suggest that further improvements are also possible.

Based on the literature review, it can be concluded that there is a gap in the application of simpler ML models, which are less computationally demanding, for student dropout prediction. Additionally, the optimization of hyper-parameters for such models for this challenge has not been sufficiently explored. Therefore, proposed work seeks to address this research gap by evaluating contemporary optimizers for the task of hyper-parameter tuning via metaheuristics. Additionally, the aim is to provide insights into the potential advantages of integrating these optimizers into dropout prediction models and introduce a modified algorithm specifically for the needs of this work.

A. THE AdaBoost MODEL

The AdaBoost [22] uses an iterative technique to combine simpler classifiers. This is achieved by combining several weaker classifiers. Using an unweighted sample as a starting point, this technique builds a set of classifiers to train the model. When a classifier classifies well, its weights are increased; when it classifies incorrectly, its weights are decreased. The error of a weak classifier, ϵ_t, may be calculated with Eq. (1).

$$\epsilon_t = \frac{\sum_{i=1}^{N} w_{i,t} \cdot \mathrm{I}(h_t(x_i) \neq y_i)}{\sum_{i=1}^{N} w_{i,t}}, \tag{1}$$

where the number of training samples is denoted by N, and the weighted error of the weak learner in the t-th iteration is represented by ϵ_t. The term $w_{i,t}$ indicates the weight of the i-th occurrence in the t-th iteration. The expression $h_t(x_i)$ represents the weak learner's prediction for the i-th occurrence in the t-th iteration. The variable y_i denotes the true label of the i-th occurrence. In addition, the function $\mathrm{I}(\cdot)$ yields 1 in the absence of parenthesis and 0 in the case of a false instance.

Further classifiers are built and the weight modification process is repeated based on the acquired weights. Large groups of classifiers are frequently built in order to develop accurate models. A linear model is produced by adding the scores of each of these sub-models. The classifier weight for the ensemble can be determined using Eq. (2).

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right), \tag{2}$$

where each weak learner in the final ensemble is assigned a weight α_t according to its performance, while the weighted error ϵ_t determines the weak learner's contribution to the overall model output. The *ln* denotes the natural logarithm.

The Eq. shown below is used to update the weights:

$$w_{i,t+1} = w_{i,t} \cdot \exp\left(-\alpha_t \cdot y_i \cdot h_t(x_i)\right), \tag{3}$$

where *exp* stands for the exponential function.

The AdaBoost method is effective for binary classification problems. It does, however, struggle with problems related to multi-class classification. One of the challenges tackled in this research is framed as a binary classification problem, which is why this model was chosen for optimization.

B. THE XGBoost MODEL

The XGBoost is a popular ML approach due to its efficacy [20]. To obtain the best performance levels, the XGBoost settings have to be adjusted. The XGBoost aims to combine many subpar models to produce an accurate prediction model. Notable performance benefits are obtained when the aforesaid optimization is combined with regularization and gradient boosting. The model makes predictions to understand complex interactions between the target and inputs based on the patterns it has identified.

Due to the numerous hyper-parameters of XGBoost, each with large value ranges, trial-and-error approach is impractical. In such complex scenarios, optimization offers a reliable solution. The XGBoost optimization process focuses on three objectives: enhancing generalization, accuracy, and speed.

For XGBoost to achieve the best results an iterative approach is required [20], that tries to optimize the objective function. The XGBoost objective function is given by Eq. (4).

$$\mathrm{obj}(\Theta) = L(\theta) + \Omega(\Theta), \tag{4}$$

where objective function $obj(\theta)$ is determined as an intersection between the loss and regular functions. The regularization term $\Omega(\Theta)$, the loss function $L(\Theta)$, and the set of hyperparameters Θ control the complexity of the model.

The mean square error (MSE) is used to compute the loss function $L(\theta)$, as shown in Eq. (5).

$$\mathrm{L}(\Theta) = \sum_i (y_i - \hat{y_i})^2, \tag{5}$$

where y_i and $\hat{y_i}$ indicate target and predicted values, respectively for each iteration i.

Finally, the Eq. (6) describes the process of differentiation between the actual and predicted values. The minimization of the overall loss function results in an increase of classification accuracy.

$$L(\Theta) = \sum_i [\, y_i \ln (1 + e^{-\hat{y}_i}) + (1 - y_i) \ln (1 + e^{\hat{y}_i})] \tag{6}$$

C. METAHEURISTIC OPTIMIZATION

It has been demonstrated that metaheuristic algorithms are effective for tackling non-deterministic polynomial-time hard (NP-hard) problems. These algorithms are intriguing because they can handle such challenges with a minimal set of rules applied to a population of candidate solutions. By following these rules, global behaviors emerge that guide the algorithm towards promising areas of the search space, leading to optimal solutions.

When considering optimization algorithms, it is important to note the "No Free Lunch" (NFL) theorem [29], which states that no single algorithm performs the best for all test cases. Therefore, empirical testing is required to determine a suitable approach for a given optimization task. Due to this, several optimizers have been developed to address challenges using a wide range of strategies. Some of notable examples that draw inspiration from nature include the firefly algorithm (FA) [30] and GA [31]. While both optimizers show impressive performance, the FA is well known for its powerful exploitation potential. This can in practice lead to premature convergence and lack of exploration needed to locate better solutions. Similarly, the GA has a major advantage in its simple solutions' encoding scheme and implementation. However, a good balance between mutation and crossover operators is often difficult to determine, and that can hinder performance in practical applications.

A more abstract approach is taken by the Particle Swarm Optimization (PSO) [32], Variable Neighborhood Search (VNS) [33] and COLSHADE [34] algorithms drawing inspiration from several sources to formulate a search strategy. While the PSO draws inspiration from a flock of birds, or school of fish behavior, the search space exploitation mechanisms are enforced by velocity and speed rules adopted from physics. Although the PSO is able to strike a good balance between exploration and exploitation, more adaptive approaches have emerged that often overcome some of observed drawbacks of the original optimizer [35], [36]. Besides those mentioned, some more recent additions to metaheuristics family include the Reptile Search Algorithm (RSA) [37] and Elk Heard Optimizer (EHO) [38]. The potential of those algorithms is yet to be fully explored in literature. However, the reported outcomes on the CEC benchmarking suite [39] suggest its decent potential.

Hybrid metaheuristic algorithms have demonstrated exceptional performance across various real-world problems. For instance, in the healthcare industry, notable examples have been documented [40], [41]. Furthermore, hybrid algorithms have enhanced methodologies in fields such as computer security [42], time series forecasting including oil production [43] and stock prices [44], [45]. They have also been applied effectively in anomaly detection [46].

III. PROPOSED METHODOLOGY

This section first describes the original SCHO optimizer used as a baseline for the modified version of this algorithm. Once the core optimizer is explained, some of its shortcomings are highlighted. Finally, modifications and the pseudo-code of the introduced modified approach are provided at the end of this section.

A. ORIGINAL SINH COSH OPTIMIZER

The SCHO algorithm, introduced by [8], is a relatively new metaheuristics approach. It is technically founded and inspired by the characteristics of sinh and cosh, two basic hyperbolic functions. The metaheuristic leverages two salient features of sinh and cosh. To begin with, the constant values of cosh greater than one serve as an important boundary between exploration and exploitation. Second, qualities related to exploration and exploitation are enhanced when sinh values in the interval $[-1, 1]$ are near zero.

With a high-chaotic initial population, the mathematical formulation of population (P) of this metaheuristics is shown in Eq (7):

$$P = \begin{bmatrix} A_{1,1} \ldots A_{1,j} \ldots A_{1,D} \\ A_{1,2} \ldots A_{2,j} \ldots A_{2,D} \\ A_{N,1} \ldots A_{N,j} \ldots A_{N,D} \end{bmatrix} \tag{7}$$

where the location of each agent A is determined by its position in the P and value of its parameters', e.g. notation $A_{2,j}$ denotes j-th parameter of the 2nd agent. All agents are initialized by Eq. (8).

$$A = rnd(N, D) \times (ub - lb) + lb, \tag{8}$$

where rnd represents an arbitrary uniform value within the range $[0, 1]$, ub and lb represent the upper and lower limits of the search domain, while variables D and N denote the number of dimensions in the search space and the total number of solutions, respectively.

Once initialized, the algorithm has to find a balance between exploration and exploitation to guide solutions toward regions of interest in the search space. This equilibrium is governed by Eq. (9), where S splits exploration into two strategies (phases):

$$S = floor(\frac{T}{ct}) \tag{9}$$

where T signifies the maximum count of rounds, and ct represents a control parameter, empirically determined to have a value of 3.6 [8] and $floor()$ function rounds down results.

In the first exploration phase, solutions are updated as defined by Eq (10):

$$A_{(i,j)}^{t+1} = \begin{cases} A_{best}^{(j)} + r_1 \times W_1 \times A_{(i,j)}^{t} & r_2 > 0.5 \\ A_{best}^{(j)} - r_1 \times W_1 \times A_{(i,j)}^{t} & r_2 < 0.5, \end{cases} \tag{10}$$

where the current iteration number is t, $A^{t+1}(i, j)$ describes the j-th dimension of the i-th agent in the subsequent iteration $t + 1$, and $A^{(j)}best$ denotes the best agent in the j dimension. Unifrom pseudo-random values are chosen for r_1 and r_2 from the range $[0, 1]$. The weighted coefficient for the specific agent, represented as W_1, is calculated according to expression (11).

$$W_1 = r_3 \times b_1 \times (cosh \cdot r_4 + \mu \times sinh \cdot r_4 - 1), \tag{11}$$

where parameter b_1 is progressively reduced over the iterations, and random selections of r_3 and r_4 are made uniformly between 0 and 1. In addition, a sensitivity coefficient μ controls the exploration accuracy and according to the findings from [8],its value is hard-coded to 0.388.

The Eq (12) models the second phase of exploration, where the best generated solution does not affect the agents.

$$A_{(i,j)}^{t+1} = \begin{cases} A_{best}^{(j)} + |\epsilon \times W_2 \times A_{best}^{(j)} - A_{i,j}^{(t)}| & r_5 > 0.5 \\ A_{best}^{(j)} - |\epsilon \times W_2 \times A_{best}^{(j)} - A_{i,j}^{(t)}| & r_5 < 0.5, \end{cases} \tag{12}$$

where, the purpose of parameter ϵ is to weaken influence of the current best solution on the agents, therefore its value is hard-coded and set to 0.003 [8]. The r_5 again denotes pseudo-random number drawn from uniform distribution between 0 and 1. The weight coefficient W_2 is calculated as follows:

$$W_2 = r_6 \times b_2, \tag{13}$$

where a uniformly randomly chosen number from [0, 1] is denoted by r_6, while a steadily declining value is indicated by b_2, calculated as shown in Eq. (14).

$$b_2 = 2 \cdot (-\frac{t}{T} + n), \tag{14}$$

where n is sensitive coefficient which guides the exploration process in the second phase.

Exploitation is a crucial stage in optimization process, focusing solutions on promising areas of the search space to come closer to the ideal state. Again, the SCHO performs exploitation by conducting two phases. In the first stage Eq. (15) is used.

$$A_{(i,j)}^{t+1} = \begin{cases} A_{best}^{(j)} + r_7 \times W_3 \times A_{(i,j)}^{t} & r_8 > 0.5 \\ A_{best}^{(j)} - r_7 \times W_3 \times A_{(i,j)}^{t} & r_8 < 0.5 \end{cases} \tag{15}$$

the parameters r_7 and r_8 are uniformly selected within limits [0, 1], while the weighted coefficient W_3 is established as follows:

$$W_3 = r_9 \times b_1 \times (cosh \cdot r_{10} + \mu \times sinh \cdot r_{10}), \tag{16}$$

where r_9 and r_{10} denote again uniformly arbitrarily chosen values within [0, 1]. The same parameter μ, fixed to the value of 0.388, is employed as in the exploitation phase [8].

The second exploitation strategy relies is modeled by the Eq (17):

$$A_{(i,j)}^{t+1} = A_{(i,j)}^{t} + r_{11} \times \frac{sinh \cdot r_{12}}{cosh \cdot r_{1}2} \left| W_2 \times A_{best}^{t} - A_{i,j}^{t} \right|, \tag{17}$$

where r_{11} and r_{12} are uniformly chosen pseudo-random numbers from the [0, 1] range.

More details regarding the baseline SCHO approach can be retrieved from [8].

B. INTRODUCED ADAPTIVE GUIDED BEST SCHO

While the original SCHO algorithm showcases impressive results when evaluated on Congress on Evolutionary Computation (CEC) [39] functions, as one of a recently introduced algorithms, its potential is yet to be fully explored when applied to real-world problems. Through supplementary extensive empirical simulations using basic SCHO on standard CEC constrained and bound-constrained challenges, it was concluded that there exists significant potential for SCHO's improvements.

Specifically, while the baseline SCHO performs well in diversification, it falls short in exploitation, particularly in the later iterations of the algorithm's execution. When the search process is assumed to have converged towards an optimum after some number of iterations, more intensified exploitation is required. Furthermore, it was also noticed that the agents' diversity in initial population could be better. Therefore, modified SCHO approach introduced for the purpose of this study aims towards enhancing baseline SCHO's performance by incorporating three mechanisms.

The first introduced modification is quasi-reflection learning (QRL) procedure into the algorithm's initialization stages with the goal of establishing better coverage of the search space. The initial 50% agents in the population P are initialized using standard SCHO procedure, shown for agent A and parameter j in Eq. (18), while the latter half is generated as QRL counterparts of the initially generated agents. As no additional fitness function evaluations are performed, algorithm complexity remains consistent with the original.

$$A_j = lb_j + (ub_j - lb_j) \cdot \varepsilon, \tag{18}$$

where, ε denotes pseudo-random number generated from uniform distribution between 0 and 1, while lb_j and ub_j represent lower and upper search space constraints for the agent's parameter j.

The generation of agents using the QRL procedure is handled as per Eq (19), where as an example the j-th parameter of agent A is created:

$$A_j^{qr} = \text{rnd}\left(\frac{lb_j + ub_j}{2}, A_j\right), \tag{19}$$

where, A_j^{qr} is QRL counterpart of agent's A j-th parameter and rnd denotes a pseudo-random number drawn from the interval $\left[\frac{lb_j + ub_j}{2}, A_j\right]$.

Secondly, a balancing operator is introduced, which is used to alternate between the SCHO and FA search mechanisms. The FA search expression is chosen since it exhibits outstanding exploitation abilities [47]. Incorporated balancing parameter θ is dynamically updated throughout one algorithm's run to maintain the ratio between exploitation and exploration. Therefore, the firefly search (FS) mechanism, described in Eq (20) [30], is integrated into the baseline SCHO in order to boost exploitation.

$$A_i(t+1) = A_i(t) + \beta e^{-\gamma r_{i,k}^2}(A_k(t) - A_i(t)) + \alpha \epsilon_i(t), \tag{20}$$

where A_i and A_k are two agents from population, with indexes i and k, respectively, t and $t + 1$ represent current and subsequent iteration, respectively, β (agent attraction coefficient) signifies the separation between agents, serving as a metric for their mutual attraction, $r_{i,k}^2$ is square of Euclidean distance between individuals i and k, γ denotes the light absorption coefficient, α controls the degree of randomness, and $\epsilon_i(t)$ represents a stochastic vector.

However, Eq. (20) is commonly swapped with Eq. (21) to improve computational performance [30].

$$\beta(r) = \frac{\beta_0}{\left(1 + \gamma \times r^2\right)}, \tag{21}$$

where β_0 represents the attractiveness at $r = 0$.

More details about the FA approach can be captured from [30].

To avoid premature convergence resulting from the exceptional FS abilities, the θ parameter is dynamically adjusted to promote exploration in the early stages and exploitation in the later stages. At the end of each iteration, the θ value is updated in accordance with Eq (22):

$$\theta = 1 - (\frac{t}{T}), \tag{22}$$

with T denoting the maximum number of iterations in a run.

To alternate execution between the FA search and basic SCHO expression, in each iteration, a uniform random value rnd_1 is selected from the range [0,1]. If this value exceeds θ, the FS is used to update the agents' positions. Otherwise, SCHO search mechanisms are employed for the solutions' update. This approach ensures that exploitation is prioritized in later iterations as the probability of utilizing the FS mechanisms increases.

Finally, third alternation refers to introduction of additional control parameter ψ, that also promotes intensification in later phases of algorithm's execution with the goal of generating high quality solutions, when the search process has already converged towards optimum regions. Similarly to the previous approach, this parameter is updated at the end of every iteration, as per Eq (23).

$$\psi = 1 - (\frac{t}{T}) \tag{23}$$

Following each iteration, an arbitrary uniformly distributed value rnd_2 is selected from a range [0, 1]. Should this value exceeds ψ, a new solution (A_{new}) is generated by applying a uniform genetic crossover operator (smilary as in [48]) between the two best solutions from the population (A_{best_1} and A_{best_2}). The crossover probability (p_c) was empirically determined and hard-coded to the value of 0.1. The worst solution is then replaced with the new one. No further evaluations are conducted on the new solution and thus computational complexity is maintained.

In summary, the QRL population initialization enhances the diversity of the starting population and expands the coverage of the search space. This mechanism also improves diversification in early stages. The two introduced dynamic control parameters (operators) address the issue of an inadequate exploitation-exploration trade-off by promoting exploitation in the later iterations, leading to higher quality solutions. Notably, the computational complexity of the introduced algorithm remains consistent with the original SCHO algorithm, as no additional evaluations are introduced.

The described algorithm is named the adaptive guided best SCHO (AGbSCHO) algorithm. The pseudo-code for this algorithm is presented in Algorithm 1.

Algorithm 1 The AGbSCHO Algorithm Pseudo-Code

```
Initialize parameters $\theta = 1$ and $\psi = 1$
Set population size $N$, maximum number of iterations $T$, current iteration counter $t = 0$, and genetic crossover probability $p_c = 0.1$
Initialize SCHO [8] and FA [30] control parameters to values suggested in original versions
Generate $\frac{N}{2}$ of population $P$ using Eq. (18)
Generate the rest of the population by applying the QRL (Eq.(19))
while $t < T$ do
  for all Agents $A$ in $P$ do
    Assess $A$ based on objective function
    Assign fitness value to $A$
  end for
  Rank agents based on fitness
  Acquire random value $rnd_1$ drawn from uniform distribution within the range [0, 1]
  if $rnd_1 > \theta$ then
    for all Agents $A$ in $P$ do
      Update $A$'s position using FS mechanism
    end for
  else
    for all Agents $A$ in $P$ do
      Update $A$'s position using SCHO mechanism
    end for
  end if
  Acquire random value $rnd_2$ drawn from uniform distribution within the range [0, 1]
  if $rnd_2 > \psi$ then
    Generate a new solution $A_{new}$ by applying uniform crossover operator between $A_{best\_1}$ and $A_{best\_2}$
    Replace worst agent $A_{worst}$ with $A_{new}$
  end if
  Update $\theta$ according to Eq. (22)
  Update $\psi$ according to Eq. (23)
end while
```

As previously stated, in this research, proposed modified SCHO algorithm, alongside the original optimizer, as well as several other well-established optimizers, have been applied to hyper-parameters optimization of XGBoost and AdaBoost models. Each agents' (solutions') parameter represents one hyper-parameter that was tuned. The specifics of the experimental setup are provided in the following section.

IV. EXPERIMENTAL SETUP

To assess the proposed approach, a publicly available students' dropout dataset, provided by the UC Irvine ML repository[1] is used. This dataset contains details available at the point of student registration, encompassing academic trajectory, demographic information, and socio-economic factors. Each instance in the dataset represents one particular student. All students (instances) are divided into three categories with the following class labels: dropout, enrolled, and graduate. It is noted that the used dataset is imbalanced, containing different number of observations per each class. For more details, please refer to https://archive.ics.uci.edu/dataset/697/predict+students+dropout+and+academic+success.

Three experiments are carried out in this research. The first two experiments apply AdaBoost and XGBoost to determine the difference between students who will drop out and those who are currently enrolled or have graduated (binary classification). Minimal pre-processing is required as the dataset is fairly well formatted, with no missing values and relevant categorical values already integer-encoded. These integer-encoded categorical values are then one-hot encoded. In the simulation outcomes section, graduate and currently enrolled students are grouped together under the label student. Conversely to first two simulations, the third experiment utilized XGBoost for a three-class (multi-class) classification task, according to the groups described above (dropout, enrolled and graduate). In each experiment, 70% of the available data is used to train models, while the latter 30% is used to evaluate the models.

In each simulation, several metaheuristic algorithms are applied to optimize model outcomes. Alongside introduced AGbSCHO algorithm, the original SCHO [8] is tested to provide a baseline for evaluations. Also, a set of other cutting-edge contemporary optimizes is also included in the comparison: FA [30], GA [31], PSO [32], VNS [33], RSA [37], EHO [38], and COLSHADE [34]. These algorithms are independently implemented for the conducted experiments. The parameter settings used for each algorithm are based on the suggestions provided in the original works, that introduced each respective algorithm. Simulations are carried out under identical conditions with 10 agents in population ($N = 10$), that try to iterative improve performance over 15 iterations ($T = 15$). Finally to account for randomness associated with heuristic algorithms experimentation is carried out over 30 independent runs ($runtime = 30$). Specific SCHO and FA control parameters in the proposed AGbSCHO approach were set according to suggestions from original papers [8], [30].

Optimizers are utilized to determine optimal (sub-optimal) hyper-parameters for AdaBoost and XGBoost classifiers. The selection of specific hyper-parameters for optimization is based on their significant impact on algorithm performance, and their respective ranges are determined through extensive empirical testing. Hyper-parameters and their corresponding ranges are outlined in Table 1 and Table 2 for the XGBoost and AdaBoost classifiers, respectively.

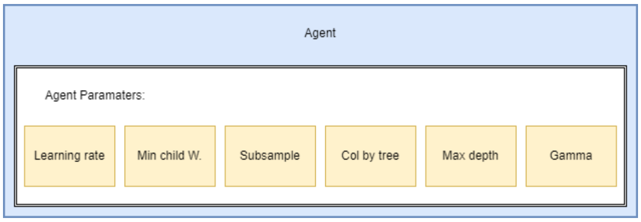


FIGURE 1. Agent solution encoding for XGBoost optimization.

TABLE 1. The AdaBoost tuned hyper-parameters with respective ranges.

Parameter	Lower Bound	Upper bound
Learning rate	0.1	2
Estimators count	10	50
Depth	1	10

TABLE 2. The XGBoost tuned hyper-parameters with respective ranges.

Parameter	Lower Bound	Upper Bound
Learning Rate	0.1	0.9
Minimum child weight	1	10
Subsample	0.01	1
Col sample by tree	0.01	1
Max depth	3	10
γ	0	0.8

To facilitate optimization, the hyper-parameters of XGBoost and AdaBoost are encoded as agent parameters, which are then optimized by each evaluated metaheuristics in an iterative process. Solution encoding schemes for XGBoost and AdaBoost are shown in figures 1 and 2 respectively.

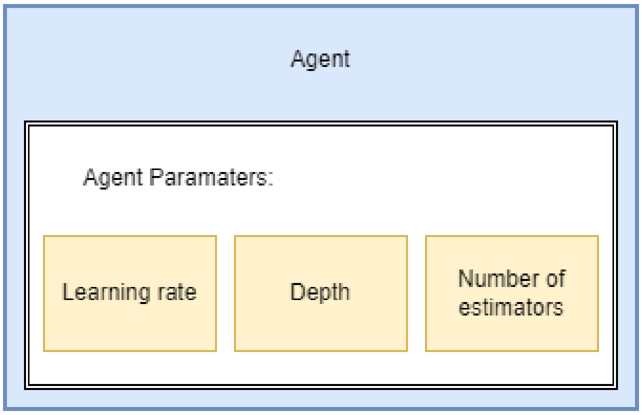


FIGURE 2. Agent solution encoding for AdaBoost optimization.

To evaluate the classification performance of each optimized model, a set of standard metrics is selected. Utilized metrics include accuracy given by Eq. (24), precision shown in Eq. (25), recall calculated as in Eq. (26), and f1-score defined by Eq. 27.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{24}$$

[1] https://archive.ics.uci.edu/dataset/697/predict+students+dropout+and+academic+success (last accessed on 15.02.2023.)

$$Precision = \frac{TP}{FP + TP} \tag{25}$$

$$Recall = \frac{TP}{FN + TP} \tag{26}$$

$$F1 - Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}, \tag{27}$$

where TP and TN indicate true positives and negatives, while FP and FN mark false positives and negatives. A supplementary metric used in this research is Cohen's Kappa defined by Eq. (28), as it is able to deal with imbalanced data. Cohen's Kappa is a common choice where inter-rater reliability is important, as it can evaluate the consistency between several observers.

$$K = \frac{P_{obs} - P_{exp}}{1 - P_{exp}}, \tag{28}$$

where the predicted agreement values are given as:

$$P\_obs = [(TP + FN) \times (TP + FP)(FP + TN) \times (FN + TN)], \tag{29}$$

the observed agreement values $(TP + TN)$ are represented as P_obs. The P_exp is used to denote the Cohen's coefficient K which falls between [0, 1], with a value nearer to 0 denotes less agreement.

Due to its ability to account for class imbalance present in the dataset used, Cohen's Kappa coefficient was selected as the objective function for the experiments. Besides, in each iteration, the classification error rate $(1 - accuracy)$ was also captured and it is referred in this research as an indicator function. The XGBoost and AdaBoost challenge is formulated as a maximization problem aimed at maximizing the Cohen's Kappa score to establish a higher agreement rate between different classes.

Besides all mentioned metrics taken into account in this study, the Matthew's correlation coefficient (MCC) is also included to evaluate the classification potential of the best-constructed models. The MCC is calculated as described in Eq. (30).

$$MCC = \frac{TP \times FN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{30}$$

Finally, to make the process of tuning XGBoost and AdaBoost for this specific challenge more comprehensible for the readers, a flowchart summarizing the utilization of the introduced optimizer, along with other evaluated meta-heuristics within the experimental framework, is provided in Figure 3.

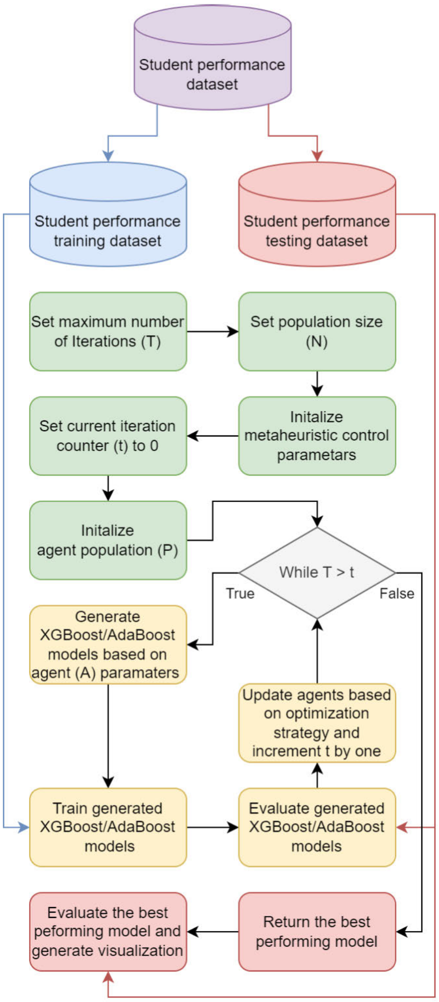

Student performance dataset
Student performance training dataset
Student performance testing dataset
Set maximum number of Iterations (T)
Set population size (N)
Set current iteration counter (t) to 0
Initalize metaheuristic control parametars
Initalize agent population (P)
While T > t
True
False
Generate XGBoost/AdaBoost models based on agent (A) paramaters
Update agents based on optimization strategy and increment t by one
Train generated XGBoost/AdaBoost models
Evaluate generated XGBoost/AdaBoost models
Evaluate the best peforming model and generate visualization
Return the best performing model

FIGURE 3. The flowchart of proposed framework used in simulations.

V. SIMULATION OUTCOMES

The following section presents the results of three conducted experiments: AdaBoost for binary, XGBoost for binary and XGBoost for multi-class dropout classification. Outcomes are compared using various indicator and objective functions. Convergence rates are monitored throughout the optimization process and displayed, while outcome distributions are logged to evaluate the reliability of the algorithms. Detailed comparisons between the best models, as generated by each optimizer in terms of the classification metrics mentioned in the previous section, are provided. Finally, to ensure the repeatability of the experiments by other researchers, the hyper-parameters' selections for the best-performing models are also included.

It is noted that in all results' tables, the best achieved result is marked with bold style. Also, to make clear distinguishing between the AdaBoost and XGBoost experiments, prefix 'AB' for AdaBoost and 'XG' for XGBoost was included before methods' names.

A. EXPERIMENT I - ADABOOST BINARY CLASSIFICATION

Comparison for dropout detection between the best constructed AdaBoost classifiers by each metaheuristics for binary classification challenge is presented in Table 3, in terms of objective, and in Table 4, in terms of indicator function for the best worst mean, median, variance (var) and standard deviation (std) metrics, throughout the course of

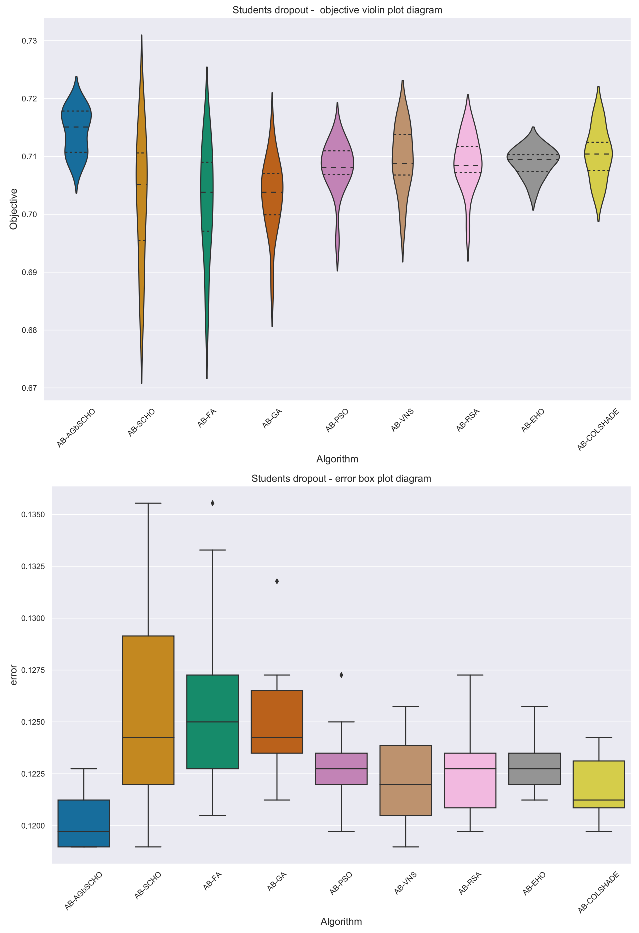


FIGURE 4. Experiment I - tuned AdaBoost objective and indicator function distribution plots for binary dropout classification.

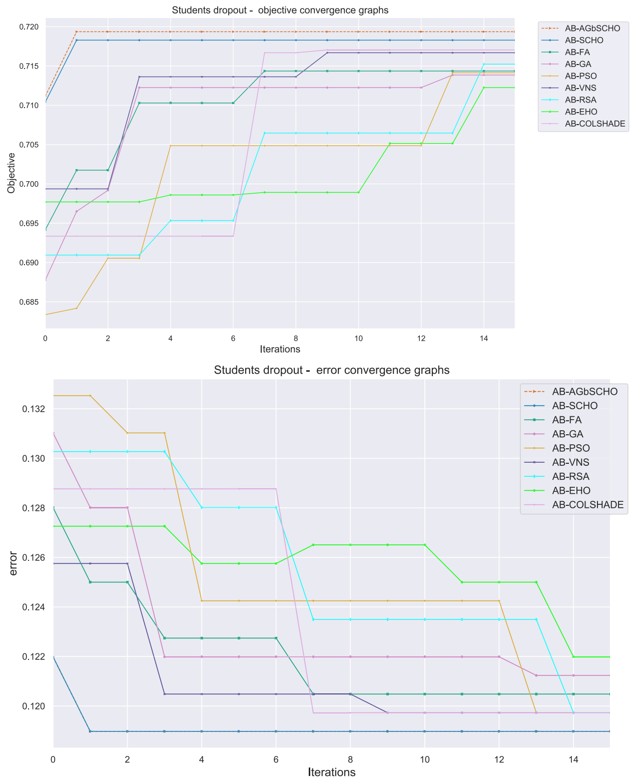


FIGURE 5. Experiment I - tuned AdaBoost objective and indicator function convergence graphs in the best run for binary dropout classification.

30 independent runs. The models optimized by the introduced AGbSCHO algorithm demonstrate the highest scores in all cases attaining the best objective function (Cohen Kappa) rating of 0.719355 and mean, worst and median results of 0.714342, 0.708068 and 0.715080 respectively. The EHO metaheuristics exhibits the best stability, achieving the best values for std and var metrics.

Similar outcomes are presented in terms of the indicator function (classification error rate), with AGbSCHO showing the best performance for the mean, worst, median, standard deviation, and variance metrics, obtaining values of 0.120231, 0.122741, 0.119729, 0.001188, and 0.000141, respectively. However, it is important to note that the original SCHO and the introduced algorithm are tied for the best model in terms of the indicator function. While the EHO and other approaches did not achieve the best results, their demonstrated stability is commendable.

By observing distribution plots a insight in to the repeatability of optimization conducted by a single optimizer can be discerned. A more diapered distribution suggests a lower stability, and therefore results of a single optimization can vary to a greater degree. A more condensed distribution suggests a higher algorithm stability, meaning that the optimization outcomes are more repeatable. Distribution outcomes for objective and indicator function, visualized as violin plots and box plots, respectively, can be observed from Figure 4. The presented figure clearly shows improvements made by the introduced algorithm over other approaches, demonstrating better grouping of outcomes compared to the original algorithm.

Algorithms convergence can help highlight some of the advantages and disadvantages of the optimizers. A high rate of convergence suggests that an algorithm has a powerfully exploitation mechanisms, at the other side, a premature convergence towards a sub-optimal region may suggest a lack of exploration. Comparing optimizers considering convergence rates also provides valuable information regarding the balance between exploration and exploitation. Comparisons of convergence rates for objective and indicator function in the best run over 15 iterations is shown in Figure 5.

The provided figure illustrates that the QRL mechanism of the introduced AGbSCHO allows the metaheuristic to identify the optimal part of the search space immediately after initialization. Furthermore, the steep ascent after a few iterations demonstrates AGbSCHO's exceptional exploitation abilities, outperforming all other methods in terms of convergence speed. The graphs also show that the AGbSCHO efficiently avoids the challenge of trapping in local optima by converging immediately towards promising regions of the search space. In contrast, for example, the RSA metaheuristic stagnates for many iterations before it converges, indicating

TABLE 3. Experiment I - objective function comparative analysis outcomes of tuned AdaBoost for binary dropout classification.

Method	Best	Worst	Mean	Median	Std	Var
AB-AGbSCHO	**0.719355**	**0.708068**	**0.714342**	**0.715080**	0.003670	1.35E-05
AB-SCHO	0.718284	0.683489	0.702859	0.705145	0.010543	1.11E-04
AB-FA	0.714354	0.682692	0.702233	0.703816	0.009205	8.47E-05
AB-GA	0.713845	0.687774	0.703215	0.703816	0.005945	3.53E-05
AB-PSO	0.714136	0.695375	0.708091	0.708068	0.004284	1.84E-05
AB-VNS	0.716681	0.698203	0.709531	0.708813	0.005345	2.86E-05
AB-RSA	0.715232	0.697322	0.709000	0.708441	0.004495	2.02E-05
AB-EHO	0.712250	0.703557	0.708993	0.709435	**0.002355**	**5.55E-06**
AB-COLSHADE	0.717041	0.703816	0.710382	0.710414	0.004188	1.75E-05

TABLE 4. Experiment I - indicator function comparative analysis of tuned AdaBoost for binary dropout classification.

Method	Best	Worst	Mean	Median	Std	Var
AB-AGbSCHO	0.118976	**0.122741**	**0.120231**	**0.119729**	**0.001188**	**1.41E-06**
AB-SCHO	0.118976	0.135542	0.125301	0.124247	0.004632	2.15E-05
AB-FA	0.120482	0.135542	0.125803	0.125000	0.004192	1.76E-05
AB-GA	0.121235	0.131777	0.124950	0.124247	0.002482	6.16E-06
AB-PSO	0.119729	0.127259	0.122892	0.122741	0.001765	3.11E-06
AB-VNS	0.119729	0.125753	0.122189	0.121988	0.002066	4.27E-06
AB-RSA	0.119729	0.127259	0.122490	0.122741	0.001957	3.83E-06
AB-EHO	0.121988	0.125753	0.122691	0.122741	0.001213	1.47E-06
AB-COLSHADE	0.119729	0.123494	0.121837	0.121235	0.001584	2.51E-06

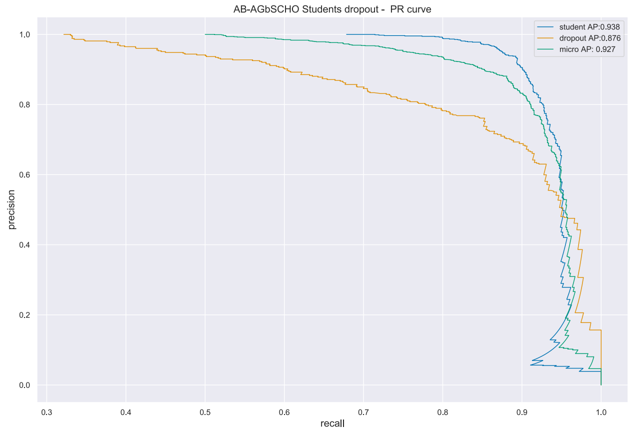


FIGURE 6. Experiment I - the best performing tuned AdaBoost model (AB-AGbSCHO) for binary dropout classification PR curve.

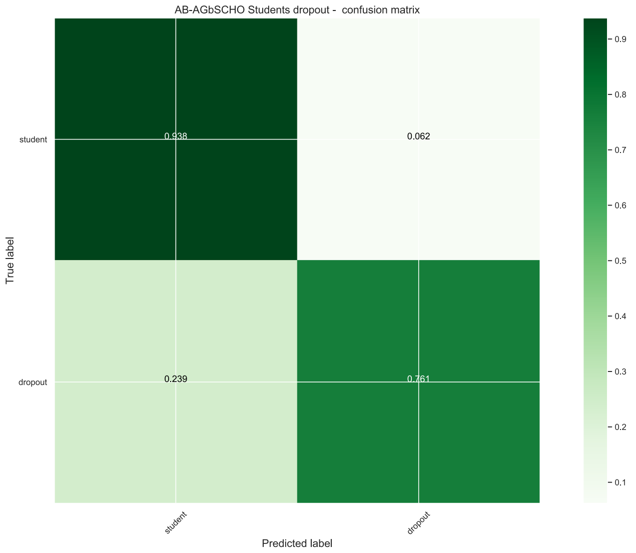


FIGURE 7. Experiment I - the best performing AdaBoost model (AB-AGbSCHO) for binary dropout classification confusion matrix.

that its search process frequently encounters local minima within the search area.

Detailed evaluations and outcomes for the best-performing models generated by each metaheuristics are provided in Table 5. The introduced algorithm exhibits clear dominance, demonstrating superior outcomes across several metrics for both classes. However, the overall accuracy is comparable to the original SCHO algorithm. The PSO also presents interesting results, with a high rate of recall, F1-score, and precision for dropout detection. This aligns with expectations and serves to further reinforce the NFL theorem.

The precision recall (PR) curve is a valuable tool for understanding the trade-offs between precision and recall at various threshold levels, while a confusion matrix is used to evaluate the performance of a classification model by summarizing the predictions compared to the actual outcomes. The PR curves and confusion matrix for the best performing model (in this case ones generated by introduced AGbSCHO metaheuristics) are provided in figures 6 and 7 respectively, followed by the parameter selections of the best performing models optimized by each algorithm in Table 6.

From the presented PR curve (Figure 6) can be seen that the AB-AGbSCHO obtained micro average precision, that takes into account number of instances per each class, of 0.927, which is relatively promising result. This result can be validated from confusion matrix (Figure 7), where the AB-AGbSCHO misclassified only 6.2% and 23.9% of student and dropout classes, respectively. Since the dropout is minority class in the used dataset, this observed misclassification difference is expected.

TABLE 5. Experiment I - detailed metrics for the best-constructed models by tuned AdaBoost for binary dropout classification.

Methods	Metrics	Student	Dropout	Macro avg	Weighted avg		
AB-AGbSCHO	precision	**0.892291**	0.853018	0.872655	0.879664	-	-
	recall	0.937847	**0.761124**	**0.849485**	0.881024	-	-
	f1-score	0.914502	**0.804455**	**0.859479**	**0.879118**	-	-
	accuracy	-	-	-	-	0.881024	-
	MCC	-	-	-	-	-	**0.721769**
AB-SCHO	precision	0.889822	0.858667	0.874244	**0.879804**	-	-
	recall	0.941176	0.754098	0.847637	0.881024	-	-
	f1-score	0.914779	0.802993	0.858886	0.878836	-	-
	accuracy	-	-	-	-	0.881024	-
	MCC	-	-	-	-	-	0.721391
AB-FA	precision	0.887958	0.857909	0.872933	0.878296	-	-
	recall	0.941176	0.749415	0.845295	0.879518	-	-
	f1-score	0.913793	0.800000	0.856897	0.877205	-	-
	accuracy	-	-	-	-	0.879518	-
	MCC	-	-	-	-	-	0.7176970
AB-GA	precision	0.890295	0.850000	0.870148	0.877339	-	-
	recall	0.936737	0.756440	0.846589	0.878765	-	-
	f1-score	0.912926	0.800496	0.856711	0.876776	-	-
	accuracy	-	-	-	-	0.878765	-
	MCC	-	-	-	-	-	0.716349
AB-PSO	precision	0.884058	**0.870166**	0.877112	0.879591	-	-
	recall	**0.947836**	0.737705	0.842770	0.880271	-	-
	f1-score	**0.914837**	0.798479	0.856658	0.877423	-	-
	accuracy	-	-	-	-	0.880271	-
	MCC	-	-	-	-	-	0.719063
AB-VNS	precision	0.889706	0.856383	0.873044	0.878991	-	-
	recall	0.940067	0.754098	0.847082	0.880271	-	-
	f1-score	0.914193	0.801993	0.858093	0.878117	-	-
	accuracy	-	-	-	-	0.880271	-
	MCC	-	-	-	-	-	0.719659
AB-RSA	precision	0.886458	0.864130	**0.875294**	0.879279	-	-
	recall	0.944506	0.744731	0.844618	0.880271	-	-
	f1-score	0.914562	0.800000	0.857281	0.877726	-	-
	accuracy	-	-	-	-	0.880271	-
	MCC	-	-	-	-	-	0.719259
AB-EHO	precision	0.890180	0.847769	0.868974	0.876543	-	-
	recall	0.935627	0.756440	0.846034	0.878012	-	-
	f1-score	0.912338	0.799505	0.855921	0.876058	-	-
	accuracy	-	-	-	-	0.878012	-
	MCC	-	-	-	-	-	0.714640
AB-COLSHADE	precision	0.890526	0.854497	0.872512	0.878942	-	-
	recall	0.938957	0.756440	0.847698	0.880271	-	-
	f1-score	0.914100	0.802484	0.858292	0.878212	-	-
	accuracy	-	-	-	-	0.880271	-
	MCC	-	-	-	-	-	0.719783
	support	901	427				

TABLE 6. Experiment I - the best constructed AdaBoost model parameters' selected by each optimizer for binary dropout classification.

Method	Number of estimators	Depth	Learning rate
AB-AGbSCHO	20	3	0.384179
AB-SCHO	16	3	0.385072
AB-FA	15	3	0.358660
AB-GA	18	3	0.565506
AB-PSO	14	3	0.303599
AB-VNS	17	3	0.316245
AB-RSA	18	3	0.250498
AB-EHO	13	3	0.560980
AB-COLSHADE	20	3	0.316954

B. EXPERIMENT II - XGBOOST BINARY CLASSIFICATION

Comparisons for binary dropout detection between the best constructed XGBoost classifiers optimized by each algorithm are presented in tables 7 and 8 in terms of objective and indicator function, respectively for the best, worst, mean, median, variance (var) and standard deviation (std) metrics for 30 executions, each starting from its own pseudo-random number seed.

Generated models by introduced AGbSCHO algorithm demonstrate the highest scores in all cases attaining a best objective function rating of 0.736784 and a mean and median results of 0.725798 and 0.724569 respectively (Table 7). The worst performing model by proposed method is also better than worst models generated by other approaches with objective function value of 0.717565. However, the PSO algorithm demonstrates superior stability, achieving the best values for the std and var metrics.

Similar results are highlighted in terms of the indicator function (Table 8), with the AGbSCHO producing the top model achieving a score of 0.111446. The mean and median scores across 30 runs were 0.116165 and 0.116717, respectively. Similar to the objective function comparison, even

TABLE 7. Experiment II - objective function comparative analysis of tuned XGBoost for binary dropout classification.

Method	Best	Worst	Mean	Median	Std	Var
XG-AGbSCHO	**0.736784**	**0.717565**	**0.725798**	**0.724569**	0.004876	2.38E-05
XG-SCHO	0.720065	0.707157	0.715174	0.716522	0.003407	1.16E-05
XG-FA	0.726935	0.706633	0.715801	0.716721	0.004480	2.01E-05
XG-GA	0.725066	0.707478	0.714957	0.714208	0.004668	2.18E-05
XG-PSO	0.721661	0.709302	0.717200	0.717925	**0.003333**	**1.11E-05**
XG-VNS	0.725207	0.712752	0.718129	0.718472	0.003440	1.18E-05
XG-RSA	0.725841	0.710658	0.717867	0.717925	0.004139	1.71E-05
XG-EHO	0.724306	0.711152	0.718050	0.718472	0.003959	1.57E-05
XG-COLSHADE	0.729905	0.710784	0.718859	0.717237	0.005785	3.35E-05

TABLE 8. Experiment II - indicator function comparative analysis of tuned XGBoost for binary dropout classification.

Method	Best	Worst	Mean	Median	Std	Var
XG-AGbSCHO	**0.111446**	**0.118976**	**0.116165**	**0.116717**	0.002066	4.27E-06
XG-SCHO	0.118976	0.121235	0.120984	0.121235	0.001053	1.11E-06
XG-FA	0.116717	0.125000	0.121285	0.121235	0.001813	3.29E-06
XG-GA	0.115964	0.124247	0.120984	0.121235	0.002106	4.44E-06
XG-PSO	0.118223	0.121988	0.119528	0.118976	**0.001388**	**1.93E-06**
XG-VNS	0.116717	0.121235	0.119327	0.119729	0.001503	2.26E-06
XG-RSA	0.114458	0.122741	0.119277	0.119729	0.002108	4.45E-06
XG-EHO	0.117470	0.121988	0.119227	0.118976	0.001523	2.32E-06
XG-COLSHADE	0.113705	0.121988	0.119076	0.119729	0.002592	6.72E-06

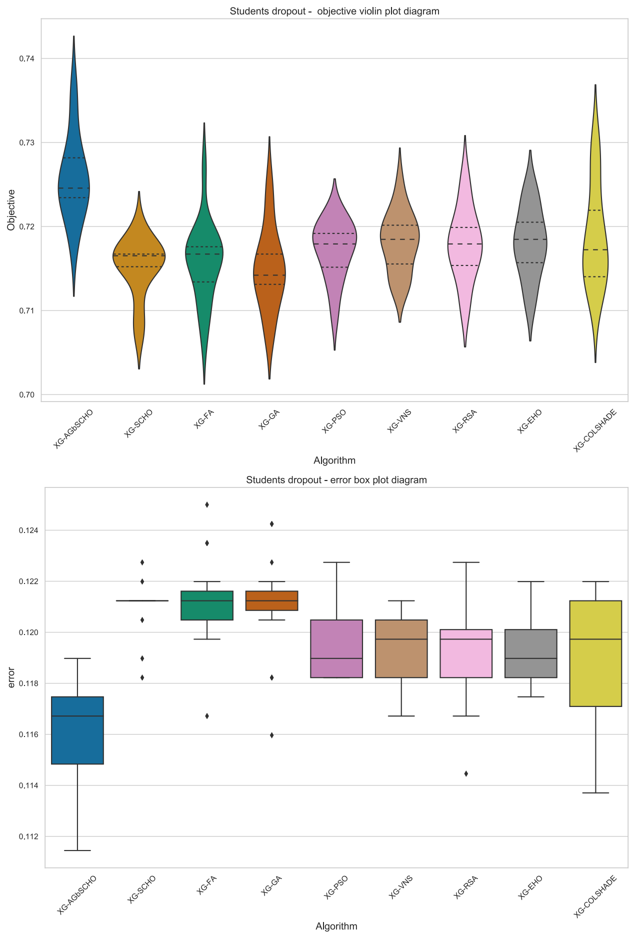


FIGURE 8. Experiment II - tuned XGBoost objective and indicator function distribution plots for binary dropout classification.

the worst-performing model by AGbSCHO outperformed the best models generated by some metaheuristics. Although PSO did not achieve the best results, its demonstrated stability is commendable.

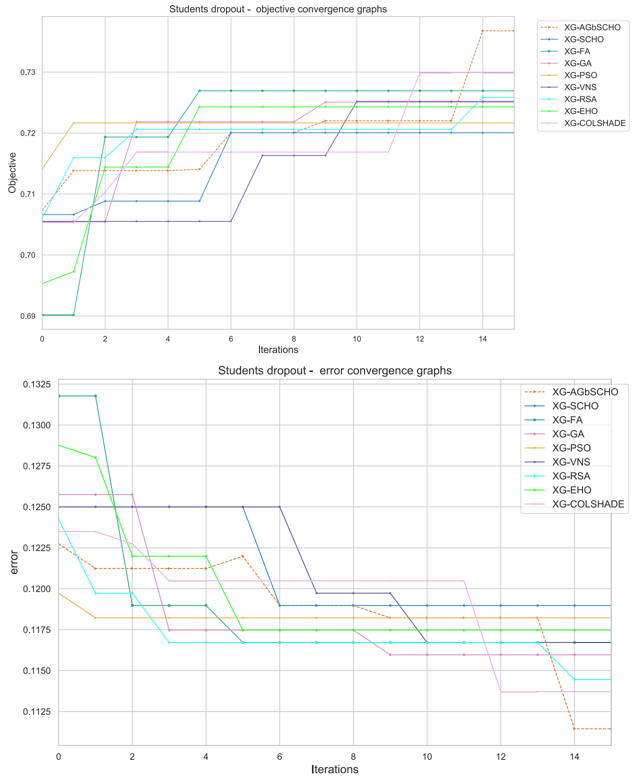


FIGURE 9. Experiment II - tuned XGBoost objective and indicator function convergence graphs in the best run for binary dropout classification.

Similar to the previous experiment (AdaBoost tuning), the stability of the optimizer over 30 runs can be clearly visualized in the distribution plots, such as box and whisker diagrams and violin plots. Distribution outcomes for objective and indicator functions can be observed in Figure 8. The figure shows that AGbSCHO exhibits relatively good

TABLE 9. Experiment II - detailed metrics for the best-constructed models by tuned XGBoost for binary dropout classification.

Methods	Metrics	Student	Dropout	Macro avg	Weighted avg		
XG-AGbSCHO	precision	0.896733	0.868074	0.882404	**0.887518**	-	-
	recall	0.944506	0.770492	**0.857499**	**0.888554**	-	-
	f1-score	**0.920000**	**0.888554**	**0.868189**	**0.886682**	-	-
	accuracy	-	-	-	-	**0.888554**	-
	MCC	-	-	-	-	-	**0.739483**
XG-SCHO	precision	0.893955	0.849351	0.871653	0.879613	-	-
	recall	0.935627	0.765808	0.850718	0.881024	-	-
	f1-score	0.914317	0.805419	0.859868	0.879302	-	-
	accuracy	-	-	-	-	0.881024	-
	MCC	-	-	-	-	-	0.722067
XG-FA	precision	**0.899358**	0.845178	0.872268	0.881937	-	-
	recall	0.932297	**0.779859**	0.856078	0.883283	-	-
	f1-score	0.915531	0.811206	0.863369	0.881987	-	-
	accuracy	-	-	-	-	0.883283	-
	MCC	-	-	-	-	-	0.728166
XG-GA	precision	0.891099	0.865952	0.878526	0.883014	-	-
	recall	0.944506	0.756440	0.850473	0.884036	-	-
	f1-score	0.917026	0.807500	0.862263	0.881809	-	-
	accuracy	-	-	-	-	0.884036	-
	MCC	-	-	-	-	-	0.728459
XG-PSO	precision	0.894068	0.851563	0.872815	0.880401	-	-
	recall	0.936737	0.765808	0.851272	0.881777	-	-
	f1-score	0.914905	0.806412	0.860658	0.880021	-	-
	accuracy	-	-	-	-	0.881777	-
	MCC	-	-	-	-	-	0.723767
XG-VNS	precision	0.895127	0.854167	0.874647	0.881957	-	-
	recall	0.937847	0.768150	0.852998	0.883283	-	-
	f1-score	0.915989	0.808878	0.862434	0.881549	-	-
	accuracy	-	-	-	-	0.883283	-
	MCC	-	-	-	-	-	0.727323
XG-RSA	precision	0.885685	**0.885154**	**0.885419**	0.885514	-	-
	recall	**0.954495**	0.740047	0.847271	0.885542	-	-
	f1-score	0.918803	0.806122	0.862463	0.882572	-	-
	accuracy	-	-	-	-	0.885542	-
	MCC	-	-	-	-	-	0.731697
XG-EHO	precision	0.896699	0.848329	0.872514	0.881146	-	-
	recall	0.934517	0.772834	0.853675	0.882530	-	-
	f1-score	0.915217	0.808824	0.862020	0.881008	-	-
	accuracy	-	-	-	-	0.882530	-
	MCC	-	-	-	-	-	0.725945
XG-COLSHADE	precision	0.891441	0.872973	0.882207	0.885503	-	-
	recall	0.947836	0.756440	0.852138	0.886295	-	-
	f1-score	0.918774	0.810540	0.864657	0.883972	-	-
	accuracy	-	-	-	-	0.886295	-
	MCC	-	-	-	-	-	0.733729
	support	901	427				

stability, achieving consistent results across all 30 runs. Similar conclusions can be drawn for the VNS, PSO, EHO, and COLSHADE metaheuristics. Conversely, FA and GA demonstrate poor stability with scattered results in different runs. The RSA approach is relatively stable, but clearly visible outliers in some executions.

As noted previously, algorithm convergence can help highlight the advantages and disadvantages of optimizers. A high rate of convergence may indicate that an algorithm has strong exploitation mechanisms; however, premature convergence to a suboptimal region can suggest a lack of exploration. Comparing optimizers based on their convergence rates provides valuable information about the balance between exploration and exploitation. The Figure 9 illustrates the comparisons of convergence rates for the objective and indicator functions in the best run. While many optimizers tend to focus on sub-optimal regions of the search space, the modifications to the SCHO algorithm enable it to overcome local traps and focus on more promising regions within the search space. With the assistance of dynamic parameters, AGbSCHO converges slowly yet steadily towards the best results, surpassing all other metaheuristics.

Detailed evaluations and outcomes for the best-performing models are provided in Table 9. The introduced algorithm shows clear dominance, delivering superior results across several metrics. Unlike the previous experiment, the overall accuracy is significantly better than that achieved by the baseline SCHO algorithm. The RSA also presents notable results, with a high rate of recall and precision for some classes, closely followed by FA. Once again, this outcome is expected and further reinforces the NFL theorem.

The PR curves and confusion matrix for the best performing model (in this case the one obtained by the AGbSCHO

TABLE 10. Experiment II - the best constructed XGBoost model parameters' selected by each optimizer for binary dropout classification.

Methods	Learning rate	Min child W.	Subsample	Col by tree	Max depth	Gamma
XG-AGbSCHO	0.515965	3.621653	0.907950	0.863299	6	0.000000
XG-SCHO	0.769270	9.930883	0.772600	1.000000	6	0.800000
XG-FA	0.590572	6.973637	0.842200	1.000000	8	0.157791
XG-GA	0.900000	10.000000	0.927848	0.372662	5	0.000000
XG-PSO	0.900000	1.000000	0.653400	0.694760	3	0.082121
XG-VNS	0.552244	5.409314	0.912545	1.000000	8	0.102573
XG-RSA	0.606682	1.000000	1.000000	0.261665	4	0.003928
XG-EHO	0.482289	6.758533	0.653890	0.999295	10	0.083886
XG-COLSHADE	0.483608	6.587873	0.765078	0.936913	10	0.374648

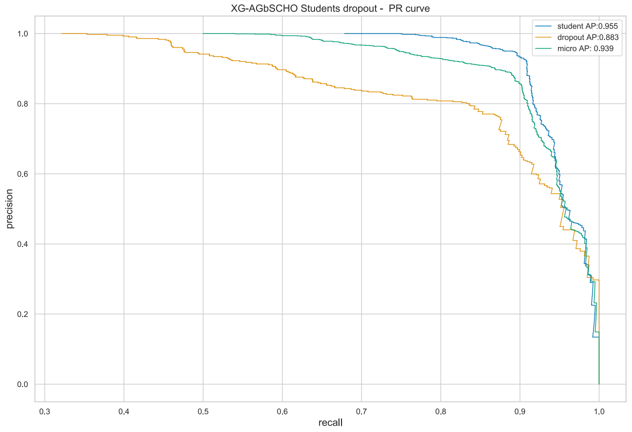


FIGURE 10. Experiment II - the best performing tuned XGBoost model (XG-AGbSCHO) for binary dropout classification PR curve.

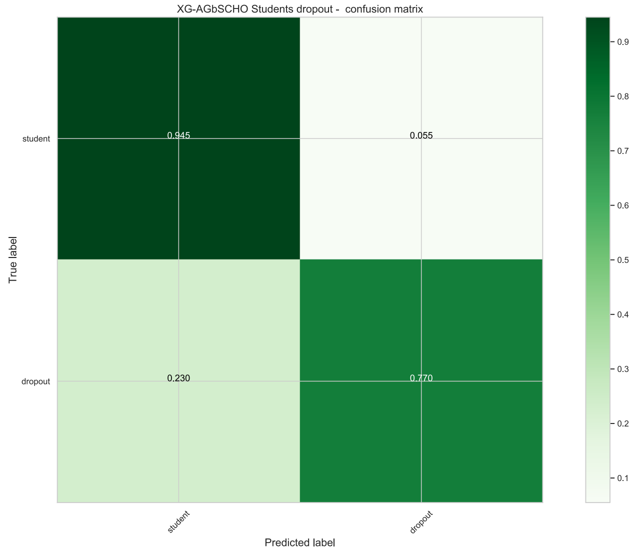


FIGURE 11. Experiment II - the best performing XGBoost model (XG-AGbSCHO) for binary dropout classification confusion matrix.

metaheuristics) are provided in Figure 10 and Figure 11 respectively. The achieved average precision for student and dropout classes of 0.955 and 0.883, respectively, along with a micro-averaged precision of 0.939, suggests that the proposed AGbSCHO algorithm can be highly effective in practical dropout detection challenges. These results are supported by the confusion matrix, where AGbSCHO achieves a true positive (TP) rate of 77% for the dropout class and 94.5% for the student class. As previously noted, given that the dropout class has significantly fewer observations than the student class in the used dataset, a higher TP rate for the student class is expected.

Finally, to enable reproducibility of the simulations by other researchers, the parameter selections for the best-performing models optimized by each algorithm are shown in Table 10.

C. EXPERIMENT III - XGBOOOST MULTI-CLASS CLASSIFICATION

Oppositely to first two simulations, the third experiment utilizes XGBoost for a multi-class classification task, according to the class labels described previously: dropout, enrolled and graduate.

Comparisons for multi-class classification among the best XGBoost classifiers optimized by each algorithm are presented in Table 11 for objective functions and in Table 12 for indicator functions, covering metrics such as best, worst, mean, median, variance (var), and standard deviation (std) calculated over 30 runs. The models optimized by the AGbSCHO algorithm consistently achieve the highest scores across almost all metrics. Specifically, they achieve a best objective function score of 0.666221, with mean and median results of 0.651079 and 0.650129, respectively. Additionally, even the worst metric outperforms those established by all other included approaches. Similar to the previous simulations, in this case, the GA method exhibits the best std and var metrics.

Similar outcomes are demonstrated in terms of the indicator function, where AGbSCHO outperformed all other methods with a best score of 0.199548, and mean and median scores of 0.208183 and 0.208584, respectively. While GA did not achieve the best outcomes, its demonstrated stability is commendable.

The distribution outcomes of objective and indicator functions can be observed in Figure 12. Unlike previous experiments, while the proposed AGbSCHO outperforms all other state-of-the-art metaheuristics in the best run for the multi-class experiment, overall stability is not as strong as in the binary simulations. From the figure, both violin plots (objective function) and box and whisker plots (indicator function) clearly show an outlier representing the best run. The FA, VNS, and GA perform similarly to the proposed

TABLE 11. Experiment III - objective function comparative analysis outcomes of tuned XGBoost for multi-class dropout classification.

Method	Best	Worst	Mean	Median	Std	Var
XG-AGbSCHO	**0.666221**	**0.645646**	**0.651079**	**0.650129**	0.004551	2.07E-05
XG-SCHO	0.653408	0.640276	0.647694	0.648351	0.003593	1.29E-05
XG-FA	0.658901	0.644092	0.648294	0.647296	0.003918	1.54E-05
XG-GA	0.655117	0.643851	0.647384	0.647138	**0.002875**	**8.26E-06**
XG-PSO	0.653424	0.637195	0.644810	0.644092	0.004770	2.28E-05
XG-VNS	0.655538	0.637851	0.646013	0.645443	0.004293	1.84E-05
XG-RSA	0.652273	0.635535	0.644102	0.643532	0.004799	2.30E-05
XG-EHO	0.654193	0.635383	0.644569	0.643022	0.005766	3.32E-05
XG-COLSHADE	0.653780	0.633860	0.645138	0.645417	0.005479	3.00E-05

TABLE 12. Experiment III - indicator function comparative analysis of tuned XGBoost for multi-class dropout classification.

Method	Best	Worst	Mean	Median	Std	Var
XG-AGbSCHO	**0.199548**	**0.210843**	**0.208183**	**0.208584**	0.002577	6.64E-06
XG-SCHO	0.207078	0.213855	0.210492	0.210843	0.001800	3.24E-06
XG-FA	0.204819	0.211596	0.210442	0.210843	0.002073	4.30E-06
XG-GA	0.206325	0.211596	0.210843	0.210843	**0.001673**	**2.80E-06**
XG-PSO	0.207078	0.214608	0.211898	0.212349	0.002620	6.86E-06
XG-VNS	0.204819	0.216867	0.211044	0.211596	0.002715	7.37E-06
XG-RSA	0.207831	0.217620	0.212450	0.212349	0.003133	9.82E-06
XG-EHO	0.207831	0.219127	0.212248	0.211596	0.003907	1.53E-05
XG-COLSHADE	0.207078	0.217620	0.211998	0.211596	0.003181	1.01E-05

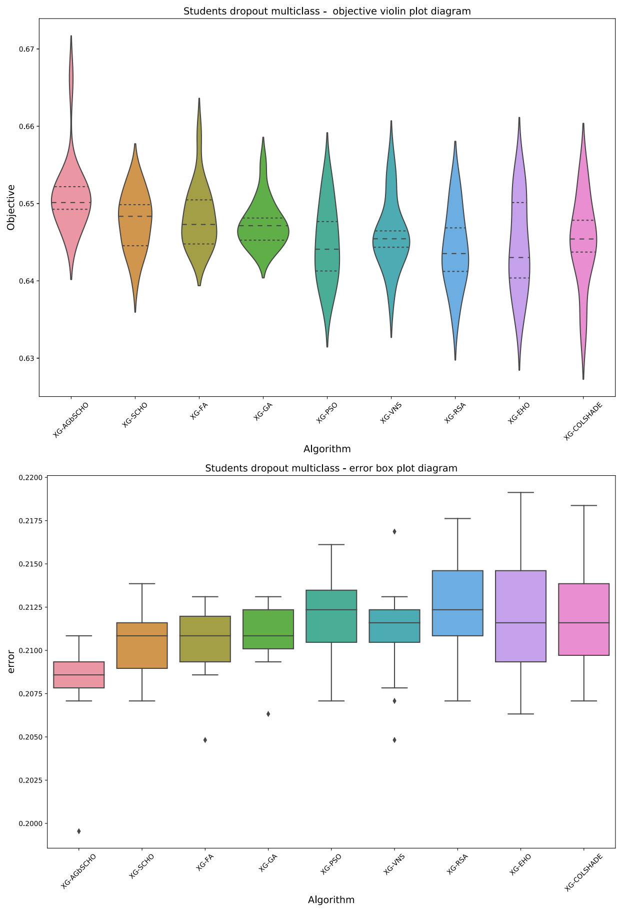


FIGURE 12. Experiment III - tuned XGBoost objective and indicator function distribution plots for multi-class dropout classification.

method, with even more observed outliers. Oppositely, in this case, PSO, RSA, EHO, and COLSHADE exhibit the best stability across 30 runs.

The comparison of convergence speed in the best run for all metaheuristics is illustrated in Figure 13. In this case, the AGbSCHO's QRL initialization did not facilitate rapid convergence towards an optimum at the beginning of the run. However, dynamic parameters θ and ψ, which gradually enhance exploitation and intensification-diversification trade-off throughout the run, enabled the algorithm to converge slowly but steadily towards the best solution. Notably, around the 7-th iteration, a significant improvement in result quality becomes apparent. Other methods, with the exception of FA, exhibited average convergence rate performance. FA successfully generated an initial population with satisfactory solution quality, but it did not converge over the course of all 15 iterations.

Detailed evaluations and outcomes for the top-performing models are presented in Table 13. The introduced algorithm shows clear superiority, achieving excellent results across various metrics. Notably, AGbSCHO excels in macro and weighted averages, as well as accuracy. However, for specific metrics such as precision, recall, and f1-score for certain classes, FA, GA, PSO, VNS, and EHO algorithms outperform the modified proposed algorithm. This once again illustrates the nuances highlighted by the NFL theorem.

The PR curves and confusion matrix for the best performing models are provided in Figure 14 and Figure 15 respectively, while the parameter selections for the best performing models optimized by each algorithm are shown in Table 14 to facilitate repeatability of experiments.

With a micro average precision of 0.864, as observed in Figure 14, it can be concluded that XG-AGbSCHO may be applied for real-world multi-class student dropout challenges. The best model achieves the highest average precision for the graduate class and the lowest for the enrolled class, with values of 0.901 and 0.564, respectively. Once again, as seen in previous simulations, this can be verified by the

TABLE 13. Experiment III - detailed metrics for the best-constructed models by tuned XGBoost for multi-class dropout classification.

Methods	Metric	Graduate	Student	Dropout	Macro avg	Weighted avg		
XG-AGbSCHO	precision	0.817089	0.639344	0.843434	**0.766623**	**0.793705**	-	-
	recall	0.923077	0.491597	0.782201	**0.732292**	**0.800452**	-	-
	f1-score	0.866856	0.555819	0.811665	**0.744780**	**0.793367**	-	-
	accuracy	-	-	-	-	-	**0.800452**	-
	MCC	-	-	-	-	-	-	0.723010
XG-SCHO	precision	0.811245	0.621469	0.834158	0.755624	0.784601	-	-
	recall	0.914027	0.462185	0.789227	0.721813	0.792922	-	-
	f1-score	0.859574	0.530120	0.811071	0.733589	0.784935	-	-
	accuracy	-	-	-	-	-	0.792922	-
	MCC	-	-	-	-	-	-	0.707546
XG-FA	precision	0.821769	0.631579	0.823821	0.759056	0.788343	-	-
	recall	0.911011	**0.504202**	0.777518	0.730910	0.795181	-	-
	f1-score	0.864092	**0.560748**	0.800000	0.741613	0.789119	-	-
	accuracy	-	-	-	-	-	0.795181	-
	MCC	-	-	-	-	-	-	0.718134
XG-GA	precision	0.809651	0.606557	**0.849624**	0.755278	0.786106	-	-
	recall	0.911011	0.466387	0.793911	0.723769	0.793675	-	-
	f1-score	0.857346	0.527316	**0.820823**	0.735162	0.786455	-	-
	accuracy	-	-	-	-	-	0.793675	-
	MCC	-	-	-	-	-	-	0.702856
XG-PSO	precision	0.808311	**0.642045**	0.830049	0.760135	0.785503	-	-
	recall	0.909502	0.474790	0.789227	0.724506	0.792922	-	-
	f1-score	0.855926	0.545894	0.809124	0.736981	0.785314	-	-
	accuracy	-	-	-	-	-	0.792922	-
	MCC	-	-	-	-	-	-	0.699821
XG-VNS	precision	0.813648	0.639053	0.826196	0.759633	0.786393	-	-
	recall	**0.935143**	0.453782	0.768150	0.719025	0.795181	-	-
	f1-score	**0.870175**	0.530713	0.796117	0.732335	0.785525	-	-
	accuracy	-	-	-	-	-	0.795181	-
	MCC	-	-	-	-	-	-	**0.729599**
XG-RSA	precision	0.808568	0.614525	0.840796	0.754630	0.784155	-	-
	recall	0.911011	0.462185	0.791569	0.721588	0.792169	-	-
	f1-score	0.856738	0.527578	0.815440	0.733252	0.784468	-	-
	accuracy	-	-	-	-	-	0.792169	-
	MCC	-	-	-	-	-	-	0.701474
XG-EHO	precision	**0.823529**	0.600000	0.825553	0.749694	0.784120	-	-
	recall	0.907994	0.478992	0.786885	0.724624	0.792169	-	-
	f1-score	0.863702	0.532710	0.805755	0.734056	0.785751	-	-
	accuracy	-	-	-	-	-	0.792169	-
	MCC	-	-	-	-	-	-	0.717671
XG-COLSHADE	precision	0.816958	0.612360	0.828010	0.752443	0.783844	-	-
	recall	0.915535	0.457983	0.789227	0.720915	0.792922	-	-
	f1-score	0.863442	0.524038	0.808153	0.731878	0.784838	-	-
	accuracy	-	-	-	-	-	0.792922	-
	MCC	-	-	-	-	-	-	0.716110
	support	663	238	427				

confusion matrix (Figure 15), where the best-performing XG-AGbSCHO model achieves TP rates of 92.3

VI. OUTCOMES VALIDATION AND INTERPRETATION

Relying solely on results may be insufficient to reach a conclusive solution. Modern computer science demands rigorous validation of outcomes through statistical methods to substantiate their reliability. Moreover, understanding the reasons behind model decisions is often as critical as the decisions themselves. Therefore, this section presents the results of statistical analysis followed by model interpretations using explainable AI techniques.

A. STATISTICAL VALIDATION

In recent years, computer scientists must assess the statistical significance of proposed enhancements since experimental data is not always sufficient to demonstrate that one algorithm is superior to its competitors. According to the literature recommendations, it is suggested by [49] that in these kinds of situations, statistical testing should involve building a representative set of results for every validated method. However, when dealing with outliers from a non-normal distribution, this strategy could be ineffective, yielding erroneous results. Also, according to [49], the question of whether or not to use the mean of objective function value across multiple runs in statistical tests when comparing stochastic techniques is still up for dispute.

To proceede with statistical validation, first a decision whether to use parametric, or non-parametric statistical tests should be made. For this purpose, a well-known Shapiro-Wilk test [50] for single-problem analysis was employed. For every conceivable combination of method and challenge (experiment), the related p-values were determined by aggregating the objective function outcomes of

TABLE 14. Experiment III - the best constructed XGBoost model parameters' selected by each optimizer for multi-class dropout classification.

Methods	Learning rate	Min child W.	Subsample	Col by tree	Max depth	Gamma
XG-AGbSCHO	0.695428	8.728771	1.000000	1.000000	8	0.428350
XG-SCHO	0.779573	4.564434	1.000000	1.000000	6	0.800000
XG-FA	0.900000	9.474133	1.000000	1.000000	9	0.800000
XG-GA	0.683497	7.964178	1.000000	1.000000	8	0.800000
XG-PSO	0.672647	1.000000	1.000000	1.000000	7	0.800000
XG-VNS	0.900000	7.108702	0.673425	1.000000	3	0.123549
XG-RSA	0.900000	4.622149	1.000000	0.993888	7	0.201291
XG-EHO	0.900000	8.793894	1.000000	0.719597	7	0.800000
XG-COLSHADE	0.658740	4.674254	0.942203	0.904612	7	0.800000

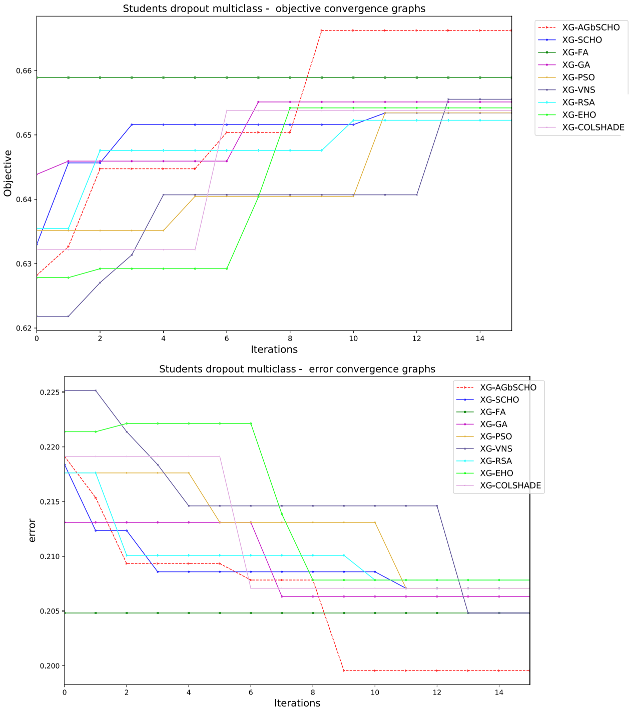


FIGURE 13. Experiment III - tuned XGBoost objective and indicator function convergence graphs in the best run for multi-class dropout classification.

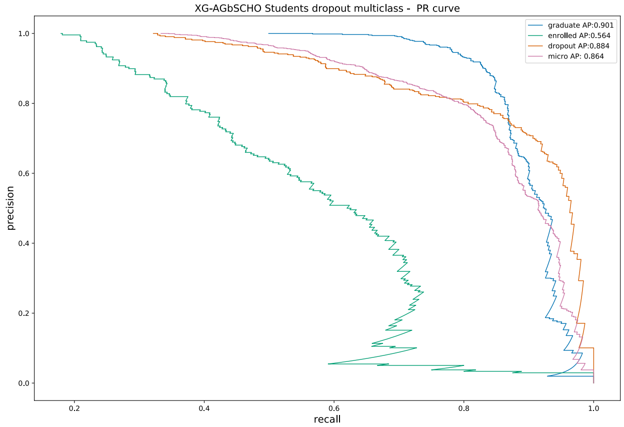


FIGURE 14. Experiment III - the best performing tuned XGBoost model (XG-AGbSCHO) for multi-class dropout classification PR curve.

each run to form a data sample for each algorithm and problem pair. The p-values derived as a result are shown in Table 15.

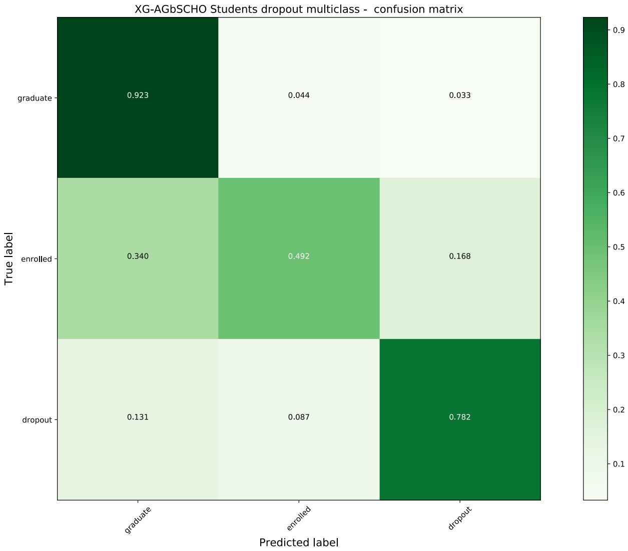


FIGURE 15. Experiment III - the best performing XGBoost model (XG-AGbSCHO) for multi-class dropout classification confusion matrix.

TABLE 15. The Shapiro-Wilk test scores for the single-problem analysis.

Algorithm	AdaBoost Binary	XGBoost Binary	XGBoost Multiclass
AGbSCHO	0.022	0.020	0.021
SCHO	0.024	0.023	0.026
FA	0.020	0.020	0.019
GA	0.018	0.022	0.033
PSO	0.026	0.029	0.032
VNS	0.030	0.019	0.023
RSA	0.033	0.022	0.030
EHO	0.020	0.021	0.022
COLSHADE	0.021	0.022	0.029

The null hypothesis, which states that results originate from normal distribution, is suitably rejected, since every p-value in Table 15 is less than the predetermined significance level, α, which is set to 0.05. Therefore, since not all solution data samples conform to a Gaussian distribution, using the average objective value in the upcoming statistical tests is not appropriate. Consequently, the best objective function results from the study were selected for more statistical analysis. Additionally, the conclusion that parametric testing was improper was based on the non-met normality assumption.

These findings are further supported by kernel density estimate (KDE) plots shown in Figure 16, illustrating the distributions of objective function outcomes for each optimizer across 30 independent runs for all three conducted simulations.

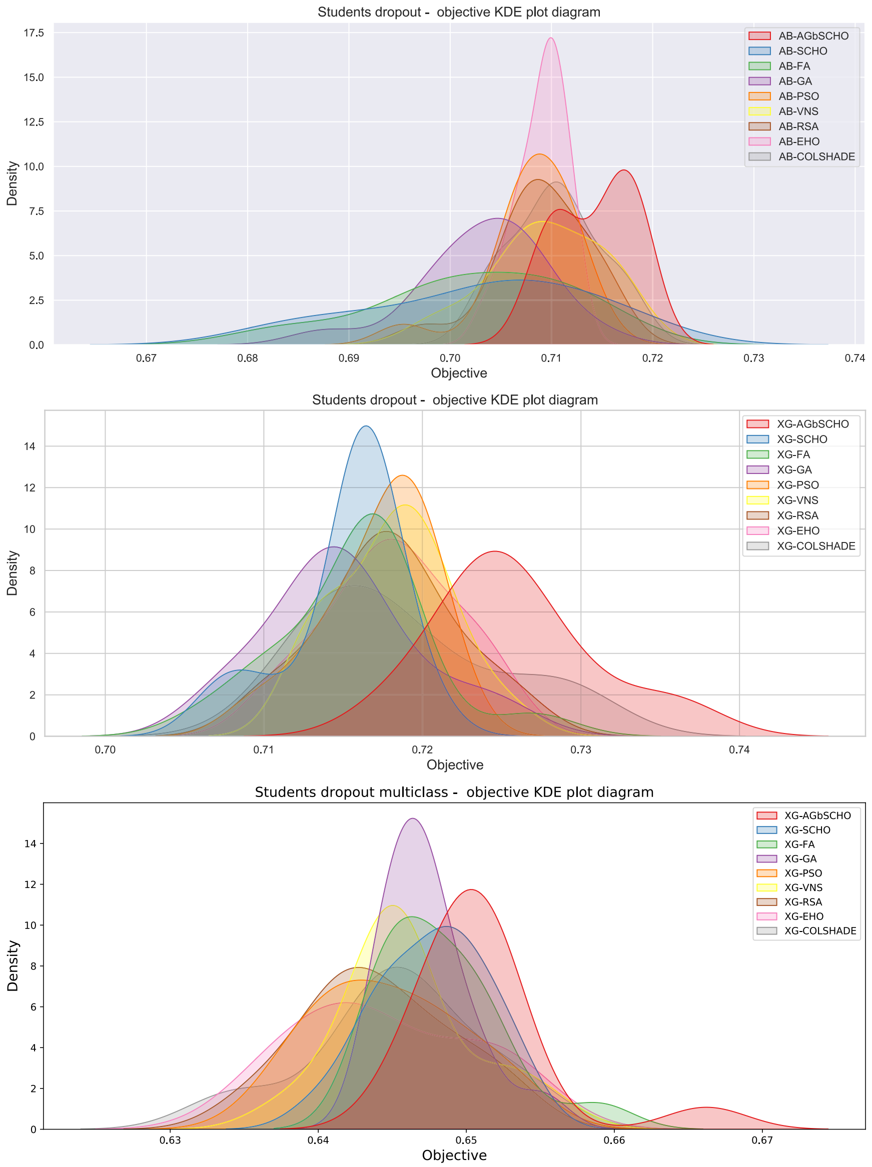


FIGURE 16. Objective function KDE plots for each experiment.

Subsequently, the Wilcoxon signed-rank test [51] was employed as a non-parametric method. Initially, the control algorithm, AGbSCHO in this case, was selected for comparison with all others based on the best objective function result in each run. The objective was to determine whether the improvements achieved by the proposed method were statistically significant. The computed p-values for each of the three experiments conducted were all below the threshold of 0.05, when AGbSCHO was compared to each other metaheuristic. These results indicate that the innovative algorithm significantly outperformed all others at the significance level of 0.05 ($\alpha = 0.05$). Table 16 presents the findings from the Wilcoxon signed-rank test.

The Wilcoxon signed-rank test revealed that the developed technique (AGbSCHO) outperformed the other algorithms statistically significantly under all three test cases. The p-values for each comparison were less than 0.05, which is far less than the other commonly used significance level of 0.1. As a conclusion, the consistency across multiple datasets demonstrates that the AGbSCHO approach consistently outperforms the other algorithms examined in this study.

TABLE 16. Outcomes of Wilcoxon signed-rank test (AGbSCHO vs. others).

Algorithm	AdaBoost Binary	XGBoost Binary	XGBoost Multiclass
SCHO	0.033	0.031	0.032
FA	0.016	0.016	0.017
GA	0.030	0.010	0.001
PSO	0.042	0.032	0.036
VNS	0.030	0.020	0.029
RSA	0.033	0.023	0.023
EHO	0.027	0.022	0.035
COLSHADE	0.030	0.033	0.031

B. BEST MODEL INTERPRETATION

The rationale behind decisions is often as crucial as the decisions themselves. This is particularly true in higher education settings, where understanding why a student may be struggling or choosing to leave academia is paramount. A fundamental understanding of the factors influencing these decisions is vital for addressing challenges promptly and improving outcomes.

Therefore, interpreting the models of student dropout is crucial for gaining deeper insights into students' behavior. There are several techniques for interpreting ML model decisions, while in recent times the application of SHAP [9] and SAGE [10] techniques has become popular among researchers due to their efficiency and humble computing resources utilization. These techniques are game theory inspired approaches, treating models as competitors in a simulated game. This not only enables accounting for the importance of individual features but also enhances understanding of their interactions at both local and global scales.

Based on experimental findings for binary classification simulations, the best results were captured by the XG-AGbSCHO, with achieved accuracy of 88.86% and Cohen Kappa score of 0.7368. Consequently, this model was employed for binary classification interpretation. The best binary dropout detection model SHAP analysis outcomes are provided in Figure 17, followed by SAGE analysis shown in Figure 18. The SHAP analysis indicates that on a local level, the number of curricular units in the second semester alongside tuition fees and curricular units in the first semester play the highest role in student dropout. Tuition fees being up to date are also a high-impact feature on model decisions. Fees not being up to date can hint at student' outlook towards their education, as well as financial struggles. Students struggling with feelings might be forced to take part-time work and thus have less time to dedicate to working on their degree. Age of enrolment is also a contributing factor. Older students often have more personal responsibilities in their lives but can also feel more pressure to complete their studies on time. Furthermore, it's also important to note that not all features are actionable. While these contribute to a model's decision little to nothing can be done to change these factors by both the students as well as institutions.

The SAGE analysis outcomes are provided in Figure 18. Global importance provides valuable insight into the features of the effect entire student body. Insight in this regard provides information on actionable factors that can influence the whole student body. Tuition fees and 2nd curriculum units have the highest importance. These indicate overall ability to tackle academic responsibilities as well as overall economic well-being. Student outlook on education also plays an important role, as it hits towards a student's devotion towards education. Certain factors that are not actionable also play a factor, such as the mother's occupation and age of enrolment. These factors cannot directly be addressed by students and educational institutions but might be affected by longer-term policy. It is vital to consider expert opinions when considering policy. While certain factors might be important to a model, such impacts can also hint at biases within the training data. To detect and understand these experts in the field of higher education should be consulted.

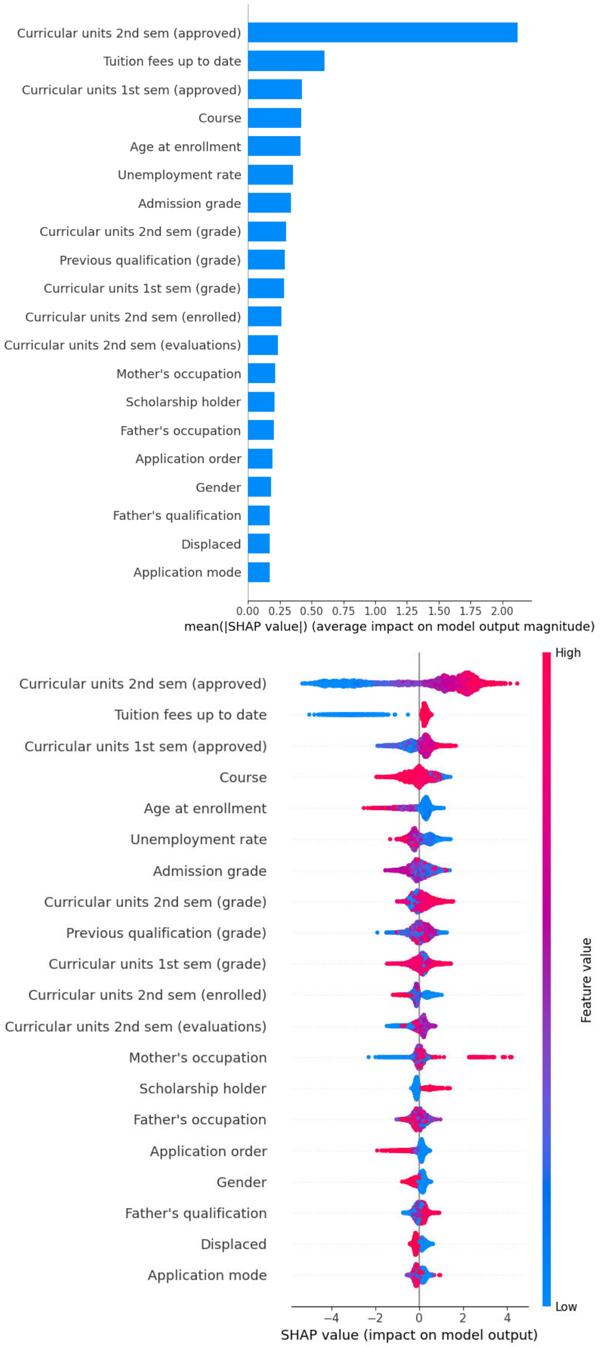


FIGURE 17. The best binary classifier SHAP bar and swarm impact plots for highest impact features.

Since the multi-class student dropout classification was conducted by using only tuned XGBoost models, the best model with accuracy of 88.0% and Cohen Kappa score of 0.6662, which was generated by XG-AGbSCHO, was employed for interpretation purposes. Multi-class classification interpretations provides an insight for each of

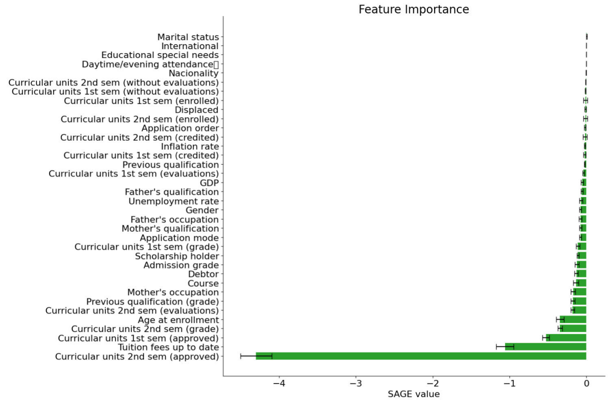


FIGURE 18. The best binary classifier SAGE feature impact plot.

the three classes in Figure 19. This allows for understanding what are the key factors that determine a graduate student from a dropout and addresses factors accordingly. Factors such as the number of approved curriculum units per semester have a high impact on students who are still enrolled and who have graduated. However, this feature shows little importance for students who are at risk of dropping out. Units under evaluation have a high impact on dropout detection, meaning that students who have not completed courses have a high chance of leaving academia. Tuition fees play a less significant role in dropout in multi-class classification. Employment rate plays an important role in the differentiation between graduating and current students. Scholarships also have a high impact on student graduation.

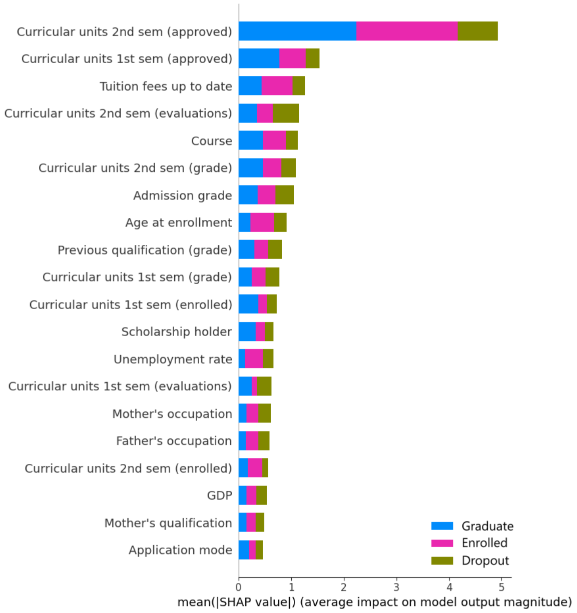


FIGURE 19. The best multi-class classifier SHAP bar plots for highest impact features.

While model interpretation outcomes can be usefully for improving outcomes ad refining models, policy makers need to be cautious at taking interpretations at face value. Other factors not accounted for in the available dataset play a role in decisions. Additionally, different models may place different weight on different factors and each interpretation is specific to the given model and needs to be treated as such.

C. PROPOSED APPROACH IMPLEMENTATION STRATEGY

Student retention is an essential part of college operations. Improving overall outcomes and institutional relations is at the core of academic success. However, various factors influence student decisions to abandon education. Economic, cultural, and personal reasons can lead students to leave school, and certain factors are outside the sphere of influence of an institution. Nevertheless, offering better study plans or payment plans to students at risk of being overwhelmed can improve student outcomes and foster student loyalty towards an institution. Continuous monitoring can help reduce student loss through timely detection and the data drive approach would allow models to adjust to the changing situations without explicit programming.

Identifying students at risk is the first step toward addressing dropout. Depending on the information available to an institution, an adapted model can be trained to detect students at risk of leaving academia. This model can then be optimized using a proposed modified optimizer to refine detection capabilities. Once a sufficiently accurate model is formulated, this classifier can be analyzed further to identify factors that influence student dropout on both a global and individual level.

Evaluations using explainable AI techniques have a twofold contribution. Firstly, major factors influencing dropout are identified. Global interpretations can help institutions implement institution-wide policies that improve overall retention. Local interpretations can be leveraged to identify the factors that influence individual students' decisions, helping formulate customized strategies with the guidance of experts and counselors, especially for high-risk students. This may include providing better payment plans, reduced workload, or more flexible hours to help working students attend lectures more often.

However, it is important to note that expert opinions are essential when implementing such a system. Each institution, administration, and student is different, and different cultures, regions, and education systems have their own specific needs and expectations. Therefore, consulting experts is crucial for proper policy implementation.

VII. CONCLUSION

This work explored the increasingly pressing challenge of student retention in higher education facilities through a data-driven approach. Detecting students dropout is a challenging task with many internal, external, student-specific factors, and personal motivations affecting student choices. Even with the severe consequences for individuals, groups, and society at large, figuring out and treating these causes is still a highly challenging task. Given the importance of identifying nuanced explanations for dropout decision reasons that faculty members may overlook, this research explored a variety of variables, such as personal issues, financial difficulties, social integration, and academic integration. Complicating matters are challenges with re-enrollment, exam recognition, and returning to academic work after a vacation.

The goal of the work was to use ML approaches tuned by metaheuristics to enhance the prediction performance for students dropout detection. Proposed research tried to fill the noticeable gap in the application of ML for predicting student dropout. Specifically, the gap in the area of applications of simpler, less computationally demanding ML models for student dropout prediction. Moreover, prior to this study, the optimization of hyper-parameters for these models in this context has not been thoroughly explored.

The proposed study examined two established classifiers, AdaBoost and XGBoost models. Given the significant impact of hyper-parameter selection on classifier performance, a modified version of the SCHO metaheuristics was introduced specifically for this study. The adapted algorithm aimed to address limitations observed in the baseline approach. Optimized models were evaluated using a publicly available dataset, demonstrating promising results with an accuracy exceeding 88.86% for binary student dropout classification and a Cohen's Kappa score of 0.7368. Similarly, for multi-class student dropout identification, the models achieved an accuracy of 88.0% with a Cohen's Kappa of 0.6662. These robust performances validate the proposed methodology's potential applicability in real-world higher education environment.

Following rigorous statistical validation, the best models were subjected to SHAP and SAGE analysis in order to highlight the key factors affecting the academic career of students and to provide further insight. Ultimately, by using the explainable AI techniques, this research advances the application of AI in higher education by giving organizations and decision-makers essential data to develop targeted student retention strategies. The findings have the potential to enhance the general effectiveness and performance of higher education institutions while fostering a more supportive and resilient learning environment. In this way, proposed study findings may have significant contributions in preventing students dropout in practice.

However, several limitations, mostly practical, affecting the scalability and efficiency of the proposed approach were identified. A primary constraint was the limited computational power available, which, combined with the high computational demands of optimization, may have restricted the performance and scalability of the proposed methodology. Furthermore, the study was constrained to a specific set of optimizers, potentially limiting the generalizability of the findings. It is also crucial to consider the influence of domain experts in interpreting results, as various factors may vary in impact across different conditions, locations, and socioeconomic environments. Additionally, evolving privacy policies and data regulations could present challenges for the practical implementation of the proposed methodology.

Future research will focus on discovering new applications for the introduced optimizer, aiming to broaden its utility beyond its current scope. This exploration may reveal additional benefits and applications, thereby advancing this field further. By addressing current limitations and pursuing these future directions, significant contributions can be made towards direction of refining the effectiveness and applicability of AI-driven solutions for addressing student dropout. These advancements are poised to profoundly impact higher education, enhancing decision-makers' understanding and capabilities in tackling student retention challenges.

ACKNOWLEDGMENT

This research was supported by the Science Fund of the Republic of Serbia, Grant No. 7373, Characterizing crises-caused air pollution alternations using an artificial intelligence-based framework—crAIRsis and Grant No. 7502, Intelligent Multi-Agent Control and Optimization applied to Green Buildings and Environmental Monitoring Drone Swarms—ECOSwarm.

REFERENCES

[1] A. Behr, M. Giese, H. D. Teguim Kamdjou, and K. Theune, "Motives for dropping out from higher education—An analysis of bachelor's degree students in Germany," *Eur. J. Educ.*, vol. 56, no. 2, pp. 325–343, Jun. 2021.

[2] A. B. Yılmaz and S. Karataş, "Why do open and distance education students drop out? Views from various stakeholders," *Int. J. Educ. Technol. Higher Educ.*, vol. 19, no. 1, pp. 1–22, Dec. 2022.

[3] A. F. Núñez-Naranjo, M. Ayala-Chauvin, and G. Riba-Sanmartí, "Prediction of university dropout using machine learning," in *Proc. Int. Conf. Inf. Technol. Syst.* Cham, Switzerland: Springer, Jan. 2021, pp. 396–406.

[4] S. Wild, S. Rahn, and T. Meyer, "Factors mitigating the decline of motivation during the first academic year: A latent change score analysis," *Motivat. Emotion*, vol. 48, no. 1, pp. 36–50, Feb. 2024.

[5] O. Lorenzo-Quiles, S. Galdón-López, and A. Lendínez-Turón, "Factors contributing to university dropout: A review," *Frontiers Educ.*, vol. 8, Mar. 2023, Art. no. 1159864.

[6] Z. A. Long and M. F. M. Noor, "Factors influencing dropout students in higher education," *Educ. Res. Int.*, vol. 2023, pp. 1–13, Feb. 2023.

[7] E. Alyahyan and D. Düştegör, "Predicting academic success in higher education: Literature review and best practices," *Int. J. Educ. Technol. Higher Educ.*, vol. 17, no. 1, pp. 1–21, Dec. 2020.

[8] J. Bai, Y. Li, M. Zheng, S. Khatir, B. Benaissa, L. Abualigah, and M. A. Wahab, "A sinh cosh optimizer," *Knowl.-Based Syst.*, vol. 282, Dec. 2023, Art. no. 111081.

[9] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Red Hook, NY, USA: Curran Associates, 2017, pp. 4765–4774.

[10] I. Covert, S. M. Lundberg, and S.-I. Lee, "Understanding global feature contributions with additive importance measures," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 17212–17223.

[11] F. Del Bonifro, M. Gabbrielli, G. Lisanti, and S. P. Zingaro, "Student dropout prediction," in *Proc. Int. Conf. Artif. Intell. Educ.* Cham, Switzerland: Springer, 2020, pp. 129–140.

[12] D. Cornell, A. Gregory, F. Huang, and X. Fan, "Perceived prevalence of teasing and bullying predicts high school dropout rates," *J. Educ. Psychol.*, vol. 105, no. 1, pp. 138–149, Feb. 2013.

[13] H. M. Abdulghani, K. Alanazi, R. Alotaibi, N. A. Alsubeeh, T. Ahmad, and S. Haque, "Prevalence of potential dropout thoughts and their influential factors among Saudi medical students," *SAGE Open*, vol. 13, no. 1, Jan. 2023, Art. no. 215824402211469.

[14] R. Banaag, J. L. Sumodevilla, and J. Potane, "Factors affecting student drop out behavior: A systematic review," *Int. J. Educ. Manage. Innov.*, vol. 5, no. 1, pp. 53–70, Jan. 2024.

[15] H. A. Mengash, "Using data mining techniques to predict student performance to support decision making in university admission systems," *IEEE Access*, vol. 8, pp. 55462–55470, 2020.

[16] L. Kemper, G. Vorhoff, and B. U. Wigger, "Predicting student dropout: A machine learning approach," *Eur. J. Higher Educ.*, vol. 10, no. 1, pp. 28–47, Jan. 2020.

[17] F. Qiu, G. Zhang, X. Sheng, L. Jiang, L. Zhu, Q. Xiang, B. Jiang, and P.-K. Chen, "Predicting students' performance in e-learning using learning process and behaviour data," *Sci. Rep.*, vol. 12, no. 1, p. 453, Jan. 2022.

[18] S. Dass, K. Gary, and J. Cunningham, "Predicting student dropout in self-paced MOOC course using random forest model," *Information*, vol. 12, no. 11, p. 476, Nov. 2021.

[19] S. R. Sihare, "Student dropout analysis in higher education and retention by artificial intelligence and machine learning," *Social Netw. Comput. Sci.*, vol. 5, no. 2, p. 202, Jan. 2024.

[20] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, and T. Zhou, "XGBoost: Extreme gradient boosting," *R Package Version*, vol. 1, no. 4, pp. 1–4, 2015.

[21] T. Zhang, H. Liu, J. Tao, Y. Wang, M. Yu, H. Chen, and G. Yu, "Enhancing dropout prediction in distributed educational data using learning pattern awareness: A federated learning approach," *Mathematics*, vol. 11, no. 24, p. 4977, Dec. 2023.

[22] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class AdaBoost," *Statist. Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[23] P. Perchinunno, M. Bilancia, and D. Vitale, "A statistical analysis of factors affecting higher education dropouts," *Social Indicators Res.*, vol. 156, nos. 2–3, pp. 341–362, Aug. 2021.

[24] M. Nagy and R. Molontay, "Predicting dropout in higher education based on secondary school performance," in *Proc. IEEE 22nd Int. Conf. Intell. Eng. Syst. (INES)*, Jun. 2018, pp. 389–394.

[25] C. L. R. Velasco, E. G. Villena, J. B. Ballester, F. Á. D. Prados, E. S. Alvarado, and J. C. Álvarez, "Forecasting of post-graduate students' late dropout based on the optimal probability threshold adjustment technique for imbalanced data," *Int. J. Emerg. Technol. Learn. (iJET)*, vol. 18, no. 4, pp. 120–155, Feb. 2023.

[26] N. M. Alruwais, "Deep FM-based predictive model for student dropout in online classes," *IEEE Access*, vol. 11, pp. 96954–96970, 2023.

[27] K.-D. Lu and Z.-G. Wu, "Genetic algorithm-based cumulative sum method for jamming attack detection of cyber-physical power systems," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.

[28] K.-D. Lu, Z.-G. Wu, and T. Huang, "Differential evolution-based three stage dynamic cyber-attack of cyber-physical power systems," *IEEE/ASME Trans. Mechatronics*, vol. 28, no. 2, pp. 1137–1148, Apr. 2023.

[29] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[30] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *Int. J. Bio-Inspired Comput.*, vol. 2, no. 2, p. 78, 2010.

[31] S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks* (Studies in Computational Intelligence), vol. 780. Cham, Switzerland: Springer, 2019, doi: 10.1007/978-3-319-93025-1_4.

[32] R. E. J. Kennedy, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, Perth, WA, Australia, Sep. 1995, pp. 1942–1948.

[33] N. Mladenović and P. Hansen, "Variable neighborhood search," *Comput. Oper. Res.*, vol. 24, no. 11, pp. 1097–1100, 1997.

[34] J. Gurrola-Ramos, A. Hernàndez-Aguirre, and O. Dalmau-Cedeño, "COLSHADE for real-world single-objective constrained optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2020, pp. 1–8.

[35] R. J. Kuo, M. F. Luthfiansyah, N. A. Masruroh, and F. E. Zulvia, "Application of improved multi-objective particle swarm optimization algorithm to solve disruption for the two-stage vehicle routing problem with time windows," *Expert Syst. Appl.*, vol. 225, Sep. 2023, Art. no. 120009.

[36] N. Bacanin, V. Simic, M. Zivkovic, M. Alrasheedi, and A. Petrovic, "Cloud computing load prediction by decomposition reinforced attention long short-term memory network optimized by modified particle swarm optimization algorithm," *Ann. Oper. Res.*, 2023, doi: 10.1007/s10479-023-05745-0.

[37] L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer," *Expert Syst. Appl.*, vol. 191, Apr. 2022, Art. no. 116158.

[38] M. A. Al-Betar, M. A. Awadallah, M. S. Braik, S. Makhadmeh, and I. A. Doush, "Elk herd optimizer: A novel nature-inspired metaheuristic algorithm," *Artif. Intell. Rev.*, vol. 57, no. 3, p. 48, Feb. 2024.

[39] J. Carrasco, S. García, M. M. Rueda, S. Das, and F. Herrera, "Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review," *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100665.

[40] M. Zivkovic, L. Jovanovic, M. Ivanovic, A. Krdzic, N. Bacanin, and I. Strumberger, "Feature selection using modified sine cosine algorithm with COVID-19 dataset," in *Evolutionary Computing and Mobile Sustainable Networks: Proceedings of ICECMSN 2021*. Singapore: Springer, 2022, pp. 15–31.

[41] P. Whig, S. Kouser, A. B. Bhatia, R. R. Nadikattu, and Y. J. Alkali, "Leveraging meta-heuristics in improving health care delivery: A comprehensive overview," in *Nature-Inspired Methods for Smart Healthcare Systems and Medical Data*, A. M. Anter, M. Elhoseny, and A. D. Thakare, Eds., Cham, Switzerland: Springer, 2024, doi: 10.1007/978-3-031-45952-8_8.

[42] M. Zivkovic, L. Jovanovic, M. Ivanovic, N. Bacanin, I. Strumberger, and P. M. Joseph, "XGBoost hyperparameters tuning by fitness-dependent optimizer for network intrusion detection," in *Communication and Intelligent Systems: Proceedings of ICCIS 2021*. Singapore: Springer, 2022, pp. 947–962.

[43] L. Jovanovic, D. Jovanovic, N. Bacanin, A. Jovancai Stakic, M. Antonijevic, H. Magd, R. Thirumalaisamy, and M. Zivkovic, "Multi-step crude oil price prediction based on LSTM approach tuned by salp swarm algorithm with disputation operator," *Sustainability*, vol. 14, no. 21, p. 14616, Nov. 2022.

[44] L. Jovanovic, N. Milutinovic, M. Gajevic, J. Krstovic, T. A. Rashid, and A. Petrovic, "Sine cosine algorithm for simple recurrent neural network tuning for stock market prediction," in *Proc. 30th Telecommun. Forum (TELFOR)*, Nov. 2022, pp. 1–4.

[45] N. Bacanin, M. Zivkovic, L. Jovanovic, M. Ivanovic, and T. A. Rashid, "Training a multilayer perception for modeling stock price index predictions using modified whale optimization algorithm," in *Computational Vision and Bio-Inspired Computing: Proceedings of ICCVBIC 2021*. Singapore: Springer, 2022, pp. 415–430.

[46] L. Jovanovic, N. Bacanin, M. Zivkovic, M. Antonijevic, A. Petrovic, and T. Zivkovic, "Anomaly detection in ECG using recurrent networks optimized by modified metaheuristic algorithm," in *Proc. 31st Telecommun. Forum (TELFOR)*, Nov. 2023, pp. 1–4.

[47] S. Saravanan, R. S. Kumar, and P. Balakumar, "Binary firefly algorithm based reconfiguration for maximum power extraction under partial shading and machine learning approach for fault detection in solar PV arrays," *Appl. Soft Comput.*, vol. 154, Mar. 2024, Art. no. 111318.

[48] N. Bacanin, T. Bezdan, F. Al-Turjman, and T. A. Rashid, "Artificial flora optimization algorithm with genetically guided operators for feature selection and neural network training," *Int. J. Fuzzy Syst.*, vol. 24, no. 5, pp. 2538–2559, Jul. 2022.

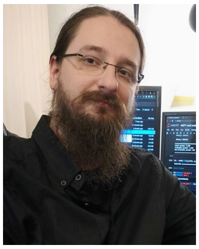
[49] T. Eftimov, P. Korošec, and B. K. Seljak, "Disadvantages of statistical comparison of stochastic optimization algorithms," in *Proc. Bioinspired Optimizaiton Methods and Their Appl. (BIOMA)*, 2016, pp. 105–118.

[50] S. S. Shapiro and R. S. Francia, "An approximate analysis of variance test for normality," *J. Amer. Stat. Assoc.*, vol. 67, no. 337, p. 215, Mar. 1972.

[51] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in Statistics*. New York, NY, USA: Springer, 1992, pp. 196–202.

RADIC GORAN received the degree from the Faculty of Electrical Engineering at Belgrade, Department of Computer Science and Information Technology, University of Belgrade.

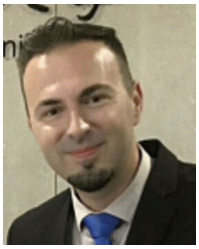
His andragogical and pedagogical skills are attributed to the teaching and other activities, he performed throughout his 20-year long academic career. At the Information Technology School—ITS, where he has been working since the founding of the school, in 2006. He actively participated in designing the academic program. Since 2010, he has been the Executive Manager of ITS. He is a member of a project development team with projects such as: LINKgroup's Distance Learning System (the first local interactive and multimedia e-Learning platform), LINKgroup's Testing and Learning Software, software solution for implementing testing, examination and training processes, an university information system, and a student service for a higher education institution.



LUKA JOVANOVIC (Member, IEEE) received the degree from the Faculty of Technical Sciences, Singidunum University. Throughout his studies, he actively contributed to various artificial intelligence research projects, resulting in over 50 peer-reviewed publications in esteemed journals and international conferences. He maintains a diverse range of research interests, covering topics, such as reinforcement learning, natural language processing, distributed computing, and signal processing. In his current capacity with the Faculty of Technical Sciences, Singidunum University, he plays a crucial role in the realm of research. His responsibilities include spearheading research initiatives, publishing scholarly articles, conceptualizing innovative ideas, and implementing experiments through programming.

NEBOJSA BACANIN received the Ph.D. degree from the Faculty of Mathematics, University of Belgrade, in 2015 (study program computer science, average grade 1000). He started his university career in Serbia 18 years ago with the Graduate School of Computer Science in Belgrade. He is currently a Full Professor, the Vice-Rector of Scientific Research, and the Head of Applied Artificial Intelligence Study Program, Singidunum University, Belgrade, Serbia. He is involved in scientific research in the field of computer science and his specialty includes stochastic optimization algorithms, swarm intelligence, soft-computing, optimization, modeling, artificial intelligence algorithms, swarm intelligence, machine learning, image processing, and cloud and distributed computing. He has published more than 380 scientific papers in high-quality journals and international conferences indexed in Clarivate Analytics JCR, Scopus, WoS, IEEEXplore, and other scientific databases. As a member of numerous editorial boards in cutting-edge international journals and committee boards of international conferences, he regularly edit and perform review activities. He has also been included in the prestigious Stanford University list of the 2% best world researchers, when the whole career is taken into account, in the field of artificial intelligence. Also, according to AD scientific index, he is currently listed as the Best Researcher from Computer Science Area in Serbia.

MILOŠ S. STANKOVIĆ received the Dipl. Ing. (M.Sc.) degree from the School of Electrical Engineering, University of Belgrade, Serbia, in 2002, and the Ph.D. degree in systems and entrepreneurial engineering from the University of Illinois at Urbana–Champaign (UIUC), USA, in 2009. From 2009 to 2012, he was a Postdoctoral Researcher with the Automatic Control Laboratory and the ACCESS Linnaeus Centre, KTH Royal Institute of Technology, Stockholm, Sweden. From 2012 to 2020, he was with the Innovation Center, School of Electrical Engineering, University of Belgrade. In 2017, he joined Singidunum University, Belgrade, Serbia, where he is currently an Associate Professor. Since 2017, he has also been with the Vlatacom Institute of High Technologies, Belgrade. His research interests include networked control systems, machine learning, dynamic game theory, optimization, and decentralized decision making, with applications to big data analytics, cyber-physical systems, and the Internet of Things. He is a member of the General Assembly of European Control Association (EUCA) and a member of Conference Editorial Board of IEEE Control Systems Society.



VLADIMIR SIMIC was born in Belgrade, Serbia, in 1983. He received a Ph.D. degree in transportation engineering from the University of Belgrade, Serbia, in 2014. Since 2020, he has been an Associate Professor of the Transport and Traffic Engineering Department at the University of Belgrade. He has conducted intensive research on operations research applications in diverse fields of specialization, with a particular focus on developing artificial intelligence systems, advanced hybrid multi-criteria decision-making tools, and real-life largescale optimization models. He published more than 240 research papers, including 180 papers in SCI-indexed International Journals (e.g., IEEE TRANSACTIONS OF FUZZY SYSTEMS, INFORMATION SCIENCES, ENGINEERING APPLICATIONS OF ARTIFICIAL INTELLIGENCE, EXPERTS SYSTEMS WITH APPLICATIONS, APPLIED SOFT COMPUTING, etc.). He regularly serves as an ad-hoc reviewer of many top-tier journals.

MILOS ANTONIJEVIC received the master's degree in engineering of organizational sciences from the Faculty of Organizational Sciences, University of Belgrade (study program E-business) and the Ph.D. degree in computer sciences (study program advanced security systems) from Singidunum University, Belgrade, Serbia.

He started his career in education 14 years ago with the High School of Graphics and Media, Belgrade. He is currently an Assistant Professor with Singidunum University and as a Certified ISO 27001 Auditor of various accreditation authorities. He is involved in scientific research in the field of computer science, with a focus on implementation of AI and optimization algorithms in various security systems.

MIODRAG ZIVKOVIC received the Ph.D. degree from the School of Electrical Engineering, University of Belgrade, in 2014. He is currently a Full Professor with the Faculty of Informatics and Computing, Singidunum University, Belgrade, Serbia. He is involved in scientific research in the field of computer science and his specialty includes artificial intelligence, machine learning, deep learning, stochastic optimization algorithms, swarm intelligence, and human–computer interaction. He has published more than 150 scientific articles in high-quality journals, articles as book chapters, and international conferences indexed in Clarivate Analytics JCR, Scopus, WoS, IEEEXplore, and other scientific databases.