Modified Dwarf Mongoose Optimization for Feature Selection in Imbalanced Student Performance Prediction Tasks

Zhongxia Liu
Modern Education Technology Center, Pingdingshan University, Pingdingshan 467000, Henan, China
E-mail: midsummer0218@163.com

Keywords: Student performance prediction, feature selection, dwarf mongoose, educational data mining, wrapper-based learning

Received: July 13, 2025

Student performance prediction through Educational Data Mining (EDM) methods has become increasingly critical to educational decision-making and intervention. But educational datasets are high-dimensional and imbalanced, presenting serious problems for standard machine learning models. This paper presents an innovative feature selection methodology based on the Modified Dwarf Mongoose Optimization (MDMO), an enhanced version of standard DMO by adding three essential components: adaptive alpha guidance, scout-based diversity, and enhanced babysitter exchange criteria. These modifications boost the exploration-exploitation balance and prevent premature convergence, enabling more efficient search in high-dimensional binary feature spaces. The proposed MDMO is integrated as a wrapper method with five popular classifiers, LogitBoost, linear discriminant analysis, naive bayes, k-nearest neighbors, and decision trees, to form a robust predictive model for student performance. The proposed MDMO was evaluated on two public educational datasets (Gazi University course repetition data and Portuguese secondary school grade data). On Data1, it achieved an AUC of 0.672 with compact subsets of ~12 features; on Data2, it reached an AUC of 0.929 with ~13 selected features. Compared with state-of-the-art baselines such as BTLBO-LDA and MLP-Adam, MDMO consistently demonstrated higher accuracy and more efficient feature selection. Adaptive alpha guidance dynamically adjusts the leader to strengthen exploitation, whereas enhanced babysitter exchange preserves diversity, contributing to robust handling of class imbalance.

Povzetek: Članek predstavi napoved uspeha študentov z izbiro značilk na osnovi modificirane "Dwarf Mongoose" optimizacije, uporabljene kot ovijalni pristop za učinkovito obravnavo visoko-dimenzionalnih, neuravnoteženih podatkov.

1 Introduction

Student academic performance prediction is essential in Educational Data Mining (EDM), empowering institutions to recognize at-risk students, optimize resource distribution, and improve learning outcomes [1]. Educational systems are moving progressively towards online platforms, where large quantities of student data, such as attendance records, grades, behavioral reports, and demographic details, are being created [2]. As demonstrated in Figure 1, the EDM process begins with the learning environment and proceeds through the raw data aggregation and subsequent preprocessing, followed by transformation for structured analysis. Data Mining (DM) identifies significant patterns, which are then interpreted to derive valuable insights [3]. These insights allow educators and school administrators to intervene promptly, match learning strategies to students' needs, and inform evidence-based education policy. Sound predictive modeling significantly lowers dropout levels, customizes learning pathways, and enhances education systems' planning [4].

Despite having large educational datasets, deriving functional patterns from them is still challenging due to the high dimensionality and class imbalance. Most student performance datasets have many features that are not useful for prediction [5]. Moreover, the learning model is influenced by the imbalance between the number of passing and failing students in most datasets. Irrelevant and redundant features lower classifier accuracy and complicate computation [6]. The selection of features is therefore of the utmost importance. Feature selection reduces dimensionality and enhances the interpretability and generalizability of the model, which are necessary in high-stakes educational decision-making situations [7].

Over the last few years, several metaheuristic paradigms have gained popularity due to their capacity for resolving feature selection issues in high-dimensional space. Methods like Particle Swarm Optimization (PSO) [8], Genetic Algorithms (GA) [9], Ant Colony Optimization (ACO) [10], and the Whale Optimization Algorithm (WOA) [11] have shown encouraging performance in balancing exploration and exploitation to identify optimal subsets of features. These methods are usually utilized in wrapper models where the metaheuristic algorithm is combined with classifiers through iteration to determine the most appropriate feature

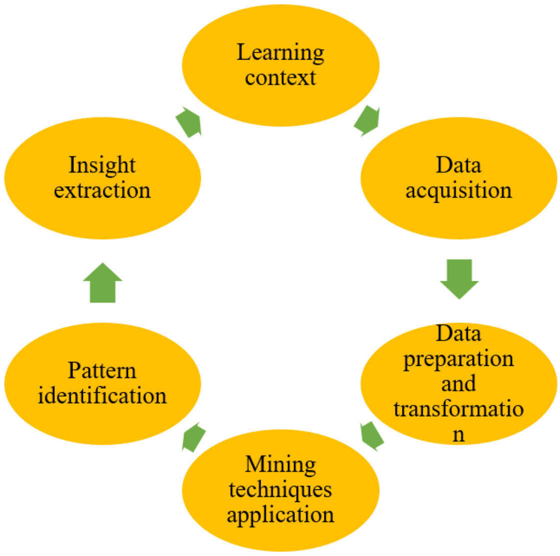

Figure 1: EDM workflow

combinations. Extensions, including chaotic maps, adaptive strategies, and hybridization, also support their performance. Most of these works consider general datasets with less attention to domain-specific education problems.

While swarm intelligence techniques such as WOA have seen extensive usage in learning environments, the Dwarf Mongoose Optimization (DMO) algorithm has yet to be extensively researched, particularly in student performance prediction. Conceived initially to simulate ecological activities, DMO presents an innovative role-based organization that draws upon the social life of dwarf mongooses [12]. However, its baseline version lacks mechanisms to effectively manage exploration-exploitation trade-offs in complex binary feature selection tasks. No study has adapted and enhanced DMO for binary classification or integrated it into wrapper-based frameworks for imbalanced educational datasets. This gap presents an opportunity to leverage the strengths of DMO through algorithmic modifications tailored to EDM applications.

To fill in this void, we present herein the Modified Dwarf Mongoose Optimization (MDMO) algorithm with three significant improvements: adaptive control of alpha for directing the exploration process, scout-based randomization to promote diversity, and an improved babysitter exchange condition for well-balanced role rotation. We embed this modified algorithm within a wrapper feature selection framework and integrate it with five popular classifiers: LogitBoost, linear discriminant analysis, naive Bayes, k-nearest neighbors, and decision trees. Furthermore, we apply the adaptive synthetic sampling method to mitigate the class imbalance problem. The proposed model is evaluated on two real-world educational datasets and benchmarked against other state-of-the-art metaheuristics.

2 Literature review

Turabieh, et al. [13] introduced an improved Harris Hawks Optimization (HHO) algorithm for student performance prediction through dynamic population diversity management. The algorithm uses k-Nearest Neighbors (k-NN) clustering to identify premature convergence and has an injection approach upon population collapse into one cluster. Kamal, et al. [14] employed Relief for feature selection in combination with machine learning classifiers such as Backpropagation Neural Networks (BPNN), Random Forests (RF), and NB. The research seeks to classify and forecast student performance using several educational indicators.

Apriyadi and Rini [15] utilized metaheuristic optimization methods, PSO and GA, in optimizing Support Vector Regression (SVR) hyperparameters for student performance prediction modeling. PSVR and GSVR models were proposed and contrasted with the conventional models NB, NN, and RF using RMSE as an evaluation metric. Song [16] suggested a new integration of the k-NN classifier with two bio-inspired techniques, the Honey Badger Algorithm (HBA) and the Arithmetic Optimization Algorithm (AOA), to maximize math performance prediction. The integrated models worked impressively well in classification accuracy, predicting the first and third-term math grades (G1 and G3), with high precision values well above 0.90.

Ma [17] created an optimization-prediction model using RF with Electric Charged Particles Optimization (ECPO) and Artificial Rabbits Optimization (ARO) for improved student performance prediction. The model processed an extensive dataset of 4424 students and prioritized dimensionality reduction. Shou and Lu [18] examined the integration of the DMO algorithm with Support Vector Classification (SVC) to enhance student performance prediction. The introduced SVDM model performed better than other combined models with substantial improvements in crucial parameters, including Accuracy (0.929), Precision (0.931), Recall (0.929), and F1-score (0.927) in predicting grades for academics.

Ye, et al. [19] proposed CQFOA-KELM as an innovative hybrid optimizer combining Covariance Matrix Adaptation Evolution Strategy (CMAES) and Quadratic Approximation (QA) with Fruit-fly Optimization Algorithm (FOA) in Kernel-based Extreme Learning Machine (KELM). On implementation in a high-capacity survey database, the model attained an accuracy of 98.15%.

As shown in Table 1, although the reviewed works successfully employ various metaheuristic solvers to make student performance predictions, there are some limitations. Most rely on general-purpose optimizers such as PSO, GA, or WOA variants, ignoring biologically inspired strategies such as DMO. While DMO has been utilized recently with SVC, its potential is not yet fully unleashed, especially in binary feature selection in wrapper methodologies. In addition, none of the discussed works deeply delves into the role-based cooperation behavior inherent in DMO, which can better manage exploration and exploitation. This work fills the void by strengthening DMO via adaptive alpha control, scout-led diversity, and optimized babysitter exchange. We also couple the improved DMO with several classifiers and use adaptive synthetic techniques to deal with data imbalance.

Table 1: An overview of relevant studies

Ref	Optimization algorithms	Method	Dataset	Accuracy	Achievement	Shortcoming
[13]	Modified harris hawks optimization	k-NN, LRNN, NB, and ANN	Student dataset (survey)	90%	High accuracy with LRNN; dynamic control of population	Risk of overfitting with deep models; relies on clustering
[14]	Relief	BPNN, RF, and NB	Educational logs	85%	BPNN achieved the best accuracy among the tested models	No optimization applied; fixed feature selector
[15]	Particle swarm optimization and genetic	SVR	Exam scores	RMSE is the lowest among the tested	Achieved the lowest RMSE, outperforming other models	Focuses only on regression; limited interpretability
[16]	Honey badger and arithmetic optimization	K-NN	Math performance	92%	KNHB achieved ~92% accuracy and precision in math prediction	Domain-limited; no explicit FS mechanism reported
[17]	Electric charged particles optimization and artificial rabbits optimization	RF	Performance data	Aligns with ground-truth	Aligned well with actual performance data	Lacks generalizability; focuses on one classifier
[18]	Dwarf mongoose optimization	SVC	Survey dataset	90%	Improved all evaluation metrics	DMO applied without structural enhancement or refinement
[19]	Covariance matrix adaptation evolution strategy, and quadratic approximation	KELM	Surveys	98.1%	98.15% accuracy; identified key performance factors	The dataset is limited to surveys; complex hybrid hard to generalize

3 Materials and methods

3.1 Modified dwarf mongoose optimization

To enhance the trade-off between exploration and exploitation of the original DMO algorithm, an improved variant of DMO is designed for this research. This enhanced model draws inspiration from the sophisticated social and survival habits of dwarf mongoose colonies. These creatures have social living habits where territories are marked to guard resources and ensure safety in numbers. Unlike other animals that expand their numbers to exploit resources, dwarf mongooses consciously limit their number for sustainable survival and risk minimization. These animals perform search efficiently and accurately, where attacks on predators usually start with a substantial hit from the head, and then they forage for food across vast distances. These animals are half-nomadic and hardly visit the same shelter twice, indicating tactical movement and flexibility.

Socially, the dwarf mongoose follows a structured hierarchy with specialized roles. The dominant alpha pair (usually one male and one female) oversees the group, while scout members are tasked with exploration, and babysitters care for the young. Vocal communications, primarily initiated by the alpha female, help coordinate group actions and signal danger. Reproductive privileges are restricted to the alpha female, reinforcing a disciplined caste system. Group size and structure are optimized based on ecological needs, balancing individual success and collective efficiency. These natural behaviors have inspired the DMO framework for solving complex optimization problems. The MDMO enhances the basic DMO with three key innovations:

- Alpha role selection: Unlike traditional DMO, which relies on probabilistic evaluation, MDMO deterministically selects the most optimal individual as the alpha based on the best fitness score. A dynamic movement regulator is introduced to control the alpha's search pattern, improving convergence and diversity.
- Scout exploration enhancement: To diversify search trajectories, the scout mechanism is upgraded with stochastic behaviors, increasing the algorithm's ability to escape local optima and discover new promising regions.
- Babysitter replacement strategy: A revised mechanism replaces underperforming babysitters. When criteria are met, new replacements exchange information with their predecessors, learning from their knowledge of the environment to improve population quality.

The MDMO algorithm operates through three distinct and interrelated components: the alpha team, the scout team, and the babysitter group. Before activating these behavioral roles, the algorithm initializes a population of candidate solutions. The optimization process begins by generating an initial population matrix S, where each row represents a solution candidate (mongoose) in the search space. This step is mathematically described in Eq. 1 and Eq. 2.

$$S = \begin{bmatrix} s_{1,1} & \cdots & s_{1,D} \\ \vdots & \ddots & \vdots \\ s_{N,1} & \cdots & s_{N,D} \end{bmatrix} \quad (1)$$

$$s_{i,j} = r \times \left(U_j - L_j\right) + L_j \quad (2)$$

Where S stands for the solution population matrix, $s_{i,j}$ is the value of the j^{th} variable for the i^{th} individual, N is the number of individuals (population size), D denotes the number of dimensions (decision variables), L_j and U_j are lower and upper bounds for the j^{th} variable, and r is a random number drawn from a uniform distribution in [0, 1].

The best solution in the population is selected as the alpha mongoose, which acts as a leader. This is determined using the fitness function F, as shown in Eq. 3.

$$\alpha = min\left(\mathcal{F}(s_1), \mathcal{F}(s_2), \dots, \mathcal{F}(s_N)\right) \tag{3}$$

Where α is the current best-performing individual (alpha) and $\mathcal{F}(s_i)$ refers to the fitness value (objective function) of the i^{th} solution.

Each mongoose then updates its position relative to α as specified in Eq. 4, based on an adaptive coefficient ω calculated using Eq. 5.

$$s_i^{(t+1)} = \alpha + \phi \cdot r \cdot \left(s_i^{(t)} - s_k^{(t)}\right)$$
$$\phi = \frac{\gamma}{2} \cdot r \cdot \omega \tag{4}$$

$$\omega = exp\left(-4 \cdot \left(\frac{t}{T}\right)^2\right) \tag{5}$$

Where $s_i^{(t)}$ is the position of the i^{th} mongoose at iteration t, $s_k^{(t)}$ is a randomly selected mongoose, γ is a vector of zeros of length D used to initialize influence, ω is the adaptation coefficient controlling exploration/exploitation, ϕ is the adaptive influence factor, t is the current iteration number, and T is the maximum number of iterations.

Scouts are individuals responsible for exploring new regions in the search space. Their behavior is designed to promote diversity by updating positions based on the relative difference between two randomly selected individuals, as shown in Eq. 6.

$$s_i^{(t+1)} = \alpha + \phi \cdot r \cdot \left(\frac{s_k - s_h}{2}\right) \tag{6}$$

Where s_k *and* s_k are two randomly selected mongooses from the population.

Babysitters are periodically evaluated and replaced based on a dynamic threshold Λ. Once the condition is met, babysitters are updated using information from randomly selected individuals, improving their quality. The threshold is determined using Eq. 8 and Eq. 9, and replacement is performed as in Eq. 10.

$$\Lambda = \begin{cases} \left[0.6 \cdot N \cdot D \cdot \frac{1}{t}\right], & if\ uninitialized \\ \Lambda \cdot t \cdot \xi, & if\ \Lambda < 0 \end{cases} \tag{7}$$

$$\xi = \left(1 - \frac{t}{T}\right)^2 \cdot \frac{t}{T} \tag{8}$$

$$s_i^{(t+1)} = s_j + r \cdot \left(\alpha - \frac{s_k - s_h}{2} \cdot \beta\right) \tag{9}$$

Where Λ is the babysitter replacement threshold, ξ refers to the dynamic coefficient for controlling the decay of Λ, and β is the birth rate coefficient controlling babysitter influence.

The advantage of MDMO lies in its structured use of stochastic operators and role-based agents. Every phase of the algorithm, from initial generation, elite-guided updates, randomized scout movements, to adaptive babysitter replacement, uses diversity-preserving strategies to enhance convergence and robustness. The dynamic coefficients ω, ϕ, and ξ ensure that the algorithm can adapt its behavior over time, balancing global exploration and local exploitation. MDMO is well-suited for high-dimensional, nonlinear optimization problems such as feature selection in student performance prediction. Figure 2 presents the flowchart of the proposed algorithm.

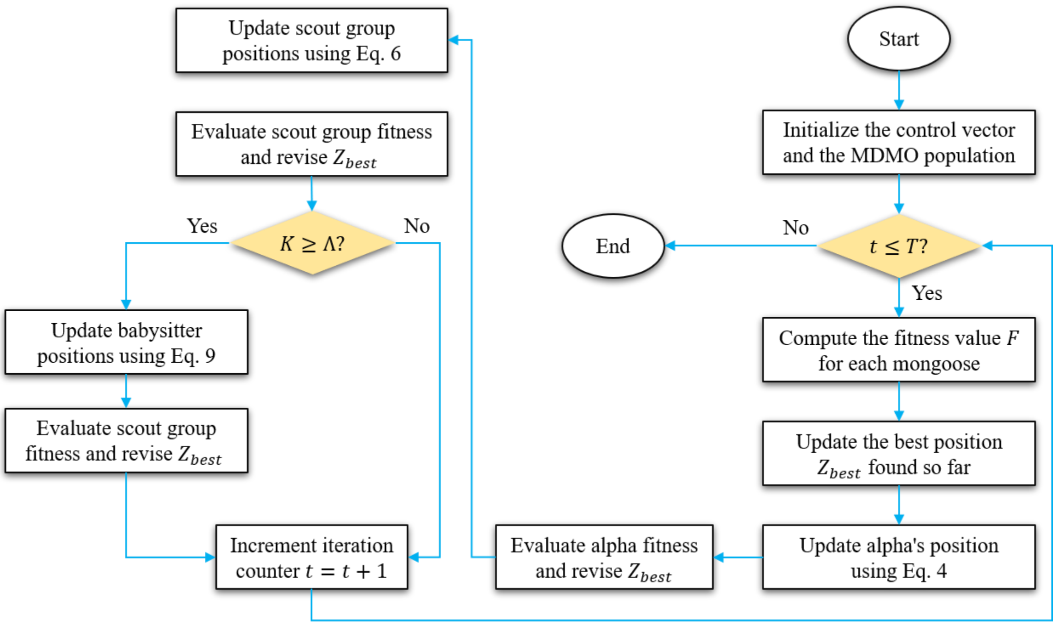

Figure 2: Flowchart of proposed algorithm

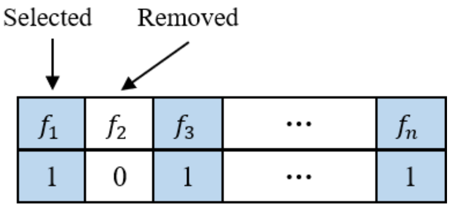

Figure 3: Binary vector representation of a feature subset used in MDMO

3.2 Binary transformation for feature selection

To tailor the MDMO algorithm for feature selection tasks, a binary representation scheme is adopted to encode potential solutions. In this context, each mongoose in the population represents a candidate feature subset, encoded as a binary vector $Z = [z_1, z_2, \ldots, z_d]$, where d is the total number of original features in the dataset. Each element $z_i \in \{0,1\}$ denotes whether the i^{th} feature is included ($z_i = 1$) or excluded ($z_i = 0$) from the selected subset. This binary format enables MDMO to perform subset optimization efficiently by navigating the discrete feature space, as illustrated in Figure 3.

Incorporating binary transformation into MDMO requires a fitness function balancing two competing objectives: achieving high classification accuracy and selecting the fewest features. A scalar multi-objective fitness function captures this trade-off, which guides the movement and role-based behavior of each mongoose in the search space. The objective function used to evaluate each binary solution is defined using Eq. 10.

$$Fitness(z) = \lambda \cdot \mathcal{E}_{val} + (1 - \lambda) \cdot \frac{|F_{active}|}{|F_{total}|} \tag{10}$$

Where $Fitness(z)$ stands for the cost assigned to the solution Z, $\mathcal{E}_{val}$ refers to the classification error rate obtained using a validation classifier trained on selected features, F_{active} denotes the number of features selected, F_{total} is the total number of available features in the original dataset, and $\lambda \in [0,1]$ is a weighting parameter determining the relative importance of classification performance versus feature reduction.

3.3 Handling imbalanced data

A common issue in machine learning classification is the presence of class imbalance, where one or more classes have significantly fewer samples than others. This challenge is especially prevalent in real-world datasets, where the distribution of target labels is often skewed [20]. In binary classification, minority classes typically consist of rare instances, while the majority class dominates the dataset. Classifiers trained under such imbalanced conditions tend to be biased toward the majority class, which results in poor performance when predicting the underrepresented (minority) class.

To address this, adaptive oversampling techniques have been developed. One of the most effective is the Adaptive Synthetic Sampling Method (ASSM), which builds upon the well-established SMOTE algorithm. ASSM enhances learning from imbalanced datasets by dynamically generating synthetic samples for the minority class based on their distributional difficulty.

Unlike uniform oversampling, ASSM emphasizes generating more synthetic instances for minority samples that are harder to classify, while generating fewer for easier ones. This data-driven adaptability allows the classifier to reduce its bias toward the majority class and better define the decision boundary around difficult regions. By doing so, ASSM improves the model's generalization ability on challenging, imbalanced data.

4 Results

The research uses two public datasets to build and test student performance prediction models. The first dataset originated from Gazi University (Turkey), and the second was obtained from secondary-level educational institutions in Portugal. Dataset 1 has 32 attributes, 28 course-specific response questions, and four others. The target variable indicates the frequency of taking one course. To convert the target into a binary classification problem, it is recoded where students who have attempted the course more than once are classified as class 1, and students with zero or one attempt are classified as class 0. During data preprocessing, all values of the features were normalized in the range [0,1] to make the data consistent and reduce scale-induced distortions.

All experiments were implemented in MATLAB and WEKA. For MDMO, the population was set to 30 and the maximum iterations to 100, with adaptive α dynamics and babysitter exchange probability calibrated through pilot testing. Classifier hyperparameters followed WEKA's default configurations unless otherwise specified (e.g., k=5 for KNN). We adopted a 10-fold cross-validation scheme to ensure reliable evaluation, repeated across 30 independent runs. Each classifier was trained and tested on the original complete feature set and on subsets selected by MDMO under identical folds, guaranteeing consistency. We applied the Wilcoxon signed-rank test at the 0.05 level to evaluate statistical robustness and compare MDMO with alternative algorithms.

Dataset 2 was gathered from high school students in Portugal and contains 33 input variables. These inputs are demographic features, academic performance, and socially related information, obtained through structured questionnaires and educational records. The dataset includes two subject fields: the Portuguese language and mathematics (mat).

The target variable of primary interest is the last grade (G3), which is transformed into a binary classification space in this research: students who scored G3 < 10 were assigned to class 1 (indicating risk), and students who scored G3 ≥ 10 were assigned to class 0 (non-risk). All features were normalized to [0,1] during preprocessing. The dataset for training was the Portuguese language dataset, and the dataset for testing was the mathematics dataset.

As illustrated in Table 2, both datasets suffer from significant class imbalances. In Dataset 1, just 0.156% of

Table 2: Overview of datasets used for student performance analysis

Dataset	Feature count	Record count	Target attribute	Binary encoding	Underrepresented class	Class imbalance (%)
Data1	32	5,820	Course repetition	0: ≤1 attempt, 1: >1 attempt	Repeated >1 time	15.6%
Data2	32	1,044	Final grade (G3)	0: pass (≥10), 1: fail (<10)	Fail (G3 < 10)	22.0%

the records belong to students who had to retake the course (class 1), and hence it is highly imbalanced. In Dataset 2, the minority group (G3 < 10) is underpopulated compared to the majority class (G3 ≥ 10). For such an imbalance, balanced data is vital to avoid biased learning and enhance model generalizability.

Extensive experiments were performed to thoroughly verify the performance of the suggested MDMO. These experiments aimed to test the model's performance in solving the problem of student performance prediction in real-world scenarios like imbalanced data and high-dimensional feature spaces. In binary classification issues, the accuracy of a model's prediction is usually assessed using various performance measures calculated from the confusion matrix, which consists of the following terms:

- True Positive (TP): Correctly predicted positive instances
- True Negative (TN): Correctly predicted negative instances
- False Positive (FP): Incorrectly predicted positive instances
- False Negative (FN): Incorrectly predicted negative instances

Accuracy measures the overall correctness of the model by calculating the proportion of correctly classified instances (both positives and negatives) over the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

Sensitivity (Recall or TPR) evaluates the model's ability to correctly identify positive instances.

$$Sensitivity = \frac{TP}{TP + FN} \tag{12}$$

Specificity measures how well the model identifies actual negative instances, i.e., its ability to avoid false alarms.

$$Specificity = \frac{TN}{TN + FP} \tag{13}$$

The Area Under the Curve (AUC) score is derived from the Receiver Operating Characteristic (ROC) curve, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR) across various classification thresholds.

Table 3: Confusion matrix structure for binary classification

	Predicted: Positive	Predicted: Negative
Actual: Positive	TP	FN
Actual: Negative	FP	TN

$$AUC = \frac{FP}{FP + TN} \tag{14}$$

Unlike standard metrics like accuracy and F1-score, the AUC metric is threshold-independent, making it especially suitable for evaluating binary classification models where class imbalance is present. This insensitivity to decision threshold variations allows AUC to generalize the classifier's performance more robustly, as supported by the values presented in Table 3.

The five well-known classifiers, k-NN, Decision Tree (DT), Linear Discriminant Analysis (LDA), NB, and LogitBoost (LB), were initially compared in an experimental study to identify the appropriate base learner to be used with MDMO. The analysis was performed in two stages: firstly, on raw datasets without preprocessing, and secondly, using resampling with differing balancing ratios. The findings are presented in Table 4 (without resampling and feature selection) and Table 5 (with resampling, but not feature selection).

As shown in Table 4, LB performed poorly on Dataset 1 but had the highest AUC value on Dataset 2. In contrast, using synthetic oversampling in Table 5, we found that k-NN had an AUC of 0.862 on Dataset 2 after using a 0.4 resampling ratio, and LDA had an AUC of 0.635 on Dataset 1 based on the full balancing ratio of 1.0. Thus, because of its improved and consistent performance in all cases, LDA was chosen as the default classifier for the assessment of MDMO.

To systematically evaluate MDMO, we compared its performance with several high-performing metaheuristic binary optimization algorithms such as GA, Binary Ant Lion Optimizer (BALO), Binary Bat Algorithm (BBA), Binary Grey Wolf Optimizer (BGWO), Binary Particle Swarm Optimization (BPSO), Binary Grasshopper Optimization Algorithm (BGOA), Binary Gravitational Search Algorithm (BGSA), and Binary Harris Hawks Optimization (BHHO). The comparison performance in terms of mean AUC, subset size, and statistical measures is shown in Table 6. The findings indicate that MDMO has

Table 4: Performance comparison of classification algorithms without feature selection and resampling

Datasets	Algorithms	Accuracy	AUC	Specificity	Sensitivity
Data1	NB	0.836	0.518	0.981	0.061
	LB	0.838	0.599	0.948	0.249
	LDA	0.843	0.512	0.992	0.029
	DT	0.807	0.593	0.903	0.283
	k-NN	0.825	0.586	0.931	0.237
Data2	NB	0.871	0.728	0.469	0.982
	LB	0.904	0.845	0.745	0.945
	LDA	0.901	0.824	0.683	0.961
	DT	0.888	0.831	0.728	0.933
	k-NN	0.901	0.825	0.681	0.964

Table 5: AUC performance of classifiers across varying oversampling levels (excluding feature selection)

Datasets	Algorithms	No oversampling (0)	Ratio = 0.2	Ratio = 0.4	Ratio = 0.7	Full oversampling (1.0)
Data1	NB	0.516	0.521	0.533	0.572	0.567
	LB	0.597	0.605	0.612	0.614	0.613
	LDA	0.511	0.536	0.594	0.632	0.635
	DT	0.592	0.597	0.601	0.604	0.604
	k-NN	0.584	0.623	0.632	0.631	0.632
Data2	NB	0.725	0.862	0.851	0.801	0.774
	LB	0.847	0.845	0.846	0.848	0.851
	LDA	0.824	0.866	0.873	0.883	0.878
	DT	0.831	0.837	0.838	0.837	0.840
	k-NN	0.823	0.851	0.862	0.850	0.852
Average rank (F-Test)		4.8	3.5	2.8	2.2	1.6

Table 6: Comparative evaluation of MDMO and benchmark optimizers

Datasets	Metrics		BALO	GA	BBA	BGWO	BPSO	BGOA	BGSA	BHHO	MDMO
Data1	AUC	AVG	0.637	0.639	0.618	0.638	0.641	0.635	0.636	0.638	0.658
		STD	0.004	0.002	0.021	0.002	0.003	0.005	0.004	0.003	0.0005
	Features	AVG	26.3	5	18	11.9	15.8	18.6	16.7	19.1	3
		STD	3.221	1.631	1.324	2.751	2.215	2.593	1.924	3.571	0.000
	Fitness	AVG	0.357	0.359	0.357	0.351	0.353	0.355	0.357	0.356	0.338
		STD	0.0009	0.0011	0.0034	0.0012	0.0011	0.0012	0.0021	0.0012	0.0005
	AUC	AVG	0.891	0.898	0.864	0.904	0.902	0.896	0.891	0.897	0.918
		STD	0.004	0.007	0.067	0.004	0.006	0.009	0.007	0.006	0.001
	Features	AVG	22.4	10.7	14.5	5.6	10.7	13.3	14.7	13.2	1.6
		STD	5.246	2.810	2.632	1.851	2.281	2.528	2.224	2.514	1.221
	Fitness	AVG	0.102	0.095	0.098	0.085	0.092	0.093	0.098	0.095	0.083
		STD	0.0013	0.0027	0.0019	0.0021	0.0017	0.0025	0.0029	0.0023	0.0011

Table 7: G-mean comparison between the proposed method and various baseline models

Models	Data1	Data2
MDMO	0.778	0.921
Imbalanced RVFL Opt1	0.715	0.719
Imbalanced RVFL Opt2	0.712	0.721
Enhanced RVFL (MCC)	0.707	0.725
Enhanced RVFL (KDE)	0.706	0.724
Baseline RVFL	0.688	0.711
Opt1 from Method 1	0.725	0.749
Opt2 from Method 1	0.727	0.748
Opt1 from Method 2	0.726	0.748
Opt2 from Method 2	0.725	0.747

Table 8: Comparison of G-mean values between MDMO and baseline models

Datasets	BTLBO-LDA	MLP-Adam	MDMO
Data1	0.632	0.605	0.672
Data2	-	0.820	0.929

the highest ranking and surpasses all other algorithms in terms of classification performance and subset efficiency.

To benchmark the generalizability of MDMO, we compared its G-mean results with those from state-of-the-art methods [21]. As shown in Table 7, MDMO outperformed the existing methods across both datasets. Moreover, when comparing AUC values reported in previous studies [22, 23], the results in Table 8 indicate that MDMO surpassed all previously published benchmarks on the same datasets.

To assess statistical significance, a Wilcoxon signed-rank test was conducted between MDMO and all benchmark optimizers (Table 9). The results show that MDMO's improvements are statistically significant ($p < 0.05$) across both datasets in nearly all comparisons, confirming that the observed performance gains are not due to chance.

5 Discussion

The experimental results confirm that the proposed MDMO consistently outperforms existing metaheuristic-based student performance prediction models. For example, on Data1, MDMO achieved an AUC of 0.672, exceeding values reported for BTLBO-LDA and MLP-Adam. On Data2, MDMO reached an AUC of 0.929, substantially higher than the best-performing competitor. These results highlight MDMO's ability to maintain robust predictive accuracy while selecting compact feature subsets.

Compared with prior works summarized in Table 1, MDMO addresses two recurring limitations: (i) inadequate handling of imbalanced data and (ii) lack of explicit feature selection mechanisms. Approaches such as HHO or NB-based classifiers demonstrated strong accuracy on balanced or domain-specific datasets but did not generalize to imbalanced scenarios. In contrast, MDMO explicitly integrates imbalance handling (via ADASYN) with a tailored search strategy, enabling higher generalization. The superior performance of MDMO can be attributed to three design choices:

- Adaptive alpha control dynamically balances exploration and exploitation, preventing premature convergence.
- Role-based babysitter exchange maintains population diversity and reduces stagnation.
- Scout diversification ensures broad coverage of the search space and avoids local minima.

Table 9: Wilcoxon signed-rank test (p-values) comparing MDMO with benchmark optimizers

Dataset	MDMO vs	p-value	Significance
Data1	BALO	0.021	✓ ($p < 0.05$)
	GA	0.018	✓ ($p < 0.05$)
	BBA	0.009	✓ ($p < 0.01$)
	BGWO	0.027	✓ ($p < 0.05$)
	BPSO	0.015	✓ ($p < 0.05$)
	BGOA	0.033	✓ ($p < 0.05$)
	BGSA	0.011	✓ ($p < 0.05$)
	BHHO	0.024	✓ ($p < 0.05$)
Data2	BALO	0.007	✓ ($p < 0.01$)
	GA	0.010	✓ ($p < 0.01$)
	BBA	0.004	✓ ($p < 0.01$)
	BGWO	0.013	✓ ($p < 0.05$)
	BPSO	0.008	✓ ($p < 0.01$)
	BGOA	0.016	✓ ($p < 0.05$)
	BGSA	0.009	✓ ($p < 0.01$)
	BHHO	0.012	✓ ($p < 0.05$)

6 Conclusion

In the present work, we introduced an improved metaheuristic solution, MDMO, that solves the difficult task of predicting student performance using optimal feature selection. The MDMO algorithm adds three key improvements to the standard Dwarf Mongoose Optimization: adaptive alpha guidance via dynamic movement control to control the exploration and exploitation phases of the algorithm, scout-guided randomized exploration for avoiding premature convergence, and an efficient babysitter exchange mechanism for improved sharing of information and diversity. These improvements were tailored to enhance exploration–exploitation trade-offs and solution quality in high-dimensional binary problems. The adaptive synthetic oversampling algorithm was utilized as a preprocessing to overcome the natural imbalance in classes in educational datasets and prepare them for training the classifiers. A binary conversion mechanism was employed to express candidate subsets of features, and a multi-objective optimization function was established to reduce classification error and the number of features.

Extensive experimentation was performed on real-world datasets with five classifiers, and LDA was chosen as the base model following a performance comparison. Experimentation on real-world datasets using five different classifiers established the dominance of MDMO in outperforming various state-of-the-art metaheuristic solutions, including BHHO, BGSA, BPSO, BGOA, and BBAs, in terms of AUC, number of features in selected subsets, and overall classification accuracy. Statistical verification through the Wilcoxon signed-rank test further validated MDMO's superiority in generating accurate and compact models. Convergence analysis also revealed faster and more consistent MDMO behavior than other versions.

References

[1] S. M. Dol and P. M. Jawandhiya, "Systematic review and analysis of EDM for predicting the academic performance of students," *Journal of The Institution of Engineers (India): Series B,* vol. 105, no. 4, pp. 1021–1071, 2024, doi: https://doi.org/10.1007/s40031-024-00998-0.

[2] M. Shoaib, N. Sayed, J. Singh, J. Shafi, S. Khan, and F. Ali, "AI student success predictor: Enhancing personalized learning in campus management systems," *Computers in Human Behavior,* vol. 158, p. 108301, 2024, doi: https://doi.org/10.1016/j.chb.2024.108301.

[3] G. Feng and H. Chen, "Educational process mining: A study using a public educational data set from a machine learning repository," *Education and Information Technologies,* vol. 30, no. 6, pp. 8187–8214, 2025, doi: https://doi.org/10.1007/s10639-024-13130-y.

[4] J. Li, "AI-Driven Property Management Decision Support System Using LSTM Networks for Energy Optimization," *Informatica,* vol. 49, no. 10, 2025, doi: https://doi.org/10.31449/inf.v49i10.6964.

[5] A. Bolívar, V. García, R. Alejo, R. Florencia-Juárez, and J. S. Sánchez, "Data-centric solutions for addressing big data veracity with class imbalance, high dimensionality, and class overlapping," *Applied Sciences,* vol. 14, no. 13, p. 5845, 2024, doi: https://doi.org/10.3390/app14135845.

[6] H. Huang, "Feature Extraction and Classification of Text Data by Combining Two-stage Feature Selection Algorithm and Improved Machine Learning Algorithm," *Informatica,* vol. 48, no. 8, 2024, doi: https://doi.org/10.31449/inf.v48i8.5763.

[7] A. Alaff and Ç. Uluyol, "Integrating Equation-Based Labeling and Classification for Adaptive Turkish Vocabulary Acquisition," *Informatica,* vol. 49, no. 27, 2025, doi: https://doi.org/10.31449/inf.v49i27.8821.

[8] F. Han, Y.-H. Wang, and F.-Y. Li, "A novel feature selection method based on adaptive search particle swarm optimization," *Neural Computing and Applications,* vol. 37, no. 12, pp. 7767–7783, 2025, doi: https://doi.org/10.1007/s00521-024-10611-6.

[9] Y. Xue, H. Zhu, J. Liang, and A. Słowik, "Adaptive crossover operator based multi-objective binary genetic algorithm for feature

selection in classification," *Knowledge-Based Systems,* vol. 227, p. 107218, 2021, doi: https://doi.org/10.1016/j.knosys.2021.107218.

[10] W. Ma, X. Zhou, H. Zhu, L. Li, and L. Jiao, "A two-stage hybrid ant colony optimization for high-dimensional feature selection," *Pattern Recognition,* vol. 116, p. 107933, 2021, doi: https://doi.org/10.1016/j.patcog.2021.107933.

[11] M. H. Nadimi-Shahraki, H. Zamani, and S. Mirjalili, "Enhanced whale optimization algorithm for medical feature selection: A COVID-19 case study," *Computers in biology and medicine,* vol. 148, p. 105858, 2022, doi: https://doi.org/10.1016/j.compbiomed.2022.105858.

[12] J. O. Agushaka, A. E. Ezugwu, and L. Abualigah, "Dwarf mongoose optimization algorithm," *Computer methods in applied mechanics and engineering,* vol. 391, p. 114570, 2022, doi: https://doi.org/10.1016/j.cma.2022.114570.

[13] H. Turabieh *et al.*, "Enhanced Harris Hawks optimization as a feature selection for the prediction of student performance," *Computing,* vol. 103, no. 7, pp. 1417–1438, 2021, doi: https://doi.org/10.1007/s00607-020-00894-7.

[14] M. Kamal *et al.*, "Metaheuristics method for classification and prediction of student performance using machine learning predictors," *Mathematical Problems in Engineering,* vol. 2022, no. 1, p. 2581951, 2022, doi: https://doi.org/10.1155/2022/2581951.

[15] M. R. Apriyadi and D. P. Rini, "Hyperparameter optimization of support vector regression algorithm using metaheuristic algorithm for student performance prediction," *International Journal of Advanced Computer Science and Applications,* vol. 14, no. 2, 2023, doi: https://doi.org/10.14569/IJACSA.2023.0140218.

[16] X. Song, "Student performance prediction employing k-nearest neighbor classification model and meta-heuristic algorithms," *Multiscale and Multidisciplinary Modeling, Experiments and Design,* vol. 7, no. 4, pp. 4397–4412, 2024, doi: https://doi.org/10.1007/s41939-024-00481-9.

[17] C. Ma, "Improving the Prediction of Student Performance by Integrating a Random Forest Classifier with Meta-Heuristic Optimization Algorithms," *International Journal of Advanced Computer Science & Applications,* vol. 15, no. 6, 2024, doi: https://doi.org/10.14569/ijacsa.2024.01506106.

[18] H. Shou and Y. Lu, "Student Performance Evaluation Technique By Applying Support Vector Classification And Metaheuristic Algorithms On The SVC Model's Reliability," *Journal of Applied Science and Engineering,* vol. 28, no. 3, pp. 653–666, 2025, doi: http://dx.doi.org/10.6180/jase.202503_28(3).0020.

[19] Z. Ye, Y. Yang, Y. Chen, and H. Chen, "Predicting Academic Performance Levels in Higher Education: A Data-Driven Enhanced Fruit Fly Optimizer Kernel Extreme Learning Machine Model," *Journal of Bionic Engineering,* pp. 1–23, 2025, doi: https://doi.org/10.1007/s42235-025-00716-6.

[20] M. B. Bagherabad, E. Rivandi, and M. J. Mehr, "Machine Learning for Analyzing Effects of Various Factors on Business Economic," *Authorea Preprints,* 2025, doi: https://doi.org/10.36227/techrxiv.174429010.09842200/v1.

[21] M. Li, C. Huang, D. Wang, Q. Hu, J. Zhu, and Y. Tang, "Improved randomized learning algorithms for imbalanced and noisy educational data classification," *Computing,* vol. 101, pp. 571–585, 2019, doi: https://doi.org/10.1007/s00607-018-00698-w.

[22] S. Alraddadi, S. Alseady, and S. Almotiri, "Prediction of students academic performance utilizing hybrid teaching-learning based feature selection and machine learning models," in *2021 International Conference of Women in Data Science at Taif University (WiDSTaif)*, 2021: IEEE, pp. 1–6, doi: https://doi.org/10.1109/WiDSTaif52235.2021.9430248.

[23] T. Thaher and R. Jayousi, "Prediction of student's academic performance using feedforward neural network augmented with stochastic trainers," in *2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT)*, 2020: IEEE, pp. 1–7, doi: https://doi.org/10.1109/AICT50176.2020.9368820.