

## Article

# Intelligent System for Student Performance Prediction: An Educational Data Mining Approach Using Metaheuristic-Optimized LightGBM with SHAP-Based Learning Analytics

Abdalhmid Abukader, Ahmad Alzubi and Oluwatayomi Rereloluwa Adegboye \*

Business Administration Department, Institute of Graduate Research and Studies, University of Mediterranean Karpasia, Mersin 10, Northern Cyprus, Lefkosa 99010, Turkey

\* Correspondence: oluwatayomi.rereloluwa@akun.edu.tr

## Abstract

Educational data mining (EDM) plays a crucial role in developing intelligent early warning systems that enable timely interventions to improve student outcomes. This study presents a novel approach to student performance prediction by integrating metaheuristic hyperparameter optimization with explainable artificial intelligence for enhanced learning analytics. While Light Gradient Boosting Machine (LightGBM) demonstrates efficiency in educational prediction tasks, achieving optimal performance requires sophisticated hyperparameter tuning, particularly for complex educational datasets where accuracy, interpretability, and actionable insights are paramount. This research addressed these challenges by implementing and evaluating five nature-inspired metaheuristic algorithms: Fox Algorithm (FOX), Giant Trevally Optimizer (GTO), Particle Swarm Optimization (PSO), Sand Cat Swarm Optimization (SCSO), and Salp Swarm Algorithm (SSA) for automated hyperparameter optimization. Using rigorous experimental methodology with 5-fold cross-validation and 20 independent runs, we assessed predictive performance through comprehensive metrics including Coefficient of Determination ( $R^2$ ), Root Mean Squared Error (RMSE), Mean Squared Error (MSE), Relative Absolute Error (RAE), and Mean Error (ME). Results demonstrate that metaheuristic optimization significantly enhances educational prediction accuracy, with SCSO-LightGBM achieving superior performance with  $R^2$  of 0.941. SHapley Additive exPlanations (SHAP) analysis provides crucial interpretability, identifying Attendance, Hours Studied, Previous Scores, and Parental Involvement as dominant predictive factors, offering evidence-based insights for educational stakeholders. The proposed SCSO-LightGBM framework establishes an intelligent, interpretable system that supports data-driven decision-making in educational environments, enabling proactive interventions to enhance student success.

**Keywords:** educational data mining (EDM); Light Gradient Boosting Machine; machine learning; artificial intelligence in education; SHAP interpretability; predictive modeling in education



Academic Editors: Eloy López Meneses and César Bernal-Bravo

Received: 2 September 2025

Revised: 26 September 2025

Accepted: 26 September 2025

Published: 10 October 2025

**Citation:** Abukader, A.; Alzubi, A.; Adegboye, O.R. Intelligent System for Student Performance Prediction: An Educational Data Mining Approach Using Metaheuristic-Optimized LightGBM with SHAP-Based Learning Analytics. *Appl. Sci.* **2025**, *15*, 10875. <https://doi.org/10.3390/app152010875>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Historically, economic expansion was primarily attributed to the accumulation of physical capital, with labor treated merely as an inherent factor of production. Productivity

was expected to rise through investments in tangible assets such as machinery and equipment [1]. In contrast, the modern growth paradigm unequivocally recognizes the central role of human capital in driving economic development. Numerous studies have demonstrated a strong correlation between human capital accumulation and long-term economic progress [2–4]. Within empirical research, years of schooling are frequently employed as a proxy for measuring human capital [1]. Individuals with higher levels of education gain a comparative advantage in understanding and integrating new technologies and ideas into production processes. Consequently, policymakers in many countries seek to invest in education as a strategy to stimulate economic growth [5]. However, the impact of such policies varies across contexts because of disparities in educational quality. Student academic performance is a key indicator of educational quality, serving as a direct measure of how effectively the institution supports learning and development [6]. Student performance refers to a student's level of achievement in their educational pursuits. It is commonly evaluated through grades, examination scores, and overall scholastic accomplishments. Predicting student performance has attracted growing attention in education due to its practical value. Early predictions allow instructors to promptly identify underperforming students and intervene before further decline, thereby increasing the likelihood of success [7]. Applications of student performance prediction have proven useful in identifying at-risk students and forecasting potential dropout rates. At the administrative level, such techniques support the efficient allocation of resources, ensuring that funding, staff, and specialized courses are directed to students who need them most. From the learner's perspective, predictive analytics facilitate individualized learning by tailoring instruction to personal strengths, weaknesses, and learning styles, thereby improving engagement and achievement [8]. Beyond the classroom, performance prediction also informs educational policy. By analyzing aggregate trends, policymakers can optimize investments in teacher training, infrastructure, technology, and pedagogical reforms, ensuring more effective and equitable educational systems.

Educational Data Mining (EDM) is an emerging discipline dedicated to developing methods for analyzing the rapidly growing volumes of data produced by educational institutions and applying these methods to better understand students and their learning environments [9]. Machine learning has found application in several domains [10–12]. In practice, EDM often relies on machine learning methods to examine student records, identify trends, and forecast academic outcomes [13]. As shown in the literature, a wide range of machine learning algorithms have been successfully applied for this purpose, including Decision Trees (DT) [14], Random Forests (RF) [15], Logistic Regression [16], Linear Regression [17], Naïve Bayes (NB) [18], Support Vector Machines (SVM) [19], K-Nearest Neighbors (KNN) [20], Artificial Neural Networks (ANN) [21], Bayesian Networks (BN), Discriminant Analysis (DA) [22], and Principal Component Analysis (PCA) [23]. While highly effective, previous studies showed that the performance of machine learning models is highly sensitive to the choice of hyperparameters [24]. Inadequate or suboptimal hyperparameter settings often lead to poor convergence, overfitting, or entrapment in local optima during training, thereby limiting the model's generalization ability. This limitation can be addressed by integrating metaheuristic algorithms (MAs) that systematically tune the hyperparameters of a given model to enhance its accuracy. By mimicking natural processes, MAs provide robust exploration and exploitation capabilities that directly address the above limitations of standalone ML models [25].

The present study seeks to enhance student performance prediction by integrating metaheuristic optimization with LightGBM, a gradient boosting framework known for efficient data processing, reduced memory consumption, and rapid training [26]. Specifically, five swarm-based optimization algorithms—FOX, GTO, PSO, SCSO, and SSA—were

employed to fine-tune the hyperparameters of LightGBM, addressing the limitations of conventional machine learning models such as poor convergence, overfitting, and sensitivity to parameter selection. Despite the promising predictive capabilities of ML models, most existing studies have provided limited justification for their outputs, with little attention paid to feature-level interpretability in the educational domain. To address this gap, this study incorporated Explainable Artificial Intelligence (XAI), with SHAP serving as the interpretability layer. SHAP systematically decomposes each prediction into the contribution of individual input variables, thereby improving transparency, fostering trust, and supporting evidence-based decision-making within educational institutions.

Accordingly, this study compared five LightGBM models optimized with swarm-based algorithms. The major contributions of this study are as follows:

- Development of hybrid predictive models that integrate LightGBM with FOX, GTO, PSO, SCSO, and SSA for optimized hyperparameter tuning.
- Application of SHAP for interpretability, enabling detailed analysis of feature importance and its influence on prediction outcomes.
- Empirical comparison of five swarm-based optimizers on student performance data, highlighting their relative strengths and weaknesses.
- Provision of a reliable and transparent framework that combines high predictive accuracy with interpretability, thereby supporting learners, educators, and policymakers.

The remainder of this paper is structured as follows. Section 2 presents a comprehensive review of the existing literature on student performance prediction, machine learning models, and metaheuristic optimization techniques. Section 3 details the research methodology, including descriptions of the FOX, GTO, PSO, SCSO, and SSA algorithms, the LightGBM model, and the proposed optimization framework, as well as the dataset employed in this study. Section 4 reports and discusses the experimental results, covering model evaluation metrics and predictive performance analysis. Finally, Section 5 concludes the paper by summarizing the main findings, highlighting practical implications, and outlining directions for future research.

## 2. Literature Review

Researchers have increasingly explored machine learning models combined with optimization methods to better predict student performance. These studies aim to help educators understand and support students more effectively. Xu and Kim [27] developed a combination prediction model for student performance by integrating Decision Tree (DT), Support Vector Regression (SVR), and Backpropagation Neural Network (BP). The contribution of each base model was weighted using an Ant Colony Optimization (ACO) algorithm. The hybrid model showed superior performance over standalone models. Fang et al. [28] introduced a hybrid MDBO-BP-Adaboost framework, where a modified Dung Beetle Optimization Algorithm was used to enhance the learning capacity of a backpropagation neural network combined with Adaboost. Experimental comparisons with baseline machine learning models demonstrated that this approach achieved superior prediction accuracy on a student performance dataset. Cheng et al. [29] investigated student performance prediction by comparing several machine learning classifiers, including Random Forest, Decision Tree, K-Nearest Neighbors, Multi Layer Perceptron, and eXtreme Gradient Boosting (XGBoost). The use of the Support Vector Machine with the Synthetic Minority Oversampling Technique (SVM-SMOTE) technique improved performance on imbalanced data, with XGBoost emerging as the most effective model. To further enhance results, metaheuristic algorithms were integrated with XGBoost, achieving significant improvements across accuracy, precision, recall, and F1-score. Among these, the Enhanced Artificial Ecosystem-Based Optimization-based XGBoost hybrid achieved the

highest accuracy. Emima and Amalarethinam [30] developed a hybrid Enhanced Teacher-Learner-Based Optimization with the Particle Swarm Optimization (ETLBO-PSO) model algorithm to fine-tune a Machine Learning model for student performance prediction. The model was tested on a secondary education dataset in Mathematics using XGBoost, Light Gradient-Boosting Machine (LightGBM), and Categorical Boosting (CatBoost) classifiers, where it achieved a robust F1-score of 82.43% while reducing redundancy and efficiency loss. Li et al. [31] developed a LightGBM with Genetic Algorithm (GA) Model for an academic performance prediction model for college students. The model achieved high accuracy and low prediction error. From the observed results, predicted grades closely matched actual student performance, demonstrating the model's reliability.

Apriyadi et al. [32] proposed a student performance prediction model by optimizing SVR with metaheuristic algorithms. Both PSO and GA were applied to tune SVR hyperparameters, resulting in PSVR and GSVR models. Experimental results showed that the PSVR model achieved the lowest RMSE (1.608), outperforming several traditional models such as Naïve Bayes, Decision Tree, and Random Forest. Kamal et al. [33] proposed a metaheuristic and machine learning-based approach for classifying and predicting student performance. Features were first selected using a Relief algorithm, and classifiers including Back Propagation Neural Network (BPNN), Random Forest, and Naïve Bayes were applied. Among these, BPNN achieved the highest accuracy in both classification and prediction tasks. The study demonstrates the effectiveness of combining feature selection with machine learning models for reliable student performance forecasting.

Xu [34] introduced a hybrid IDA-SVR model to predict student performance, incorporating an Improved Duel Algorithm (IDA) for hyperparameter optimization. Unlike traditional approaches, the study emphasized student behavioral data and compared IDA-SVR with DT, ANN, standard SVR, and PSO-SVR. Results showed that IDA-SVR achieved the lowest MSE (0.0089), significantly outperforming competing models. The findings highlight IDA's ability to avoid local optima and accelerate convergence, making it a strong tool for enhancing prediction accuracy. Song [35] integrated the K-Nearest Neighbor Classification (KNNC) model with two bio-inspired optimization techniques, namely the Honey Badger Algorithm (HBA) and the Arithmetic Optimization Algorithm (AOA). Their approach aimed to improve forecasting precision and reliability to support better educational outcomes. The proposed KNHB model showed superior performance in categorizing final grades with an accuracy of 0.921 and a precision of 0.92. Similarly, it achieved strong results for first-period grade prediction. Ma [36] proposed a Random Forest Classifier (RFC) model enhanced with specifically Electric Charged Particles Optimization (ECPO) and Artificial Rabbits Optimization (ARO), for predicting student performance. The study analyzed input variables to determine their impact on academic outcomes, helping to identify key areas for educational improvement. The optimized RFEC model demonstrated superior predictive accuracy across 4424 students, aligning closely with actual performance measurements. Ali et al. [37] developed a hybrid Principal Component Analysis with Cuckoo Search Neural Network (PCACSNN) to improve student performance prediction and overcome overfitting issues in large datasets. The model combines ANN with the PCACSNN algorithm to enhance convergence and prediction accuracy. Experimental results on the UCI Student Performance datasets for Mathematics and Portuguese showed superior performance compared to ANN, BPNN, and CSBP. Punitha and Devaki [38] proposed a deep learning-based predictive approach for student performance that integrates feature selection with the Modified Red Deer Algorithm (MRDA) and a Deep Ensemble Network (DEnsNet). The MRDA algorithm was used to optimize both feature selection and network parameters, ensuring improved prediction accuracy. Experimental results showed that the proposed model outperformed several comparative models with accuracy improvements

ranging from 3% to 9.91% across two datasets. Hai and Wang [39] proposed a hybrid Multilayer Perceptron Classification (MLPC) approach for student performance prediction. The study integrated ensemble techniques and metaheuristic optimizers, specifically the Pelican Optimization Algorithm (POA) and Crystal Structure Algorithm (CSA), to improve model accuracy. Experimental results showed that the hybrid model achieved a 95.78% success rate, outperforming other approaches.

The existing body of research highlights the increasing effectiveness of integrating machine learning with metaheuristic optimization techniques for student performance prediction. Nonetheless, most prior studies placed a predominant emphasis on predictive accuracy, often neglecting rigorous interpretability diagnostics, and frequently restricted their analyses to a limited set of optimization algorithms. Such constraints reduce both the comprehensiveness and the practical applicability of their findings. To address these gaps, the present study conducted a systematic comparative evaluation of five swarm-intelligence-based optimization algorithms, FOX, GTO, PSO, SCSO, and SSA, in enhancing the predictive capability of the LightGBM model. Beyond accuracy, we incorporated SHAP analysis to provide post hoc interpretability, offering insight into the relative importance, variance, and interaction of features driving student outcomes. This dual emphasis on performance and interpretability contributes to the development of predictive analytics frameworks that are not only accurate and robust but also transparent and reliable.

### 3. Methodology

#### 3.1. Fox Optimization Algorithm (FOX)

The Fox Optimization Algorithm (FOX) is a bio-inspired metaheuristic that emulates the predatory behavior of foxes in snow-covered environments [40]. Foxes rely on auditory cues to detect prey beneath the snow and execute pouncing maneuvers with high precision. The algorithm abstracts this behavior into a computational framework for solving continuous numerical optimization problems. Let the population of candidate solutions (agents) be represented by a matrix  $X \in \mathbb{R}^{N \times D}$ , where  $N$  represents the number of search agents,  $D$  denotes the dimensionality of the problem,  $x_{i,j}$  represent the value of the  $j$ -th variable of the  $i$ -th agent as defined in Equation (1).

$$X = \begin{bmatrix} x_{1,1} & \dots & x_{1,D} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \dots & x_{N,D} \end{bmatrix} \quad (1)$$

Each row  $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,D}]$  represents a solution vector in the  $D$ -dimensional search space. Initial positions are generated randomly within the feasible bounds as given in Equation (2):

$$x_{i,j} = lb_j + r_{i,j} \cdot (ub_j - lb_j) \quad (2)$$

where  $lb_j$  and  $ub_j$  denote the lower and upper bounds of the  $j$ -th dimension.  $r_{i,j}$  is a random number drawn from a uniform distribution over the interval  $(0, 1)$ . To balance exploration and exploitation, a control parameter  $a \in [0, 2]$  is introduced, which decreases linearly over iterations as expressed in Equation (3):

$$a = 2 \left( 1 - \frac{t}{T} \right) \quad (3)$$

where  $t$  is the current iteration,  $T$  is the maximum number of iterations. Additionally, a random parameter  $r$  which is drawn from a uniform distribution over the interval  $(0, 1)$



determines the phase of operation. In the exploration phase, the fox detects prey through sound propagation under snow. The propagation distance is computed in Equation (4):

$$d_s(t) = v_s \cdot T_s(t) \quad (4)$$

where  $T_s(t)$  is a vector of random sound propagation times within interval  $[0, 1]^D$ ,  $v_s$  is the effective speed of sound in the medium,  $d_s(t)$  represent the total sound propagation distance. The speed of sound is estimated based on the best-known position as expressed in Equation (5):

$$v_s = \frac{X_{\text{best}}(t)}{T_s(t)} \quad (5)$$

The fitness of each agent is evaluated using the objective function  $f(\cdot)$ . The best solution found so far is denoted  $X_{\text{best}}$ , with corresponding fitness  $f_{\text{best}}$ . The fox–prey distance is approximated as half the round-trip propagation distance, which is computed according to Equation (6):

$$d_{\text{fox-prey}}(t) = 0.5 \cdot d_s(t) \quad (6)$$

The pouncing motion is modeled as a parabolic jump influenced by gravity as expressed in Equation (7):

$$J(t) = \frac{1}{2}gt^2 \quad (7)$$

where  $g$ , which is the gravitational acceleration, is set to  $9.81 \text{ m/s}^2$ ,  $t = \frac{1}{D} \sum_{j=1}^D T_s^{(j)}$  represent the average sound propagation time across dimensions. A random variable  $p$  which is drawn from  $\mathcal{U}(0, 1)$  determines the update rule. If  $p > 0.18$ , the rule in Equation (8) applies:

$$X_i(t+1) = d_{\text{fox-prey}}(t) \cdot J(t) \cdot c_1 \quad (8)$$

Otherwise, the Equation (9) update rule is applied to the current individual:

$$X_i(t+1) = d_{\text{fox-prey}}(t) \cdot J(t) \cdot c_2 \quad (9)$$

where  $c_1$  is a random number within  $[0, 0.18]$ ,  $c_2$  is a random number within  $[0.18, 1]$ , and both variables are learning coefficients reflecting pounce success probability. During exploitation, a time-controlled random walk refines the search near to the best solution. The position update is expressed in Equation (10):

$$X_i(t+1) = X_{\text{best}}(t) \cdot r_1 \cdot t_{\min} \cdot a \quad (10)$$

where  $r_1$  is a random vector within  $[0, 1]^D$ ,  $t_{\min}$  is the minimum average time across agents.

### 3.2. Giant Trevally Optimizer (GTO)

The Giant Trevally Optimizer (GTO) is inspired by the hunting behavior of *Caranx ignobilis*, a marine predator known for leaping from water to capture airborne prey [41]. This behavior involves long-range movement (exploration), area selection, and precise aerial attacks (exploitation). The population matrix as given in Equations (1) and (2), and the best solution is denoted  $X_{\text{best}}$ . The extensive Search via Lévy Flight simulates long-range foraging (exploration) using Lévy flights as expressed in Equation (11):

$$X_i(t+1) = X_{\text{best}}(t) \cdot r_2 + [(ub - lb) \cdot r_3 + lb] \cdot \text{Levy}(D) \quad (11)$$

where  $r_2$  and  $r_3$  are uniform random variables chosen within  $[0, 1]$ , and Levy ( $D$ ) is a Lévy-distributed random vector. The Lévy step is generated as expressed in Equation (12):

$$\text{Levy}(D) = \text{step} \cdot \frac{u \cdot \sigma}{|v|^{1/\beta}}, \text{ step} = 0.01, \beta = 1.5 \quad (12)$$

With  $\sigma$ , which is the scaling parameter defined in Equation (13):

$$\sigma = \left( \frac{\Gamma(1 + \beta) \cdot \sin(\pi\beta/2)}{\Gamma\left(\frac{1+\beta}{2}\right) \cdot \beta \cdot 2^{(\beta-1)/2}} \right)^{1/\beta} \quad (13)$$

$u, v \sim \mathcal{N}(0, 1)$  represent the standard and normal variables. The optimizer models the predator's ability to identify prey-rich zones as given in Equation (14):

$$X_i(t+1) = X_{\text{best}}(t) \cdot A \cdot r_4 + M(t) - X_i(t) \cdot r_5 \quad (14)$$

where  $A \in [0.3, 0.4]$  is the area selection coefficient,  $r_4$  and  $r_5$  are uniformly distributed random numbers within  $[0, 1]$ , and  $M(t)$ , which denotes the mean position of the swarm, is calculated as follows in Equation (15):

$$M(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \quad (15)$$

The Exploitation phase mimics the leap from water, accounting for visual distortion due to refraction, using Snell's Law, which is calculated as given in Equation (16):

$$\sin(\theta_1) = \frac{\eta_1}{\eta_2} \cdot \sin(\theta_2) \quad (16)$$

where,  $\theta_1$  is the angle of incidence in water,  $\theta_2$  which is chosen from the range  $[0^\circ, 360^\circ]$  is the refracted angle in air,  $\eta_1$  representing the refractive index of air is set to 1.00029.  $\eta_2$  which is the refractive index of water is set to 1.33. The visual distortion of prey is computed as follows, as given in Equation (17):

$$V = \sin(\theta_1^\circ) \cdot d_i, d_i = \|X_{\text{best}}(t) - X_i(t)\| \quad (17)$$

Finally, the position update is expressed in Equation (18):

$$X_i(t+1) = L + V + H \quad (18)$$

The launching vector  $L$  is calculated as follows in Equation (19):

$$L = X_i(t) \cdot \sin(\theta_2^\circ) \cdot f(X_i(t)) \quad (19)$$

The Adaptive slope  $H$  is computed as follows in Equation (20):

$$H = r_6 \cdot \left( 2 - 2 \frac{t}{T} \right) \quad (20)$$

$r_6$  represents a random number within  $[0, 1]$ . As  $t$  tends to  $T$  and  $H$  tends to zero, the optimizer shifts focus from exploration to exploitation.

### 3.3. Particle Swarm Optimization (PSO)

PSO is a population-based stochastic optimization method inspired by social dynamics in bird flocks and fish schools. Particles adjust their trajectories based on per-

sonal and global experience [42]. Each particle  $i$  ( $i = 1, 2, \dots, N$ ) has the following: Position vector  $X_i = [x_{i,1}, \dots, x_{i,D}]$ , Velocity vector  $V_i = [v_{i,1}, \dots, v_{i,D}]$ , Personal best  $P_i = [p_{i,1}, \dots, p_{i,D}]$ , and Global best  $G = [g_1, \dots, g_D]$ . At iteration  $t$ , updates are as given in Equations (21) and (22):

$$v_{i,j}(t+1) = \omega \cdot v_{i,j}(t) + c_3 r_7 \cdot (p_{i,j} - x_{i,j}(t)) + c_4 r_8 \cdot (g_j - x_{i,j}(t)) \quad (21)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \quad (22)$$

where  $\omega$  is the inertia weight,  $c_3$  and  $c_4$  are cognitive and social acceleration coefficients,  $r_7$  and  $r_8$  are uniformly distributed random numbers within  $[0, 1]$ . Variable  $\omega$  is computed as follows in Equation (23):

$$\omega = \omega_{\max} - \frac{t}{T}(\omega_{\max} - \omega_{\min}) \quad (23)$$

where  $\omega_{\max}$  is set to 0.9, and  $\omega_{\min}$  is set to 0.4. Higher  $\omega$  favors exploration; lower  $\omega$  enhances exploitation.

### 3.4. Sand Cat Swarm Optimization (SCSO)

SCSO models the foraging behavior of sand cats (*Felis margarita*), which use acute hearing to detect subterranean prey [43]. Two phases, prey detection (exploration) and predation (exploitation), are emulated. Population  $X \in \mathbb{R}^{N \times D}$  initialized as expressed in Equations (1) and (2). During the prey detection phase (Exploration). The sensitivity magnitude  $S_M = 2$ . Global sensitivity parameter is updated according to Equation (24):

$$r_G = S_M \left(1 - \frac{t}{T}\right) \quad (24)$$

The random control parameter is calculated as follows in Equation (25):

$$R = 2r_G r_9 - r_G \quad (25)$$

$r_9$  are uniformly distributed random numbers within  $[0, 1]$ . The Individual sensitivity span  $r$  is calculated as follows in Equation (26):

$$r = r_G \cdot r_{10} \quad (26)$$

$r_{10}$  are uniformly distributed random numbers within  $[0, 1]$ . Finally, the exploration update rule is expressed in Equation (27):

$$X_i(t+1) = r \cdot [X_b(t) - r_{11} \cdot X_i(t)] \quad (27)$$

$r_{11}$  is a uniformly distributed random number within  $[0, 1]$ , the best solution is denoted by  $X_b$ , and  $X_i(t)$  represents the current solution. For the prey attack (Exploitation), the distance to the prey and sand cat is calculated as given in Equation (28):

$$X_{\text{rnd}} = |r_{12} \cdot X_b(t) - X_i(t)| \quad (28)$$

$r_{12}$  is an abstract uniformly distributed random number within  $[0, 1]$ . The exploitation update is expressed in Equation (29):

$$X_i(t+1) = X_b(t) - r \cdot X_{\text{rnd}} \cdot \cos(\alpha) \quad (29)$$



The random angle  $\alpha$  is chosen within  $[0, 2\pi]$ . The Adaptive Switching Mechanism to transition between phases is governed by  $|R|$ , and the update is done as follows in Equation (30):

$$X_i(t+1) = \begin{cases} r \cdot [X_b(t) - r_{11} \cdot X_i(t)], & \text{if } |R| > 1 \quad (\text{Exploration}) \\ X_b(t) - r \cdot X_{rnd} \cdot \cos(\alpha), & \text{if } |R| \leq 1 \quad (\text{Exploitation}) \end{cases} \quad (30)$$

This ensures dynamic balance between global search and local refinement.

### 3.5. Salp Swarm Algorithm (SSA)

SSA, proposed by Mirjalili et al. [44], mimics the swarming and foraging behavior of salp marine invertebrates that move in chains to efficiently navigate ocean currents and locate food sources. The swarm is represented by matrix  $X \in \mathbb{R}^{N \times D}$ , where  $N$  is the number of salps, and  $D$  is the dimensionality as expressed in Equations (1) and (2). Food source  $F = [F_1, F_2, \dots, F_D]$  represents the target (optimal solution). The first salp (leader) guides the chain toward the food source as given in Equation (31):

$$x_{1,j}(t+1) = \begin{cases} F_j + c_3 \cdot [c_4 \cdot (ub_j - lb_j) + lb_j], & \text{if } c_5 \geq 0.5 \\ F_j - c_3 \cdot [c_4 \cdot (ub_j - lb_j) + lb_j], & \text{if } c_5 < 0.5 \end{cases} \quad (31)$$

$c_3$  the convergence coefficient (decreases over time) as expressed in Equation (32):

$$c_3 = 2e^{-(4t/T)^2} \quad (32)$$

$c_4$ , and  $c_5$  are uniformly distributed random numbers within  $[0, 1]$ . Large  $c_3$  promotes exploration; small  $c_3$  enables exploitation. Remaining salps follow the one ahead using a Newtonian motion model as given in Equation (33):

$$x_{i,j}(t+1) = \frac{1}{2}(x_{i,j}(t) + x_{i-1,j}(t)), \quad i \geq 2 \quad (33)$$

This ensures smooth, coordinated movement toward the leader and ultimately the food source.

### 3.6. Light Gradient Boosting Machine (LightGBM)

The Light Gradient Boosting Machine (LightGBM) is a boosting algorithm based on decision trees, for applications in regression, classification, and feature selection [45]. It is characterized by high computational efficiency, reduced memory usage, and improved predictive accuracy. The fundamental principle of LightGBM involves iteratively enhancing model performance through the sequential addition of weak learners. Let  $F_{t-1}(x)$  denote the learned model from the previous iteration. Given a loss function  $L(y, F_{t-1}(x))$ , the objective in the current iteration is to identify a weak learner  $h_t(x)$  that minimizes the overall loss when added to the existing model. This optimization can be formally expressed as given in Equation (34):

$$h_t(x) = \arg \min_{h \in \mathcal{H}} \sum L(y, F_{t-1}(x) + h_t(x)) \quad (34)$$

To facilitate optimization, the loss function is approximated using the negative gradient (also known as the functional gradient) evaluated at the current model output. This residual, denoted as  $r_{ti}$ , represents the pseudo-residuals for each instance  $x_i$  and is computed in Equation (35):

$$r_{ti} = -\frac{\partial L(y, F_{t-1}(x_i))}{\partial F_{t-1}(x_i)} \quad (35)$$

The weak learner  $h_t(x)$  is then trained to minimize the squared error between these pseudo-residuals and its predictions according to Equation (36):

$$h_t(x) = \arg \min_{h \in \mathcal{H}} \sum (r_{ti} - h_t(x))^2 \quad (36)$$

Upon obtaining  $h_t(x)$ , the model is updated to form the new strong learner as described in Equation (37):

$$F_t(x) = F_{t-1}(x) + h_t(x) \quad (37)$$

This iterative process continues until a predefined stopping criterion is met, resulting in a final predictive model with enhanced accuracy and robustness.

### 3.7. Proposed Model Optimization Framework

LightGBM demonstrates strong efficiency for prediction tasks but suffers from sensitivity to hyperparameter settings [46–48]. Parameters such as `max_depth`, `n_estimators`, and `learning_rate` strongly influence the model's ability to balance bias, variance, and computational efficiency. Without systematic tuning, LightGBM often converges to suboptimal solutions, leading to reduced predictive accuracy and poor generalization. To address this, a metaheuristic-based optimization framework is proposed, integrating swarm intelligence algorithms with LightGBM to automatically search for near-optimal hyperparameters. The framework integrates five swarm intelligence algorithms, FOX, GTO, PSO, SCSO, and SSA, to search for the optimal combination of LightGBM hyperparameters systematically. These algorithms mimic natural collective behaviors to explore the parameter space efficiently, avoiding local optima and converging toward globally superior solutions. The key hyperparameters optimized in this framework include the following:

- `max_depth`: Controls the maximum depth of individual decision trees. A well-tuned depth prevents overfitting by limiting model complexity. The search range is set to (3,15).
- `learning_rate`: Regulates the contribution of each tree during boosting. A value too low results in slow convergence, while a value too high may lead to overshooting the optimum. The range (0.01, 0.5) is explored.
- `n_estimators` determines the number of boosting rounds in LightGBM. It directly influences model performance, generalization, and training. The range (10,150) is explored.

The proposed optimization workflow is structured as follows:

- Step 1: Preprocess the educational dataset by normalizing input features, encoding categorical variables.
- Step 2: Partition the dataset into training and testing.
- Step 3: Initialize the LightGBM model with default hyperparameters and define the search bounds for each tunable parameter as specified above.
- Step 4: Employ each metaheuristic algorithm (FOX, GTO, PSO, SCSO, SSA) to optimize the hyperparameter configuration. Optimizers apply different population update mechanisms as defined in Section 3.1, 3.2, 3.3, 3.4 and 3.5; steps of each optimizer are iterated in their original literature. The algorithm iteratively updates candidate solutions based on fitness evaluation using the MSE score as the objective function. In this study, the objective function is defined as the mean of the MSE over a 2-fold cross-validation, computed on the training dataset for each candidate hyperparameter set. This approach ensures a robust assessment of model performance while maintaining computational efficiency during the search process.

- Step 5: At each iteration, compute the fitness of the current hyperparameter set, update the global best and individual best positions, and assess convergence criteria. The process continues until the maximum number of iterations is reached or convergence is achieved.
- Step 6: Retrieve the optimal hyperparameter set identified by each optimizer. Evaluate the final model on the test set using multiple performance metrics.
- Step 7: Perform SHAP analysis to interpret model predictions, identify key influencing features, and validate the results against pedagogical insights.

This proposed framework not only enhances the predictive power of LightGBM but also provides a reproducible and scalable methodology for hyperparameter tuning in an educational dataset. By leveraging swarm intelligence, the approach ensures robust, data-driven model calibration, making it highly suitable for real-world academic performance prediction systems. The selection of the five metaheuristic optimizers FOX, GTO, PSO, SCSO, and SSA was guided by three criteria: (1) representation of distinct biological inspiration paradigms, (2) demonstrated empirical efficacy particularly exceptional performance in recent machine learning hyperparameter optimization literature [49–53], and (3) algorithmic diversity in search strategy. PSO, inspired by avian flocking, serves as the established baseline for social swarm intelligence. SSA, modeling marine invertebrate chains, introduces a leader–follower exploitation mechanism distinct from PSO’s velocity-based updates. SCSO, emulating subterranean prey detection in felines, features a unique sensitivity-based switching mechanism between exploration and exploitation. Finally, FOX, simulating auditory prey localization in canids, employs a gravity-influenced pounce model for exploitation. The framework is illustrated in Figure 1.

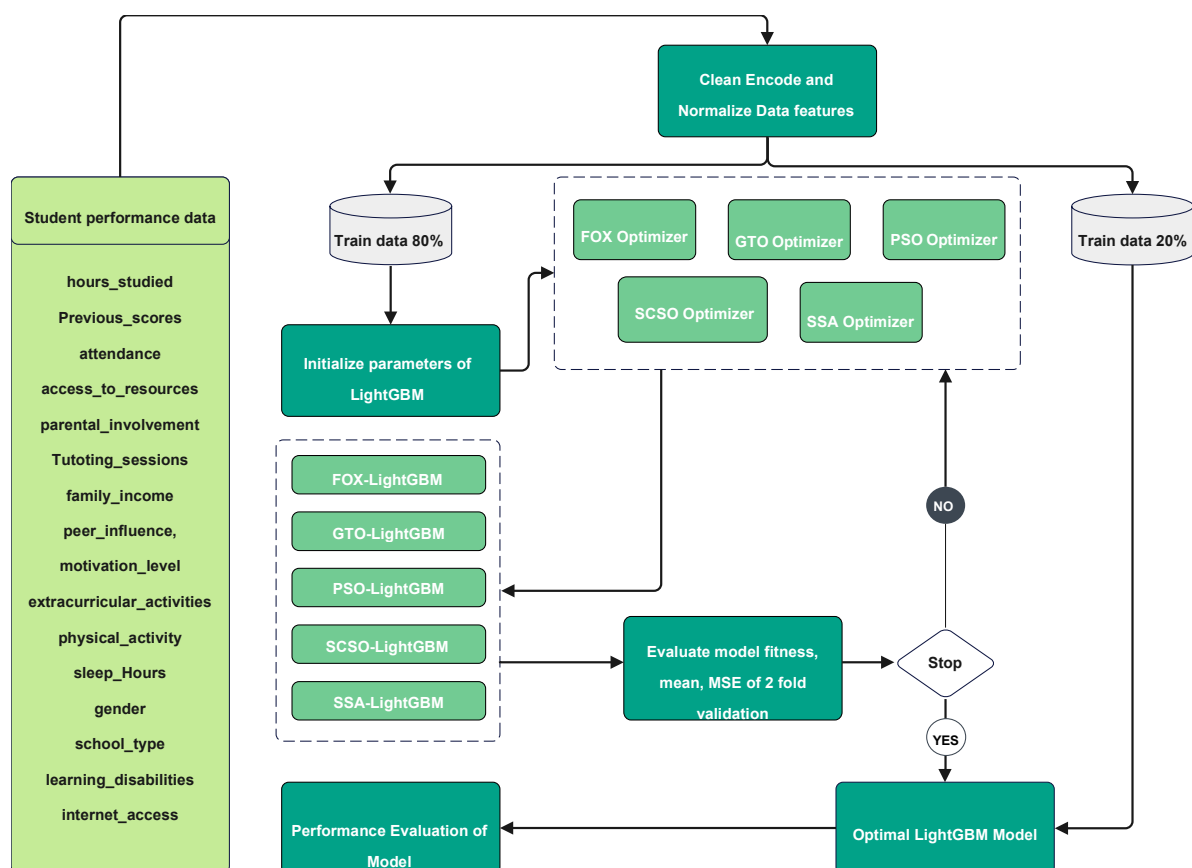


Figure 1. Proposed framework optimizer tuned LightGBM.

### 3.8. Data

The dataset utilized in this study was sourced from Kaggle [54] and contains a comprehensive set of variables reflecting diverse factors influencing student academic performance, with the primary outcome being the exam score, with 6607 data samples before data clean up. It comprises 16 predictor variables after the data clean-up process, encompassing academic, behavioral, socio-economic variables; namely the variables are hours\_studied, attendance, parental\_involvement, access\_to\_resources, extracurricular\_activities, sleep\_hours, previous\_scores, motivation\_level, internet\_access, tutoring\_sessions, family\_income, school\_type, peer\_influence, physical\_activity, learning\_disabilities, and gender. Prior to using LightGBM models to predict, a systematic data cleaning procedure were employed to ensure data quality and modeling reliability. Columns containing missing values were excluded, and duplicate records were removed to maintain data integrity. Categorical variables were transformed into numerical form using label encoding, assigning unique integer values to each category. To mitigate the influence of outliers, the Z-score method was applied, thereby retaining data points within a defined normal range. The dataset was subsequently partitioned into training (80%) and testing (20%) subsets using a random split. Feature normalization was performed using Min-Max scaling, which linearly transforms each feature to a standard range [0, 1] according to Equation (38):

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (38)$$

where  $X$  is the original feature value, and  $X_{\min}$  and  $X_{\max}$  are the minimum and maximum values of the feature in the training set, respectively. This transformation was applied independently to each feature using parameters derived from the training data and subsequently used to scale the test set, preventing data leakage. The target variable is exam score, while other variables are predictors. Table 1 provides a summary statistics table for the 16 variables from the dataset, specifically showing the distribution of variables related to student performance. It presents standard statistical measures for each variable across 4313 observations, after removing columns with empty values, encoding categorical variables into numerical values, and handling outliers without scaling.

**Table 1.** Summary of features.

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previous_Scores	Motivation_Level	
count	4313	4313	4313	4313	4313	4313	4313	4313	
mean	19.94	79.96	1.09	0.9	0.6	7.02	75.01	0.91	
std	5.24	11.45	0.69	0.7	0.49	1.2	14.34	0.69	
min	8	60	0	0	0	5	50	0	
25%	16	70	1	0	0	6	63	0	
50%	20	80	1	1	1	7	75	1	
75%	24	90	2	1	1	8	87	1	
max	31	100	2	2	1	9	100	2	
	Internet_Access	Tutoring_Sessions	Family_Income	School_Type	Peer_Influence	Physical_Activity	Learning_Disabilities	Gender	Exam_Score
count	4313	4313	4313	4313	4313	4313	4313	4313	4313
mean	0	1.29	0.79	0.3	1	2.96	0	0.42	67.14
std	0	0.97	0.74	0.46	0.89	0.98	0	0.49	3.11
min	0	0	0	0	0	1	0	0	60
25%	0	1	0	0	0	2	0	0	65
50%	0	1	1	0	1	3	0	0	67
75%	0	2	1	1	2	4	0	1	69
max	0	3	2	1	2	5	0	1	75

### 3.9. Model Evaluation Metrics

To assess the performance of the predictive models, several regression evaluation metrics were employed: the  $R^2$ , RMSE, MSE, ME, and RAE. These metrics collectively provide insights into model accuracy and error magnitude, enabling robust comparison across different modeling approaches. The  $R^2$  measures the proportion of variance in the dependent variable (exam score) that is predictable from the independent variables. It is defined as expressed in Equation (39):

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (39)$$

where  $y_i$  is the observed value,  $\hat{y}_i$  is the predicted value, and  $\bar{y}$  is the mean of the observed values.  $R^2$  ranges from  $-\infty$  to 1, with values closer to 1 indicating a better fit. A value of 1 implies perfect prediction, while values below 0 suggest the model performs worse than a simple mean predictor. The MSE quantifies the average squared difference between predicted and actual values, penalizing larger errors more heavily due to the squaring operation as defined in Equation (40):

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (40)$$

where  $n$  is the number of observations. MSE is sensitive to outliers and provides a measure of overall model precision in squared units of the target variable. The RMSE is the square root of MSE and expresses prediction error in the same units as the target variable. It is calculated as expressed in Equation (41):

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (41)$$

Lower RMSE values indicate higher accuracy and tighter clustering of predictions around the true values. The ME captures the largest absolute residual across all predictions as expressed in Equation (42):

$$\text{ME} = \max_{i=1, \dots, n} |y_i - \hat{y}_i| \quad (42)$$

This metric highlights the worst-case prediction error and is useful for identifying potential model failure cases or sensitivity to extreme values. Finally, the RAE normalizes the total absolute prediction error by the total absolute deviation from the mean according to Equation (43):

$$\text{RAE} = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{\sum_{i=1}^n |y_i - \bar{y}|} \quad (43)$$

RAE provides a relative measure of model performance, where values less than 1 indicate that the model performs better than a naive baseline (predicting the mean), and smaller values reflect superior predictive accuracy. Together, these metrics offer a multi-faceted evaluation of model performance, balancing interpretability, sensitivity to error magnitude, and comparative benchmarking against baseline models.

### 3.10. Proposed Comparative Optimization Framework

In this study, the LightGBM model was optimized using five metaheuristic algorithms: FOX, GTO, PSO, and SCSO. The hyperparameter settings for each optimizer are detailed in Table 2. The search boundaries of parameters tuned by optimizers are as detailed in Section 3.7. For the baseline LightGBM model, the following fixed parameters are

employed: `n_estimators` is set to 150, `learning_rate` is set to 0.3, and `max_depth` is set to 15. The baseline Support Vector Regression (SVR) model is configured with a radial basis function (RBF) kernel, a regularization parameter  $C$  of 100, and a kernel coefficient  $\gamma$  of 0.1, which are commonly used settings for achieving stable performance in regression tasks. The baseline model's parameter configurations are selected from widely used parameters reported in the established literature [55–57]. All metaheuristic optimizers are configured with a population size of 30 and a maximum of 100 iterations to ensure a balanced trade-off between computational efficiency and convergence to optimal solutions. The optimization process is guided by minimizing the MSE evaluated via 2-fold cross-validation within each run. The overall experimental framework consists of two phases: first, a 5-fold cross-validation test is done to ensure robust model evaluation and reduce bias in performance estimation; second, the entire process is repeated over 20 independent runs to account for the stochastic nature of the metaheuristic algorithms and to enhance the reliability of the results. This rigorous evaluation protocol enables a comprehensive comparison of the optimization techniques in terms of both predictive accuracy and generalization capability.

**Table 2.** Optimization algorithm settings.

Optimizer	Parameter Settings
FOX	$a = [0, 2]$
GTO	$A = [0.3, 0.4], H = [2, 0]$
PSO	$\omega_{\max} = 0.9, \omega_{\min} = 0.4$
SCSO	$\alpha = [0, 2\pi], C = 0.35, S_M = 2$
SSA	$c_3 = [2/e, 2]$

## 4. Experiment and Discussion

### 4.1. Student Performance Prediction

#### 4.1.1. Cross-Validation Analysis

To obtain unbiased estimates of model generalization, we employed a cross-validation strategy. In each split of the outer 5-fold CV, the training partition was used for hyperparameter optimization. After selecting the optimal hyperparameters, the model was evaluated once on the corresponding outer held-out fold. Performance metrics were then aggregated across the five outer folds. Notably, no cross-validation was applied to the held-out folds, ensuring that the data used for hyperparameter optimization were entirely separate from those used for evaluation. For the results, training-fold CV performance is reported in Table 3, while outer held-out fold performance is presented in Table 4.

Tables 3 and 4 present the training and held-out performance outcomes of the metaheuristic-optimized LightGBM models and baseline models using 5-fold cross-validation. The training performance metrics presented in Table 3 illustrate that the metaheuristic-optimized LightGBM variants substantially outperformed the baseline LightGBM and SVR models in terms of predictive accuracy and error minimization. Among the optimized models, FOX-LightGBM attained the highest average  $R^2$  score of 0.9693, indicating superior fit to the training data, followed by SCSO-LightGBM (0.9609), SSA-LightGBM (0.9606), PSO-LightGBM (0.9598), and GTO-LightGBM (0.9581). This enhanced accuracy is corroborated by lower error metrics: the optimized models achieved RMSE values ranging from 0.0361 to 0.0423, significantly lower than the baseline LightGBM (0.0578) and SVR (0.0523). Similarly, MSE values for the optimizers were lower compared to the baseline models. FOX-LightGBM also recorded the lowest RAE at 0.0695, underscoring its effectiveness in calibrating hyperparameters to capture the underlying data patterns. However, FOX-LightGBM exhibited greater variability across folds, as evidenced by its



higher standard deviations in  $R^2$ , RMSE, and MSE, relative to more stable performers like SCSO-LightGBM and PSO-LightGBM, which demonstrated tighter deviations. Notably, SVR achieved the lowest maximum error (ME) at 0.1008, suggesting reduced peak deviations in training predictions, though this did not translate to overall superiority given its higher aggregated errors.

**Table 3.** Training performance of LightGBM models using 5-fold cross-validation.

		FOX-LightGBM	GTO-LightGBM	PSO-LightGBM	SCSO-LightGBM	SSA-LightGBM	LightGBM	SVR
$R^2$	AVG	<b>0.9693</b>	0.9581	0.9598	0.9609	0.9606	0.9224	0.9363
	STD	$7.212 \times 10^{-3}$	$5.647 \times 10^{-3}$	$1.114 \times 10^{-3}$	$5.173 \times 10^{-4}$	$8.283 \times 10^{-4}$	$8.680 \times 10^{-4}$	<b><math>4.054 \times 10^{-4}</math></b>
RMSE	AVG	<b><math>3.606 \times 10^{-2}</math></b>	$4.234 \times 10^{-2}$	$4.155 \times 10^{-2}$	$4.100 \times 10^{-2}$	$4.113 \times 10^{-2}$	$5.775 \times 10^{-2}$	$5.231 \times 10^{-2}$
	STD	$4.198 \times 10^{-3}$	$2.884 \times 10^{-3}$	$4.934 \times 10^{-4}$	$3.221 \times 10^{-4}$	$4.349 \times 10^{-4}$	$3.102 \times 10^{-4}$	<b><math>2.069 \times 10^{-4}</math></b>
MSE	AVG	<b><math>1.318 \times 10^{-3}</math></b>	$1.801 \times 10^{-3}$	$1.727 \times 10^{-3}$	$1.681 \times 10^{-3}$	$1.692 \times 10^{-3}$	$3.335 \times 10^{-3}$	$2.737 \times 10^{-3}$
	STD	$3.073 \times 10^{-4}$	$2.392 \times 10^{-4}$	$4.124 \times 10^{-5}$	$2.650 \times 10^{-5}$	$3.595 \times 10^{-5}$	$3.581 \times 10^{-5}$	<b><math>2.145 \times 10^{-5}</math></b>
ME	AVG	$1.365 \times 10^{-1}$	$1.467 \times 10^{-1}$	$1.436 \times 10^{-1}$	$1.396 \times 10^{-1}$	$1.440 \times 10^{-1}$	$1.943 \times 10^{-1}$	<b><math>1.008 \times 10^{-1}</math></b>
	STD	$1.515 \times 10^{-2}$	$7.976 \times 10^{-3}$	$8.224 \times 10^{-3}$	$7.349 \times 10^{-3}$	$7.337 \times 10^{-3}$	$1.046 \times 10^{-2}$	<b><math>5.541 \times 10^{-4}</math></b>
RAE	AVG	<b><math>6.947 \times 10^{-2}</math></b>	$8.156 \times 10^{-2}$	$8.005 \times 10^{-2}$	$7.900 \times 10^{-2}$	$7.925 \times 10^{-2}$	$1.113 \times 10^{-1}$	$1.008 \times 10^{-1}$
	STD	$8.158 \times 10^{-3}$	$5.286 \times 10^{-3}$	$7.716 \times 10^{-4}$	$7.940 \times 10^{-4}$	$1.060 \times 10^{-3}$	$8.750 \times 10^{-4}$	<b><math>6.880 \times 10^{-4}</math></b>

**Table 4.** Held-out fold performance of LightGBM models using 5-fold cross-validation.

		FOX-LightGBM	GTO-LightGBM	PSO-LightGBM	SCSO-LightGBM	SSA-LightGBM	LightGBM	SVR
$R^2$	AVG	0.9331	0.9343	0.9406	<b>0.9412</b>	0.9409	0.9074	0.9257
	STD	$4.131 \times 10^{-3}$	$4.544 \times 10^{-3}$	<b><math>1.238 \times 10^{-3}</math></b>	$2.060 \times 10^{-3}$	$1.963 \times 10^{-3}$	$1.672 \times 10^{-3}$	$2.589 \times 10^{-3}$
RMSE	AVG	$5.354 \times 10^{-2}$	$5.306 \times 10^{-2}$	$5.049 \times 10^{-2}$	<b><math>5.021 \times 10^{-2}</math></b>	$5.034 \times 10^{-2}$	$6.303 \times 10^{-2}$	$5.647 \times 10^{-2}$
	STD	$1.549 \times 10^{-3}$	$2.054 \times 10^{-3}$	$6.341 \times 10^{-4}$	$1.011 \times 10^{-3}$	<b><math>5.022 \times 10^{-4}</math></b>	$9.757 \times 10^{-4}$	$1.193 \times 10^{-3}$
MSE	AVG	$2.869 \times 10^{-3}$	$2.820 \times 10^{-3}$	$2.550 \times 10^{-3}$	<b><math>2.522 \times 10^{-3}</math></b>	$2.534 \times 10^{-3}$	$3.974 \times 10^{-3}$	$3.190 \times 10^{-3}$
	STD	$1.679 \times 10^{-4}$	$2.177 \times 10^{-4}$	$6.447 \times 10^{-5}$	$1.031 \times 10^{-4}$	<b><math>5.030 \times 10^{-5}</math></b>	$1.236 \times 10^{-4}$	$1.353 \times 10^{-4}$
ME	AVG	$1.755 \times 10^{-1}$	$1.694 \times 10^{-1}$	<b><math>1.620 \times 10^{-1}</math></b>	$1.637 \times 10^{-1}$	$1.638 \times 10^{-1}$	$2.071 \times 10^{-1}$	$2.117 \times 10^{-1}$
	STD	$1.543 \times 10^{-2}$	$1.391 \times 10^{-2}$	<b><math>6.862 \times 10^{-3}</math></b>	$8.939 \times 10^{-3}$	$1.586 \times 10^{-2}$	$2.015 \times 10^{-2}$	$2.909 \times 10^{-2}$
RAE	AVG	$1.032 \times 10^{-1}$	$1.022 \times 10^{-1}$	$9.731 \times 10^{-2}$	<b><math>9.677 \times 10^{-2}</math></b>	$9.701 \times 10^{-2}$	$1.215 \times 10^{-1}$	$1.088 \times 10^{-1}$
	STD	$3.456 \times 10^{-3}$	$3.861 \times 10^{-3}$	$2.182 \times 10^{-3}$	$2.745 \times 10^{-3}$	<b><math>1.314 \times 10^{-3}</math></b>	$3.084 \times 10^{-3}$	$3.314 \times 10^{-3}$

The held-out set results in Table 4 provide critical insights into model generalization, revealing shifts in relative performance that highlight the optimizers' robustness to unseen data. Here, SCSO-LightGBM emerged as the top performer with the highest average  $R^2$  (0.9412), followed closely by SSA-LightGBM (0.9409) and PSO-LightGBM (0.9406), while GTO-LightGBM (0.9343) and FOX-LightGBM (0.9331) lagged slightly behind. These rankings are mirrored in the error metrics: SCSO-LightGBM achieved the lowest RMSE (0.0502) and MSE ( $2.522 \times 10^{-3}$ ), with SSA-LightGBM and PSO-LightGBM exhibiting comparable values (RMSE: 0.0503 and 0.0505, respectively). In contrast, the baselines yielded higher errors, with LightGBM at RMSE 0.0630 and SVR at 0.0565. PSO-LightGBM recorded the lowest ME (0.1620), indicating minimal systematic bias, whereas SVR and unoptimized LightGBM displayed larger biases (0.2117 and 0.2071, respectively). RAE values further affirm the optimizers' advantages, with SCSO-LightGBM at the lowest (0.0968), compared to 0.1215 for LightGBM and 0.1088 for SVR. Standard deviations on the test set were generally low across models, but SCSO-LightGBM and PSO-LightGBM maintained particularly tight variances underscoring their consistency. Comparing training

and test performances reveals the optimizers' capacity for balanced generalization, with modest degradations in  $R^2$  and proportional increases in RMSE and MSE, indicative of effective hyperparameter tuning that mitigates overfitting. FOX-LightGBM, despite its training dominance, experienced the most pronounced relative drop in  $R^2$  increase in RMSE, suggesting a propensity for overfitting due to aggressive parameter optimization. Collectively, these results substantiate the efficacy of metaheuristic algorithms in enhancing LightGBM's hyperparameter optimization, with SCSO-LightGBM standing out as the most robust variant. It integrates superior accuracy, error minimization, generalization, and stability, making it a promising approach for the student performance prediction task.

#### 4.1.2. Independent Run Analysis

During independent run analysis, the data split is fixed throughout the experiment; the sole intended source of variability is the metaheuristic optimizer's stochastic initialization and update dynamics. To ensure the robustness and reliability of the performance evaluation, each metaheuristic-optimized LightGBM model was subjected to 20 independent runs using the same hyperparameter settings and experimental configuration from the previous experiment. The results from 20 independent runs, for training as detailed in Table 5, demonstrate the efficacy of metaheuristic-optimized LightGBM models in enhancing predictive performance on the training set. FOX-LightGBM achieved the highest average  $R^2$  of 0.9682, reflecting superior hyperparameter optimization, followed by GTO-LightGBM (0.9610), SCSO-LightGBM (0.9609), SSA-LightGBM (0.9608), and PSO-LightGBM (0.9590). These values substantially outperform the baselines, with unoptimized LightGBM and SVR recording  $R^2$  scores of 0.9229 and 0.9371, respectively. Error metrics further underscore the optimizers' advantages: FOX-LightGBM yielded the lowest average RMSE and MSE, compared to the unoptimized LightGBM and SVR. The relative absolute error (RAE) was minimized by FOX-LightGBM to 0.0703, outperforming unoptimized LightGBM (0.1106) and SVR (0.0999). The best-case performance of FOX-LightGBM, with an  $R^2$  of 0.9828, RMSE of 0.0273, and MSE of  $7.450 \times 10^{-4}$ , highlights its capability to achieve exceptional fits. However, its higher standard deviations indicate greater variability compared to SCSO-LightGBM and SSA-LightGBM, which exhibited greater stability. Notably, SVR achieved the lowest maximum error (ME) at 0.1005, suggesting fewer extreme deviations, though its overall error metrics remained less competitive.

Test set results in Table 6 reveal the generalization capabilities of the models. SSA-LightGBM and SCSO-LightGBM led with average  $R^2$  values of 0.9388 and 0.9386, respectively, followed by PSO-LightGBM (0.9364), while FOX-LightGBM (0.9332) and GTO-LightGBM (0.9316) trailed slightly. Error metrics show that SSA-LightGBM achieved the lowest RMSE of 0.0506 and MSE of  $2.559 \times 10^{-3}$ , with SCSO-LightGBM close behind with RMSE of 0.0507, and MSE of  $2.570 \times 10^{-3}$ . PSO-LightGBM remained competitive, whereas FOX-LightGBM and GTO-LightGBM exhibited higher errors. The baselines underperformed, with unoptimized LightGBM achieving the lowest  $R^2$  of 0.9032 and the highest RMSE of 0.0636, while SVR outperformed LightGBM but lagged behind the optimizers. RAE values further confirm the optimizers' superiority, with SSA-LightGBM (0.0995) and SCSO-LightGBM (0.0998) achieving the lowest, compared to 0.1252 for unoptimized LightGBM and 0.1141 for SVR. The best-case test performance highlights SCSO-LightGBM's edge, with an  $R^2$  of 0.9406 and MSE of  $2.486 \times 10^{-3}$ . SCSO-LightGBM also recorded the lowest ME (0.1610), indicating minimal systematic bias, while unoptimized LightGBM and SVR exhibited larger biases. Comparing training and test results reveals distinct generalization patterns. FOX-LightGBM, despite its training dominance, experienced a notable performance drop. The standard deviations across 20 runs highlight the reliability of SCSO-LightGBM and SSA-LightGBM, which exhibited the lowest variability. FOX-LightGBM

and GTO-LightGBM showed higher variability, indicating less consistent performance. The near-zero standard deviations for the baselines reflect their deterministic nature but do not compensate for their inferior accuracy. In conclusion, while all metaheuristic optimizations enhanced LightGBM's performance over the baselines, SCSO-LightGBM and SSA-LightGBM emerged as the most effective, combining superior predictive accuracy, robust generalization, and high stability across 20 independent runs.

**Table 5.** Training results from 20 independent runs using optimized LightGBM models.

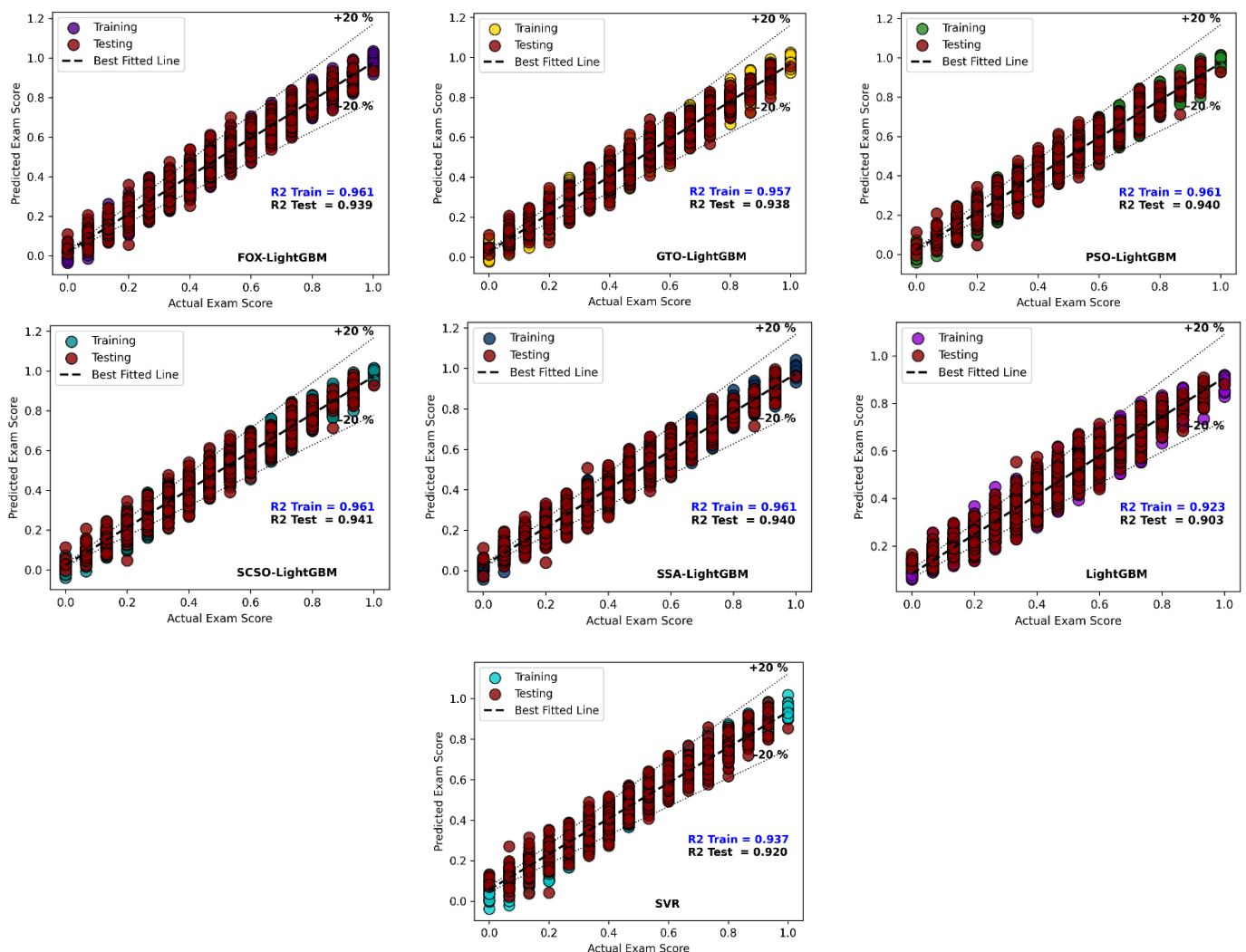
		FOX-LightGBM	GTO-LightGBM	PSO-LightGBM	SCSO-LightGBM	SSA-LightGBM	LightGBM	SVR
$R^2$	AVG	<b>0.9682</b>	0.9610	0.9590	0.9609	0.9608	0.9229	0.9371
	STD	$8.981 \times 10^{-3}$	$8.017 \times 10^{-3}$	$2.782 \times 10^{-3}$	$5.743 \times 10^{-4}$	$8.170 \times 10^{-4}$	$2.220 \times 10^{-16}$	<b>0</b>
	Best	<b>0.9828</b>	0.9732	0.9633	0.9618	0.9622	0.9229	0.9371
RMSE	AVG	<b><math>3.667 \times 10^{-2}</math></b>	$4.085 \times 10^{-2}$	$4.210 \times 10^{-2}$	$4.111 \times 10^{-2}$	$4.111 \times 10^{-2}$	$5.772 \times 10^{-2}$	$5.217 \times 10^{-2}$
	STD	$5.348 \times 10^{-3}$	$4.342 \times 10^{-3}$	$1.398 \times 10^{-3}$	$3.012 \times 10^{-4}$	$3.012 \times 10^{-4}$	<b><math>6.939 \times 10^{-18}</math></b>	$2.776 \times 10^{-17}$
	Best	<b><math>2.730 \times 10^{-2}</math></b>	$3.402 \times 10^{-2}$	$3.986 \times 10^{-2}$	$4.065 \times 10^{-2}$	$4.065 \times 10^{-2}$	$5.772 \times 10^{-2}$	$5.217 \times 10^{-2}$
MSE	AVG	<b><math>1.373 \times 10^{-3}</math></b>	$1.688 \times 10^{-3}$	$1.774 \times 10^{-3}$	$1.690 \times 10^{-3}$	$1.697 \times 10^{-3}$	$3.332 \times 10^{-3}$	$2.722 \times 10^{-3}$
	STD	$3.884 \times 10^{-4}$	$3.467 \times 10^{-4}$	$1.203 \times 10^{-4}$	$2.472 \times 10^{-5}$	$3.530 \times 10^{-5}$	$8.674 \times 10^{-19}$	<b><math>4.337 \times 10^{-19}</math></b>
	Best	<b><math>7.450 \times 10^{-4}</math></b>	$1.158 \times 10^{-3}$	$1.589 \times 10^{-3}$	$1.653 \times 10^{-3}$	$1.636 \times 10^{-3}$	$3.332 \times 10^{-3}$	$2.722 \times 10^{-3}$
ME	AVG	$1.340 \times 10^{-1}$	$1.483 \times 10^{-1}$	$1.521 \times 10^{-1}$	$1.444 \times 10^{-1}$	$1.432 \times 10^{-1}$	$1.980 \times 10^{-1}$	<b><math>1.005 \times 10^{-1}</math></b>
	STD	$1.588 \times 10^{-2}$	$1.516 \times 10^{-2}$	$1.253 \times 10^{-2}$	$7.535 \times 10^{-3}$	$8.347 \times 10^{-3}$	<b>0</b>	$4.163 \times 10^{-17}$
	Best	$1.006 \times 10^{-1}$	$1.248 \times 10^{-1}$	$1.323 \times 10^{-1}$	$1.343 \times 10^{-1}$	$1.331 \times 10^{-1}$	$1.980 \times 10^{-1}$	<b><math>1.005 \times 10^{-1}</math></b>
RAE	AVG	<b><math>7.028 \times 10^{-2}</math></b>	$7.829 \times 10^{-2}$	$8.069 \times 10^{-2}$	$7.880 \times 10^{-2}$	$7.895 \times 10^{-2}$	$1.106 \times 10^{-1}$	$9.999 \times 10^{-2}$
	STD	$1.025 \times 10^{-2}$	$8.323 \times 10^{-3}$	$2.679 \times 10^{-3}$	$5.773 \times 10^{-4}$	$8.205 \times 10^{-4}$	<b><math>2.776 \times 10^{-17}</math></b>	$4.163 \times 10^{-17}$
	Best	<b><math>5.233 \times 10^{-2}</math></b>	$6.521 \times 10^{-2}$	$7.640 \times 10^{-2}$	$7.791 \times 10^{-2}$	$7.751 \times 10^{-2}$	$1.106 \times 10^{-1}$	$9.999 \times 10^{-2}$

**Table 6.** Test results from 20 independent runs using optimized LightGBM models.

		FOX-LightGBM	GTO-LightGBM	PSO-LightGBM	SCSO-LightGBM	SSA-LightGBM	LightGBM	SVR
$R^2$	AVG	0.9332	0.9316	0.9364	0.9386	<b>0.9388</b>	0.9032	0.9196
	STD	$3.275 \times 10^{-3}$	$3.517 \times 10^{-3}$	$2.868 \times 10^{-3}$	$1.296 \times 10^{-3}$	$6.900 \times 10^{-4}$	$4.441 \times 10^{-16}$	<b><math>2.220 \times 10^{-16}</math></b>
	Best	0.9391	0.9382	0.9404	<b>0.9406</b>	0.9399	0.9032	0.9196
RMSE	AVG	$5.283 \times 10^{-2}$	$5.348 \times 10^{-2}$	$5.156 \times 10^{-2}$	$5.069 \times 10^{-2}$	<b><math>5.058 \times 10^{-2}</math></b>	$6.363 \times 10^{-2}$	$5.799 \times 10^{-2}$
	STD	$1.294 \times 10^{-3}$	$1.366 \times 10^{-3}$	$1.152 \times 10^{-3}$	$5.344 \times 10^{-4}$	$2.851 \times 10^{-4}$	$1.388 \times 10^{-17}$	<b>0</b>
	Best	$5.048 \times 10^{-2}$	$5.086 \times 10^{-2}$	$4.995 \times 10^{-2}$	<b><math>4.986 \times 10^{-2}</math></b>	$5.015 \times 10^{-2}$	$6.363 \times 10^{-2}$	$5.799 \times 10^{-2}$
MSE	AVG	$2.793 \times 10^{-3}$	$2.862 \times 10^{-3}$	$2.660 \times 10^{-3}$	$2.570 \times 10^{-3}$	<b><math>2.559 \times 10^{-3}</math></b>	$4.048 \times 10^{-3}$	$3.363 \times 10^{-3}$
	STD	$1.370 \times 10^{-4}$	$1.471 \times 10^{-4}$	$1.199 \times 10^{-4}$	$5.428 \times 10^{-5}$	$2.898 \times 10^{-5}$	<b>0</b>	<b>0</b>
	Best	$2.548 \times 10^{-3}$	$2.586 \times 10^{-3}$	$2.495 \times 10^{-3}$	<b><math>2.486 \times 10^{-3}</math></b>	$2.515 \times 10^{-3}$	$4.048 \times 10^{-3}$	$3.363 \times 10^{-3}$
ME	AVG	$1.743 \times 10^{-1}$	$1.816 \times 10^{-1}$	$1.758 \times 10^{-1}$	<b><math>1.610 \times 10^{-1}</math></b>	$1.657 \times 10^{-1}$	$2.201 \times 10^{-1}$	$2.037 \times 10^{-1}$
	STD	$9.157 \times 10^{-3}$	$1.867 \times 10^{-2}$	$1.419 \times 10^{-2}$	$1.162 \times 10^{-2}$	$1.029 \times 10^{-2}$	<b><math>2.776 \times 10^{-17}</math></b>	$5.551 \times 10^{-17}$
	Best	$1.502 \times 10^{-1}$	$1.608 \times 10^{-1}$	$1.510 \times 10^{-1}$	$1.508 \times 10^{-1}$	<b><math>1.476 \times 10^{-1}</math></b>	$2.201 \times 10^{-1}$	$2.037 \times 10^{-1}$
RAE	AVG	$1.040 \times 10^{-1}$	$1.053 \times 10^{-1}$	$1.015 \times 10^{-1}$	$9.976 \times 10^{-2}$	<b><math>9.955 \times 10^{-2}</math></b>	$1.252 \times 10^{-1}$	$1.141 \times 10^{-1}$
	STD	$2.546 \times 10^{-3}$	$2.688 \times 10^{-3}$	$2.268 \times 10^{-3}$	$1.052 \times 10^{-3}$	$5.610 \times 10^{-4}$	<b>0</b>	$6.939 \times 10^{-17}$
	Best	$9.935 \times 10^{-2}$	$1.001 \times 10^{-1}$	$9.831 \times 10^{-2}$	<b><math>9.814 \times 10^{-2}</math></b>	$9.870 \times 10^{-2}$	$1.252 \times 10^{-1}$	$1.141 \times 10^{-1}$

The scatter plots in Figure 2 illustrate the relationship between the actual and predicted exam scores for both training and testing sets, using the best test outcome with the corresponding training result from the 20 independent trials for each model. FOX-LightGBM

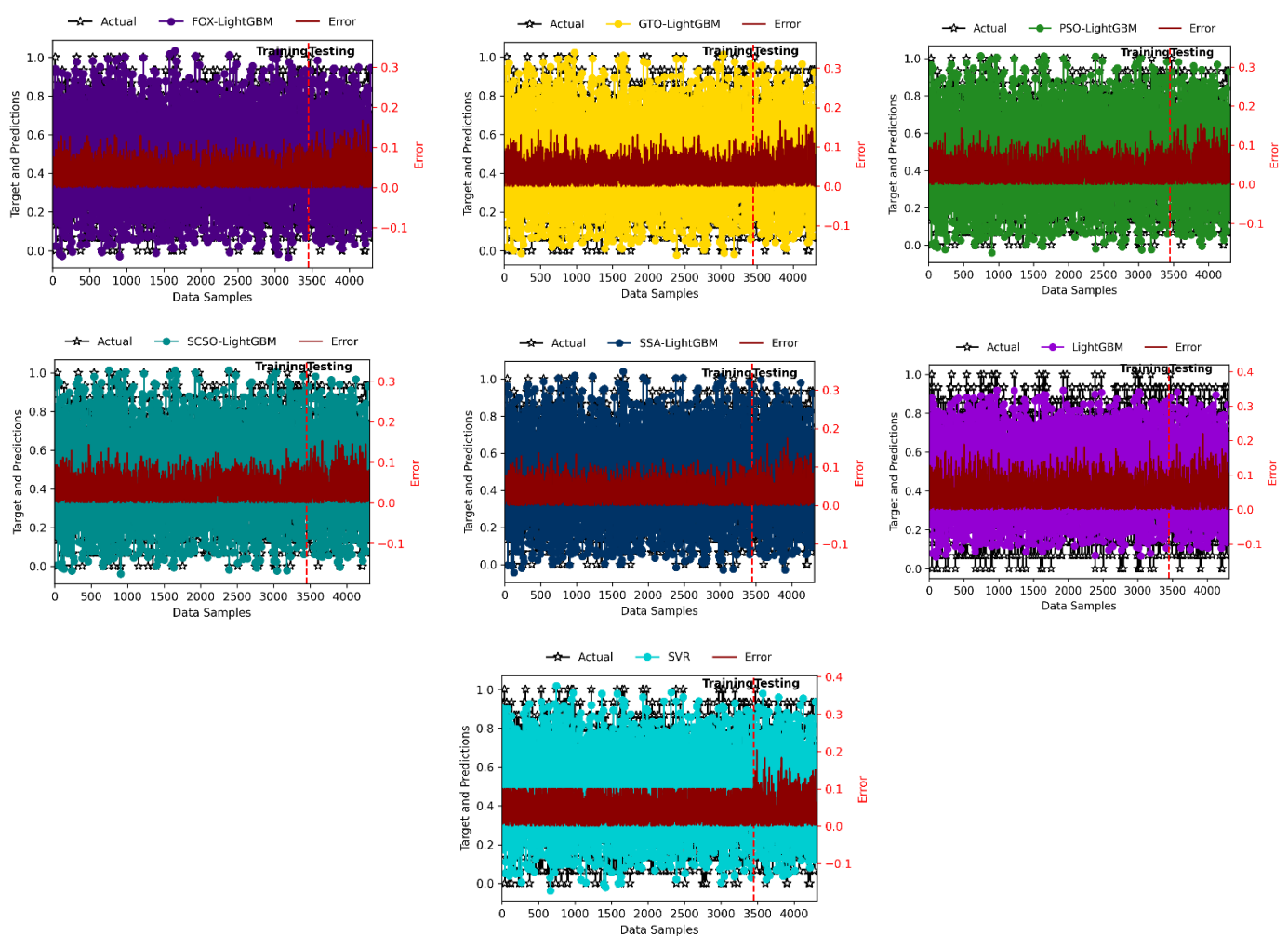
achieved the highest training but exhibited a slightly lower testing  $R^2$ , suggesting stronger fitting on the training data but reduced generalization compared to other optimized variants. GTO-LightGBM followed a similar pattern with a training  $R^2$  of 0.957 and a testing  $R^2$  of 0.938, reflecting adequate performance but a modestly larger train-test gap. PSO-LightGBM balanced training and testing outcomes well, while SSA-LightGBM performed, demonstrating consistent predictive capacity across both phases. Notably, SCSO-LightGBM achieved the highest testing accuracy  $R^2$  of 0.941 with a matching training  $R^2$  of 0.961, making it the best generalizing model. The alignment of its predicted points along the fitted line in both training and testing highlights minimal deviation and reduced bias compared with other optimizers. In contrast, the baseline models underperformed: unoptimized LightGBM yielded the lowest test  $R^2$  (0.903), with points more widely dispersed around the fitted line, while SVR achieved moderate accuracy but still lagged behind the optimized LightGBM models. Collectively, these plots confirm that while FOX-LightGBM dominated training, SCSO-LightGBM and SSA-LightGBM achieved superior test generalization, with SCSO-LightGBM emerging as the most reliable and effective method overall.



**Figure 2.** Actual vs. predicted plots of compared models.

Figure 3 presents the prediction trends of each model over the entire dataset, partitioned into training and testing phases. The black markers represent the actual values, while the colored markers indicate the model predictions. The dark red curves show the absolute prediction error, providing an intuitive view of both accuracy and stability across

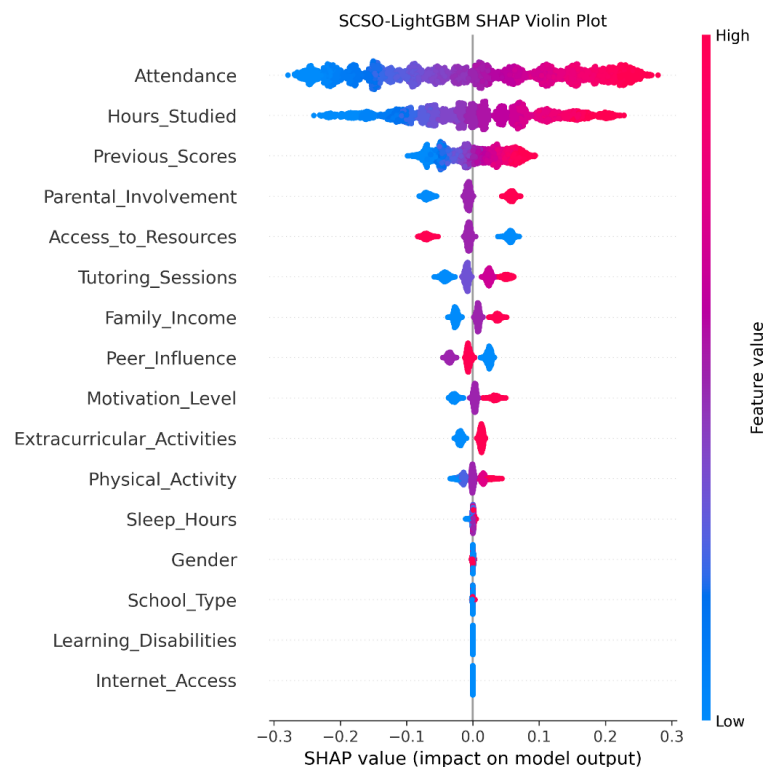
samples. The plot shows the absolute error between the best test result of 20 independent runs, including the corresponding training output. For the optimized LightGBM models, the prediction lines closely follow the actual values across both training and testing sets, confirming that the metaheuristic-tuned models successfully captured the underlying data distribution. Among them, SCSO-LightGBM and SSA-LightGBM achieved the most stable error bands, with the red error line remaining nearly flat and centered around zero across samples, which reflects high robustness and minimal bias. PSO-LightGBM also exhibited a consistently narrow error band, demonstrating reliable prediction quality. FOX-LightGBM, despite strong training accuracy, displayed slightly larger fluctuations in the testing region, suggesting reduced stability compared with SCSO-LightGBM and SSA-LightGBM models. Similarly, GTO-LightGBM performed reasonably well but showed slightly higher deviations in its error distribution. The baseline models demonstrated weaker predictive capacity. Unoptimized LightGBM presented the widest dispersion between actual and predicted values, accompanied by a visibly higher error band, particularly in the testing phase. SVR performed better than unoptimized LightGBM, with reduced deviation, but its error curve remained more variable compared to the metaheuristic-optimized LightGBM models. The consistently low and stable error distributions in SCSO-LightGBM and SSA-LightGBM confirm their superior generalization and reliability. These results underscore that, while all metaheuristic optimizations improve LightGBM compared to the baselines, SCSO-LightGBM exhibits the most accurate and stable performance across both training and testing samples, making it the best overall model in this study.



**Figure 3.** Absolute error plots of training and test results of compared models.



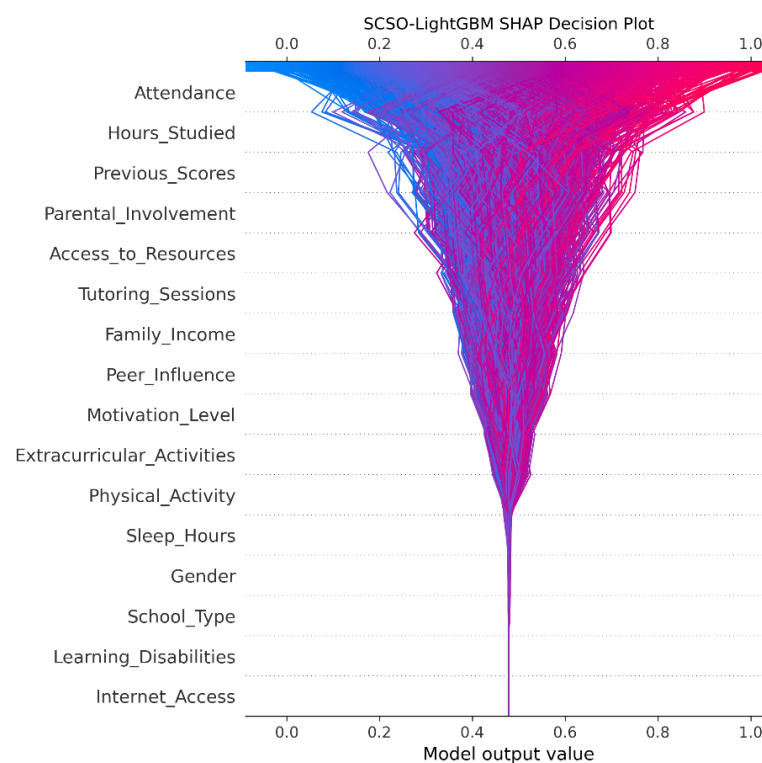
The SHAP violin plot in Figure 4 provides a detailed view of how individual feature values influenced the model predictions in SCSO-LightGBM. The SCSO-LightGBM model is used to understand the most contributive features during the LightGBM prediction process due to its robust performance. Each distribution shows the spread of SHAP values, with the color indicating whether the feature value is high (red) or low (blue). This allows for an examination not only of the importance of features but also of the direction and magnitude of their influence. Attendance exerted the strongest largest positive influence: higher attendance (red points) was consistently associated with positive SHAP values, increasing the predicted exam score, while lower attendance (blue points) shifted predictions downward. Hours Studied showed a similar pattern, with more study hours leading to higher predicted scores, underscoring the strong contribution of academic effort to performance outcomes. Previous Scores also had a positive influence, though with a narrower distribution, reflecting their role as a stable baseline indicator of performance. Moderately influential features such as Parental Involvement, Access to Resources, and Tutoring Sessions exhibited both positive and negative SHAP contributions, suggesting that their impact varies across students depending on context. Features including Family Income, Peer Influence, and Motivation Level produced smaller yet consistent effects, often reinforcing or moderating the impact of the primary academic factors. By contrast, features such as Sleep Hours, School Type, Gender, Learning Disabilities, and Internet Access displayed SHAP values centered closely around zero, indicating negligible influence on model outputs. This further validates that the model prioritized behavioral and academic engagement variables over demographic or structural factors in predicting exam performance. Taken together, the violin plot emphasizes that academic engagement (attendance, study hours, previous scores) and supportive contexts (resources, parental involvement) drive the largest shifts in predictions, whereas demographic variables contribute minimally. This reinforces the interpretability of SCSO-LightGBM, confirming its alignment with intuitive educational predictors while providing nuanced insights into how features interact to shape performance outcomes.



**Figure 4.** SHAP violin plot of SCSO-LightGBM model.



The SHAP decision plot in Figure 5 provides a detailed visualization of how individual features contribute to the cumulative prediction process in the SCSO-LightGBM model. Each line represents a single sample's path from the model's base value (average prediction) toward the final output probability, with contributions from features added sequentially. Features displayed at the top, such as Attendance and Hours Studied, exert the largest shifts in prediction, consistent with the importance rankings shown in the SHAP violin plot. Specifically, high attendance and increased study hours consistently push predictions upward (toward higher exam scores), while low values in these features shift the predictions downward. Previous Scores, Access to Resources, and Parental Involvement also exert moderate influence, though their impact is more variable across samples, as indicated by the spread of lines. Features such as Tutoring Sessions, Family Income, and Peer Influence provide smaller but still noticeable shifts in prediction, reflecting their secondary role in determining outcomes. The remaining features, including Physical Activity, Sleep Hours, School Type, Gender, Learning Disabilities, and Internet Access, contributed minimally to decision paths, consistent with their low SHAP importance scores. Conclusively, the decision plot illustrates the interpretability strength of the SCSO-LightGBM model: predictions are largely driven by academic effort (study hours, previous scores), behavioral engagement (attendance, parental involvement), and resource availability. Importantly, the convergence of most decision paths around the correct prediction zone indicates that the model's decisions are not only accurate but also stable across different samples, thereby validating the reliability of SCSO-LightGBM for educational outcome prediction. The SHAP decision plot (Figure 5) further highlights interaction effects, where Parental Involvement and Access to Resources often amplify the positive contributions of Attendance and Previous Scores, while attenuating their negative impacts when these core features are low. Subgroup-level analysis confirms this pattern: among students; SHAP gains are driven primarily by Attendance and Hours Studied.



**Figure 5.** SHAP decision plot of SCSO-LightGBM model.

## 5. Conclusions

This study evaluated the efficacy of metaheuristic hyperparameter optimization in enhancing the performance of LightGBM for student performance prediction. Five swarm-based optimization algorithms FOX, GTO, PSO, SCSO, and SSA were benchmarked against baseline models. The experimental design employed 5-fold cross-validation, with 20 independent runs to mitigate stochastic variability, while performance was assessed using multiple metrics ( $R^2$ , RMSE, MSE, RAE, and ME). Results demonstrated that metaheuristic optimization consistently outperformed the baseline models. SCSO-LightGBM exhibited superior generalization on test datasets. In contrast, unoptimized LightGBM and SVR consistently underperformed across all metrics, reinforcing the value of systematic hyperparameter tuning. SHAP further validated the models' reliability. For the best-performing models, the most influential predictors were Attendance, Hours Studied, and Previous Scores, with secondary contributions from Access to Resources and Parental Involvement. These findings, consistent across mean importance, decision, and violin plots, align with educational domain knowledge, providing credible and actionable insights for policy development.

It is important to acknowledge that while the findings offer valuable insights. The findings of this study are constrained by reliance on a single dataset, which inherently limits the generalizability of the results across diverse domains, noise regimes, and feature distributions. Furthermore, the computational overhead associated with the deployment of metaheuristic optimizers constitutes a significant practical limitation, particularly in resource-constrained or latency-sensitive applications. Additionally, the comparative evaluation of metaheuristic algorithms was conducted under a fixed iteration, which may not fully capture their convergence behavior or relative efficiency under dynamic or adaptive stopping criteria. Future research should prioritize the development and evaluation of multi-objective optimization frameworks that jointly optimize predictive accuracy, training efficiency, model complexity, and fairness-aware metrics such as bias mitigation. To enhance scalability and practical utility, multi-fidelity search strategies incorporating early stopping mechanisms will be investigated. Robustness evaluations under realistic distributional shifts, adversarial noise, and outlier contamination are also critical to assess real-world applicability. Subsequent studies will extend the current analysis by benchmarking metaheuristic optimizers against established non-metaheuristic baselines, including grid search, random search, and Bayesian optimization. This comparative framework will elucidate the trade-offs inherent in different optimization paradigms. Additionally, future experiments will systematically vary the number of optimization iterations to evaluate convergence properties.

**Author Contributions:** A.A. (Abdalmid Abukader): Methodology, Formal Analysis, Original Draft, A.A. (Ahmad Alzubi): Supervision, Resources, Editing. O.R.A.: Conceptualization, Supervision, Resources, Editing, Methodology, Original Draft. All authors have read and agreed to the published version of the manuscript.

**Funding:** Authors declare no funding was received.

**Data Availability Statement:** The data obtained through the experiments are available upon request from the corresponding author.

**Conflicts of Interest:** The authors declare that there are no conflicts of interest.

## References

1. Saksiriruthai, S. Human Capital as a Determinant of Long-Term Economic Growth. In *Research Anthology on Preparing School Administrators to Lead Quality Education Programs*; IGI Global Scientific Publishing: Hershey, PA, USA, 2021; pp. 1518–1533. [\[CrossRef\]](#)
2. Baumann, C.; Hamin. The role of culture, competitiveness and economic performance in explaining academic performance: A global market analysis for international student segmentation. *J. Mark. High. Educ.* **2011**, *21*, 181–201. [\[CrossRef\]](#)
3. Barro, R.J.; McCleary, R.M. Religion and Economic Growth across Countries. *Am. Sociol. Rev.* **2003**, *68*, 760–781. [\[CrossRef\]](#)
4. Barro, R.J. *Determinants of Economic Growth: A Cross-Country Empirical Study*; National Bureau of Economic Research: Cambridge, MA, USA, 1996; p. 5698. [\[CrossRef\]](#)
5. Sultana, T.; Dey, S.R.; Tareque, M. Exploring the linkage between human capital and economic growth: A look at 141 developing and developed countries. *Econ. Syst.* **2022**, *46*, 101017. [\[CrossRef\]](#)
6. Ahmed, W.; Wani, M.A.; Plawiak, P.; Meshoul, S.; Mahmoud, A.; Hammad, M. Machine learning-based academic performance prediction with explainability for enhanced decision-making in educational institutions. *Sci. Rep.* **2025**, *15*, 26879. [\[CrossRef\]](#)
7. Vaarma, M.; Li, H. Predicting student dropouts with machine learning: An empirical study in Finnish higher education. *Technol. Soc.* **2024**, *76*, 102474. [\[CrossRef\]](#)
8. Dwivedi, D.N.; Mahanty, G.; Dwivedi, V.N. The Role of Predictive Analytics in Personalizing Education: Tailoring Learning Paths for Individual Student Success. In *Enhancing Education with Intelligent Systems and Data-Driven Instruction*; IGI Global Scientific Publishing: Hershey, PA, USA, 2024; pp. 44–59. [\[CrossRef\]](#)
9. Pelima, L.R.; Sukmana, Y.; Rosmansyah, Y. Predicting University Student Graduation Using Academic Performance and Machine Learning: A Systematic Literature Review. *IEEE Access* **2024**, *12*, 23451–23465. [\[CrossRef\]](#)
10. Ahmed, A.A.; Sayed, S.; Abdoulhalik, A.; Moutari, S.; Oyedele, L. Applications of machine learning to water resources management: A review of present status and future opportunities. *J. Clean. Prod.* **2024**, *441*, 140715. [\[CrossRef\]](#)
11. Ke, W.; Chan, K.-H. A Multilayer CARU Framework to Obtain Probability Distribution for Paragraph-Based Sentiment Analysis. *Appl. Sci.* **2021**, *11*, 11344. [\[CrossRef\]](#)
12. Xing, Z.; Lam, C.-T.; Yuan, X.; Im, S.-K.; Machado, P. MMQW: Multi-Modal Quantum Watermarking Scheme. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 5181–5195. [\[CrossRef\]](#)
13. Alnasyan, B.; Basher, M.; Alassafi, M. The power of Deep Learning techniques for predicting student performance in Virtual Learning Environments: A systematic literature review. *Comput. Educ. Artif. Intell.* **2024**, *6*, 100231. [\[CrossRef\]](#)
14. Hasan, R.; Palaniappan, S.; Raziff, A.R.A.; Mahmood, S.; Sarker, K.U. Student Academic Performance Prediction by using Decision Tree Algorithm. In Proceedings of the 2018 4th International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia, 13–14 August 2018; pp. 1–5.
15. Alamri, H.L.; S. Almuslim, R.; S. Alotibi, M.; K. Alkadi, D.; Ullah Khan, I.; Aslam, N. Predicting Student Academic Performance using Support Vector Machine and Random Forest. In Proceedings of the 2020 3rd International Conference on Education Technology Management, London, UK, 17–19 December 2020; Association for Computing Machinery: New York, NY, USA, 2021; pp. 100–107.
16. Salah Hashim, A.; Akeel Awadh, W.; Khalaf Hamoud, A. Student Performance Prediction Model based on Supervised Machine Learning Algorithms. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *928*, 032019. [\[CrossRef\]](#)
17. Sravani, B.; Bala, M.M. Prediction of Student Performance Using Linear Regression. In Proceedings of the 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 5–7 June 2020; pp. 1–5.
18. Tripathi, A.; Yadav, S.; Rajan, R. Naive Bayes Classification Model for the Student Performance Prediction. In Proceedings of the 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, India, 5–6 July 2019; Volume 1, pp. 1548–1553.
19. Huang, C.; Zhou, J.; Chen, J.; Yang, J.; Clawson, K.; Peng, Y. A feature weighted support vector machine and artificial neural network algorithm for academic course performance prediction. *Neural Comput. Appl.* **2023**, *35*, 11517–11529. [\[CrossRef\]](#)
20. Ahmed, S.T.; Al-Hamdani, R.; Croock, M.S. Enhancement of student performance prediction using modified K-nearest neighbor. *TELKOMNIKA (Telecommun. Comput. Electron. Control)* **2020**, *18*, 1777–1783. [\[CrossRef\]](#)
21. Baashar, Y.; Alkawsi, G.; Mustafa, A.; Alkahtani, A.A.; Alsariera, Y.A.; Ali, A.Q.; Hashim, W.; Tiong, S.K. Toward Predicting Student's Academic Performance Using Artificial Neural Networks (ANNs). *Appl. Sci.* **2022**, *12*, 1289. [\[CrossRef\]](#)
22. Parrales-Bravo, F.; Caicedo-Quiroz, R.; Barzola-Monteses, J.; Cevallos-Torres, L. Applying Bayesian Networks to Predict and Understand the Student Academic Performance. In Proceedings of the 2024 Second International Conference on Advanced Computing & Communication Technologies (ICACCTech), Sonipat, India, 16–17 November 2024; pp. 738–743.
23. Yang, S.J.H.; Lu, O.H.T.; Huang, A.Y.Q.; Huang, J.C.H.; Ogata, H.; Lin, A.J.Q. Predicting Students' Academic Performance Using Multiple Linear Regression and Principal Component Analysis. *J. Inf. Process.* **2018**, *26*, 170–176. [\[CrossRef\]](#)
24. Wilson, A.; Anwar, M.R. The Future of Adaptive Machine Learning Algorithms in High-Dimensional Data Processing. *Int. Trans. Artif. Intell.* **2024**, *3*, 97–107. [\[CrossRef\]](#)

25. Yaqoob, A.; Verma, N.K.; Aziz, R.M. Metaheuristic Algorithms and Their Applications in Different Fields. In *Metaheuristics for Machine Learning*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2024; pp. 1–35. [\[CrossRef\]](#)
26. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2017. Available online: [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf) (accessed on 29 August 2025).
27. Xu, H.; Kim, M. Combination prediction method of students' performance based on ant colony algorithm. *PLoS ONE* **2024**, *19*, e0300010. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Fang, R.; Zhou, T.; Yu, B.; Li, Z.; Ma, L.; Luo, T.; Zhang, Y.; Liu, X. Prediction model of middle school student performance based on MBSO and MDBO-BP-Adaboost method. *Front. Big Data* **2025**, *7*, 1518939. [\[CrossRef\]](#) [\[PubMed\]](#)
29. Cheng, B.; Liu, Y.; Jia, Y. Evaluation of students' performance during the academic period using the XG-Boost Classifier-Enhanced AEO hybrid model. *Expert Syst. Appl.* **2024**, *238*, 122136. [\[CrossRef\]](#)
30. Lowast, A.E.; Amalarethinam, D.I.G. A Hybrid Model of Enhanced Teacher Learner Based Optimization (ETLBO) with Particle Swarm Optimization (PSO) Algorithm for Predicting Academic Student Performance. *INDJST* **2025**, *18*, 772–783. [\[CrossRef\]](#)
31. Li, G.; Cui, J.; Fu, H.; Sun, Y. Light GBM and GA Based Algorithm for Predicting and Improving Students' Performance in Higher Education in the Context of Big Data. In *Proceedings of the 2024 7th International Conference on Education, Network and Information Technology (ICENIT)*, Dalian, China, 16–18 August 2024; pp. 1–5. [\[CrossRef\]](#)
32. Apriyadi, M.R.; Ermatita; Rini, D.P. Hyperparameter Optimization of Support Vector Regression Algorithm using Metaheuristic Algorithm for Student Performance Prediction. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 144–150. [\[CrossRef\]](#)
33. Kamal, M.; Chakrabarti, S.; Ramirez-Asis, E.; Asis-López, M.; Allauca-Castillo, W.; Kumar, T.; Sanchez, D.T.; Rahmani, A.W. Metaheuristics Method for Classification and Prediction of Student Performance Using Machine Learning Predictors. *Math. Probl. Eng.* **2022**, *2022*, 2581951. [\[CrossRef\]](#)
34. Xu, H. Prediction of Students' Performance Based on the Hybrid IDA-SVR Model. *Complexity* **2022**, *2022*, 1845571. [\[CrossRef\]](#)
35. Song, X. Student performance prediction employing k-Nearest Neighbor Classification model and meta-heuristic algorithms. *Multiscale Multidiscip. Model. Exp. Des.* **2024**, *7*, 4397–4412. [\[CrossRef\]](#)
36. Ma, C. Improving the Prediction of Student Performance by Integrating a Random Forest Classifier with Meta-Heuristic Optimization Algorithms. | EBSCOhost. Available online: <https://openurl.ebsco.com/contentitem/doi:10.14569%2Fijacsa.2024.01506106?sid=ebsco:plink:crawler&id=ebsco:doi:10.14569%2Fijacsa.2024.01506106> (accessed on 30 August 2025).
37. Ali, M.; Liaquat, M.D.; Atta, M.N.; Khan, A.; Lashari, S.A.; Ramli, D.A. Improving Student Performance Prediction Using a PCA-based Cuckoo Search Neural Network Algorithm. *Procedia Comput. Sci.* **2023**, *225*, 4598–4610. [\[CrossRef\]](#)
38. Punitha, S.; Devaki, K. A high ranking-based ensemble network for student's performance prediction using improved meta-heuristic-aided feature selection and adaptive GAN for recommender system. *Kybernetes* **2024**. [\[CrossRef\]](#)
39. Hai, Q.; Wang, C. Optimizing Student Performance Prediction: A Data Mining Approach with MLPC Model and Metaheuristic Algorithm. | EBSCOhost. Available online: <https://openurl.ebsco.com/contentitem/doi:10.14569%2Fijacsa.2024.0150407?sid=ebsco:plink:crawler&id=ebsco:doi:10.14569%2Fijacsa.2024.0150407> (accessed on 30 August 2025).
40. Mohammed, H.; Rashid, T. FOX: A FOX-inspired optimization algorithm. *Appl. Intell.* **2023**, *53*, 1030–1050. [\[CrossRef\]](#)
41. Sadeeq, H.T.; Abdulazeez, A.M. Giant Trevally Optimizer (GTO): A Novel Metaheuristic Algorithm for Global Optimization and Challenging Engineering Problems. *IEEE Access* **2022**, *10*, 121615–121640. [\[CrossRef\]](#)
42. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Proceedings of the Proceedings of ICNN'95—International Conference on Neural Networks*, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
43. Seyyedabbasi, A.; Kiani, F. Sand Cat swarm optimization: A nature-inspired algorithm to solve global optimization problems. *Eng. Comput.* **2022**, *36*, 2627–2651. [\[CrossRef\]](#)
44. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [\[CrossRef\]](#)
45. Wang, J.; Chi, J.; Ding, Y.; Yao, H.; Guo, Q. Based on PCA and SSA-LightGBM oil-immersed transformer fault diagnosis method. *PLoS ONE* **2025**, *20*, e0314481. [\[CrossRef\]](#) [\[PubMed\]](#)
46. Florek, P.; Zagdański, A. Benchmarking state-of-the-art gradient boosting algorithms for classification. *arXiv* **2023**, arXiv:2305.17094. [\[CrossRef\]](#)
47. Yang, H.; Chen, Z.; Yang, H.; Tian, M. Predicting Coronary Heart Disease Using an Improved LightGBM Model: Performance Analysis and Comparison. *IEEE Access* **2023**, *11*, 23366–23380. [\[CrossRef\]](#)
48. Alshari, H.; Saleh, A.; Odabas, A. Comparison of Gradient Boosting Decision Tree Algorithms for CPU Performance. *Erciyes Tip Derg.* **2021**, 157–168.
49. Farhan, M.; Chen, T.; Rao, A.; Shahid, M.I.; Xiao, Q.; Salam, H.A.; Ma, F. An experimental study of knock analysis of HCNG fueled SI engine by different methods and prediction of knock intensity by particle swarm optimization-support vector machine. *Energy* **2024**, *309*, 133165. [\[CrossRef\]](#)

50. Sun, C.; Qin, W.; Yun, Z. A State-of-Health Estimation Method for Lithium Batteries Based on Fennec Fox Optimization Algorithm–Mixed Extreme Learning Machine. *Batteries* **2024**, *10*, 87. [CrossRef]
51. Tan, C.; Sun, M.; Liu, W.; Tan, W.; Zhang, X.; Zhu, C.; Li, D. An Adaptive Layering Dual-Parameter Regularization Inversion Method for an Improved Giant Trevally Optimizer Algorithm. *IEEE Access* **2024**, *12*, 160761–160775. [CrossRef]
52. Punitha, A.; Ramani, P.; Ezhilarasi, P.; Sridhar, S. Dynamically stabilized recurrent neural network optimized with intensified sand cat swarm optimization for intrusion detection in wireless sensor network. *Comput. Secur.* **2025**, *148*, 104094. [CrossRef]
53. Yang, Z.; Jiang, Y.; Yeh, W.-C. Self-learning salp swarm algorithm for global optimization and its application in multi-layer perceptron model training. *Sci. Rep.* **2024**, *14*, 27401. [CrossRef]
54. Student Performance Factors. Available online: <https://www.kaggle.com/datasets/lainguyn123/student-performance-factors> (accessed on 17 August 2025).
55. Wang, T.; Zhang, M.; Li, Z. Explainable machine learning links erosion damage to environmental factors on Gansu rammed earth Great Wall. *npj Herit. Sci.* **2025**, *13*, 366. [CrossRef]
56. Ahmed, A.H.A.; Jin, W.; Ali, M.A.H. Prediction of compressive strength of recycled concrete using gradient boosting models. *Ain Shams Eng. J.* **2024**, *15*, 102975. [CrossRef]
57. Haryadi, D.; Hakim, A.R.; Atmaja, D.M.U.; Yutia, S.N. Implementation of Support Vector Regression for Polkadot Cryptocurrency Price Prediction. *JOIV Int. J. Inform. Vis.* **2022**, *6*, 201–207. [CrossRef]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.