RESEARCH ARTICLE

# A Jellyfish Search Optimizer-Based Optimization Framework for Student Performance Prediction

**Shaoyuan Weng**[1,*], **Zimeng Liu**[1], **Yuanyuan Zheng**[1] **and Chao Zhang**[1]

[1] Xiamen Institute of Software Technology, Xiamen 361024, China

## Abstract

**Student performance prediction represents a core task in educational data mining, facilitating early interventions, personalized learning support, and data-driven decision-making. While machine learning models have demonstrated strong predictive capabilities in this domain, their effectiveness remains constrained by hyperparameter selection. To overcome this limitation, we introduce an automated hyperparameter optimization framework that leverages the jellyfish search optimizer to identify optimal configurations. To mitigate the variability introduced by data partitioning, we adopt 10-fold cross-validation with 10 repeated trials. Experimental results indicate that the proposed framework significantly enhances the performance of baseline models across all evaluated metrics. Leveraging this superior performance, the framework provides a robust tool for student performance prediction and a wide array of educational analytics applications.**

**Keywords**: automatic parameter optimization, jellyfish search optimizer, student performance prediction, educational data mining.

**Submitted:** 30 September 2025
**Accepted:** 11 October 2025
**Published:** 22 November 2025

**Vol.** 1, **No.** 1, 2025.
10.62762/TEDM.2025.736642

***Corresponding author:**
Shaoyuan Weng
wengshaoyuany@163.com

## 1 Introduction

As the volume of educational data collected from the learning management system and student assessment platforms continues to grow, new opportunities for Educational Data Mining (EDM) have emerged [1, 2]. EDM refers to the application of data-driven techniques to analyze educational datasets, with the goal of understanding learning processes and optimizing outcomes for learners [3]. In EDM, student performance prediction has become an important application, which underpins early-warning and dropout-prevention systems [4–6]. For educators, accurate prediction of student performance can help them deliver early interventions, substantially lowering dropout rates and boosting course engagement [7]. In addition, identifying students needing extra support allows better allocation of resources for them [8].

To accurately predict student performance, single and ensemble machine learning models have been employed [9]. For example, in [10], machine learning models were used, and they found that the Support Vector Machine (SVM) had the best performance after hyper-parameter tuning. For ensemble models, Hong et al. [11] utilized Random Forest (RF) to predict at-risk student status with high performance. Similarly, Cheng et al. [12] applied eXtreme Gradient Boosting (XGBoost) for student performance prediction. Their results showed

**Citation**
Weng, S., Liu, Z., Zheng, Y., & Zhang, C. (2025). A Jellyfish Search Optimizer-Based Optimization Framework for Student Performance Prediction. *ICCK Transactions on Educational Data Mining*, 1(1), 6–15.

that the XGBoost delivered the best performance. Considering the prevalence of categorical features in the educational data and that the Categorical Boosting (CatBoost) is effective in addressing such tasks, Fan et al. [13] employed CatBoost for student performance prediction, outperforming the SVM, RF, and XGBoost.

However, the hyper-parameters are critical for the performance of machine learning models. To address this problem, many swarm intelligent algorithms were introduced for hyper-parameter tuning in EDM. For example, Fan et al. [14] designed a double particle swarm optimization-based CatBoost to optimize the hyper-parameter of CatBoost and the threshold. Their results showed the best accuracy of 96.62% for Mathematics and 94.45% for Portuguese courses. Mahawar and Rattan [15] introduced an optimized decision tree classifier by using the ant colony optimization technique [16] for hyper-parameter tuning in early student academic performance prediction. Zheng et al. [? ] employed the Naive Bayes classifier to predict student performance by integrating the Jellyfish Search Optimizer (JSO) [17] and artificial rabbits optimization [18]. Their results showed improved performance through hyper-parameter optimization.

Inspired by the above study, we propose an automatic parameter optimization framework for student performance prediction that leverages the JSO to efficiently tune the hyper-parameters of machine learning models. The framework integrates JSO's exploration–exploitation mechanism with student performance datasets to identify optimal hyper-parameters that improve predictions in terms of all metrics used.

The main contributions of this work are as follows:

(1) We develop a JSO-based framework that automatically tunes hyper-parameters of prediction models for student performance prediction.

(2) By leveraging the exploration–exploitation balance of the JSO, the framework is able to improve the predictive performance.

(3) We provide a comprehensive evaluation based on the educational datasets and show improve performance by using the framework.

## 2 Methodology

In this section, we first present the jellyfish search optimizer, followed by an overview of the prediction models, and then describe the JSO-based framework for student performance prediction.

### 2.1 Jellyfish search optimizer

The Jellyfish Search Optimizer (JSO) is a nature-inspired metaheuristic algorithm that emulates the behavior of jellyfish in the ocean [19]. It features two primary movement strategies: passive drifting with ocean currents and active swimming. In the exploration phase, jellyfish drift along the current, enabling the algorithm to perform a broad search across the solution space while avoiding premature convergence. In the exploitation phase, jellyfish actively pursue food sources or interact within their swarm, which facilitates the refinement of solutions in promising regions. Through this balanced integration of exploration and exploitation phases, JSO effectively navigates complex optimization landscapes to identify optimal or near-optimal solutions. JSO has been successfully applied to optimal power flow problems, exhibiting superior performance relative to traditional swarm-intelligence optimizers such as particle swarm optimization, grey wolf optimizer, and whale optimization algorithm [20]. Moreover, JSO is recognized for its simplicity, efficiency, and versatility in addressing diverse optimization challenges. It demands relatively few parameters compared to other metaheuristics, thereby simplifying its implementation and tuning.

There are mainly five steps for JSO to search for the optimal solutions.

**(1)** Initialization. A population of $N$ jellyfish is initialized randomly within the search space:

$$X_i^0 = L_b + \text{rand}(0,1) \cdot (U_b - L_b), \quad i = 1, 2, \ldots, N \tag{1}$$

where $X_i^0$ is the initial solution for the $i$th jellyfish at 0th iteration, $L_b$ and $U_b$ are the lower and upper bounds of the search space, respectively.

**(2)** Ocean current movement (exploration). The motion of jellyfish following the ocean current is updated as:

$$X_i^{t+1} = X_i^t + c_0 \cdot \left(X_{\text{best}}^t - \beta \cdot \overline{X}^t\right) \tag{2}$$

where $X_{\text{best}}^t$ is the best solution at iteration $t$, $\overline{X}^t$ is the mean position of the population, $\beta$ is a distribution coefficient, and $c_0$ is a control parameter. This mechanism promotes global exploration, encouraging jellyfish to search widely while gradually converging toward promising regions.

**(3)** Active swimming (exploitation). In this phase, jellyfish either move toward the best solution or move randomly within the group. The motion is updated as:

$$X_i^{t+1} = X_i^t + \gamma \cdot (X_j^t - X_i^t) \tag{3}$$

where $X_j^t$ is a randomly selected solution from the population and $\gamma$ is a learning coefficient controlling the step size. This local search increases diversity while refining candidate solutions.

Alternatively, if the jellyfish moves toward the best solution, then it is updated as:

$$X_i^{t+1} = X_i^t + \eta \cdot (X_{\text{best}}^t - X_i^t) \tag{4}$$

where $\eta$ is a positive parameter that adjusts the attraction strength toward the best solution. This step reinforces exploitation, accelerating convergence around high-quality solutions.

**(4)** Switching mechanism. The decision to follow ocean currents (exploration) or perform active swimming (exploitation) is determined by a probability function:

$$P(t) = \frac{t}{T} \tag{5}$$

where $t$ is the current iteration and $T$ is the maximum number of iterations. If $\text{rand}(0, 1) > P(t)$, the algorithm favors exploration; otherwise, it favors exploitation.

**(5)** Updating and termination. At each iteration, all positions are updated according to the above process, and the best solution $X_{\text{best}}$ is recorded. The algorithm terminates after $T$ iterations or when a convergence condition is satisfied.

### 2.2 Prediction models

In this section, we introduce four machine learning models: SVM, RF, XGBoost, and CatBoost.

#### *2.2.1 SVM*

SVM is a widely adopted machine learning model. Its primary objective is to identify the optimal hyperplane that maximally separates data points of distinct classes within a high-dimensional feature space [21]. Support vectors, defined as the data points closest to this hyperplane, play a crucial role in defining the decision boundary. By maximizing the margin—the distance between the hyperplane and the nearest support vectors—SVM fosters strong generalization while minimizing the risk of overfitting. This margin-maximization principle renders SVM particularly effective for high-dimensional data or scenarios with limited training samples.

#### *2.2.2 RF*

RF is an ensemble learning method that integrates multiple decision trees to enhance predictive performance [22]. Unlike a single decision tree, which is prone to overfitting or sensitivity to minor data perturbations, RF constructs numerous decision trees and aggregates their predictions to produce the final outcome. Each tree is trained on a bootstrap sample from the dataset. At each split, only a random subset of features is considered. This randomness ensures diversity among the trees. For the final prediction, outputs from the decision trees are combined through majority voting for classification or averaging for regression. As a result, RF achieves superior generalization and reduced variance compared to individual trees.

#### *2.2.3 XGBoost*

XGBoost is a highly efficient and scalable implementation of gradient boosting, designed for supervised learning tasks like classification and regression [23]. Unlike RF, XGBoost builds trees sequentially, where each new tree corrects the errors of the previous ones by focusing on the most difficult samples to predict. It optimizes a regularized objective function that balances model fit and complexity. By doing so, it is able to reduce the risk of overfitting. Its use of second-order gradients (both first and second derivatives of the loss function) allows for more precise updates during training, which leads to faster convergence and higher predictive accuracy compared to traditional gradient boosting. The use of built-in regularization, parallel and distributed computing, handling of missing data, and support for custom loss functions make it a strong prediction model.

#### *2.2.4 CatBoost*

CatBoost is a gradient boosting algorithm that is particularly well-suited for handling categorical features directly, without requiring extensive pre-processing [24]. Like other boosting methods, CatBoost builds an ensemble of decision trees sequentially, with each new decision tree correcting the errors of the previous ones. What sets it apart is its use of ordered boosting and innovative techniques

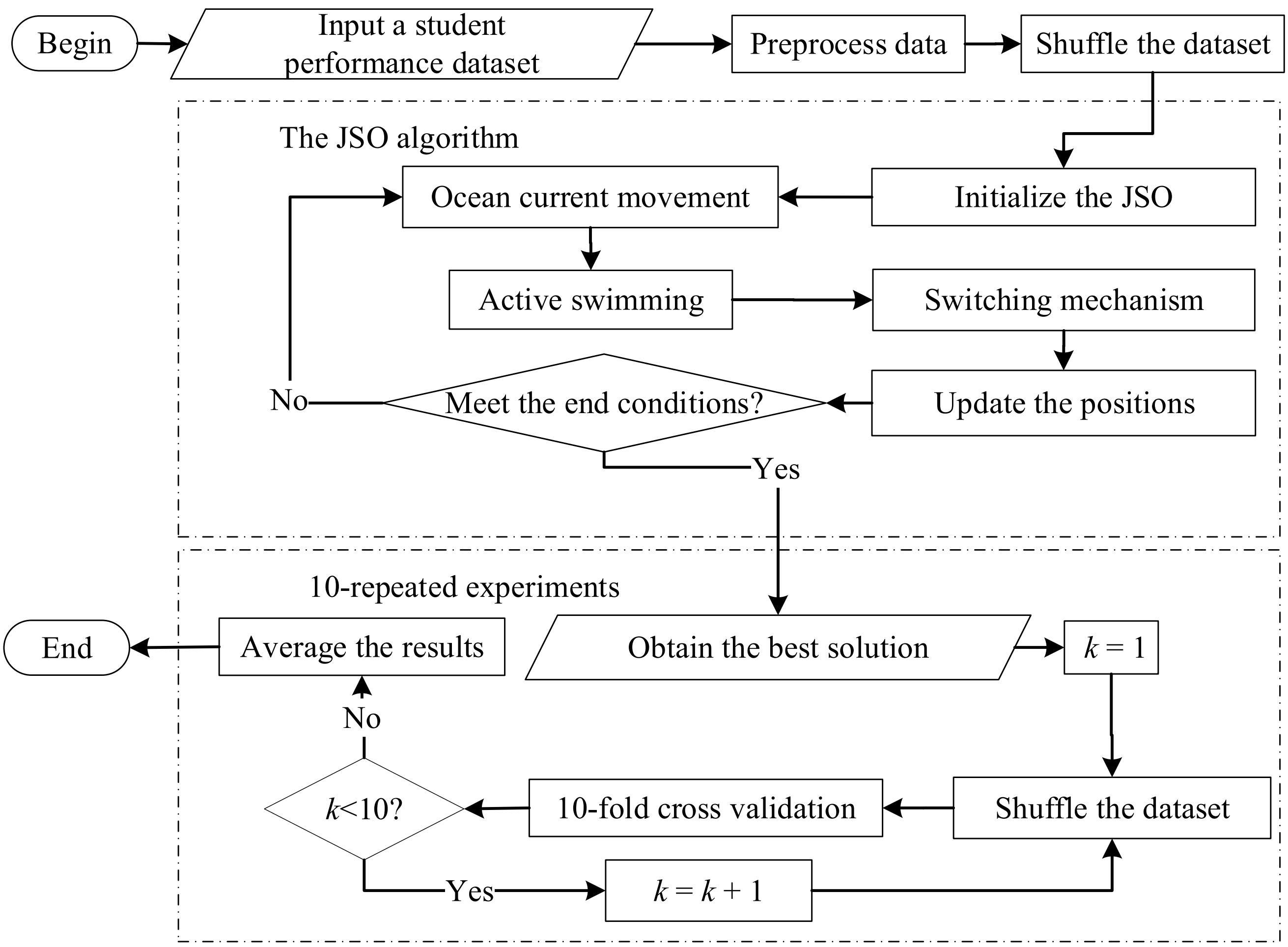


**Figure 1.** Flowchart of JSO-based framework for student performance prediction.

to reduce prediction shift, a problem that can arise when models use target statistics during training. This design makes CatBoost more stable and less prone to overfitting, while also simplifying the data preparation pipeline by handling categorical variables.

### 2.3 JSO-based framework for student performance prediction

Figure 1 shows the process of the proposed JSO-based framework for student performance prediction. The process begins with inputting a student performance dataset. As the educational datasets contain different types of values, data pre-processing is needed to prepare the data for analysis. After that, data shuffling is conducted to introduce randomness and eliminate potential ordering biases. Next, the JSO algorithm is performed with a sequence of operations. The ocean current movement is conducted to guide the search direction. After that, the active swimming behavior of jellyfish is used to explore the solution space. Next, the switching mechanism is applied to dynamically adjust between different search strategies. Finally, the positions of the jellyfish are updated to get closer to the optimal solution. These steps repeat until the end conditions are met, such as reaching a maximum number of iterations or achieving a satisfactory solution quality. Subsequently, 10-repeated experiments are conducted to alleviate the randomness caused by the data shuffling. Specifically, starting with $k = 1$, the dataset is shuffled each time, and 10-fold cross validation [25] is performed to thoroughly assess the model's performance across different data partitions. If $k < 10$, $k$ is incremented by 1, and the cycle of shuffling the dataset and conducting 10-fold cross validation repeats. Finally, after completing all 10 experiments, the results from each are averaged to yield a reliable and comprehensive outcome for student performance prediction.

## 3 Experimental results and analysis

In this section, the description of the datasets is first introduced. Next, the evaluation metrics are provided. Finally, the experimental results and discussions are

**Table 1.** Statistical properties of the student performance dataset.

| Feature | Symbol | Type | Values |
|---|---|---|---|
| School | School | Binary | Gabriel Pereira; Mousinho da Silveira |
| Sex | Sex | Binary | Female; Male |
| Age | Age | Numeric | 15–22 (in years) |
| Address | Address | Binary | Urban; Rural |
| Family Size | Famsize | Binary | $\leq 3$ (small); $> 3$ (large) |
| Parent Status | Pstatus | Binary | Living together; Living apart |
| Mother's Education | Medu | Numeric | 0 (none) – 4 (higher education) |
| Father's Education | Fedu | Numeric | 0 (none) – 4 (higher education) |
| Father's Job | Fjob | Nominal | Teacher; Health care; Civil services (e.g., administrative, police); At home; Other |
| Mother's Job | Mjob | Nominal | Teacher; Health care; Civil services (e.g., administrative, police); At home; Other |
| School Choice Reason | Reason | Nominal | Close to home; School reputation; Course preference; Other |
| Guardian | Guardian | Nominal | Mother; Father; Other |
| Travel Time | Traveltime | Numeric | 1 ($<$ 15 min); 2 (15–30 min); 3 (30 min–1 hour); 4 ($>$ 1 hour) |
| Study Time | Studytime | Numeric | 1 ($<$ 2 hours); 2 (2–5 hours); 3 (5–10 hours); 4 ($>$ 10 hours) |
| Past Failures | Failures | Numeric | 1 (1 failure); 2 (2 failures); 4 ($\geq$ 3 failures) |
| School Support | Schoolsup | Binary | Yes; No |
| Family Support | Famsup | Binary | Yes; No |
| Paid Extra Classes | Paid | Binary | Yes; No |
| Extracurricular Activities | Activities | Binary | Yes; No |
| Nursery Attendance | Nursery | Binary | Yes; No |
| Pursue Higher Education | Higher | Binary | Yes; No |
| Home Internet Access | Internet | Binary | Yes; No |
| Romantic Relationship | Romantic | Binary | Yes; No |
| Family Relationship Quality | Famrel | Numeric | 1 (very bad) – 5 (excellent) |
| Post-School Free Time | Freetime | Numeric | 1 (very low) – 5 (very high) |
| Socializing with Friends | Goout | Numeric | 1 (very low) – 5 (very high) |
| Weekday Alcohol Consumption | Dalc | Numeric | 1 (very low) – 5 (very high) |
| Weekend Alcohol Consumption | Walc | Numeric | 1 (very low) – 5 (very high) |
| Current Health Status | Health | Numeric | 1 (very bad) – 5 (very good) |
| School Absences | Absences | Numeric | 0–93 (number of days) |
| First-Period Grade (Target) | G1 | Numeric | 0–20 (grading scale: 0 = lowest, 20 = highest) |

*Note:* Numeric features with codes (e.g., Medu, Failures) are defined to clarify value interpretation. Binary and nominal features use semicolons to separate discrete categories.

presented.

### 3.1 Data description

To assess the efficacy of our proposed student performance prediction model, we obtained the educational dataset from the UCI Repository [26]. This dataset encompasses the subjects of Mathematics and Portuguese Language. Each subset incorporates student grades alongside demographic, social, and school-related features. Specifically, the Mathematics dataset comprises 395 samples, while the Portuguese Language dataset contains 649 samples. Both datasets feature 30 input attributes and one output attribute representing final exam grades. Detailed statistical descriptions of the datasets are provided in Table 1.

### 3.2 Evaluation metrics

To evaluate the effectiveness of our proposed framework, we adopted four key metrics: Mean Absolute Error ($MAE$), Standard Deviation ($SD$) of prediction errors, Root Mean Squared Error ($RMSE$), and robustness ($MAC$) [27]. Our objective is to minimize $MAE$, $SD$, and $RMSE$ while maximizing $MAC$. Their equations are as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i| \, , \tag{6}$$

**Table 2.** Hyper-parameter settings for the MLP, SVM, RF, XGBoost, and CatBoost.

| Model | **Hyper-parameter** | **Candidate values** | **Best values for (dataset1 and dataset2)** |
|---|---|---|---|
| SVM | Penalty parameter $C$<br>Gaussian kernel bandwidth $\gamma$ | [10, 100, 1000]<br>[0.001, 0.01, 0.1] | [100, 10]<br>[0.001, 0.001] |
| RF | Number of estimators<br>Maximum depth of trees | [100, 150, 200]<br>[3, 4, 5] | [200, 150]<br>[5, 5] |
| XGBoost | Number of estimators<br>Maximum depth of trees | [100, 150, 200]<br>[3, 4, 5] | [100, 100]<br>[3, 4] |
| CatBoost | Number of estimators<br>Maximum depth of trees | [100, 150, 200]<br>[3, 4, 5] | [200, 100]<br>[3, 3] |

Note: Each pair of best values corresponds to the first and second datasets, respectively. For example, SVM achieves its best performance with $(C, \gamma) = (100, 0.001)$ for the first dataset and $(10, 0.001)$ for the second dataset. The ensemble models (RF, XGBoost, CatBoost) follow the same order.

$$SD = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(e_i - \bar{e})^2}, \tag{7}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}, \tag{8}$$

$$MAC = \frac{(\boldsymbol{y}^T\hat{\boldsymbol{y}})^2}{(\boldsymbol{y}^T\boldsymbol{y})(\hat{\boldsymbol{y}}^T\hat{\boldsymbol{y}})}, \tag{9}$$

where $n$ is the number of testing samples, $y_i$ is the true target value of the $i$-th sample, $\hat{y}_i$ is the predicted value of the $i$-th sample, $e_i = \hat{y}_i - y_i$ is the prediction error for the $i$-th sample, and $\bar{e} = \frac{1}{n}\sum_{i=1}^{n} e_i$ is the average prediction error.

### 3.3 Parameter settings

The hyper-parameters used in the grid search can be found in Table 2. Specifically, for SVM, candidate values of [10, 100, 1000] were considered for the penalty parameter and values of [0.001, 0.01, 0.1] were considered for the Gaussian kernel's bandwidth. For the ensemble models (RF, XGBoost, CatBoost), candidate values of [100, 150, 200] were considered for the number of base estimators and values of [3, 4, 5] were considered for a tree's maximum depth. The best hyper-parameters are [[100 0.001], [200 5], [100 3], [200 3]] for the first dataset, while [10 0.001], [150 5], [100 4], [100 3]] for the second dataset. While for the JSO, the lower bound and upper bound are used for each model, the number of jellyfish is set to 10, and the number of iterations is set to 50.

### 3.4 Experimental results

Table 3 shows the experimental results for predicting student performance in the final Mathematics exam using SVM, RF, XGBoost, and CatBoost, along with their improved versions optimized through the proposed JSO-based framework. The best results are highlighted in bold. The optimized versions of all models achieve performance gains over their original counterparts. The improvements are most pronounced in CatBoost, where JSO-CatBoost reduces $MAE$ from 2.3962 to 2.3262 and $RMSE$ from 2.9610 to 2.8422, while increasing MAC from 0.9327 to 0.9388. These results highlight that the optimization process not only enhances predictive accuracy but also improves model stability, as reflected in lower SD values. Although the gains for SVM, RF, and XGBoost are relatively smaller, they still demonstrate consistent improvements across most metrics. Overall, JSO-CatBoost emerges as the most effective model for early prediction of Mathematics performance, demonstrating the capability of the proposed framework to fine-tune gradient boosting algorithms more effectively than other model families.

Figure 2 visualizes the experimental results reported in Table 3, comparing the performance of four baseline models (SVM, RF, XGBoost, and CatBoost) and their JSO-optimized counterparts across four evaluation metrics. For all models, the JSO-optimized versions consistently reduce $MAE$, $SD$, and $RMSE$ compared to their baselines, indicating lower prediction errors and improved stability.

Table 4 presents the experimental outcomes for early prediction of the Portuguese Language final exam using the same set of baseline models and

**Table 3.** Experimental results for early prediction of Mathematics course in the final exam using the SVM, RF, XGBoost, CatBoost, and their improved version based on the proposed framework.

| Model | $MAE$ | $SD$ | $RMSE$ | $MAC$ |
|---|---|---|---|---|
| SVM | 2.4720 | 3.0274 | 3.0785 | 0.9284 |
| JSO-SVM | 2.4111 | 2.9317 | 2.9691 | 0.9330 |
| RF | 2.4876 | 2.9888 | 3.0102 | 0.9304 |
| JSO-RF | 2.4799 | 2.9321 | 2.9852 | 0.9329 |
| XGBoost | 2.5141 | 3.0840 | 3.1217 | 0.9265 |
| JSO-XGBoost | 2.4822 | 2.9999 | 3.0528 | 0.9299 |
| CatBoost | 2.3962 | 2.9470 | 2.9610 | 0.9327 |
| JSO-CatBoost | 2.3262 | 2.7979 | 2.8422 | 0.9388 |

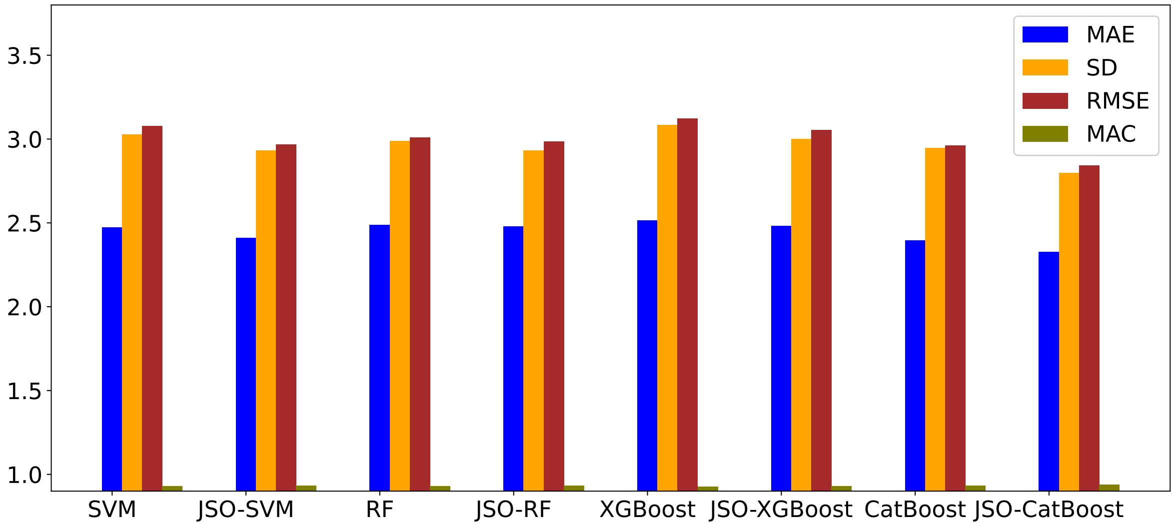


**Figure 2.** Experimental results based on the results in Table 3.

their JSO-optimized variants. The results from the Portuguese Language course again confirm the advantage of the proposed framework. Similar to the Mathematics course, JSO-based optimization consistently enhances all baseline models, with JSO-CatBoost delivering the strongest improvements. It achieves the lowest error rates ($MAE = 1.7333$, $RMSE = 2.2124$) and the highest MAC (0.9646), surpassing all other models. Notably, the optimized models also show reduced variability in predictions, evidenced by decreased SD values. This suggests that the proposed framework not only increases accuracy but also contributes to more stable and reliable performance across folds. The fact that similar trends are observed in both Mathematics and Portuguese courses demonstrates the generalizability of the method across different subjects, reinforcing its potential for practical deployment in early student performance prediction.

Figure 3 illustrates the experimental results reported in Table 4, comparing the performance of the baseline models (SVM, RF, XGBoost, and CatBoost) and their JSO-optimized counterparts across four metrics. Similar to the previous results, the JSO-enhanced models consistently outperform their baseline versions across all metrics.

The superior performance of the JSO-CatBoost model can be attributed to three main factors. First, the JSO dynamically balances exploration and exploitation, enabling more precise parameter tuning and avoiding local optima during optimization. Second, the CatBoost inherently handles categorical and numerical features efficiently, and the JSO-based search enhances this ability by aligning hyperparameters with data distribution characteristics. Third, the JSO framework adaptively updates model parameters through oscillatory motion and active/passive movement strategies, ensuring smoother convergence and reducing overfitting.

## 4 Conclusion

In this work, we proposed an automatic parameter optimization framework for early student performance prediction, leveraging the JSO to enhance the

**Table 4.** Experimental results for early prediction of Portuguese Language course in the final exam using the SVM, RF, XGBoost, CatBoost, and their improved version based on the proposed framework.

| Model | $MAE$ | $SD$ | $RMSE$ | $MAC$ |
|---|---|---|---|---|
| SVM | 1.8305 | 2.3126 | 2.3374 | 0.9607 |
| JSO-SVM | 1.7906 | 2.2618 | 2.2858 | 0.9623 |
| RF | 1.8168 | 2.2925 | 2.3031 | 0.9612 |
| JSO-RF | 1.8127 | 2.2805 | 2.3003 | 0.9617 |
| XGBoost | 1.9618 | 2.4132 | 2.4420 | 0.9573 |
| JSO-XGBoost | 1.8917 | 2.3717 | 2.3901 | 0.9586 |
| CatBoost | 1.8100 | 2.2670 | 2.2977 | 0.9620 |
| JSO-CatBoost | 1.7333 | 2.1958 | 2.2124 | 0.9646 |

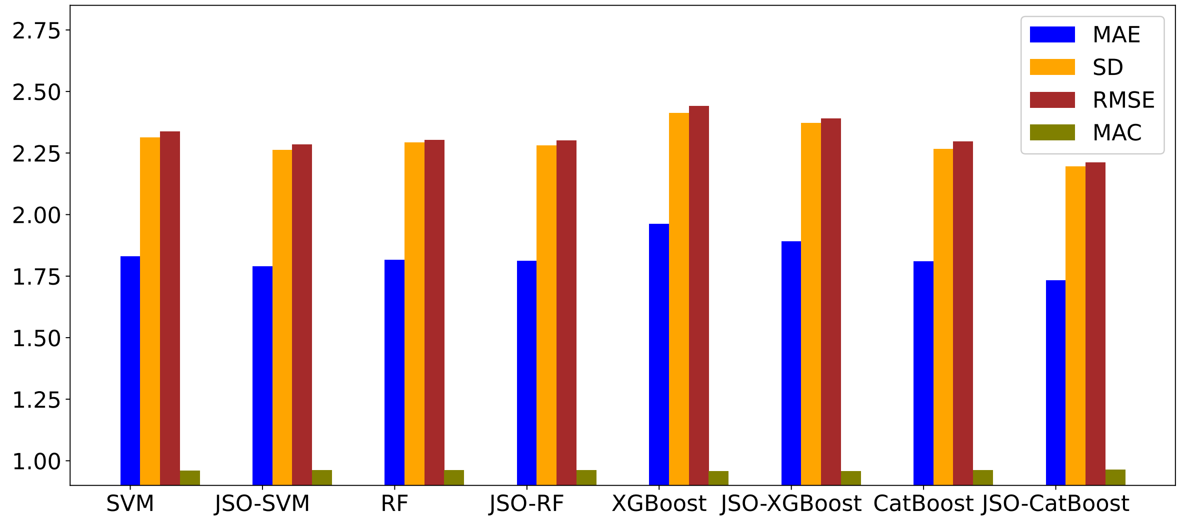


**Figure 3.** Experimental results based on the results in Table 4.

effectiveness of baseline machine learning models. The experimental results on both the Mathematics and Portuguese Language courses demonstrated the proposed framework consistently improved the predictive performance of SVM, RF, XGBoost, and CatBoost models across all evaluation metrics. Among these, CatBoost benefited the most from optimization, with JSO-CatBoost achieving the lowest prediction errors and the highest accuracy in both datasets. The findings highlight the potential of the proposed framework as a reliable tool for early student performance prediction. Beyond student performance prediction, the framework can be extended to broader educational data mining tasks (e.g., early warning systems for at-risk students, personalized curriculum and learning path recommendations) and other application domains where accurate and stable predictive modeling is essential.

While the proposed framework demonstrates strong performance, several directions merit further exploration. First, integrating deep learning models into the optimization pipeline could enhance the ability to capture complex learning patterns. Second, incorporating multi-source heterogeneous educational data would allow for richer and more personalized predictions. Third, extending the framework to continual learning settings may improve adaptability to evolving student behaviors over time. Finally, investigating the interpretability of optimized models could provide more actionable insights for educators and policymakers.

## Data Availability Statement

Data will be made available on request.

## Funding

This work was supported by the Fujian Provincial Young and Middle-aged Teachers' Educational Research Project (Science and Technology Category), China under Grant JAT241390.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Ethical Approval and Consent to Participate

Not applicable.

## References


[1] Bai, X., Zhang, F., Li, J., Guo, T., Aziz, A., Jin, A., & Xia, F. (2021). Educational big data: Predictions, applications and challenges. *Big Data Research*, *26*, 100270. [CrossRef]

[2] Rabelo, A., Rodrigues, M. W., Nobre, C., Isotani, S., & Zárate, L. (2024). Educational data mining and learning analytics: A review of educational management in e-learning. *Information Discovery and Delivery*, *52*(2), 149-163. [CrossRef]

[3] Kalita, E., Oyelere, S. S., Gaftandzhieva, S., Rajesh, K. N., Jagatheesaperumal, S. K., Mohamed, A., ... & Ali, T. (2025). Educational data mining: a 10-year review. *Discover Computing*, *28*(1), 81. [CrossRef]

[4] Hemdanou, A. L., Sefian, M. L., Achtoun, Y., & Tahiri, I. (2024). Comparative analysis of feature selection and extraction methods for student performance prediction across different machine learning models. *Computers and Education: Artificial Intelligence*, *7*, 100301. [CrossRef]

[5] Öz, E., Bulut, O., Cellat, Z. F., & Yürekli, H. (2025). Stacking: An ensemble learning approach to predict student performance in PISA 2022. *Education and Information Technologies*, *30*(6), 7753-7779. [CrossRef]

[6] Cao, W., & Mai, N. (2025). Predictive Analytics for Student Success: AI-Driven Early Warning Systems and Intervention Strategies for Educational Risk Management. *Educational Research and Human Development*, *2*(2), 36-48. [CrossRef]

[7] Bañeres, D., Rodríguez-González, M. E., Guerrero-Roldán, A. E., & Cortadas, P. (2023). An early warning system to identify and intervene online dropout learners. *International Journal of Educational Technology in Higher Education*, *20*(1), 3. [CrossRef]

[8] Maiya, A. K., & Aithal, P. S., (2023). A Review based Research Topic Identification on How to Improve the Quality Services of Higher Education Institutions in Academic, Administrative, and Research Areas. *International Journal of Management, Technology, and Social Sciences (IJMTS)*, *8*(3), 103-153. [CrossRef]

[9] Fan, Z., Gou, J., & Wang, C. (2025). An error complementarity-based iterative learning approach via categorical boosting for student performance prediction. *Engineering Applications of Artificial Intelligence*, *161*, 112192. [CrossRef]

[10] Ahmed, E. (2024). Student performance prediction using machine learning algorithms. *Applied computational intelligence and soft computing*, *2024*(1), 4067721. [CrossRef]

[11] Hong, Y. Z., Rani, M. N. A., Radzuan, N. F. M., Yen, L. H., & Nagalingam, S. A. R. A. S. V. A. T. H. I. (2024). An early warning system for students at risk using supervised machine learning. *Journal of Engineering Science and Technology*, *19*(1), 131-139.

[12] Cheng, B., Liu, Y., & Jia, Y. (2024). Evaluation of students' performance during the academic period using the XG-Boost Classifier-Enhanced AEO hybrid model. *Expert Systems with Applications*, *238*, 122136. [CrossRef]

[13] Fan, Z., Gou, J., & Weng, S. (2025). Complementary CatBoost based on residual error for student performance prediction. *Pattern Recognition*, *161*, 111265. [CrossRef]

[14] Fan, Z., Gou, J., & Wang, C. (2023). Predicting secondary school student performance using a double particle swarm optimization-based categorical boosting model. *Engineering Applications of Artificial Intelligence*, *124*, 106649. [CrossRef]

[15] Mahawar, K., & Rattan, P. (2025). Empowering education: Harnessing ensemble machine learning approach and ACO-DT classifier for early student academic performance prediction. *Education and Information Technologies*, *30*(4), 4639-4667. [CrossRef]

[16] Roy, P., & Chowdhury, C. (2024). A region-wise indoor localization system based on unsupervised learning and ant colony optimization technique. *Applied Soft Computing*, *157*, 111509. [CrossRef]

[17] Mohamedshareef, R., Abapour, M., Hosseini, S. H., & Seyydi, H. (2024). Optimizing directional overcurrent relay coordination with jellyfish search algorithm. *Electrical Engineering*, *106*(6), 7103-7115. [CrossRef]

[18] Alwakeel, A. S., Ismail, M. I., Fouda, M. M., Shaheen, A. M., & Khaled, A. (2024). Improved artificial rabbits algorithm for positioning optimization and energy control in RIS multiuser wireless communication systems. *IEEE Internet of Things Journal*, *11*(11), 20605-20618. [CrossRef]

[19] Ahmed, D., Ebeed, M., Kamel, S., Nasrat, L., Ali, A., Shaaban, M. F., & Hussien, A. G. (2024). An enhanced jellyfish search optimizer for stochastic energy management of multi-microgrids with wind turbines, biomass and PV generation systems considering uncertainty. *Scientific Reports*, *14*(1), 15558. [CrossRef]

[20] Nadimi-Shahraki, M. H., Banaie-Dezfouli, M., & Zamani, H. (2025). Conscious Neighborhood-Based Jellyfish Search Optimizer for Solving Optimal Power Flow Problems. *Mathematics*, *13*(19), 3068. [CrossRef]

[21] Tanveer, M., Tiwari, A., Akhtar, M., & Lin, C. T. (2025). Enhancing imbalance learning: A novel slack-factor fuzzy SVM approach. *IEEE Transactions on Emerging Topics in Computational Intelligence*. [CrossRef]

[22] Speiser, J. L., Miller, M. E., Tooze, J., & Ip, E. (2019). A comparison of random forest variable selection methods for classification prediction modeling. *Expert systems with applications*, *134*, 93-101. [CrossRef]

[23] Zhang, P., Jia, Y., & Shang, Y. (2022). Research and application of XGBoost in imbalanced data.

*International Journal of Distributed Sensor Networks*, *18*(6), 15501329221106935. [CrossRef]

[24] Fan, Z., Gou, J., & Weng, S. (2024). A feature importance-based multi-layer catboost for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering*, *36*(11), 5495-5507. [CrossRef]

[25] Fan, Z., Wu, F., & Tang, Y. (2023). A hierarchy-based machine learning model for happiness prediction. *Applied Intelligence*, *53*(6), 7108-7117. [CrossRef]

[26] Cortez, P. (2008). Student Performance [Dataset]. UCI Machine Learning Repository. [CrossRef]

[27] Fan, Z., Chiong, R., Hu, Z., & Lin, Y. (2020). A fuzzy weighted relative error support vector machine for reverse prediction of concrete components. *Computers & Structures*, *230*, 106171. [CrossRef]

[28] Zheng, X., & Li, C. (2024). Predicting students' academic performance through machine learning classifiers: a study employing the Naive Bayes classifier (NBC). *International Journal of Advanced Computer Science and Applications*, *15*(1). [CrossRef]

**Shaoyuan Weng** received the master's degree from the College of Computer Science and Technology, Huaqiao University, Xiamen, China, in 2021. She is currently a lecturer with the Xiamen Institute of Software Technology. Her main research interests include in the areas of analysis of data mining and collaborative task. (Email: wengshaoyuany@163.com)

**Zimeng Liu** is currently the Director of the Teaching and Research Section for Big Data Technology and Cloud Computing Technology Application at the Xiamen Institute of Software Technology. He is also a Senior Engineer (Information System Project Manager), Senior Technician, and Dual-Qualified Teacher. (Email: liuzimenglzm@163.com)

**Yuanyuan Zheng** received the master's degree from the College of Computer Science and Technology, Huaqiao University, Xiamen, China, in 2021. She is currently a lecturer with the Xiamen Institute of Software Technology. Her main research interests include in the areas of analysis of data mining and collaborative task. (Email: zhengyuanyuan1025@163.com)

**Chao Zhang** received his M.S. degree from the School of Computer Science and Technology, Harbin University of Science and Technology, in 2018. He is currently a lecturer at Xiamen Institute of Software Technology. His research interests include data analysis and data mining (Email: zhangchao636@163.com)