Research on Predicting Flotation Concentrate Grade Based on Improved COA Optimized ELM Neural Network

Jia Ren, Yong Zhang*

***Abstract*—The concentrate grade is a crucial economic indicator for evaluating flotation production. Establishing an effective prediction model is of great significance for improving resource recovery and reducing production costs. Given the complex nonlinear relationships and high-dimensional data in flotation prediction, which are difficult to deduce using simple mathematical formulas, this study proposes an Extreme Learning Machine (ELM) model optimized by an improved Crayfish Optimization Algorithm (COA) for predicting concentrate grade. First, to enhance the optimization performance of the COA, an opposition-based learning strategy and an optimal cave strategy are introduced. Additionally, a Lévy flight strategy with short-and long-distance random jumps is incorporated to expand the search space and balance solution distribution, allowing the improved COA to escape local optima. Second, the improved COA algorithm is compared with four other algorithms on the CEC2022 benchmark functions, demonstrating its superior performance in various aspects. Finally, the improved COA is used to optimize the parameters of the ELM, forming the RCOA-ELM model for flotation concentrate grade prediction. Compared with other ELM optimization models, the proposed model achieves a mean absolute error (MAE) of 0.0774, a root mean square error (RMSE) of 0.1045, and a coefficient of determination (R^2) of 0.8529, all of which outperform competing models. The results indicate that the proposed model effectively improves the prediction accuracy of concentrate grade and accurately reflects its variation trends, meeting practical industrial requirements.**

***Index Terms*—Crayfish Optimization Algorithm, Extreme Learning Machine, Concentrate Grade, Flotation Prediction.**

I. Introduction

FLOTATION technology plays a crucial role in the mineral processing industry, and the concentrate grade directly affects product quality, making it a key economic indicator for evaluating flotation production. A low concentrate grade leads to a decline in product quality, while an excessively high concentrate grade increases the tailings grade, reducing metal recovery rates and causing metal losses, which negatively impact economic benefits. Therefore, maintaining the concentrate grade within a reasonable range is of great significance for improving resource recovery and reducing production costs.

The flotation production process is characterized by multiple couplings and multiple stages, making it difficult to predict the flotation concentrate grade using simple mathematical formulas due to its complex nonlinear relationships and high-dimensional data. With the rapid development of soft sensing technology, neural network-based prediction methods have emerged and been increasingly applied in industrial production, gradually becoming an effective approach for optimizing the flotation process. To establish a flotation concentrate grade prediction model, some researchers have utilized intelligent algorithm-optimized neural network soft sensing models to measure the flotation concentrate grade, providing a novel method. Al-Thyabat et al. [1] used a multilayer feedforward neural network to study the effects of feed particle size, reagent dosage, and impeller speed on concentrate grade and flotation recovery. Hoseinian et al. [2] employed a hybrid neural network combining Genetic Algorithm-based Neural Networks (GANN) and Multiple Linear Regression (MLR) to achieve a more accurate prediction of ion removal efficiency in the ion flotation process. Rachel Cook et al. [3] integrated a random forest model with the Firefly Algorithm to develop a hybrid machine learning (ML) model for predicting the flotation efficiency of galena and chalcopyrite. Their results demonstrated that the hybrid ML model outperformed standalone ML models and exhibited superior predictive capability.

The Crayfish Optimization Algorithm (COA), proposed by Jia et al. [4], is a novel swarm intelligence optimization algorithm inspired by the foraging, shelter-seeking, and competitive behaviors of crayfish. Based on these behaviors, the COA was developed to simulate how crayfish search for food, find shelters, and compete for survival. While COA exhibits strong global search capabilities, is easy to implement, and depends on the initial population distribution, it suffers from slow local convergence and unstable search efficiency. These limitations become particularly evident when dealing with high-dimensional complex problems, where COA is prone to falling into local optima, leading to suboptimal optimization results. To address these issues, researchers have proposed various improvements to COA. Shikoun N H et al. [5] introduced a new binary Crayfish Optimization Algorithm (BinCOA) by integrating refraction-based opposition learning and crossover strategies for feature selection. Jia H et al. [6] enhanced COA by incorporating an environmental update mechanism based on water quality factors and a ghost-adversarial learning strategy, proposing the Modified Crayfish Optimization Algorithm (MCOA) to solve multiple engineering application problems. Chaib L et al. [7] developed a novel method combining COA with fractional-order chaotic maps (FC-maps) for photovoltaic model parameter estimation.

Manuscript received April 1, 2025; revised June 5, 2025.

Jia Ren is a graduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051, China. (e-mail: 18341275676@163.com).

Yong Zhang is a Professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051, China. (*Corresponding author, phone: 86-0412-889699; e-mail: zy9091@163.com).

The Extreme Learning Machine (ELM) algorithm, proposed by Huang et al. [8], is a highly efficient learning algorithm that does not require adjustments to the input weights and hidden biases during execution. Instead, it only requires setting the number of hidden layer nodes to obtain the optimal solution. In recent years, ELM has been widely used for classification and regression tasks due to its advantages of short training time, high execution efficiency, and strong generalization ability [9]. Li et al. [10, 11] accurately predicted the transient stability of power systems using an improved ELM model. However, the initialization of input weights and hidden biases in the ELM network is random, which affects prediction stability. To address this issue, Miao et al. [12] proposed an improved ELM model optimized by the Whale Optimization Algorithm (WOA) for gas outburst risk level prediction, effectively enhancing ELM's predictive accuracy. Jia Z et al. [13, 14] applied the Grey Wolf Optimizer (GWO) to optimize the input weights and hidden layer thresholds of ELM, developing a GWO-ELM model for predicting rock blasting fragmentation.

Based on the above background, this study proposes an improved Crayfish Optimization Algorithm (RCOA) to optimize the Extreme Learning Machine (ELM) for flotation concentrate grade prediction. First, to enhance the effectiveness of COA, several improvements are introduced. The opposition-based learning strategy is employed to replace random initialization, ensuring population diversity [15, 16]. A nonlinear convergence factor is designed to better balance the exploration and exploitation capabilities of the algorithm. The optimal cave strategy is adopted to perturb and update the best position, improving the quality of individual position selection and enhancing global search ability. Finally, the Lévy flight strategy is incorporated to help the improved COA escape local optima, achieving higher solution accuracy and better problem-solving capability. Next, comparative experiments are conducted using three classical algorithms and the original COA algorithm on the CEC2022 benchmark functions. The results demonstrate the effectiveness of the proposed improvements to the COA algorithm. Finally, the RCOA algorithm is applied to optimize the ELM, establishing the RCOA-ELM prediction model. A comparative analysis is conducted by optimizing ELM using four different algorithms. The experimental results indicate that the RCOA-ELM model exhibits superior predictive performance.

II. CRAYFISH OPTIMIZATION ALGORITHM

Intelligent optimization algorithms are proposed based on principles such as simulating natural biological evolution and swarm intelligence. Inspired by the social behaviors of crayfish in foraging, summer shelter-seeking, and competition, Jia et al. [4] introduced the Crayfish Optimization Algorithm (COA) in 2023. The algorithm operates in three stages: shelter-seeking, competitive cave selection, and foraging. Foraging and competitive behaviors represent the development phase of COA, while shelter-seeking behavior is the exploration phase. The change in environmental temperature directly affects crayfish behavior, causing them to enter different stages. When the temperature exceeds 30°C, crayfish enter the shelter-seeking phase, where they prefer cooler environments. In this phase, they may encounter empty caves and enter them directly or may need to compete to obtain a cave. In suitable temperatures (ranging from 15°C to 30°C), crayfish enter the foraging phase, and their food intake is adjusted based on the temperature. The optimal foraging temperature is 25°C, and the crayfish's intake follows an approximately normal distribution.

$$temp = rand \times 15 + 20 \tag{1}$$

$$p = C_1 \times \left(\frac{1}{\sqrt{2 \times \pi} \times \sigma} \times \exp\left(-\frac{(temp - \mu)^2}{2\sigma^2} \right) \right) \tag{2}$$

Here, rand is a random number between 0 and 1, μ represents the optimal foraging temperature for crayfish, which is 25°C, C_1 is a constant with a value of 0.2, and σ is set to 3. By adjusting C_1 and σ, the food intake of crayfish under different temperatures can be controlled.

A. *Shelter-seeking phase*

When the temperature of the environment where the crayfish are located exceeds 30°C, they will seek caves and enter the shelter-seeking phase. The COA algorithm first randomly generates position information in the search space and calculates the fitness of each individual. Assuming the individual's optimal position is X_G and the current population's optimal position is X_L, the position information of the cave X_{cave} is defined as follows.

$$X_{\text{cave}} = \frac{X_G + X_L}{2} \tag{3}$$

The event of crayfish competing for and occupying caves is treated as a probabilistic event in COA. When rand<0.5, it indicates that no other crayfish are competing for the cave, and the crayfish will directly enter the cave to seek shelter.

$$X_{i,j}^{t+1} = X_{i,j}^{t} + C_2 \times rand \times \left(X_{\text{cave}} - X_{i,j}^{t} \right) \tag{4}$$

$$C_2 = 2 - \left(\frac{t}{T} \right) \tag{5}$$

In the equation, t represents the current iteration, and T represents the maximum number of iterations. C_2 is a control parameter that linearly decreases from 2 to 0, balancing exploration and exploitation. During the shelter-seeking phase, the goal of the crayfish is to move closer to the cave, bringing the individual closer to the optimal solution, which enhances the exploration capability of COA, thereby speeding up the convergence of the algorithm.

B. *Competition phase*

When the temperature of the environment where the crayfish are located exceeds 30°C and rand $\geq$ 0.5, it indicates that the crayfish will compete for the cave.

$$X_{i,j}^{t+1} = X_{i,j}^{t} - X_{z,j}^{t} + X_{\text{cave}} \tag{6}$$

$$Z = \text{round}\big(rand \times (N-1) \big) + 1 \tag{7}$$

In the equation, Z represents a randomly selected crayfish individual, and N denotes the total population of crayfish. During the competition phase, crayfish compete with each other, and crayfish $X_{i,j}$ update their position based on a randomly selected individual $X_{z,j}$. This expands the search range of COA and enhances the algorithm's global optimization capability.

C. Foraging phase

When the temperature is less than or equal to 30°C, reaching a suitable temperature for crayfish feeding, the crayfish will move toward the food location to forage. The food position is defined as X_f. After finding the food, the crayfish will determine the food quantity Q based on the following equation.

$$X_{\mathrm{f}} = X_G \tag{8}$$

$$Q = C_3 \times rand \times \left(\frac{fitness_i}{fitness_{\mathrm{f}}}\right) \tag{9}$$

C_3 represents the food factor, which is a constant value set to 3. $fitness_i$ denotes the fitness value of the ith crayfish, while $fitness_f$ represents the fitness value of the food position. During feeding, the crayfish will decide whether to tear the food based on its size. When $Q > \frac{C_3+1}{2}$, it indicates that the food is too large, and the crayfish will use its first claw to tear the food.

$$X_{\mathrm{f}} = \exp\left(-\frac{1}{Q}\right) \times X_{\mathrm{f}} \tag{10}$$

When the food is torn into smaller pieces, the second and third claws will alternately grasp and consume the food. A combination of sine and cosine functions is used to simulate this alternating process.

$$\begin{aligned} X_{i,j}^{t+1} =& X_{i,j}^{t} + X_{\mathrm{f}} \times Q \times \Big(\cos(2\pi \times rand) \\ & - \sin(2\pi \times rand)\Big) \end{aligned} \tag{11}$$

When the food quantity $Q \leq \frac{C_3+1}{2}$, the crayfish will move toward the food position and consume it directly. The equation is as follows:

$$X_{i,j}^{t+1} = \left(X_{i,j}^{t} - X_{\mathrm{f}}\right) \times Q + Q \times rand \times X_{i,j}^{t} \tag{12}$$

III. Improved COA Algorithm

The classical Crayfish Optimization Algorithm (COA) already possesses good global search capability and is easy to implement. However, it suffers from a slow local convergence rate and unstable search efficiency, making it prone to getting trapped in local optima. To address these issues, this paper introduces improvements focusing on the initial population distribution, nonlinear convergence factors, and search strategies of the COA. The specific improvements and the enhanced algorithm's workflow are as follows.

A. Opposition-Based Learning for Population Initialization

Population initialization plays a crucial role in metaheuristic algorithms. In the population initialization phase of the Crayfish Optimization Algorithm (COA), individuals are randomly generated within a predefined search space. However, due to the randomness of position updates, this may lead to an uneven population distribution, reducing the effectiveness of global search. Therefore, this paper adopts an opposition-based learning strategy for population initialization. By generating a mirror individual for each original individual with respect to the center of the search space, the search range is expanded, helping to prevent the algorithm from getting trapped in local optima. The corresponding equation is as follows:

$$X_{\mathrm{ol}} = lb + ub - X \tag{13}$$

Where: X_{ol} is the opposition-based mirror individual, X is an individual from the original population, and lb and ub represent the lower and upper bounds of the search space, respectively.

B. Nonlinear Convergence Factor

In COA, the control parameter C_2 affects the algorithm's performance. If C_2 is greater than 2 or less than 1, it will quickly cause the algorithm to diverge. In the COA algorithm, C_2 decreases linearly during the iteration process, which results in slow convergence and a tendency to fall into local optima, making it difficult to balance the algorithm's global search and local exploitation abilities. To improve the algorithm's solving capability and address the issue of slow convergence in COA, a nonlinear C_2 is designed to better balance exploration and exploitation in the algorithm. The equation is as follows:

$$C_2 = \frac{(t-T)^2}{T^2} + 1 \tag{14}$$

The curve of C_2 variation with iterations before and after the improvement is shown in Figure 1. Compared to the traditional linear decay method, this nonlinear decay factor is smoother, decreasing rapidly in the early stages and slowing down in the later stages. It better balances the global exploration and local exploitation abilities, effectively avoiding premature convergence and improving both the global search capability and convergence accuracy of the algorithm.

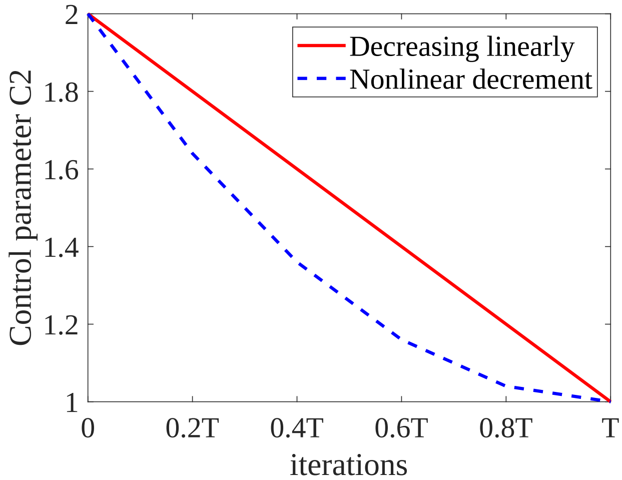

Fig. 1: Comparison of control parameter C_2

C. Optimal Cave Strategy

In the original Crayfish Optimization Algorithm (COA), the shading and competition stages tend to focus on searching near the caves. As the number of iterations increases, the positions of crayfish individuals gradually converge, potentially leading to local optima and limiting the effective exploration of the solution space. Inspired by the hunting phase of the

Grey Wolf Optimizer (GWO) [17] and incorporating diverse individual movement patterns, the optimal cave strategy is introduced. This strategy selects the best-performing individuals and guides others toward these high-quality solutions to improve the overall quality of the population. The optimal cave is defined as follows:

$$X_{\text{cave\_best}} = \{X_L, X_G, X_2, X_3, \overline{X}, X_0\} \tag{15}$$

$$\overline{X} = \frac{X_L + X_2 + X_3}{3} \tag{16}$$

$$X_0 = \frac{\sum X_i}{N} \tag{17}$$

Here, X_G represents the historically best position, which is the optimal solution obtained after multiple iterations. X_L denotes the best individual in the current population. X_2 and X_3 are the individuals ranked second and third in fitness within the current population. Their mean value balances exploration and exploitation. X_0 represents the center of the entire population, providing global distribution information. Compared to the traditional COA algorithm, which relies solely on a single cave, the optimal cave strategy offers six possible solutions. This allows individuals to select different positions during the search process, enhancing global search capability and reducing the likelihood of getting trapped in local optima. The position is updated according to the following equation. During the iteration process, an individual is randomly selected from the optimal caves, where all candidate solutions have an equal probability of being chosen.

$$X_{i,j}^{t+1} = X_{i,j}^{t} + C_2 \times rand \times (X_{\text{cave\_best}} - X_{i,j}^{t}) \tag{18}$$

$$X_{i,j}^{t+1} = X_{i,j}^{t} - X_{z,j}^{t} + X_{\text{cave\_best}} \tag{19}$$

This strategy leverages the guiding effect of the optimal individuals to dynamically balance exploitation and exploration. By introducing diverse guiding points, such as the population center and the mean of the top three best individuals, it further enhances the flexibility of individual movement. As a result, the algorithm effectively avoids falling into local optima, improves global search performance, and accelerates convergence speed.

D. Lévy Flight

Lévy flight is a stochastic walk strategy used to simulate the foraging behavior of many organisms in nature. In the foraging phase, when the food size is appropriate, crayfish will feed. However, updating the position based on the difference between the current individual position and the food position is easily influenced by the food location, which may cause individuals to miss potential optimal solutions in other directions. Therefore, this paper adopts Lévy flight to perform food searches before feeding, randomly switching between long and short distance movements [18]. This approach helps the algorithm conduct effective global searches in the search space, avoiding local optima and enhancing search diversity and exploration capability.

$$X(t+1) = r \times (X_{\text{bc}}(t) - s \times X(t)) \tag{20}$$

$$s = \frac{u}{v^{1/\beta}} \tag{21}$$

$$u \sim N(0, \sigma_u^2), \quad v \sim N(0, \sigma_v^2) \tag{22}$$

$$\sigma_u = \left[\frac{\Gamma(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)\beta \cdot 2^{\frac{\beta-1}{2}}} \right]^{\frac{1}{\beta}}, \quad \sigma_v = 1 \tag{23}$$

In the formula: r is the parameter vector used to control exploration and exploitation; X_{bc} is the position vector of the best candidate in the current population; s is the Lévy flight path; the parameter β ranges between 0 and 2, typically taken as $\beta = 1.5$; the parameters μ and v are normally distributed random numbers; σ_u and σ_v are the standard deviations of the normal distributions.

E. Structure of the RCOA algorithm

To clearly illustrate the structure of the proposed RCOA, Algorithm 1 presents its pseudo-codes, which visualize the procedural framework of RCOA. These pseudo-codes facilitate a better understanding of the core steps and operational mechanisms of RCOA.

Algorithm 1: Pseudocode of RCOA

```
Initialize parameters.
Initialize the population according to Equation (13).
Evaluate the population fitness and obtain the best value.
t = 1;
while t ≤ T do
Compute temperature temp according to Equation (1).
Calculate control parameter C using Equation (14).
if temp > 30 then
Identify elite cave individuals according to Equation (15).
if rand < 50 then
Update position via cooling phase using Equation (18).
else
Update position via competition phase using Equation (19).
end if
else
Perform foraging using Lévy flight mechanism.
end if
Apply boundary control and update fitness values accordingly.
t = t + 1;
end while
```

IV. Performance Testing of Improved Crayfish Optimization Algorithm

A. Test Function

To verify the effectiveness of the improved crayfish optimization algorithm (RCOA) in global search and convergence speed, the performance of RCOA is evaluated using the CEC2022 benchmark function set [19]. CEC2022 provides four types of functions: unimodal functions, multimodal functions, hybrid functions, and composition functions, as shown in Table 1. Different types of functions can comprehensively and effectively test the optimization capability and stability of the algorithm.

B. Parameter setting

To ensure a fair comparison experiment, the parameter settings are as follows: the population size is set to 50, the number of iterations is 500, and each test function is independently executed 30 times. The optimal value, mean, and standard deviation are recorded. The search domain for all functions is $[-100, 100]$, and their dimensions are

TABLE I: Test functions

	No.	Function	F_i^*
Unimodal Function	1	Shifted and full Rotated Zakharov Function	300
Basic Functions	2	Shifted and full Rotated Rosenbrock's Function	400
	3	Shifted and full Rotated Expanded Schaffer's f6 Function	600
	4	Shifted and full Rotated Non-Continuous Rastrigin's Function	800
	5	Shifted and full Rotated Levy Function	900
Hybrid Functions	6	Hybrid Function 1 (N=3)	1800
	7	Hybrid Function 2 (N=6)	2000
	8	Hybrid Function 3 (N=5)	2200
Composition Functions	9	Composition Function 1 (N = 5)	2300
	10	Composition Function 2 (N = 4)	2400
	11	Composition Function 3 (N = 5)	2600
	12	Composition Function 4 (N = 6)	2700

10. The programming software used is MATLAB 2024a, running on a PC with 16GB of RAM and a 2.50GHz processor, with Windows 11 as the operating system. The improved algorithm, RCOA, is compared with four other algorithms: the original Crayfish Optimization Algorithm (COA) [4], the Whale Optimization Algorithm (WOA) [20], the Harris Hawks Optimization Algorithm (HHO) [21], and the Dung Beetle Optimization Algorithm (DBO) [22]. These algorithms have been widely validated and applied in previous studies, demonstrating strong optimization performance, which further highlights the effectiveness of the improved algorithm proposed in this paper.

C. Experimental Results and Analysis

RCOA is compared with three classical algorithms and COA, with the results detailed in Table 2. The comparison includes three performance metrics: mean value, standard deviation, and optimal value. The improved Crayfish Optimization Algorithm demonstrates excellent performance across various benchmark functions. For unimodal function F1, RCOA ranks first in performance. In multimodal functions (F2-F5), it consistently achieves the best performance. In hybrid functions (F6-F8), RCOA ranks first across all cases. For composition and asymmetric functions (F9-F12), it ranks first in F9, F10, and F12. Overall, for different types of test functions in the CEC2022 benchmark suite, the RCOA algorithm significantly outperforms the four comparison algorithms in terms of optimization performance. The improved RCOA achieves the best mean value, standard deviation, and optimal value for most functions, proving its effectiveness and strong problem-solving capability.

To better observe the convergence of the RCOA algorithm on the CEC2022 benchmark set, Figure 2-7 presents the iteration convergence curves of six selected functions from the CEC2022 test set for the five algorithms. The horizontal axis represents the number of iterations, while the vertical axis indicates the average fitness value after 30 independent runs. RCOA demonstrates significant advantages in both convergence speed and convergence accuracy, highlighting its strong competitiveness. In summary, the improved RCOA algorithm exhibits superior performance on the CEC2022 test set, achieving a better balance between exploration and exploitation compared to other algorithms.

D. Computational complexity analysis of RCOA

The algorithmic complexity is a key indicator for evaluating the efficiency of an algorithm. Time complexity studies the dominant computational processes of the algorithm to analyze the theoretical computation time of different algorithms. In this paper, we use Big-O notation to calculate the time complexity of COA and RCOA.

The time complexity calculation for the standard Crawling Optimization Algorithm (COA) is as follows: during the initialization phase of the algorithm, an initial population needs to be generated, which has a time complexity of $O(N \times D)$, where N is the population size and D is the problem dimension. During the iteration process, for each generation, the fitness values of all individuals must first be calculated, which has a time complexity of $O(N \times D)$; updating the positions of all individuals also has a time complexity of $O(N \times D)$. The algorithm repeats the fitness evaluation and position update until the maximum number of iterations T, is reached. Therefore, the overall time complexity of COA is $O(T \times N \times D)$.

The RCOA improvement algorithm introduces an opposition learning mechanism based on the COA algorithm. During the initialization phase, additional opposition individuals are generated and filtered, but the overall complexity remains $O(N \times D)$. After adopting the optimal cave strategy, although more guiding information is introduced, an extra $O(N \times D)$ computation for the population center is added each generation. However, this increase in complexity is still within the same order of magnitude and does not significantly affect the overall complexity. The introduction of a nonlinear convergence factor and Lévy flight does not add any extra loops or operations to the original algorithm's code. Therefore, the overall time complexity of RCOA remains $O(T \times N \times D)$, which is the same as that of COA, without increasing the runtime and remaining in the same complexity class.

V. RCOA-ELM Prediction Model

To verify the accuracy and effectiveness of the RCOA-ELM prediction model, flotation data collected from actual production in a certain region is selected as the research subject. The data is first preprocessed and then split into

TABLE II: Results of comparing algorithms on the CEC2022 benchmark function

F	Index	RCOA	COA	DBO	HHO	WOA
F1	Mean	3.0009E+02	3.6379E+03	1.7733E+03	1.1483E+03	2.9606E+04
	Std	1.9691E-01	2.8121E+03	1.5212E+03	5.1625E+02	1.2097E+04
	Min	3.0000E+02	3.2342E+02	3.0226E+02	4.1508E+02	9.8194E+03
	Rank	1	5	3	2	4
F2	Mean	4.0952E+02	4.1884E+02	4.2304E+02	4.7030E+02	4.9604E+02
	Std	1.2721E+01	2.7092E+01	2.8004E+01	8.6314E+01	9.7641E+01
	Min	4.0039E+02	4.0002E+02	4.0001E+02	4.0046E+02	4.0717E+02
	Rank	1	2	3	4	5
F3	Mean	6.0027E+02	6.0489E+02	6.0984E+02	6.4426E+02	6.4402E+02
	Std	4.6227E-01	6.3909E+00	6.9639E+00	1.0237E+01	1.3179E+01
	Min	6.0000E+02	6.0001E+02	6.0094E+02	6.1603E+02	6.1950E+02
	Rank	1	2	3	5	4
F4	Mean	8.0844E+02	8.2982E+02	8.3613E+02	8.2791E+02	8.4021E+02
	Std	2.4710E+00	6.6981E+00	1.3451E+01	7.5410E+00	1.7196E+01
	Min	8.0398E+02	8.0498E+02	8.1393E+02	8.1324E+02	8.1450E+02
	Rank	1	5	3	2	4
F5	Mean	9.0093E+02	1.0780E+03	1.0182E+03	1.3954E+03	1.4492E+03
	Std	1.4456E+00	2.3917E+02	1.4151E+02	1.7812E+02	3.1601E+02
	Min	9.0000E+02	9.0063E+02	9.0063E+02	1.1176E+03	9.7565E+02
	Rank	1	3	2	4	5
F6	Mean	2.2370E+03	4.2470E+03	4.4429E+03	5.7656E+03	6.8747E+03
	Std	1.2200E+03	2.1214E+03	1.9580E+03	3.2157E+03	4.8254E+03
	Min	1.8538E+03	1.8617E+03	1.8454E+03	2.0032E+03	2.3872E+03
	Rank	1	2	3	4	5
F7	Mean	2.0206E+03	2.0262E+03	2.0417E+03	2.0698E+03	2.0800E+03
	Std	6.4609E+00	2.3017E+01	2.0682E+01	3.1898E+01	2.5180E+01
	Min	2.0016E+03	2.0044E+03	2.0211E+03	2.0258E+03	2.0331E+03
	Rank	1	2	3	4	5
F8	Mean	2.2195E+03	2.2353E+03	2.2378E+03	2.2359E+03	2.2359E+03
	Std	7.2227E+00	3.1035E+01	3.3832E+01	1.2337E+01	8.1025E+00
	Min	2.2032E+03	2.2104E+03	2.2208E+03	2.2254E+03	2.2255E+03
	Rank	1	2	5	3	4
F9	Mean	2.5293E+03	2.5293E+03	2.5623E+03	2.6098E+03	2.6046E+03
	Std	6.7556E-13	4.0210E-03	4.3070E+01	4.1313E+01	4.7176E+01
	Min	2.5293E+03	2.5293E+03	2.5293E+03	2.5365E+03	2.5303E+03
	Rank	1	2	3	5	4
F10	Mean	2.5076E+03	2.5901E+03	2.5503E+03	2.5904E+03	2.7024E+03
	Std	2.7309E+01	1.2128E+02	6.6307E+01	1.4537E+02	3.2149E+02
	Min	2.5003E+03	2.5004E+03	2.5005E+03	2.5005E+03	2.5005E+03
	Rank	1	3	2	4	5
F11	Mean	2.8369E+03	2.8073E+03	2.7656E+03	2.8719E+03	3.0872E+03
	Std	1.1575E+02	1.2905E+02	1.3014E+02	2.5745E+02	2.2225E+02
	Min	2.6000E+03	2.6002E+03	2.6000E+03	2.6082E+03	2.7530E+03
	Rank	3	2	1	4	5
F12	Mean	2.8634E+03	2.8696E+03	2.8773E+03	2.9256E+03	2.9055E+03
	Std	1.0816E+00	1.2736E+01	1.8704E+01	5.5058E+01	3.9517E+01
	Min	2.8594E+03	2.8621E+03	2.8640E+03	2.8615E+03	2.8658E+03
	Rank	1	2	3	5	4
	Rank sum	14	30	35	46	55
	Final Standings	1	2	3	4	5

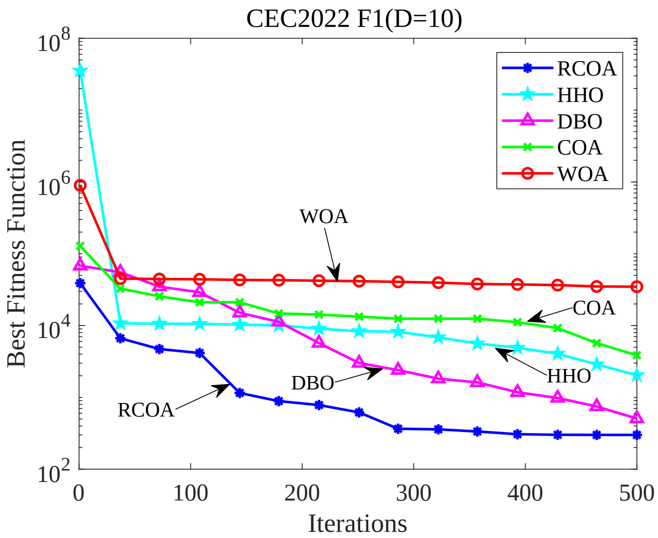

Fig. 2: Convergence curve of F1

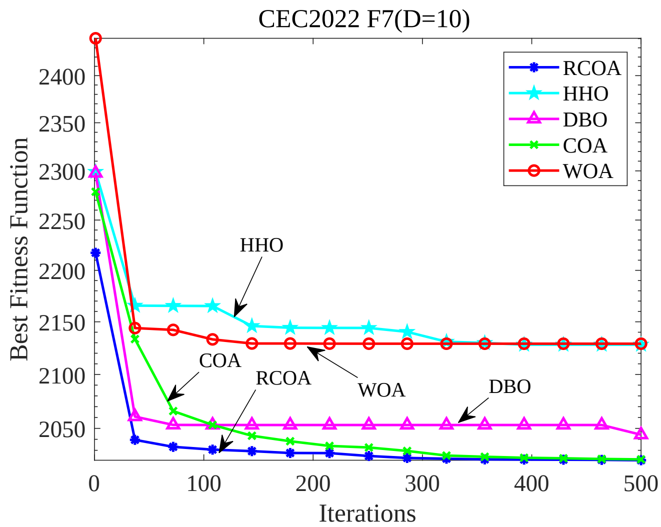

Fig. 5: Convergence curve of F7

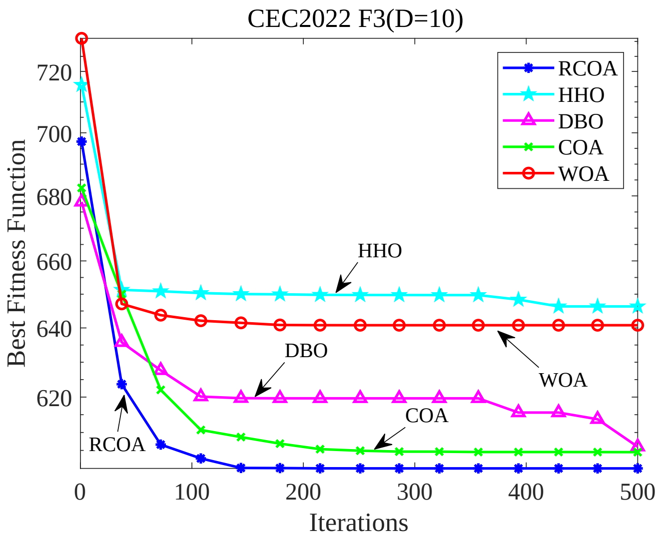

Fig. 3: Convergence curve of F3

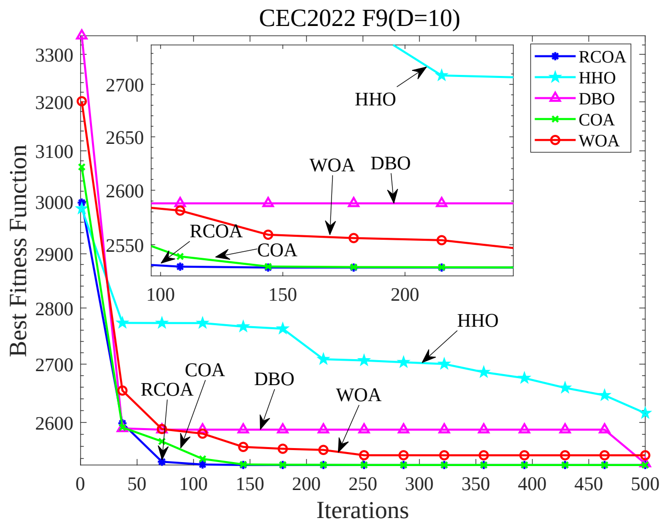

Fig. 6: Convergence curve of F9

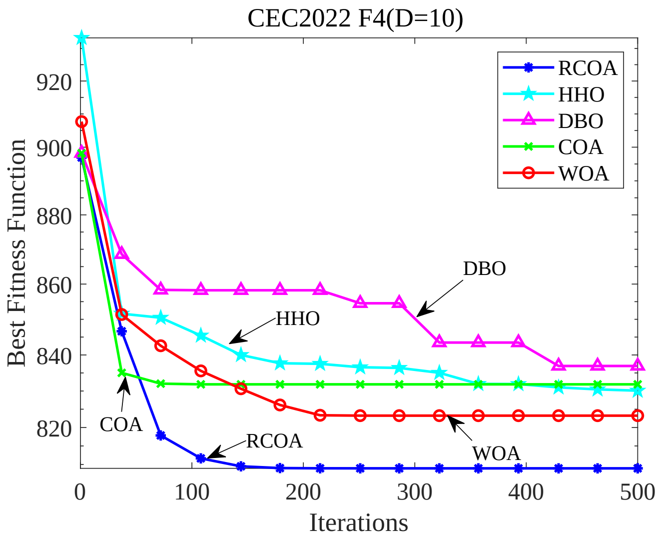

Fig. 4: Convergence curve of F4

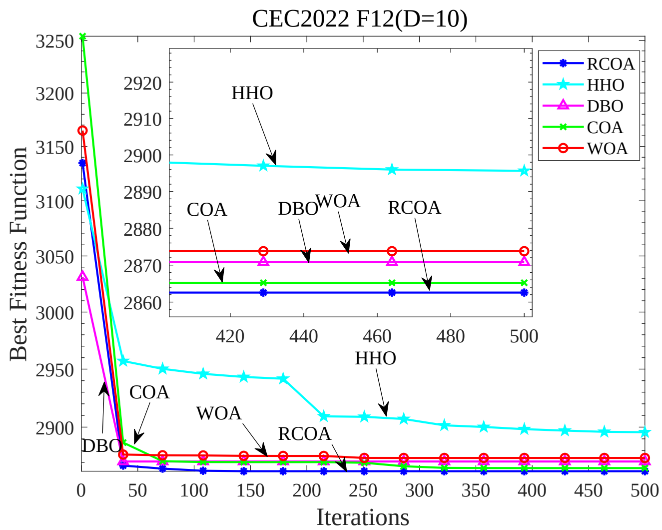

Fig. 7: Convergence curve of F12

a training set and a test set in an 8:2 ratio. The improved Crayfish Optimization Algorithm is used to optimize the ELM neural network model by obtaining the optimal initial weights and biases. A prediction model is then constructed to complete the case study on flotation concentrate grade prediction.

A. Data Preprocessing

This paper uses the 3σ-principle for outlier detection [23]. For the flotation concentrate grade data, outlier handling is performed based on the 3σ-principle using SPSS software. The basic process is as follows:

$$|x_{\text{data}} - x_{\text{mean}}| \geq 3\sigma \tag{24}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - x_{\text{mean}})} \tag{25}$$

$$x_{\text{mean}} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{26}$$

Where X_i is the value of a certain feature sample, and X_{mean} is the mean value. N is the number of samples. With the assistance of SPSS and Excel software, the original 616 data points were reduced to 446 data points, which include 5 feature dimensions such as feed grade and feed concentration. After data preprocessing, the data is split into a training set and a test set in an 8:2 ratio to establish the flotation concentrate grade prediction model for research.

B. Concentrate Grade Prediction Model

To improve the prediction accuracy of flotation concentrate grade, an RCOA-ELM hybrid model is used for flotation concentrate grade prediction research, with the structural diagram shown in Figure 8. The RCOA-ELM prediction model consists of two main modules: the RCOA module and the ELM module. The fitness evaluation is performed based on the update mechanism of the RCOA algorithm, and when the conditions are met, the global optimal value is recorded and assigned as the initial weights and biases of the ELM neural network. The ELM prediction model is then trained and calculated to complete the prediction of flotation concentrate grade.

VI. Experiment and Analysis

A. Experimental Evaluation Metrics

To verify the reliability of the flotation concentrate grade prediction model established in this paper, comparative experiments are conducted with the ELM, COA-ELM, HHO-ELM, WOA-ELM, and DBO-ELM models. A reasonable evaluation metric system is established, primarily using the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and the coefficient of determination R^2 to assess the prediction performance.

(1) Mean Absolute Error (MAE): This metric prevents errors from canceling each other out and can accurately reflect the average fluctuation between the model's predicted values and the true values of the samples.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{y}_{\text{pre}}(i) - \mathbf{y}_(i)\right| \tag{27}$$

(2) Root Mean Square Error (RMSE): The RMSE shows the deviation between the model's predicted values and the true values. The smaller the value, the higher the prediction accuracy of the model.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_{\text{pre}}(i) - y(i)\right)^2} \tag{28}$$

(3) Coefficient of Determination R^2: This describes the degree of fit of the model to the predicted problem. The closer the value is to 1, the stronger the model's ability to fit the data [24].

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(y_{\text{train}}(i) - y_{\text{pre}}(i)\right)^2}{\sum_{i=1}^{N}\left(y_{\text{train}}(i) - \overline{y}_{\text{train}}\right)^2} \tag{29}$$

B. Analysis of Experimental Results

To ensure fair experimental results, all model tests are conducted in the same testing environment, with identical hardware and software configurations. Comparative experiments are performed to predict flotation concentrate grade using the ELM network optimized by COA, HHO, DBO, WOA, and RCOA algorithms. The population size for all algorithms is set to 30, and the maximum number of iterations is 200. The upper and lower bounds of the initial positions are set to $[-1, 1]$. The performance comparison of each flotation prediction model is shown in Table 3, and the comparison between predicted values and true values is illustrated in Figures 4-9.

TABLE III: Comparison of flotation concentrate grade prediction results

Predictive models	MAE	RMSE	R^2
ELM	0.1004	0.1267	0.7836
WOA-ELM	0.0946	0.1201	0.8057
HHO-ELM	0.0933	0.1173	0.8147
DBO-ELM	0.0868	0.1119	0.8312
COA-ELM	0.0867	0.1116	0.8321
RCOA-ELM	0.0774	0.1045	0.8529

As shown in Table 3, the ELM neural network optimized by RCOA predicts flotation concentrate grade with higher accuracy. All performance metrics are significantly better than those of other algorithms, with MAE and RMSE values of 0.0774 and 0.1045, respectively, and a fitting coefficient R^2 of 0.8529, which is much higher than the models optimized by other algorithms. This indicates that the optimization effect of the improved RCOA algorithm is significant, achieving a good balance between global search ability and local search ability. Additionally, the prediction performance has improved compared to the original ELM neural network. From the comparison of the true values and predicted values in Figures 9-14, it can be seen that the RCOA-ELM neural network's predictions are much closer

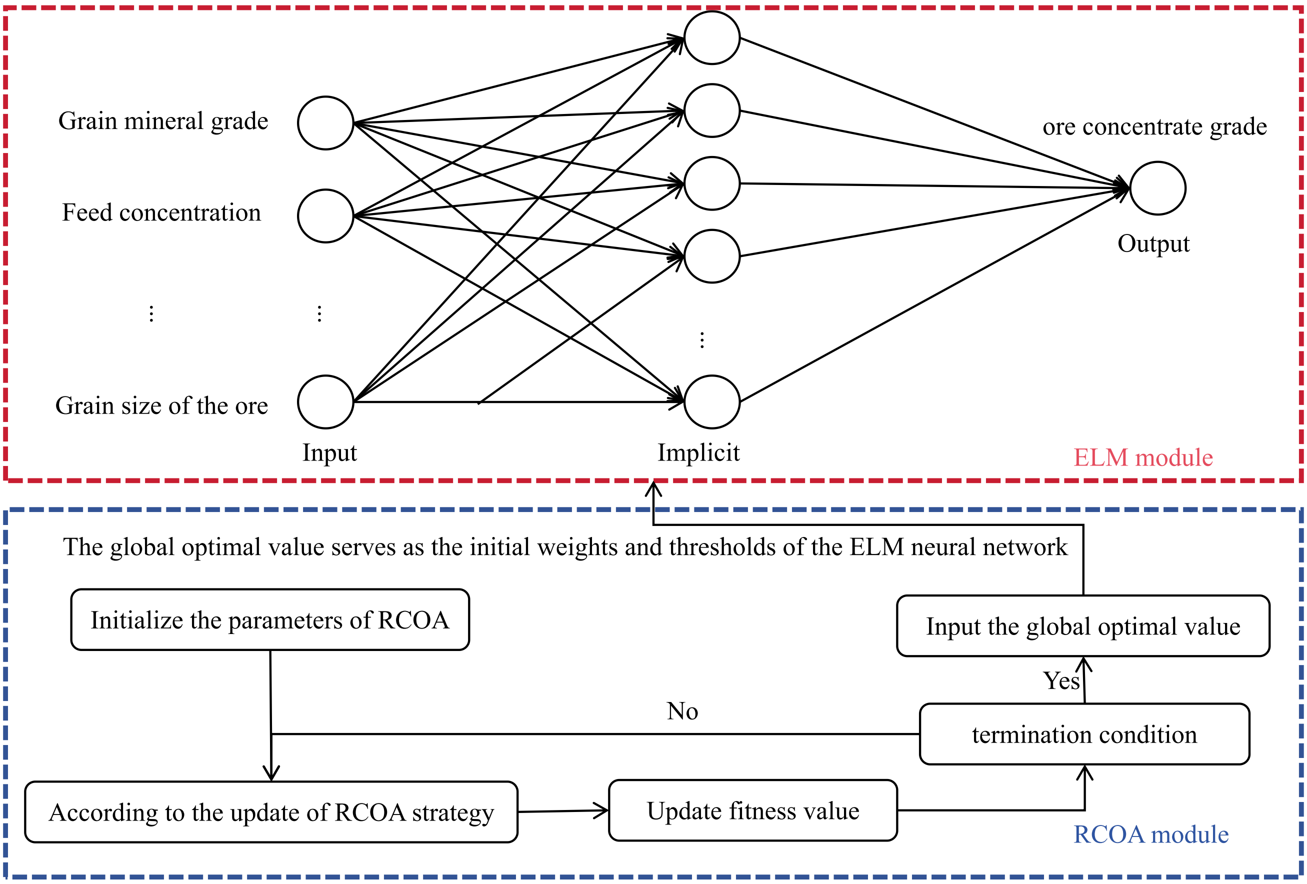

Fig. 8: The structure diagram of the RCOA-ELM flotation prediction model

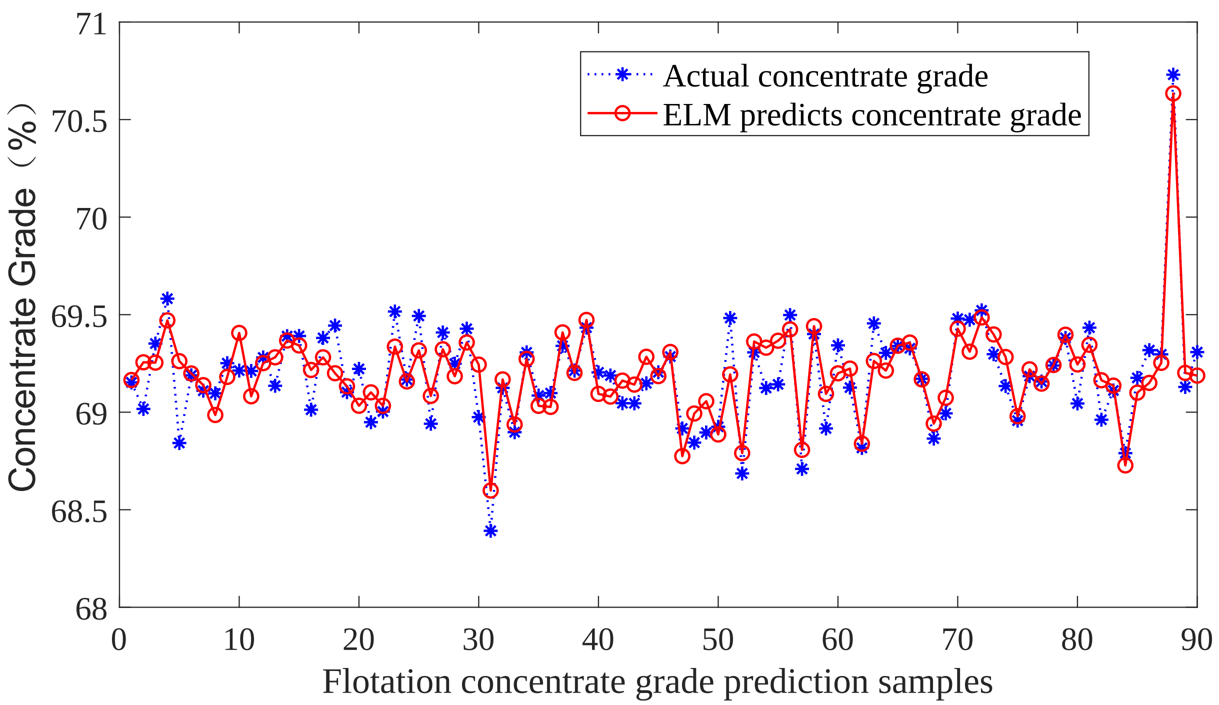

Fig. 9: Prediction result graph of ELM model

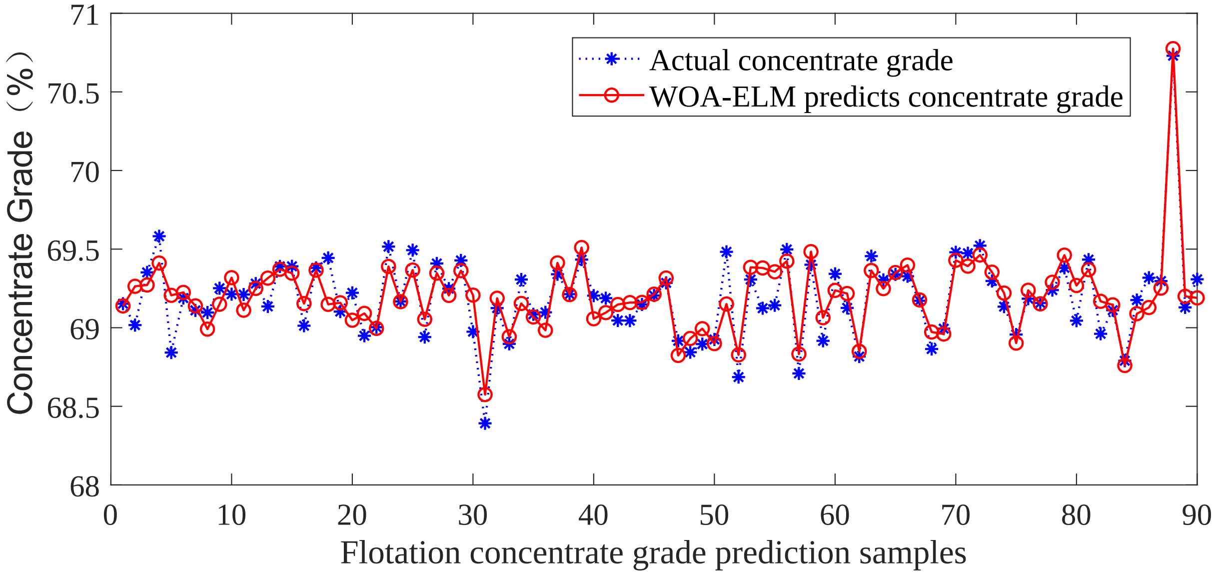

Fig. 10: Prediction result graph of WOA-ELM model

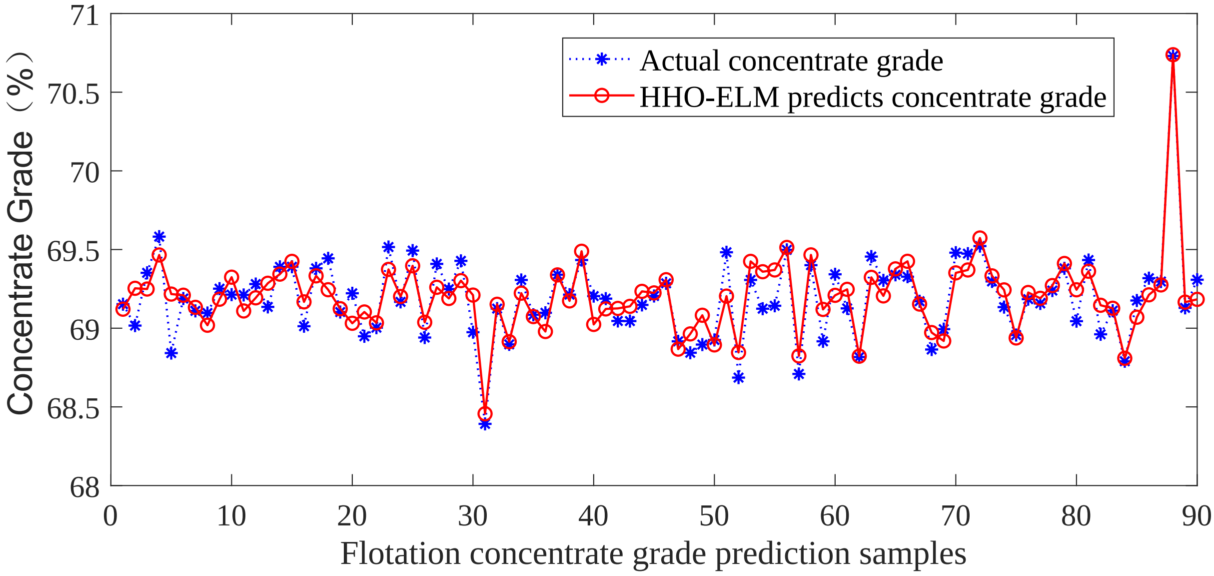

Fig. 11: Prediction result graph of HHO-ELM model

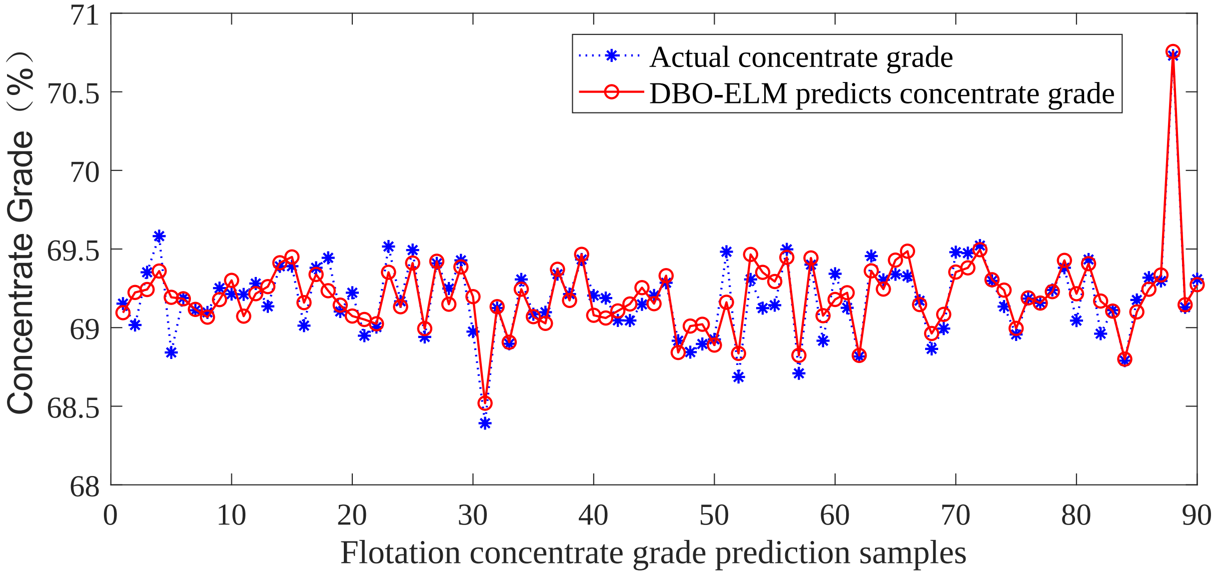

Fig. 12: Prediction result graph of DBO-ELM model

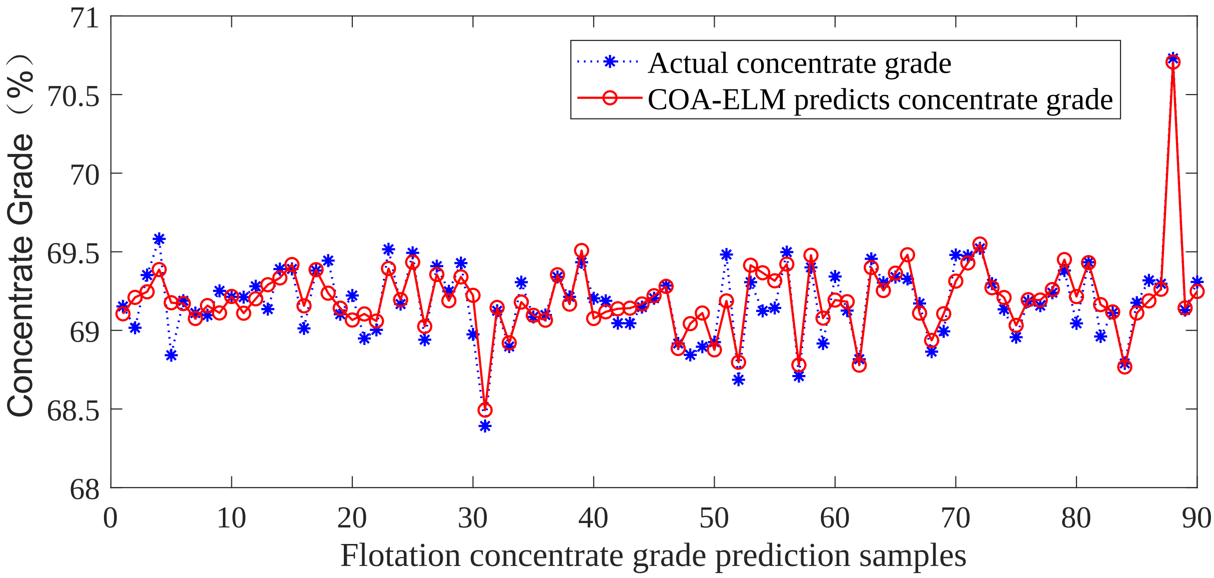

Fig. 13: Prediction result graph of COA-ELM model

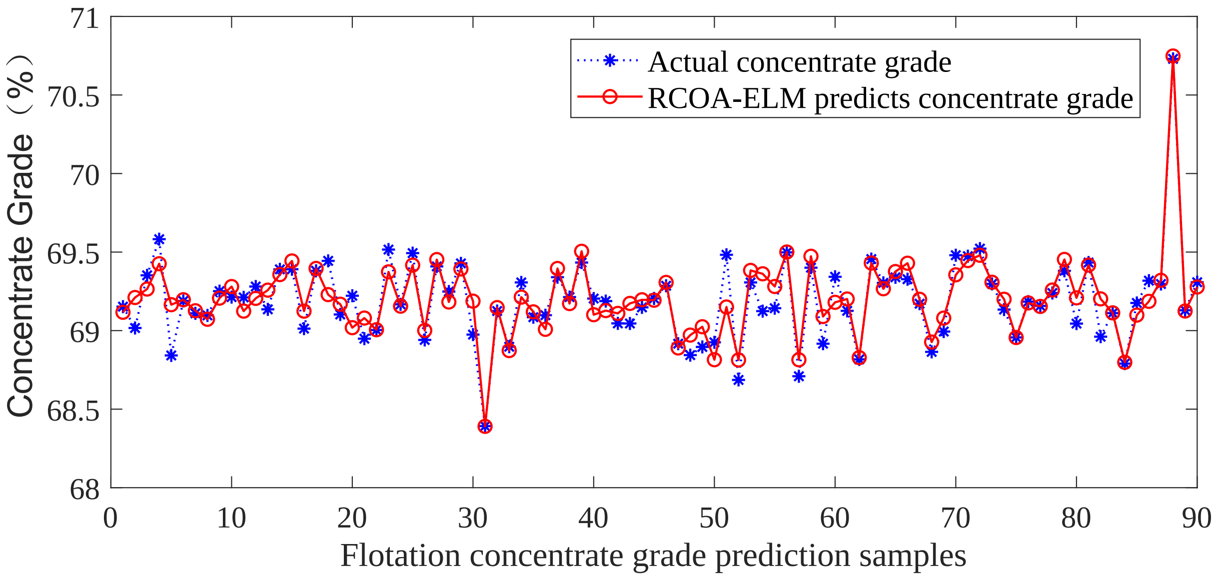

Fig. 14: Prediction result graph of RCOA-ELM model

to the true values. Through a combined analysis of the numerical results and figures, it can be concluded that the RCOA-ELM flotation concentrate grade prediction model exhibits better stability and predictive ability.

VII. Conclusion

The Crayfish Optimization Algorithm (COA) is a new swarm intelligence optimization algorithm. To address its limitations in optimization accuracy and its tendency to get stuck in local extrema, this paper proposes an improved Crayfish Optimization Algorithm (RCOA) that utilizes a nonlinear convergence factor and an optimal cave strategy. The optimal cave strategy increases the likelihood of individuals exploring the search space, while the nonlinear convergence factor balances global search ability and local exploitation ability. This improvement enhances both the convergence accuracy and speed, strengthening the optimization performance of the COA algorithm.The proposed RCOA is thoroughly compared with the original COA, WOA, HHO, and the latest swarm intelligence algorithm DBO. The numerical results and convergence curves of the CEC2022 benchmark functions show that the comprehensive performance of RCOA is significantly better than the comparison algorithms. The RCOA algorithm is used to optimize the weights and biases of the ELM model, establishing the RCOA-ELM prediction model. This model is applied to predict the flotation concentrate grade in mining, and a comparison with different algorithms optimizing the ELM prediction results shows that the optimized model has good predictive accuracy and can accurately predict the concentrate grade, proving the model's effectiveness and practical significance.

References

[1] S. Al-Thyabat, "On the optimization of froth flotation by the use of an artificial neural network," *Journal of China University of Mining and Technology*, vol. 18, no. 3, pp. 418–426, 2008.

[2] F. S. Hoseinian, B. Rezai, E. Kowsari, *et al.*, "A hybrid neural network/genetic algorithm to predict zn (ii) removal by ion flotation,"

Separation Science and Technology, vol. 55, no. 6, pp. 1197–1206, 2020.

[3] R. Cook, K. C. Monyake, M. B. Hayat, *et al.*, "Prediction of flotation efficiency of metal sulfides using an original hybrid machine learning model," *Engineering Reports*, vol. 2, no. 6, p. e12167, 2020.

[4] H. Jia, H. Rao, C. Wen, *et al.*, "Crayfish optimization algorithm," *Artificial Intelligence Review*, vol. 56, no. Suppl 2, pp. 1919–1979, 2023.

[5] N. H. Shikoun, A. S. Al-Eraqi, and I. S. Fathi, "Bincoa: an efficient binary crayfish optimization algorithm for feature selection," *IEEE Access*, vol. 12, pp. 28621–28635, 2024.

[6] H. Jia, X. Zhou, J. Zhang, *et al.*, "Modified crayfish optimization algorithm for solving multiple engineering application problems," *Artificial Intelligence Review*, vol. 57, no. 5, p. 127, 2024.

[7] L. Chaib, M. Tadj, A. Choucha, *et al.*, "Improved crayfish optimization algorithm for parameters estimation of photovoltaic models," *Energy Conversion and Management*, vol. 313, p. 118627, 2024.

[8] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[9] T. Guo, L. Zhang, and X. Tan, "Neuron pruning-based discriminative extreme learning machine for pattern classification," *Cognitive Computation*, vol. 9, no. 4, pp. 581–595, 2017.

[10] Y. Li and Z. Yang, "Application of eos-elm with binary jaya-based feature selection to real-time transient stability assessment using pmu data," *IEEE Access*, vol. 5, pp. 23092–23101, 2017.

[11] Y. Zhang, T. Li, G. Na, *et al.*, "Optimized extreme learning machine for power system transient stability prediction using synchrophasors," *Mathematical Problems in Engineering*, vol. 2015, no. 1, p. 529724, 2015.

[12] D. Miao, J. Ji, X. Chen, *et al.*, "Coal and gas outburst risk prediction and management based on woa-elm," *Applied Sciences*, vol. 12, no. 21, p. 10967, 2022.

[13] Z. Jia, Z. Song, J. Fan, *et al.*, "Prediction of blasting fragmentation based on gwo-elm," *Shock and Vibration*, vol. 2022, no. 1, p. 7385456, 2022.

[14] B. Yan, C. Xing, J. S. Wang, *et al.*, "Prediction model for complex process based on adaptive neural fuzzy inference system optimized by swarm intelligence optimization algorithms," *Engineering Letters*, vol. 33, no. 1, pp. 48–58, 2025.

[15] D. Hou, Y. Zhang, and J. Ren, "A lightweight object detection algorithm for remote sensing images," *Engineering Letters*, vol. 33, no. 3, pp. 704–711, 2025.

[16] S. Velarde-Gomez and E. Giraldo, "Multivariable nonlinear control based on exact feedback linearization with integral action for a PMSM," *Engineering Letters*, vol. 32, no. 12, pp. 2270–2277, 2024.

[17] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[18] H. Haklı and H. Uğuz, "A novel particle swarm optimization algorithm with levy flight," *Applied Soft Computing*, vol. 23, pp. 333–345, 2014.

[19] P. Bujok and P. Kolenovsky, "Eigen crossover in cooperative model of evolutionary algorithms applied to cec 2022 single objective numerical optimisation," in *2022 IEEE congress on evolutionary computation (CEC)*, pp. 1–8, IEEE, 2022.

[20] H. Deng, L. Liu, J. Fang, *et al.*, "A novel improved whale optimization algorithm for optimization problems with multi-strategy and hybrid algorithm," *Mathematics and Computers in Simulation*, vol. 205, pp. 794–817, 2023.

[21] A. A. Heidari, S. Mirjalili, H. Faris, *et al.*, "Harris hawks optimization: Algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.

[22] J. Xue and B. Shen, "Dung beetle optimizer: A new meta-heuristic algorithm for global optimization," *The Journal of Supercomputing*, vol. 79, no. 7, pp. 7305–7336, 2023.

[23] R. K. Pearson, "Outliers in process modeling and identification," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 1, pp. 55–63, 2002.

[24] M. Li and Y. Wang, "Power load forecasting and interpretable models based on gs_xgboost and shap," in *Journal of Physics: Conference Series*, vol. 2195, p. 012028, IOP Publishing, 2022.