



# Multi-strategy improved snow ablation optimizer: a case study of optimization of kernel extreme learning machine for flood prediction

Lele Cui<sup>1</sup> · Gang Hu<sup>1</sup> · Yaolin Zhu<sup>2</sup>

Accepted: 6 March 2025 / Published online: 24 March 2025  
© The Author(s) 2025

## Abstract

The Kernel Extreme Learning Machine (KELM) has the advantage of automatically extracting data features, learning and processing nonlinear problems from historical data, which can help achieve better prediction results for flood prediction problems with complex and sudden causes. Traditional flood disaster prediction usually only considers one influencing factor without considering the complex factors that affect flood occurrence. This article develops a new method for predicting the probability of flood occurrence based on 20 influencing factors. Firstly, in order to better utilize KELM performance, an improved snow ablation optimization algorithm (MESAO) was proposed for subsequent experiments by introducing a level based selection pressure mechanism, covariance matrix learning strategy, historical position based boundary adjustment strategy, and random centroid reverse learning strategy into snow ablation optimization (SAO). Secondly, MESAO is used to perform hyperparameter optimization on the regularization coefficient C and kernel function parameter S of the KELM model. Finally, the construction of a multi feature input–output model for the application of MESAO-KELM in flood prediction problems was completed. In terms of hyperparameter optimization, the numerical experimental results of this method were superior to the prediction results of 10 other intelligent algorithms and 5 regression prediction models. According to the evaluation index results, the best adaptability of MESAO optimized KELM and higher prediction accuracy and stability compared to other prediction models were demonstrated. This method overcomes the limitations of traditional prediction models based on a single influencing factor and can predict the probability of flood occurrence based on complex and variable factors. It can be said that MESAO-KELM has strong generalization ability. Accurate flood prediction can provide early warning and take measures in advance to protect and reduce the impact of floods on human and social development.

**Keywords** Flood forecasting · Prediction accuracy · Snow ablation optimizer · Boundary adjustment · Kernel extreme learning machine

Extended author information available on the last page of the article

## 1 Introduction

The harm of flood disasters is comprehensive (Zhang 2018), and there are many factors that affect the occurrence of floods. Accurately predicting floods based on multiple influencing factors is conducive to providing decision-making basis for national flood prevention and rescue and flood control system scheduling (Wen 2004), ensuring the safety of people's lives and property, improving urban management efficiency, constructing urban drainage systems, and enhancing disaster response capabilities (Xu et al. 2024; Chen 2024).

Flood detection refers to the process of identifying, monitoring, and alerting authorities or individuals to the presence or potential occurrence of floods in specific areas. It involves the use of various technologies and methods to detect, predict, and mitigate the impact of floods. At present, the modeling methods for flood prediction can be divided into physical process based flood forecasting models and data-driven flood forecasting models.

The flood forecasting model based on physical processes, also known as hydrological model, summarizes the physical conditions of the studied watershed and further combines them with hydrological empirical formulas. There are mainly three forms. The first type is the lumped expression, which removes local factors within the studied watershed for research purposes. The Stanford Basin Hydrological Model proposed by Linsley et al. (1960) is one of the most representative models, which can comprehensively consider multiple hydrological factors and has good model stability. However, the main disadvantage of this model is that it ignores the spatial differences of hydrological phenomena, has a single application scenario, and cannot flexibly adapt to the hydrological conditions at that time when facing extreme climate events. The water tank model proposed by Suga et al. (2000) has been applied to hydrological research of multiple rivers in Japan. It has the advantages of flexible structure, strong corrosion resistance, and space saving. However, this model does not use mathematical formulas to reflect nonlinear characteristics, and requires excessive equipment consumption and complex maintenance in the later stage. The second type is distributed, which represents and simulates hydrological processes based on the dynamic mechanism of the water cycle. The watershed is divided into multiple individuals, and different individuals are estimated. Finally, all studies are combined to obtain the final hydrological process. Abbott et al. (1986a, 1986b) established the SHE (System Hydrological European) model to estimate hydrological processes in European waters. This model can estimate evapotranspiration, precipitation, and runoff processes in European watersheds, but it does not take into account spatial heterogeneity caused by different terrains within the watershed. The third type is semi distributed, which means there is no specific actual division of watershed grids except for terrain. For example, the VIC (Variable Infiltration Capacity Macroscale Hydrologic Model) model divides the watershed into several networks in reference (Cherkauer et al. 2003), each with different underlying surface parameters, and requires separate calculations for each grid to obtain the final results. The multi-level soil structure of this model can well reflect the dynamic changes in soil moisture, but the technical requirements of this model are too high and the computational resources required in the simulation process are too large.

The data-driven flood forecasting model uses data as a carrier to build different watershed hydrological models based on different data characteristics, thereby achieving better prediction results. Mainly includes using neural networks and machine learning related methods for prediction. Li et al. (2024a) proposed a hybrid flood forecasting model based on

Transformer and LSTM, which achieved good prediction accuracy in the same watershed in different years. This model improved the accuracy of flood flow prediction, but this method has extremely high requirements for data quality, so there is an urgent need to reduce its computational complexity to improve stability. Sogol Moradian et al. (2024) combined fluid mechanics and machine learning (ML) models to generate flood predictions. This method can provide short-term flood forecasts for different situations of composite coastal rivers, but it can only generate accurate forecasts when high-quality training datasets are provided. Yuan et al. (2024) utilized the performance of convolutional neural networks to predict the depth of floods in cities in order to reduce economic losses. This method can identify the extent and depth of urban flooding in various situations, but the accuracy of this method in predicting extreme flood situations still needs to be improved. Liao et al. (2023) used a convolutional neural network based on clustering to quickly predict urban floods, this method has high prediction accuracy, but when the prediction time is too early, the prediction accuracy will decrease and the prediction range will gradually lose credibility.

The above studies usually predict the depth or range of flood inundation in a certain region, or provide flood forecasting and warning for a certain river or watershed. The purpose of this study is to find a method that can predict the probability of flood occurrence based on flood related information. Through research, we realized that the core of building flood prediction models is to consider complex comprehensive factors. Only considering a single or a few influencing factors can lead to prediction fluctuations and loss of credibility. The selected model must ensure strong generalization and have the ability to efficiently process such nonlinear data. Therefore, this article chooses the kernel extreme learning machine as the model basis, the kernel extreme learning machine has excellent learning speed and generalization ability, and has performed well in solving numerous problems. For example, Zhou et al. (2024) applied KELM to accurately predict the settlement of concrete faced gravel dams, which has higher predictive performance and accurate prediction intervals compared to traditional methods. Li et al. (2024b) used KELM for fault diagnosis of aircraft engines, which achieved better diagnostic errors and effectiveness compared to traditional diagnostic methods. Wu et al. (2024) applied KELM to the detection of overhead contact systems in electrified railways, and the results showed that this method improved detection accuracy and efficiency. Quan et al. (2024) combined nuclear principal component analysis with an improved sparrow optimization algorithm and applied it to the fault diagnosis of fuel cells. The results showed that the overall accuracy of this method reached 99.5%. Zhang et al. (2023) proposed a multi label classifier based on KELM and ensemble learning, which has better classification performance compared to other methods. However, when using KELM for flood prediction, the prediction results are affected by the regularization coefficient and kernel parameters of the model parameters, and appropriate parameters are beneficial for improving the accuracy of the model. When searching for the optimal parameters, this article uses intelligent algorithms to optimize the model parameters. A competitive optimization algorithm will greatly improve the predictive performance of KELM. There are some classic algorithms such as genetic algorithm (GA) (Kramer 2017), differential evolution (DE) (Storn and Price 1997), Particle Swarm Optimization (PSO) (Kennedy and Eberhart 1995), etc. There are also algorithms inspired by biological populations, such as Zebra Optimization Algorithm (ZOA) (Trojovská et al. 2022), Grey Wolf Optimization (GWO) (Mirjalili et al. 2014a), Osprey Optimization Algorithm (OOA) (Dehghani and Trojovský 2023) and others. Algorithms inspired by the physical and chemical rules

of the universe, such as Energy Valley Optimization (EVO) (Azizi et al. 2023), Kepler Optimization Algorithm (KOA) (Mohamed et al. 2023), Multi-Verse Optimization (MVO) (Mirjalili et al. 2016) and others. Inspired by human behavioral habits, some algorithms have been proposed, such as Imperialist Competitive Optimization (ICA) (Atashpaz-Gargari and Lucas 2007), Teaching Learning based Optimization (TLBO) (Rao et al. 2012), League Championship Algorithm (LCA) (Kashan 2014), etc. This study involves the Snow Ablation Optimizer (SAO) (Deng and Liu 2023), which is inspired by a series of physical changes in snow in nature. This algorithm is used to reflect the highly dispersed characteristics of individual populations in the solution space. The optimization mechanism of SAO has shown good performance in handling optimization problems and has performed well in practical applications.

We have found that there is an urgent need for a flood prediction model that meets both accuracy and comprehensive consideration of multiple influencing factors. This model can not only improve prediction accuracy but also consider influencing factors based on complex real-world situations, thereby enhancing reliability. Therefore, this article proposes an improved optimization algorithm for optimizing model parameters to help improve prediction accuracy. The main contributions are as follows. Firstly, the MESAO-KELM flood prediction model introduced in this study can simultaneously consider a large number of influencing factors and ensure prediction accuracy. The model combines the KELM model and MESAO algorithm. Secondly, the effectiveness and accuracy of the model were verified through practical case studies. The model was compared with 10 other intelligent algorithms for optimizing KELM in numerical experiments and also compared with 5 prediction models. Finally, multiple sets of experiments have verified that the algorithm proposed in this paper has excellent performance. This progress provides important scientific basis for flood prediction based on multiple influencing factors.

This article elaborates on the relevant concepts of nuclear extreme learning machines in Sect. 2. Section 3 introduces the proposed MESAO algorithm. Section 4 presents relevant experiments on flood prediction using MESAO-KELM. The last part is a summary of the entire text.

## 2 Kernel extreme learning machine (KELM)

ELM belongs to a type of feedforward neural network, known for its fast learning speed and relatively simple implementation process. ELM has the characteristics of strong generalization, wide applicability, and only optimizing output layer weights and randomly initializing hidden layer weights (Huang et al. 2012). The combination of ELM and physical information neural networks provides higher accuracy for evaluating error norms in solving three-dimensional problems (Wang et al. 2025), ELM can be used for parameter optimization of the latest proposed mobile adaptive task processing model and applied to the operation of agricultural robots under external constraints, in order to improve the adaptability and accuracy of the operation (Alrowaily et al. 2024). However, in terms of training efficiency and generalization, the effect of random vector function linking (RVFL) is very close to that of ELM. For example, a WCRVFL network model for COVID-19 prediction is proposed by combining RVFL with wavelet coupling. This model has excellent prediction accuracy and provides powerful help for controlling the spread of the epidemic (Hazarika and Gupta

2020), there is also the use of fuzzy functions to calculate the hidden layer output of deep non iterative random vector function links, which can achieve a high level of accuracy in the feature classification problem of Alzheimer's disease (Sharma et al. 2021). Although RVFL has good performance and accuracy, its parameter settings are more complex and its usage scenarios are limited. In comparison, ELM is simple and easy to use with many advantages, which is more in line with the use and purpose of this study. The schematic diagram of ELM structure is shown in Fig. 1, in the figure,  $\omega$  and  $b$  represent the weights and biases of the hidden layer, while  $\beta$  represents the weights of the output layer.

The specific expression is as follows:

$$\begin{cases} f(x) = \mathbf{h}(x)\beta = \mathbf{H}\beta, \\ \beta = \mathbf{H}^T(\mathbf{I}/C + \mathbf{H}\mathbf{H}^T)^{-1}\mathbf{T}. \end{cases} \quad (1)$$

Among them,  $x$  is the output of the training samples,  $f(x)$  is the actual output,  $\mathbf{h}(x)$  is the sample matrix, and  $\mathbf{H}$  is the output matrix of the hidden layer,  $\rho$  is the weight vector between the hidden layer and the output layer,  $\mathbf{T}$  is the output matrix of the training samples,  $C$  is the regularization factor, and  $\mathbf{I}$  is the diagonal matrix.

ELM adopts a random allocation method when generating the output matrix of the hidden layer, which results in different output matrices and changes in weights, lacking stability and robustness. Therefore, combining ELM with kernel functions improves the predictive performance of the model while retaining the advantages of ELM. The kernel function is expressed as:

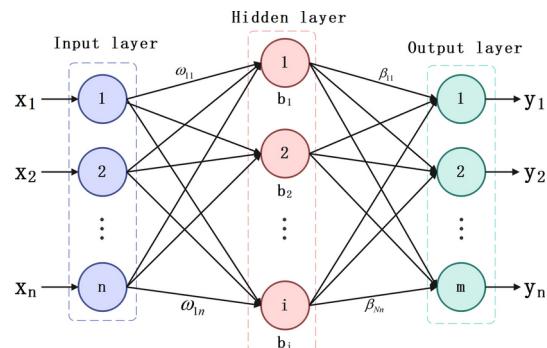
$$\Omega_{ELM} = \mathbf{H}\mathbf{H}^T = \mathbf{h}(\mathbf{x}_i)\mathbf{h}(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j). \quad (2)$$

where  $\mathbf{x}_i$ ,  $\mathbf{x}_j$  is the input vector and  $K(\mathbf{x}_i, \mathbf{x}_j)$  is the radial basis function.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2). \quad (3)$$

where  $\|\mathbf{x}_i - \mathbf{x}_j\|$  is the Euclidean distance between sample  $i$  and  $j$ , and  $\gamma$  is the kernel coefficient. Therefore, the Kernel Extreme Learning Machine (KELM) is represented as:

**Fig. 1** Schematic diagram of ELM structure



$$\mathbf{f}(x) = \begin{bmatrix} k(x, x_1) \\ \vdots \\ k(x, x_N) \end{bmatrix}^T (\mathbf{I}/C + \Omega_{ELM})^{-1} T. \quad (4)$$

where  $x_1, \dots, x_N$  is the training sample.

### 3 Improved snow ablation optimization (MESAO)

#### 3.1 Snow ablation optimizer

The ablation of snow is beneficial to the development of nature and human health. SAO is inspired by the ablation and sublimation behavior of snow during ablation. Snow becomes steam through sublimation and melts into liquid water, which can be evaporated into steam.

#### 3.2 Mathematical modeling of SAO

##### (1) Initialization

In SAO, the population is first randomly generated and modeled as a matrix with  $Num$  rows and  $D$  columns, satisfies Eq. (5).

Among them,  $Num$  is the population size,  $D$  is the dimension of the solution space,  $Low$  is the lower bound of the solution space,  $Up$  is the upper bound of the solution space, and  $\alpha$  is the random number between  $[0, 1]$ .

$$\mathbf{V} = Low + \alpha \times (Up - Low) = \begin{bmatrix} V_{1,1} & V_{1,2} & \cdots & V_{1,D} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ V_{Num-1,1} & V_{Num-1,2} & \cdots & V_{Num-1,D-1} \\ V_{Num,1} & V_{Num,2} & \cdots & V_{Num,D-1} \end{bmatrix}_{Num \times D}. \quad (5)$$

##### (2) Exploration stage

When snow or melted liquid water is converted into steam, the search agent exhibits highly dispersed characteristics due to irregular motion. Therefore, Brownian motion is used to simulate this situation, specifically:

$$f_{Bu}(h; 0, 1) = \frac{1}{\sqrt{2\pi}} \times \exp\left(-\frac{h^2}{2}\right). \quad (6)$$

The calculation formula for the position during the exploration phase is:

$$V_a(i+1) = El(i) + Bu_a(i) \otimes (\alpha_1 \times (Gol(i) - V_a(i)) + (1 - \alpha_1) \times (\bar{V}(i) - V_a(i))). \quad (7)$$

The  $V_a(i)$  here represents the  $i$  th individual in the  $a$  th iteration process,  $Bu_a(i)$  is a random vector based on Gaussian distribution, representing Brownian motion,  $Gol(i)$  represents the

current optimal solution,  $El(i)$  represents randomly selected individuals from several elite groups in the population,  $\bar{V}(i)$  is the centroid position of the population, specifically:

$$\bar{V}(i) = \frac{1}{\text{Num}} \sum_{a=1}^{\text{Num}} V_a(i). \quad (8)$$

$$El(i) \in [Gol(i), V_{second}(i), V_{third}(i), V_c(i)]. \quad (9)$$

Among them,  $V_{second}(i)$  and  $V_{third}(i)$  respectively represent the second and third best individuals in the current population,  $V_c(i)$  is the centroid position of the individuals with fitness values ranking in the top 50%, and they are referred to as leaders. The mathematical expression is:

$$V_c(i) = \frac{1}{\text{Num}_1} \sum_{a=1}^{\text{Num}_1} V_a(i). \quad (10)$$

$\text{Num}_1$  is the number of leaders, equal to half the size of the population,  $V_a(i)$  is the best leader for the  $a$  iteration, and  $El(i)$  is randomly selected from the set consisting of the current optimal solution, the second best individual, the third best individual, and the centroid position of the leader.

### (3) Development stage

When snow is converted into liquid water through the ablation process, search agents are encouraged to develop high-quality solutions around the current best solution, using the degree day method to reflect the ablation process, which generally takes the form of:

$$R = Df \times (T_{ave} - T_1). \quad (11)$$

Among them,  $R$  is the snow ablation rate, which is a key parameter for simulating snow ablation behavior during the development stage.  $T_{ave}$  is the daily average temperature, and  $T_1$  is the reference temperature, set to 0. As a result, there are,

$$R = Df \times T. \quad (12)$$

The  $Df$  here is the degree day method factor,  $Df \in [0.35, 0.6]$ , and the updated  $Df$  after each iteration is:

$$Df = 0.35 + 0.25 \times \frac{e^{\frac{i}{i_{\max}}} - 1}{e - 1}. \quad (13)$$

The  $i_{\max}$  here represents the termination condition, and the calculation method for snow ablation rate is:

$$Df = (0.35 + 0.25 \times \frac{e^{\frac{i}{i_{\max}}} - 1}{e - 1}) \times T(i), T(i) = e^{\frac{-i}{i_{\max}}}. \quad (14)$$

The formula for updating the position during the development phase is:

$$V_a(i+1) = R \times Gol(i) + Bu_a(i) \otimes (\alpha_2 \times (Gol(i) - V_a(i)) + (1 - \alpha_2) \times (\bar{V}(i) - V_a(i))). \quad (15)$$

Among them,  $\alpha_2 \in [-1, 1]$ , with the help of cross term  $-\alpha_2 \times (Gol(i) - V_a(i))$  and  $(1 - \alpha_2) \times (\bar{V}(i) - V_a(i))$  in this stage, is more conducive to individual development of areas with greater potential based on the optimal search agent and group centroid position.

#### (4) Dual population mechanism

As mentioned earlier, liquid water derived from snow can also be converted into steam for exploration, which means that individuals gradually increase the possibility of irregular movements with highly dispersed features. The algorithm gradually tends to explore the solution space, so a dual population mechanism is designed for description, as shown in Algorithm 1. Divide the entire population  $Pop$  into two subpopulations  $Pop_a$  and  $Pop_b$ , the sizes of the three populations are  $Num_a$ ,  $Num_b$  and  $Num$ , respectively,  $Pop_a$  is used for exploration, while  $Pop_b$  is used for development. As the iteration progresses, the size of  $Pop_b$  gradually decreases and the size of  $Pop_a$  correspondingly increases. Algorithm 1 provides the pseudocode for this mechanism:

In summary, the complete position update equation for SAO is as follows:

$$V_a(i+1) = \begin{cases} El(i) + Bu_a(i) \otimes (\alpha_1 \times (Gol(i) - V_a(i)) + (1 - \alpha_1) \times (\bar{V}(i) - V_a(i))), & a \in index_a \\ R \times Gol(i) + Bu_a(i) \otimes (\alpha_2 \times (Gol(i) - V_a(i)) + (1 - \alpha_2) \times (\bar{V}(i) - V_a(i))), & a \in index_b \end{cases}. \quad (16)$$

Among them, as stated in Eq. (5),  $index_a$  and  $index_b$  respectively represent a set of indices that include the row numbers of individuals in  $Pop_a$  and  $Pop_b$  in the entire position matrix.

### 3.3 The algorithm proposed in this article

SAO, as a new swarm intelligence algorithm, can perform well in simple optimization problems. However, as the problem becomes increasingly complex and high-dimensional, it requires a large amount of computational resources, resulting in high computational costs. The performance of SAO significantly decreases when solving complex multimodal and combinatorial functions, which means it is more likely to enter local optima. In addition, the balance between global exploration and local development of SAO becomes difficult to maintain in the later stages of iteration, leading to a decrease in optimization accuracy and efficiency. Therefore, this section proposes some improvement strategies and presents an improved Snow Ablation Optimizer (MESAO) that better balances exploration and development while ensuring performance, improves convergence speed while avoiding getting stuck in local optima, and maintains population diversity during the iteration process, using the improved algorithm for subsequent work can improve efficiency and obtain better experimental results.

In MESAO, the performance of SAO is enhanced through the use of three strategies proposed in this paper: a level based selection pressure mechanism, a historical position based boundary adjustment strategy, a random centroid reverse learning strategy, and an existing covariance matrix learning strategy. Use a grade based selection pressure mechanism to further explore potential areas in the search space and achieve a better balance with the development phase. The covariance matrix learning strategy is a further improvement

during the development phase to avoid individuals falling into local optima. The boundary adjustment strategy based on historical location increases the diversity of the population by processing individual positions. The random centroid reverse learning strategy improves the accuracy and convergence speed of the algorithm, enabling faster search near potentially optimal positions.

### 3.3.1 Level based selection pressure mechanism

In the exploration stage of the original SAO, four elite individuals were obtained according to formula (8), while in the position update of formula (15), one of the individuals was randomly selected for position update, resulting in a slow convergence speed towards the global optimum and a gradual decrease in convergence speed. Therefore, this article proposes a grade based selection pressure mechanism to improve the convergence speed of the algorithm.

We assign different probabilities to different elite individuals, with a probability value of  $Gol(i) > V_{second}(i) > V_{third}(i) > V_c(i)$ , expressed as follows:

$$Rank_{Ri} = \phi(4 - \mathbf{Ri}) + 1. \quad (17)$$

$$Pr_{\mathbf{Ri}} = Rank_{\mathbf{Ri}} / (Rank_1 + Rank_2 + \dots + Rank_{\mathbf{Ri}}). \quad (18)$$

where  $Rank_{\mathbf{Ri}}$  is the level of elites,  $Pr_{\mathbf{Ri}}$  is the probability value assigned to different elites, and the maximum and minimum levels are assigned to the best and worst individuals respectively,  $\mathbf{Ri} \in \{1, 2, 3, 4\}$  represents the index of four elite individuals,  $\phi$  is used to adjust the control factor based on rank selection and  $\phi = 1$  which means assigning different selection probability values to the four elite individuals to select one elite individual for position update,

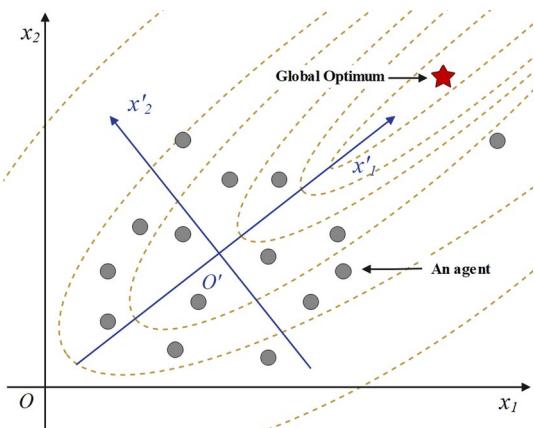
$$El\_Rank(i) \in [Gol(i) \cdot Pr_{R1}, V_{second}(i) \cdot Pr_{R2}, V_{third}(i) \cdot Pr_{R3}, V_c(i) \cdot Pr_{R4}]. \quad (19)$$

Therefore, the location update during the exploration phase is as follows:

$$V_a(i+1) = El\_Rank(i) + Bu_a(i) \otimes (\alpha_1 \times (Gol(i) - V_a(i)) + (1 - \alpha_1) \times (\bar{V}(i) - V_a(i))). \quad (20)$$

### 3.3.2 Covariance matrix learning strategy

The covariance matrix learning strategy (CML) reestablishes a feature coordinate system for the current population, reducing the algorithm's dependence on the original fixed coordinate system, enhancing the algorithm's rotation without deformation, and thereby strengthening the algorithm's performance in solving indivisible problems (i.e. problems that are difficult to decompose into independent sub problems or partial problems, and have high coupling between different parts of the problem) (Chen et al. 2016). The feature coordinate system is shown in Fig. 2. In SAO, the optimal position of each individual guides the entire population towards the global optimal  $Gol(i)$ , but individual optima face the dilemma of lacking independent learning and experience, making it difficult to improve and prone to falling into local optima. The CML strategy is often used to solve complex problems, allowing the

**Fig. 2** Feature coordinate system

population to learn from the constructed covariance matrix. Therefore, this section applies this strategy to improve the SAO development phase, avoiding the algorithm from getting stuck in local optima and providing a better balance between exploration and development.

Among them, the dashed line in the figure represents the contour lines of the function. Compared with the original fixed coordinate system  $Ox_1x_2$ , the search individuals in the feature coordinate system  $Ox_1x_2'$  are more suitable for the contour lines of the function and are more likely to find the global optimal solution.

The covariance matrix learning strategy mainly includes two core steps. The first step is the decomposition of covariance features and coordinate transformation. Firstly, we need to construct a covariance matrix. In this paper, we directly calculate the covariance matrix of the entire population. For a given population  $v_1, v_2, \dots, v_{Num}$  with  $D$  decision variables, the covariance matrix of the population in the  $i$  th iteration is as follows:

$$\text{cov}(m, n) = \frac{\sum_{k=1}^{Num} (V_{(k, m)}^i - \bar{V}_m^i)(V_{(k, n)}^i - \bar{V}_n^i)}{Num - 1}, \quad m = 1, \dots, D; n = 1, \dots, D. \quad (21)$$

where,

$$\bar{V}_m^i = \frac{\sum_{k=1}^{Num} V_{(k, m)}^i}{Num}. \quad (22)$$

The mean of the  $m$  th dimension is represented, and the covariance matrix of the entire population is expressed as:

$$\mathbf{Cov}(V) = [\text{cov}(m, n)]_{D \times D}. \quad (23)$$

The second step is to perform feature decomposition on the covariance matrix. In fact, the covariance matrix of a population is a real symmetric matrix. According to the knowledge of linear algebra, any real symmetric matrix can be decomposed into the following form:

$$\mathbf{Cov}(V) = \mathbf{Q}\Lambda\mathbf{Q}^T. \quad (24)$$

This is an important property of symmetric matrices, called the spectral theorem. Among them,  $\mathbf{Q}$  is an orthogonal matrix and also the eigenvector matrix of the covariance matrix  $\mathbf{Cov}(V)$ , with a diagonal matrix in the middle that stores the eigenvalues of  $\mathbf{Cov}(V)$ . In addition, multiplying an orthogonal matrix by a vector, whose vector length remains constant, geometrically represents a certain angle rotation of the vector. This also indicates that the eigenvectors in Fig. 2 are rotated from the original coordinate system, thereby enhancing the rotation invariance of the algorithm and improving its performance.

In MESAQ, the position of the development phase has been updated as follows:

$$\mathbf{QQ} = \mathbf{V} \times \mathbf{Q}. \quad (25)$$

$$V_a(i+1) = V_a(i) + \mathbf{QQ}_a(i). \quad (26)$$

Among them,  $\mathbf{QQ}$  is the product of the population generation matrix of formula (5) multiplied by the orthogonal matrix.

Therefore, the complete position update equation of MESAQ is as follows:

$$V_a(i+1) = \begin{cases} El\_Rank(i) + Bu_a(i) \otimes (\alpha_1 \times (Gol(i) - V_a(i)) + (1 - \alpha_1) \times (\bar{V}(i) - V_a(i))), & a \in \text{index}_a \\ V_a(i) + QQ_a(i), & a \in \text{index}_b \end{cases}. \quad (27)$$

### 3.3.3 Boundary adjustment strategy based on historical location

Based on Eqs. (7) and (15), we have obtained the position of the entire population. Next, we will perform boundary processing on the obtained positions to ensure the correctness and robustness of the algorithm by applying specific processing methods to the boundaries or extreme cases. In SAQ, the boundary treatment used is,

$$V_{a,j}(i+1) = \begin{cases} Up, & V_{a,j}(i+1) > Up, \\ Low, & V_{a,j}(i+1) < Low. \end{cases} \quad (28)$$

When individuals exceed the upper or lower boundaries, they will be fixed indiscriminately on the upper and lower boundaries, causing individuals on the boundaries to fall into a local aggregation state, resulting in a decrease in population diversity and affecting the search for the global optimal solution, which has a negative impact on optimization. Therefore, this article proposes a boundary adjustment strategy based on historical location, namely

$$V_{a,j}(i+1) = \begin{cases} (Up + V_{a,j}(i+1))/2, & V_{a,j}(i+1) > Up, \\ (Low + V_{a,j}(i+1))/2, & V_{a,j}(i+1) < Low. \end{cases} \quad (29)$$

This strategy links the improved update location with the population location generated at the beginning of the current generation, assigning individuals beyond the upper and lower

boundaries to corresponding search range areas. Compared to the original boundary method, it more effectively utilizes population information and increases population diversity.

### 3.3.4 Random centroid reverse learning strategy

Before optimization, SAO used a random initialization population method to increase the range of feasible solutions and prolong the search time. Therefore, this paper proposes a random centroid reverse learning strategy, which takes the reverse solution of the current partial solution based on the idea of reverse learning, and considers the opposite solution while considering the existing solution. By comparing with the reverse solution, selecting the better solution guides other individuals in the population to seek optimization, thereby obtaining the global optimal solution. While balancing the concepts of adversarial learning and centroid, the robustness of the original algorithm is enhanced by introducing randomness elements, in the following specific form:

Firstly, after each iteration, a random integer  $B \in [2, Num]$  is generated, which is the number of randomly selected populations. Secondly, randomly select  $B$  individuals from the current population, and then calculate the centroid of the  $B$  individuals, as follows:

$$M = \frac{\sum_{a=1}^B V_a}{B}. \quad (30)$$

Then, generate the ortho solutions of each individual in the population regarding the centroid position, as shown below:

$$V_a^* = 2 \times M - V_a. a = 1, 2, \dots, Num \quad (31)$$

Finally, the greedy rule is used to select the top  $Num$  individual with the best fitness value from the original improved population and the population  $V_a \cup V_a^*$  that has undergone random centroid reverse learning as the new generation population.

## 3.4 Specific steps for improving snow ablation optimizer

Introducing a level based selection pressure mechanism, a boundary adjustment strategy based on historical location, a random centroid reverse learning strategy, and a covariance matrix learning strategy can effectively enhance the convergence accuracy and speed of the algorithm. Algorithm 1 provides the pseudocode of MESAO as follows:

---

**Input:** dimension  $D$ , population  $V_a (a=1,2,\dots,Num)$ , population size  $Num$ , maximum number of iterations  $i_{\max}$ , snowmelt rate  $Df$ ,  $Num_a = Num_b = Num/2$

**Output:** optimal fitness value  $fit_{best}$

- 1: Randomly initialize the individual positions of the population using equation (5)
- 2: Calculate the fitness value of each individual and record the current individual's optimal  $Gol(i)$
- 3: Record four elite individuals using equation (8) and obtain  $Num_a$  and  $Num_b$  based on algorithm 1
- 4: **while** ( $i < i_{\max}$ ) **do**
- 5:     Calculate the snow ablation rate according to equation (13)
- 6:     Randomly divide the overall population  $Pop$  into  $Pop_a$  and  $Pop_b$
- 7:     **for**  $a = 1: Num$  **do**
- 8:          $Pr_{Ri} = Rank_{Ri}/(Rank_1 + Rank_2 + \dots + Rank_{Ri})$
- 9:          $El\_Rank(i) \in [Gol(i) \cdot Pr_{R1}, V_{second}(i) \cdot Pr_{R2}, V_{third}(i) \cdot Pr_{R3}, V_c(i) \cdot Pr_{R4}]$
- 10:          $V_a(i+1) = El\_Rank(i) + Bu_a(i) \otimes (\alpha_1 \times (Gol(i) - V_a(i)) + (1 - \alpha_1) \times (\bar{V}(i) - V_a(i)))$
- 11:     **end for**
- 12:     **if**  $Num_a < Num$  **do**
- 13:          $Num_a = Num_a + 1$
- 14:          $Num_b = Num_b - 1$
- 15:     **end if**
- 16:     **for**  $a = 1: Num_b$  **do**
- 17:          $\mathcal{QQ} = V \times Q$
- 18:          $V_a(i+1) = V_a(i) + \mathcal{QQ}_a(i)$
- 19:     **end for**
- 20:     **for**  $a = 1: Num$  **do**
- 21:          $V_{a,j}(i+1) = \begin{cases} (Up + V_{a,j}(i+1))/2, & V_{a,j}(i+1) > Up \\ (Low + V_{a,j}(i+1))/2, & V_{a,j}(i+1) < Low \end{cases}$
- 22:          $V_a^* = 2 \times M - V_a, a = 1, 2, \dots, Num$
- 23:         Calculate the fitness value of the reverse solution and compare it with the fitness value of the improved population. Use the greedy rule to select excellent individuals and update the optimal value
- 24:     **end for**
- 25:     Update Elite Pool
- 26:      $i = i + 1$
- 27: **end while**
- 28: Return  $Gol(i)$

---

**Algorithm 1** The proposed MESAO algorithm

**Step 1:** Set algorithm parameters. Population size  $Num$ , maximum iteration number  $i_{\max}$ , snow ablation rate  $Df$ , etc.;

**Step 2:** Initialization. Randomly generate an initialization population according to Eq. (5);

- Step 3:** Calculate the fitness value of each individual and record  $Gol(i)$ ;
- Step 4:** Population classification. Select elite individuals according to Eq. (8) and calculate the snow melting rate according to Eq. (13);
- Step 5:** Exploration phase. Update the position of the population according to Eq. (20);
- Step 6:** Development phase. Update the position of the population according to Eq. (26);
- Step 7:** Boundary processing. Perform boundary processing on the obtained new population position according to Eq. (29);
- Step 8:** Random centroid reverse learning. Select the better individual according to Eq. (31);
- Step 9:** Update the Elite Pool. Update the new elite pool and consider whether the iteration should be terminated. If it is not terminated, proceed to **Step 4**. If the iteration is terminated, output the optimal solution and the algorithm runs;

To further describe the MESAO algorithm, the specific flowchart is shown in Fig. 3.

Based on the above improvement strategies, MESAO overcomes the shortcomings of SAO. Its advantage is that it can maintain population diversity throughout the iteration process, achieve better results in balancing exploration and development, and accelerate convergence while avoiding falling into local optima. This improves the performance of the algorithm and gives it better problem-solving capabilities.

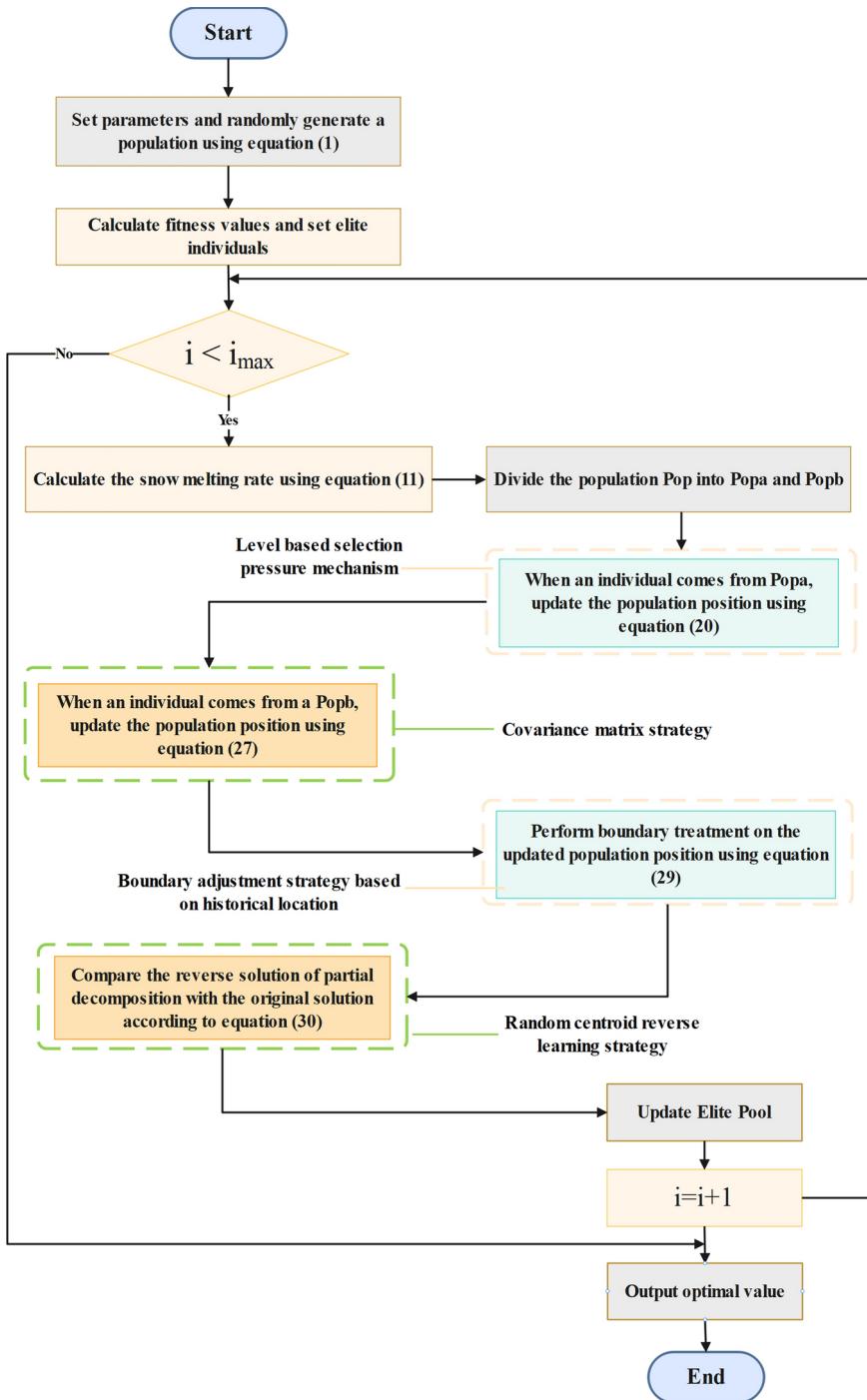
### 3.5 Experiment on the proposed algorithm

In this section, we mainly conducted two sets of comparative experiments to evaluate the proposed MESAO from multiple aspects based on experimental indicators. Firstly, the first set of experiments compares 11 newly proposed algorithms, including MESAO and the original SAO, to highlight the superiority of the proposed algorithm over the new algorithm. Secondly, in order to further confirm the performance of the proposed algorithm, a comparative experiment was conducted in the second group of experiments on 11 high-performance algorithms, including MESAO and the original SAO, which included 3 winner algorithms from CEC competitions, 2 variant algorithms from DE, and 5 variant algorithms from PSO.

Some of the charts mentioned in the following text can be found in the supplementary materials. Figures S.1–S.9, S.10–S.12, and S.13 are the convergence charts, box plots, and radar plots of MESAO and the new high-performance algorithm experiments, respectively. Figures S.14–S.22, S.23–S.25, and S.26 are the convergence graphs, box plots, and radar plots of MESAO and high-performance winner algorithm experiments, respectively. Tables S.1, S.3, and S.5 present the experimental results of MESAO and the novel high-performance algorithm. Tables S.2, S.4, and S.6 show the Wilcoxon rank sum test of MESAO and the novel high-performance algorithm. Tables S.8, S.10, and S.12 show the experimental results of MESAO and the high-performance winner algorithm. Tables S.9, S.11, and S.13 show the Wilcoxon rank sum test of MESAO and the novel high-performance algorithm. Tables S.7 and S.14 respectively calculate the running time of MESAO and the novel high-performance algorithm as well as the high-performance winner algorithm.

#### 3.5.1 Description of the test functions used

Next, we will explain the testing functions used in the two experiments. Both experiments were conducted on the CEC2017 benchmark suite, which consists of 30 test functions.

**Fig. 3** Proposed MESAO flowchart

These 30 questions consist of four types of functions: unimodal functions, simple multimodal functions, mixed functions, and combination functions. The effective search space for all questions is  $[-100, 100]$ .

Among them, F1-F3 are unimodal functions, which are used to test the basic performance of the proposed algorithm. F4-F10 are simple multimodal functions that require lower search capabilities for the proposed algorithm. F11-F20 are mixed functions used to test the performance of the proposed algorithm when facing complex problems with a large number of local minima. F21-F30 are combination functions used to test the performance of the proposed algorithm in the face of highly oscillatory features, which have the highest testing dimensions and complexity.

### 3.5.2 Experimental setup

This article tested the algorithm on two sets of experiments for each test function in the CEC2017 test suite. In order to ensure the fairness and randomness of the experiment, this article sets the overall population size of the algorithm to 30, the maximum number of iterations to 500, and each algorithm runs independently 30 times to obtain the average and standard deviation of the problem. The experiments involved were conducted on MATLAB 2023a and run on a computer with 13th Gen Intel (R) Core (TM) i5-13400F @ 2.50 GHz and 16.0 GB RAM.

In order to verify the performance and effectiveness of the proposed MESAQ algorithm, the first set of experiments in this paper selected the Coati Optimization Algorithm (COA) as the comparison object (Dehghani et al. 2023), Dung Beetle Optimization (DBO)(Xue and Shen 2023), Optical Microscope Algorithm (OMA) (Cheng and Sholeh 2023), Rime Optimization Algorithm (RIME) (Su et al. 2023), Subtraction Average Based Optimizer (SABO) (Trojovský and Dehghani 2023), Parrot Optimization (PO) (Lian et al. 2024), Crested Porcupine Optimizer (CPO) (Abdel-Basset et al. 2024), Newton Raphson based Optimizer (NRBO) (Sowmya et al. 2024), Goose Optimizer (GOOSE) (Hamad and Rashid 2024), Red billed Blue Magpie Optimizer (RBMO) (Fu et al. 2024), the original Snow Ablation Optimizer (SAO), Football Team Training Algorithm (FTTA) (Tian and Gai 2024) and Triangle Topology Aggregation Optimization (TTAO) (Zhao et al. 2024). The parameter settings of the algorithm are listed in Table 1.

## 3.6 Analysis of the impact of proposed algorithms

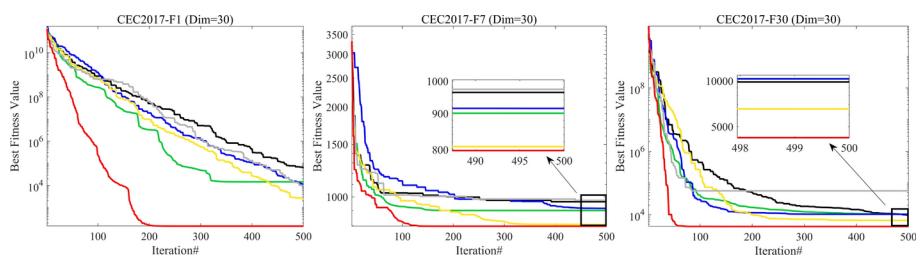
Before starting this section, we first define SAO1 as introducing a level based selection pressure mechanism separately on SAO, SAO2 as introducing a covariance matrix learning strategy separately on SAO, SAO3 as introducing a boundary adjustment strategy based on historical position separately on SAO, and SAO4 as introducing a random centroid reverse learning strategy separately on SAO. We conducted impact analysis on the proposed MESAQ, SAO, and four improved strategies (SAO1, SAO2, SAO3, and SAO4), and discussed the degree of impact of each proposed strategy on the original SAO and MESAQ. The experimental results are shown in Figs. 4, 5 and 6. According to the unimodal functions of different dimensions, it can be seen that most strategies enhance the original SAO, although the effect of covariance matrix learning strategies is not very significant. For multimodal and composite functions, the effect of the random centroid reverse learning

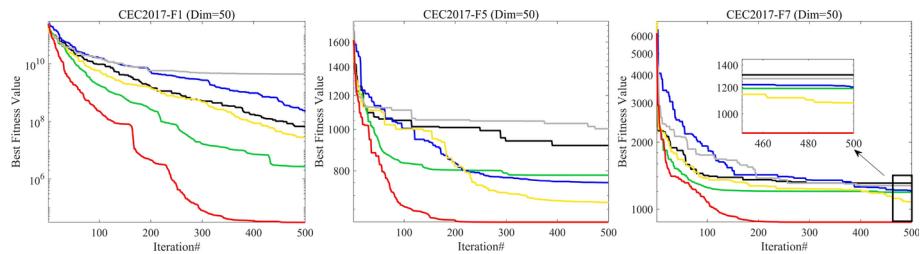
**Table 1** Parameter settings

Algorithm	Parameter	Value
COA	$r$ : random number	$r \in [0, 1]$
	I	I : randomly select 1 or 2
DBO	$\alpha$ : direction adjustment factor	$\alpha = 1$ : not deviating from the original direction $\alpha = -1$ : deviation from the original direction
	$k$ : deflection factor	$k \in (0, 0.2]$
	$b$ : constant	$b \in (0, 1)$
	$b_1, b_2$ : random vector	$b_1, b_2$ representing two independent random vectors of size $1 \times D$
OMA	$r$ : random number	$r \in [0, 1]$
	$m^r$ : random vector	$m^r \in [0, 1]$
RIME	$w$ : control the number of segments in the step function	$w = 5$
	$E$ : coefficient of adhesion	increase with the increase of iterations
SABO	$\vec{v}$ : a vector with dimension $m$	$\vec{v} \in [1, 2]$
	$ri$	random values that follow a normal distribution
PO	$r$ : random number	$r \in [0, 1]$
CPO	$\bar{r}$ : random number	$r \in [0, 1]$
	$\tau_1$	random numbers based on normal distribution
	$\tau_2$ : random number	$\tau_2 \in [0, 1]$
	$\tau_3$ : random number	$\tau_3 \in [0, 1]$
	$r_1, r_2$ : random number	$r_1, r_2 \in [1, Num]$
NRBO	$r$ : random number	$r \in (0, 1)$
	$\delta$ : balance parameters	$\delta \in [-1, 1]$
GOOSE	$pro$ : variable	$pro \in [0, 1]$
RBMO	$p$ : search for the number of randomly selected individuals	$p \in [2, 5]$
	$q$ : number of search agents	$q \in (10, Num)$
	$\alpha$ : equilibrium population coefficient	$\alpha = 0.5$
SAO	$Bu$ : brownian random number vector	$Bu \in (Num, D)$
	$T1$ : basic temperature	$T1 = 0$
MDE	$C$ : crossover probability	$C = 0.2$
	$F$ : scale factor	$F = 0.5$
BeSD	$K$	$K = 5$
	$Up$ : upper boundary	$Up = 100$
	$Low$ : lower boundary	$Low = -100$
LSHADE-SPACMA	$F$ : scale factor	$F = 0.5$
	$H$ : historical memory size	$H = 5$
	$C$ : crossover probability	$C = 0.8$
	$p$ : probability of initial control mutation strategy	$p = 0.3$
	$p$ : probability of minimum control mutation strategy	$p = 0.15$
LSHADE-cnEpSin	$\mu F, \mu CR, \mu freq$	$\mu F = \mu CR = \mu freq = 0.5$

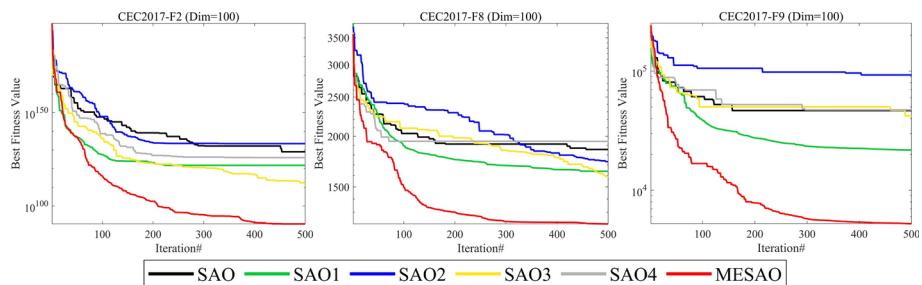
**Table 1** (continued)

Algorithm	Parameter	Value
	$r$	$r = 1.4$
	$p$	$p = 0.11$
	$H$	$H = 0.5$
	$G_{ls}$	$G_{ls} = 250$
MadDE	$q\_cr\_rate$ : optimal cross probability in personal history	$q\_cr\_rate = 0.01$
	$p\_best\_rate$ : optimal probability of personal history	$p\_best\_rate = 0.18$
	$arc\_rate$ : archiving rate	$arc\_rate = 2.3$
	$memory\_size$ : size	$memory\_size = 2.3$
	$max\_pop\_size$ : maximum population size	$max\_pop\_size = 30$
	$min\_pop\_size$ : minimum population size	$min\_pop\_size = 4$
ADFPSO	$K$ : particle novelty factor	$K = 2$
	$w$	$w = 0.9 \sim 0.4$
	$C1, C2$ : learning factor	$C1, C2$ adjustable
AGPSO	$C1, C2$ : learning factor	$C1 = C2 = 2$
	$\omega_{\max}$ : maximum weight value	$\omega_{\max} = 0.9$
	$\omega_{\min}$ : minimum weight value	$\omega_{\min} = 0.4$
EAPSO	$\eta$	$\eta = \text{sqrt}(0.001^2 D)$
	$r$	$r = 10D$
	$a$ : coefficient	$a = 0.5$
	$r_w$	$r_w = 0.995$
IPSO	$\omega_{\max}$ : maximum weight value	$\omega_{\max} = 0.9$
	$\omega_{\min}$ : minimum weight value	$\omega_{\min} = 0.4$
	$C1, C2$ : learning factor	$C1 = C2 = 2$
PPSO	$dx$	200
FTTA	$t$ : parameter	Random numbers that satisfy the t-distribution
TTAO	$r_1, r_2, r_3$ : parameter	$r_1, r_2, r_3 \in [0, 1]$ and $r_1 + r_2 + r_3 = 1$

**Fig. 4** Comparison of different strategies in CEC2017 (30 dimensions)



**Fig. 5** Comparison of different strategies in CEC2017 (50 dimensions)



**Fig. 6** Comparison of different strategies in CEC2017 (100 dimensions)

strategy on SAO is relatively small compared to other strategies. However, only MESAO, which combines four strategies, is far superior to SAO in avoiding premature convergence and falling into local optima. Therefore, we can say that the single strategy contained in MESAO is crucial for its development.

### 3.7 Experimental analysis of MESAO and new high-performance algorithms

This section summarizes the test results of COA, DBO, OMA, RIME, SABO, PO, CPO, NRBO, GOOSE, RBMO, FTTA, TTAO, SAO, and MESAO algorithms in different dimensions ( $D=30/50/100$ ) in Tables S.1, S.3, and S.5. The evaluation metrics obtained by processing the functions on the CEC2017 test suite, including optimal value, mean, standard deviation, and ranking, are included. The Wilcoxon rank sum test results at different dimensions are also provided, where the symbol "+" indicates that MESAO is superior to the comparison algorithm at the significance level, and "=" indicates that MESAO is superior to the comparison algorithm at the significance level. The symbol '-' indicates that MESAO outperforms the comparative algorithm at the significance level, and the specific results are presented in Tables S.2, S.4, and S.6. Based on the summary, we have conducted the following analysis.

For unimodal functions, MESAO is the smallest in all three dimensions of F1. Facing function F2, MESAO is only slightly inferior to RBMO at  $D=30$ .

For simple multimodal functions, MESAO has weaker optimization ability than SAO and RBMO on F3 at  $D=30$ , but ranks first on other functions. As the dimensionality increases, the advantages of MESAO become more significant. MESAO performs stably on  $D=30$  and  $D=50$ , ranking first in all functions.

For mixed functions, MESAO shows overall stability in low dimensions, consistently ranking in the top 4 among all compared algorithms, especially in F11, F12, F14, F15, F17, F18, F19, and F20. The standard SAO still has certain advantages in F13 and F16. OMA also seems to have some advantages. MESAO ranks first in F12 and F13 at  $D=50$ , and second in F11, F14, F15, F16, F17, and F18, slightly inferior to the RBMO algorithm. On  $D=100$ , MESAO ranks first in all problems except for F11, F14, and F18, with the best optimization results.

For composite functions, there are some comparative algorithms such as DBO, OMA, SABO, and SAO that have advantages on F22 in low dimensions. In addition, MESAO even shows the best performance on  $D=50$  and  $D=100$ .

Compare the statistical significance of MESAO and comparative algorithms on CEC2017 using Wilcoxon rank sum test in Tables S.2, S.4, and S.6. MESAO outperforms COA, DBO, OMA, RIME, SABO, PO, CPO, NRBO, GOOSE, RBMO, SAO, FTTA and TTAO on 30, 29, 25, 23, 29, 27, 27, 29, 28, 13, 10, 17 and 21 questions respectively at  $D=30$ . MESAO outperforms COA, DBO, OMA, RIME, SABO, PO, CPO, NRBO, GOOSE, RBMO, SAO, FTTA and TTAO on 30, 30, 28, 29, 30, 29, 21, 23, 24 and 24 questions respectively at  $D=50$ . On  $D=100$ , MESAO outperforms COA, DBO, OMA, RIME, SABO, PO, CPO, NRBO, GOOSE, RBMO, SAO, FTTA and TTAO on 29, 29, 30, 29, 30, 26, 30, 27 and 29 questions, respectively.

In order to further investigate the convergence of MESAO and comparative algorithms during the iteration process, convergence graphs of different types of functions in three dimensions are provided in Figures S.1–S.9, respectively.

For unimodal functions, MESAO has advantages in convergence accuracy and speed on F1 and F2, but its convergence accuracy on F3 is slightly inferior to other comparative algorithms.

For simple multimodal functions, it can be observed that MESAO generally has better convergence. MESAO has better convergence accuracy on F4, F5, F6, F7, F8, and F9, especially on F10 with  $D=30$ , F5 with  $D=100$ , and F9 with  $D=50$ . Only MESAO shows continuous convergence, while other algorithms fall into local optima.

For mixed functions, RBMO, OMA, SAO, DBO, and GOOSE have slightly better convergence accuracy than MESAO on functions F14 and F18. However, MESAO still has better convergence accuracy than the vast majority of algorithms on other functions, and the convergence curves of other algorithms show similar convergence behavior and results.

For composite functions, although the topology of such problems is relatively complex, with the increase of dimensions, the performance of MESAO significantly improves compared to other algorithms. The results of RIME, SAO, RBMO, DBO, and MESAO are similar on F25, F27, and F29 functions, but the relative advantages of MESAO are obvious. Therefore, it can be said that MESAO has good consistency on the vast majority of issues.

Subsequently, we present box plots of MESAO and the comparative algorithm at  $D=100$  in Figures S.10–S.12 to validate the robustness and stability of the algorithm. By observation, the data distribution of MESAO is more concentrated on the vast majority of functions, which indicates that MESAO has better robustness. This is due to the proposed strategy continuously adjusting the search state.

MESAO performs better in high dimensions than in low dimensions, so we present the radar images of MESAO and the comparison algorithm at  $D=100$  in Figure S.13. The shaded area of the radar chart is plotted based on the ranking information in Table S.5, and

the quality of each algorithm is evaluated by the size of the shadows. We can most intuitively see that the shadow area of MESAO is the smallest, while the shadow areas of COA, CPO, and SABO are all large. The ranking levels of GOOSE, OMA, RIME, and DBO vary greatly, indicating their weak stability. In the radar charts of NRBO, PO, RBMO, and SAO, it can be seen that they rank lower on certain functions. Overall, we can confirm that MESAO has better comprehensiveness.

The length of algorithm running time reflects the performance of the algorithm. Table S.7 summarizes the running time of all algorithms in three dimensions and provides the ranking of running time in different dimensions. After observation, it can be found that the running time of MESAO increases correspondingly with the increase of dimensions, which is in line with our expectations.

### 3.8 Comparison of MESAO with high performance and winner algorithms

In the previous section, we compared several new high-performance algorithms, and in this section, we further compared 11 high-performance, winner algorithms including MESAO and the original SAO. These algorithms include multi-population differential evolution algorithms (MDE) (Karkinli 2023), Bezier search differential evolution (BeSD) (Akgungor and Korkmaz 2022), Improvement of L-SHADE using semi parametric adaptive approach (LSHADE-SPACMA) (Mohamed et al. 2017), Improved LSHADE algorithm (LSHADE-cnEpSin) (Awad et al. 2017), Improving differential evolution through Bayesian hyperparameter optimization (MadDE) (Biswas et al. 2021), Adjustable drive force particle swarm optimization (ADFPSO) (Yu et al. 2022), Autonomous group particle swarm optimization (AGPSO) (Mirjalili et al. 2014b), Ensemble of surrogates assisted particle swarm optimization (EAPSO) (Li et al. 2019), Improved particle swarm optimization (IPSO) (Riazi and Saraeian 2023) and Phasor particle swarm optimization (PPSO) (Ghasemi et al. 2018), LSHADE-SPACMA, LSHADE-cnEpSin and MadDE, as the winner algorithms of the CEC competition, provide more reliable verification of the superiority and inferiority of MESAO. The experimental environment is the same as the previous experiment, and the experimental parameters are set in Table 1.

#### 3.8.1 Experimental analysis of MESAO and high performance, winner algorithm

In Figures S.14–S.22 show the convergence graphs of different types of functions in three dimensions on the CEC2017 test suite.

MESAO has better convergence accuracy and speed than other algorithms in unimodal functions F1 and F2, while F3's convergence accuracy at  $D=50$  is not as good as LSHADE-SPACMA, and its convergence accuracy at  $D=100$  is weaker than most comparison algorithms in terms of performance.

The convergence accuracy of MESAO on simple multimodal functions F5, F7, F8, and F9 is significantly better than other algorithms, while ADFPSO and BeSD have fallen into local optima early on in multiple problems. MESAO shows a slight decrease in convergence speed towards F10 at high latitudes.

The convergence degree of MESAO algorithms on mixed functions F11 and F17 is very close, but MESAO can achieve the best convergence accuracy. On F18, LSHADE-

SPACMA and LSHADE cnEpSin have better convergence accuracy and optimization ability than the proposed algorithm.

The convergence degree of MESAO algorithms on the  $D=30$  combination functions F25, F27, F28, and F29 is very close. The convergence effect of ADFPSO on F26 with  $D=50$  and  $D=100$  is better than the proposed algorithm. The convergence effect of F30 in three dimensions is significantly better than other algorithms in terms of MESAO accuracy.

The boxplots of each algorithm on CEC2017 ( $D=100$ ) are shown in Figure S.23–S.25. Based on the observation of the box plot size of each algorithm, MESAO's box shape is smaller in size among all algorithms, indicating its better robustness in the dataset. MESAO has small medians on F5, F7, F8, F9, F10, F20, F21, F22, F23, F25, and F29. MESAO exhibits a right skewed distribution on F5, F6, F9, F14, F21, F25, and F26, a left skewed distribution on F7, F10, F16, F22, F23, and F29, and a symmetrical distribution on other functions. Therefore, we can say that MESAO has good stability.

Figure S.26 shows the high-dimensional radar images of each algorithm. The MESAO radar image has the smallest shadow area, and the ADFPSO, EAPSO, LSHADE.cnEpSin, LSHADE\_SPACMA, and MDE levels vary greatly, resulting in poor stability. AGPSO, BeSD, IPSO, MadDE, MDE, SAO, and PPSO can be seen to rank lower on certain functions.

Tables S.8, S.10, and S.12 present the experimental results of MESAO with high-performance and winner algorithms on CEC2017.

For unimodal functions, MESAO can converge to the global optimal solution of the F1 function in three dimensions, and all three winner algorithms outperform the proposed algorithm in F2. As the dimension increases, the optimization ability of MESAO weakens when facing F3 problems.

For simple multimodal functions, ADFPSO and MDE outperform MESAO in F6 function among the three dimensions, and ADFPSO has better optimization ability than MESAO at  $D=50$ . In low dimensions, the original SAO appears to perform better than the proposed algorithm on the F9 function.

For mixed functions, the optimization performance of EAPSO, ADFPSO, LSHADE-SPACMA, LSHADE cnEpSin, and original SAO on F14, F16, and F18 is superior to MESAO at  $D=30$  and  $D=50$ . At  $D=100$ , the performance of the proposed algorithm will make significant progress, only slightly inferior to LSHADE-SPACMA and LSHADE cnEpSin on the F14 function, and only inferior to ADFPSO on F16 and F17.

For composite functions, LSHADE cnEpSin, MadDE, ADFPSO, SAO, and MDE do not perform as well on F22, F26, and F28 with  $D=30$ . The optimization performance of F22 and F26 at  $D=50$  is inferior to ADFPSO, while MESAO outperforms the majority of algorithms at  $D=100$ .

The results of Wilcoxon signed rank test are listed in Tables S.9, S.11, and S.13, evaluating the performance of MESAO based on different dimensions.

- (1) On  $D=30$ , MESAO outperforms MDE, BeSD, LSHADE-SPACMA, LSHADE-CneP-Sin, MadDE, ADFPSO, AGPSO, EAPSO, IPSO, PPSO, and SAO on functions 22(4), 28(1), 23(5), 28(7), 22(4), 21(6), 29(0), 23(1), 29(0), 25(3), and 15(4). Overall, MESAO outperforms all comparison objects.
- (2) On  $D=50$ , MESAO performs better than MDE, BeSD, LSHADE-SPACMA, LSHADE-CnePSin, MadDE, ADFPSO, AGPSO, EAPSO, IPSO, PPSO, and SAO on functions

- 27(1), 29(1), 25(4), 24(3), 28(2), 21(6), 30(0), 23(0), 29(0), 28(1), and 22(1). Overall, MESAO outperforms all comparison objects.
- (3) On  $D=100$ , MESAO outperforms MDE, BeSD, LSHADE-SPACMA, LSHADE-CnePSin, MadDE, ADFPSO, AGPSO, EAPSO, IPSO, PPSO, and SAO on functions 29(1), 29(1), 26(4), 26(4), 27(3), 21(8), 29(1), 28(0), 30(0), 28(1), and 30(0). Overall, MESAO outperforms all comparison objects.

The average running time statistics of the three dimensions in Table S.14 show that MESAO has the longest running time among all algorithms, but this is within our acceptable range. We allow for the drawback of longer running time for the proposed algorithm while ensuring performance.

## 4 MESAO-KELM flood prediction model

### 4.1 Optimize the kernel extreme learning machine

The quality of model prediction results in KELM depends on the optimal regularization coefficient  $C$  and kernel function parameter  $S$  settings. The regularization coefficient is used to control the complexity of the model and prevent overfitting. Appropriate kernel function parameters are beneficial for achieving optimal prediction performance on different datasets. This article will use the proposed MESAO to optimize these two parameters, and then input the optimal parameter values into KELM to obtain a kernel extreme learning machine model with the optimal parameter configuration to improve prediction accuracy. The specific steps are as follows, and the optimization flowchart is shown in Fig. 7. In addition, for KELM, radial basis function (RBF) is used as the kernel function in this article.

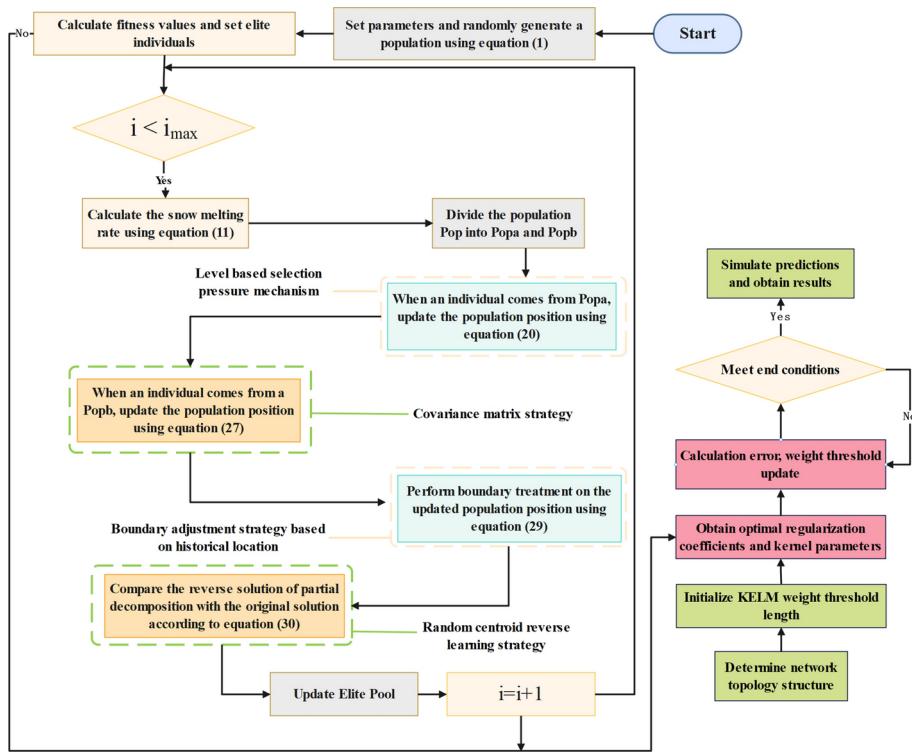
- (1) Set the network structure and initialization parameter range of KELM.
- (2) Initialize the MESAO population and calculate fitness values.
- (3) Determine whether the stopping condition is met, and if it is, pass the optimal regularization coefficient and kernel function parameters into KELM.
- (4) Use KELM with optimal parameters for regression prediction and output the prediction results.

### 4.2 Evaluation indicators for predictive models

The role of evaluation indicators for regression prediction models is to quantify the predictive performance of the model and help us choose the best model when facing practical problems. The specific evaluation indicators are given in Table 2.

### 4.3 Data collection

To predict the probability of flood occurrence, we chose the <Regression with a Flood Prediction Dataset> dataset from the Kaggle competition. Detailed data can be found in <https://www.kaggle.com/datasets/brijlaldhankour/flood-prediction-factors>. Download. We selected 3000 sets of data samples from a dataset of 50,000 sets that affect flood prediction



**Fig. 7** MESAO-KELM prediction flowchart

as the experimental data for this article. We present the prediction results for different sample ratios in Table 3. Overall, an 8:2 ratio can achieve the best fitting effect with small errors in all aspects. Therefore, in the MESAO-KELM model, we set 80% of the experimental data as training samples and 20% as testing samples. The probability of flood occurrence is predicted through simulation training of a large number of data samples influenced by various factors. Table S.15 provides the respective feature meanings of these 20 feature vectors that affect the probability of flood occurrence. MESAO-KELM, as a type of multi input single output regression prediction model, will predict the probability of flood occurrence based on 20 types of feature vectors.

#### 4.4 Comparison of multiple algorithms for optimizing KELM

All experiments were conducted on MATLAB 2023a on a 13th Gen Intel (R) Core (TM) i5-13400F computer at 2.50 GHz and 16.0 GB RAM. In the MESAO-KELM model, set the population size to 10, the number of iterations to 10, the regularization coefficient  $C$  to [1, 1], the kernel function parameter  $S$  to [50, 50], and the kernel function type to radial basis function (RBF).

Some of the charts mentioned below can be found in the supplementary materials, where Figures S.27-S.31 show the prediction results of KELM optimized by different algorithms.

**Table 2** Evaluation indicators

Index	Formula	Illustrate
Mean squared error (MSE)	$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$	Calculate the average square between the predicted value and the true value, the smaller the value, the better the accuracy
Mean absolute error (MAE)	$\frac{1}{n} \sum_{i=1}^n  \hat{y}_i - y_i $	Measure the average absolute error between the predicted value and the true value, which is non negative. The smaller the value, the better the accuracy
Root mean square error (RMSE)	$\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$	Indicate how much error will occur in the model prediction, and the smaller the value, the better the accuracy
Mean absolute percent error (MAPE)	$\frac{100\%}{n} \sum_{i=1}^n \left  \frac{\hat{y}_i - y_i}{y_i} \right $	Similar to MAE, it is standardized based on MAE, and the smaller the value, the better the accuracy
Coefficient of determination ( $R^2$ )	$R^2 = 1 - (SSE / SST)$	An indicator used to simulate the goodness of fit of a model, with a value range between 0 and 1

Predicted value:  $\hat{y} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n\}$ , actual value:  $y = \{y_1, y_2, \dots, y_n\}$ ,  $n$  is the total sample size, with a range of  $[0, +\infty)$ . SSE and SST are residual squared and total squared, respectively

Table S.15 provides an explanation of the factors influencing flood prediction. Table S.16 is an abbreviation table for various algorithm optimization KELM.

The optimal values are shown in Fig. 8, which shows the correlation between the training and testing sets of MESAO-KELM and KELM in flood prediction. On the training set result graph in Figure (a), it can be found that the data points of MESAO-KELM are basically consistent with the true values, and the fitting degree of the training set is very high. Observing the training set error plot in Fig. (b), it can be concluded that KELM has significant errors and fluctuations.

However, the error line of MESAO-KELM's training set is almost a straight line with an y-axis of 0 that is obscured by KELM's error line. On Fig. (d), it can be clearly seen that KELM has a large error, while MESAO-KELM has an error of almost 0. Table 4 presents the error evaluation metrics of MESAO-KELM on both the training and testing sets. The MAE of MESAO-KELM is lower than KELM on both sets, and the errors of other error evaluations are much smaller than KELM. Overall, we have demonstrated that MESAO-KELM indeed has higher prediction accuracy, better performance, and smaller errors than KELM.

We display the values of regularization coefficients and kernel function parameters obtained when MESAO and other intelligent algorithms optimize KELM in Table 5. In Table 6, a comparison of the fit between the training and testing sets obtained by optimizing KELM with different intelligent algorithms is presented. We can see that the fit of MESAO's training set is 0.99229, which is the same as that of SO's training set. However, the fit of MESAO's testing set reaches 0.97268, which is much higher than that of other algorithm optimization models. Therefore, it can be seen that MESAO-KELM has the best overall fit.

**Table 3** Prediction results of MESAO-KELM with different sample ratios

Data set	Index	8:2	2:8	5:5	7:3	3:7	6:4	4:6
Training set	R <sup>2</sup>	0.99229	0.98976	0.99096	0.98974	0.99063	0.99150	0.98999
	MSE	2.2983E-09	2.3223E-09	1.7942E-09	3.7152E-08	5.4022E-09	2.0510E-09	3.0762E-08
	MAE	3.0447E-05	7.6796E-05	3.2655E-05	1.4355E-04	5.9107E-05	3.5941E-05	1.1981E-04
	RMSE	4.7941E-05	5.0594E-05	4.2358E-05	1.9275E-04	7.3499E-05	4.5288E-05	1.7539E-04
	MAPE	0.006123%	0.015486%	0.006627%	0.029052%	0.011996%	0.007274%	0.023859%
	R <sup>2</sup>	0.97268	0.89063	0.92468	0.89137	0.911269	0.93957	0.93103
Test set	MSE	2.1785E-09	2.6922E-09	2.4098E-09	4.6077E-08	5.3507E-09	2.1601E-09	3.2388E-08
	MAE	2.9504E-05	9.0615E-05	3.6708E-05	1.5797E-04	5.8315E-05	3.6563E-05	1.2324E-04
	RMSE	4.6674E-05	4.9014E-05	4.9090E-05	2.1465E-04	7.3149E-05	4.6477E-05	1.8080E-04
	MAPE	0.005882%	0.018298%	0.007343%	0.031624%	0.01189%	0.007482%	0.025098%

We can say that the overfitting of MESAO-KELM is not obvious. In future experiments, we will appropriately increase the training set data to enable the model to better learn the true patterns in the data and improve its generalization ability to new data. Alternatively, we can divide the dataset into multiple subsets, select one subset for validation, and use the rest for training to more accurately estimate the performance of the model on unseen data and prevent overfitting.

In order to verify the superiority of MESAO-KELM in predicting flood accuracy, we conducted a set of experiments and selected 10 algorithms with excellent performance to optimize KELM's parameter parallel prediction. These algorithms include Whale Optimization Algorithm (WOA) (Mirjalili and Lewis 2016), Newton Raphson Optimization (NRBO), Improved Particle Swarm Optimization (IPSO), White Whale Optimization (BWO) (Zhong et al. 2022), Autonomous Group Particle Swarm Optimization (AGPSO), Snake Optimization (SO) (Hashim and Hussien 2022), Sand Cat Swarm Optimization (SCSO) (Seyyedab-basi and Kiani 2022), Runge Kutta Optimization (RUN) (Ahmadianfar et al. 2021), Phasor Particle Swarm Optimization (PPSO), and Goose Optimization Algorithm (GOOSE). Some algorithm settings are given earlier, and the settings that are not given are given in Table 7, Table S.16 provides an abbreviation table for each algorithm optimized KELM.

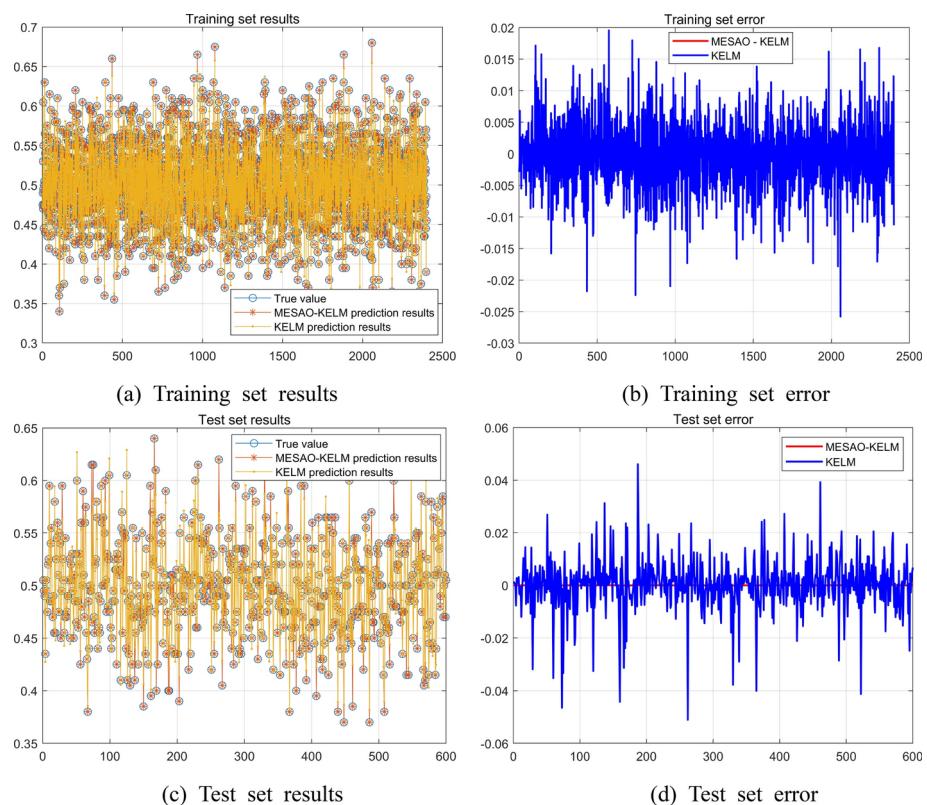
Figures 9, 10, 11, 12, 13 and 14 and Figures S.27–S.31 show the result and error graphs on the training and testing sets obtained by optimizing KELM using these 10 comparative algorithms, respectively. The fitting degree of MESAO is the best on both the training and testing sets of KELM optimized by various algorithms, and according to the error graph, it can be seen that MESAO-KELM exhibits the smallest error and the highest prediction accuracy.

Table 8 summarizes the evaluation metrics obtained by optimizing KELM using various algorithms. The minimum MAPE of MESAO-KELM on the training set is 0.006123%, which is much smaller than the errors of other algorithms. The minimum MSE, MAE, and RMSE of MESAO-KELM are 2.2983E-09, 3.0447E-05, and 4.7941E-05, respectively. On the test set, the MAPE, MSE, MAE, and RMSE of MESAO-KELM are the smallest, with values of 0.005882%, 2.1785E-09, 2.9504E-05, 4.6674E-05, respectively. Overall, compared to MESAO-KELM, MESAO-KELM has higher prediction accuracy and smaller errors in various aspects.

In the above experiment, we compared MESAO-KELM with other intelligent algorithms to optimize KELM for flood prediction problems. The results showed that the various errors of MESAO-KELM were the smallest in the vast majority of comparisons, which is closely related to the performance of MESAO itself. MESAO calculated the optimal parameter values for KELM to achieve the highest prediction accuracy. This is also related to our preparation work before the experiment, which tried various ratios of training and testing sets. Choosing an 8:2 ratio can obtain the best results. The above experimental results indicate that the model we selected is suitable, KELM is indeed applicable to this dataset, and MESAO has better optimization ability than other competitors, so the appearance of such results is in line with our expectations.

#### 4.5 Comparison between MESAO-KELM and other regression prediction models

Next, in order to verify that MESAO-KELM has better accuracy in solving flood prediction problems compared to other regression prediction models, we conducted a second set



**Fig. 8** Prediction results of MESAO-KELM and KELM

**Table 4** Evaluation indicators of MESAO-KELM and KELM

Data Set	Index	MESAO-KELM	KELM
Training set	MSE	2.2983E-09	1.9533E-05
	MAE	3.0447E-05	3.1328E-03
	RMSE	4.7941E-05	4.4196E-03
	MAPE	0.006123%	0.633350%
Test set	MSE	2.1785E-09	1.0090E-04
	MAE	2.9504E-05	6.4250E-03
	RMSE	4.6674E-05	8.0156E-02
	MAPE	0.005882%	1.275100%

**Table 5** Optimization results of KELM model parameters using various algorithms

Hyperparameter	MESAO	WOA	NRBO	IPSO	BWO	AGPSO	SO	SCSO	RUN	PPSO	GOOSE
C	11,230	50	50	50	34.6142	50	47.5355	50	50	50	39.8627
S	5945.6	1	50	50	50	1	1	1	1	50	50

**Table 6** Optimize the fitting degree of KELM model using various algorithms

R <sup>2</sup>	MESAO	WOA	NRBO	IPSO	BWO	AGPSO	SO	SCSO	RUN	PPSO	GOOSE
Train set	0.99229	0.94237	0.99218	0.99225	0.98216	0.98237	0.99229	0.93232	0.90201	0.91215	0.93225
Test set	0.97268	0.93872	0.94446	0.95329	0.92717	0.94252	0.93547	0.95653	0.90239	0.89731	0.90257

of experiments. This section compares MESAO-KELM with five other regression prediction models, namely Support Vector Machine (SVM), Particle Swarm Optimization Neural Network (PSO-BP), Random Forest (RF), Extreme Learning Machine (ELM), and Convolutional Neural Network (CNN), all of which are used for flood prediction. The specific model parameters are given in Table 9. For all regression prediction models, we use the same samples as MESAO-KELM, with 80% of the samples input into the above 5 models for training and 20% of the samples input into the trained models for prediction.

The prediction results and scatter plots of different models on the training and testing sets are shown in Figs. 15, 16, 17, 18, 19 and 20. On the training set, MESAO-KELM has the highest degree of fit between predicted values and true values. The scatter plot of MESAO-KELM shows the best fit and strong relationship between the predicted values and the true values, with no outliers observed. The predicted values of MESAO-KELM on the test set are basically consistent with the true values, and the scatter plot also fits well. From the above analysis, it can be seen that the MESAO-KELM model has better prediction performance and performance than other models.

According to the summary of the data results in Table 10, MESAO-KELM has the smallest evaluation metrics in both the training and testing sets, indicating that other regression prediction models do not have the highest prediction accuracy and the best prediction performance.

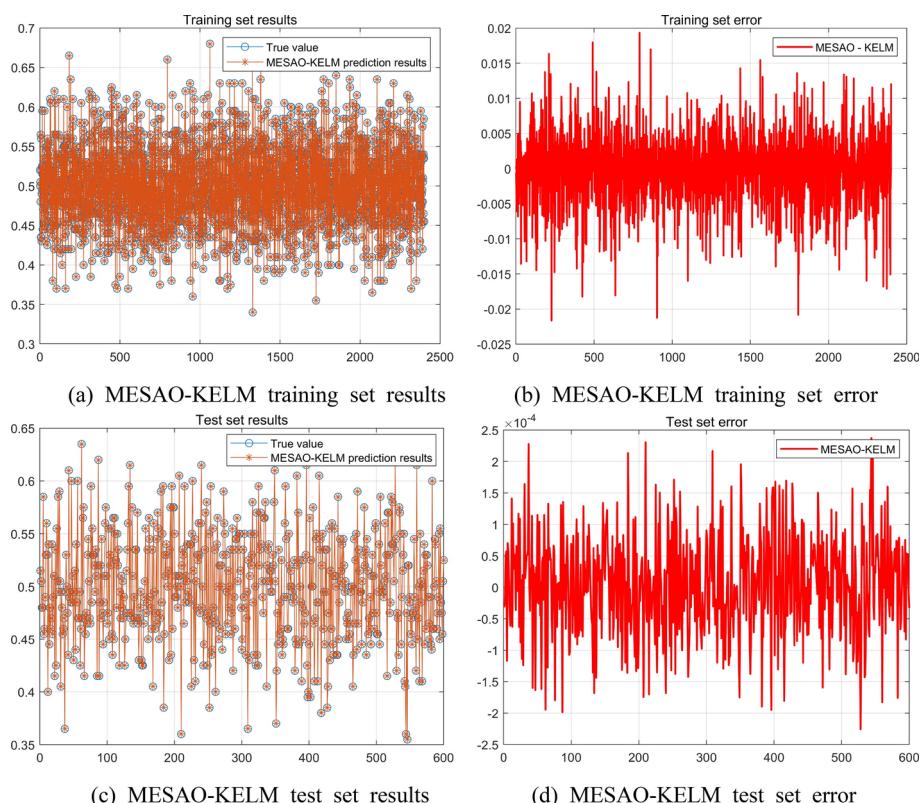
## 5 Conclusion

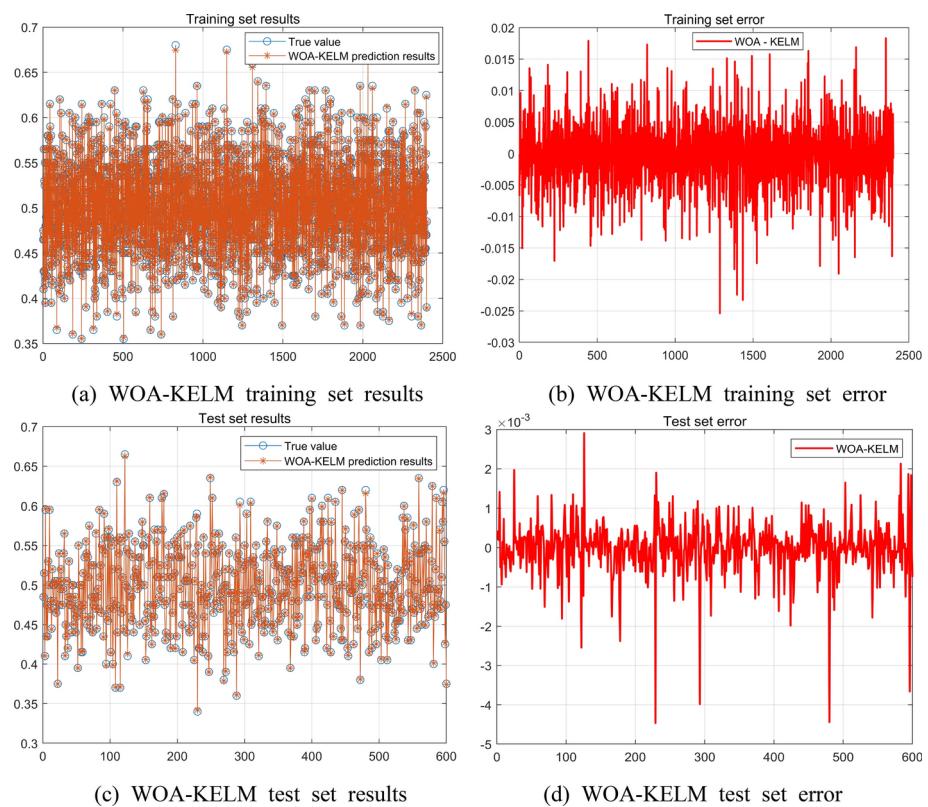
This study uses the KELM model to predict floods and constructed the MESAO-KELM flood prediction model. Numerical simulations have verified that MESAO-KELM has higher accuracy and performance in optimizing KELM and other prediction models compared to other algorithms. In addition, the MESAO algorithm proposed in this article has been validated as a competitive algorithm through comparison with classical, newly proposed, and high-performance algorithms. In future research, with the excellent performance and accuracy of MESAO-KELM, it can handle predictions for other real-world problems, such as drilling speed prediction (Su et al. 2024), remaining tool life prediction (Wu et al. 2024), industrial production prediction (Han et al. 2024), and short-term wind power prediction (Wang et al. 2024).

There is still room for improvement in MESAO-KELM, and future research plans to incorporate the concept of time series, use real-time statistical data to develop models with real-time prediction capabilities, in order to achieve research on flood prediction within ultra-short periods.

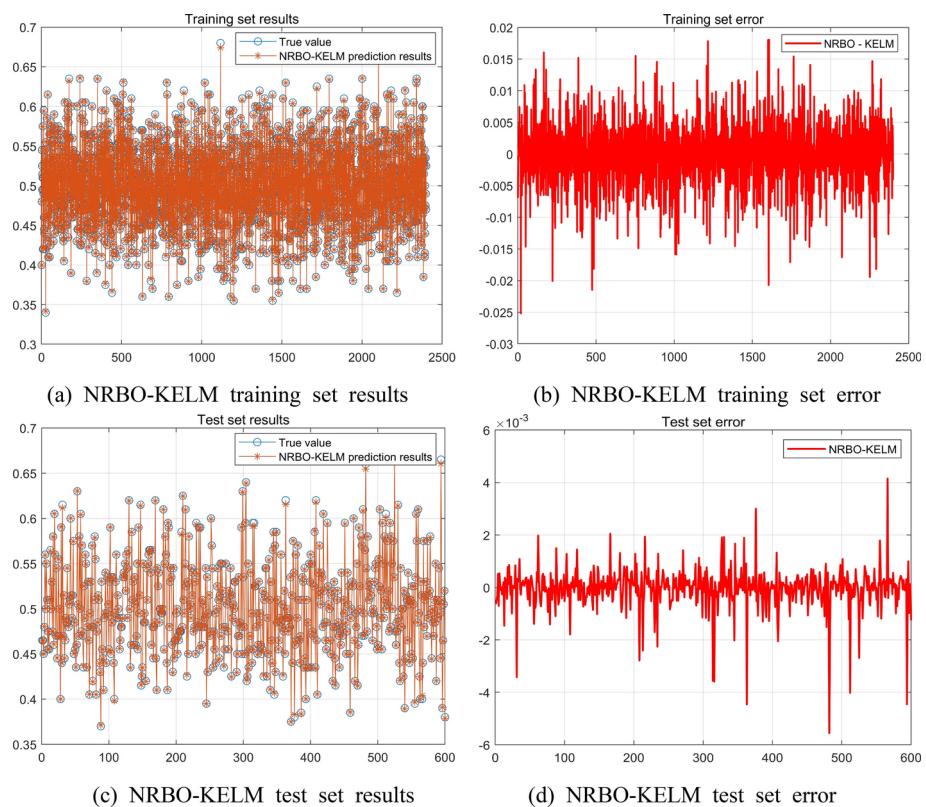
**Table 7** Parameter settings

Algorithm	Parameter	Value
WOA	$\vec{r}_1, \vec{r}_2$	Random numbers between (0,1)
	$\vec{a}_1$	Linear decrease from 2 to 0
	Constant $b$	1
BWO	$B_0$	Random numbers between [1, -1]
	$p$	Random numbers between (0, 1)
	$r_1, r_2, r_3, r_4, r_5, r_6, r_7$	Random numbers between [0, $D$ ]
	Falling probability $W_f$	Decrease from 0.1 to 0.05
SO	Constant $c_1$	0.5
	$Q$ threshold for total food quantity	0.25
	Control temperature $T$ emp threshold	0.6
SCSO	Perception $rG$	Linear decrease from 2 to 0
	Auditory Characteristics $S_M$	2
	Utilization rate $R$	Random numbers between $[-2rG, 2rG]$
RUN	Direction factor $r$	1 or -1
	$rand_1, rand_2$	Random numbers between (0,1)

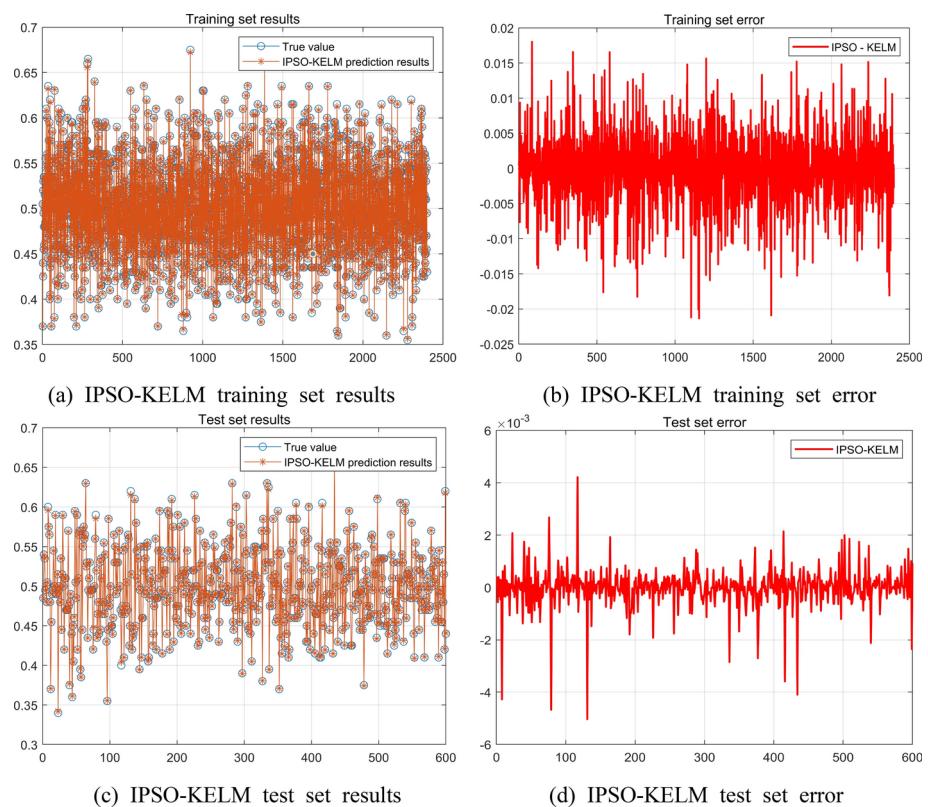
**Fig. 9** Prediction results of MESAO-KELM



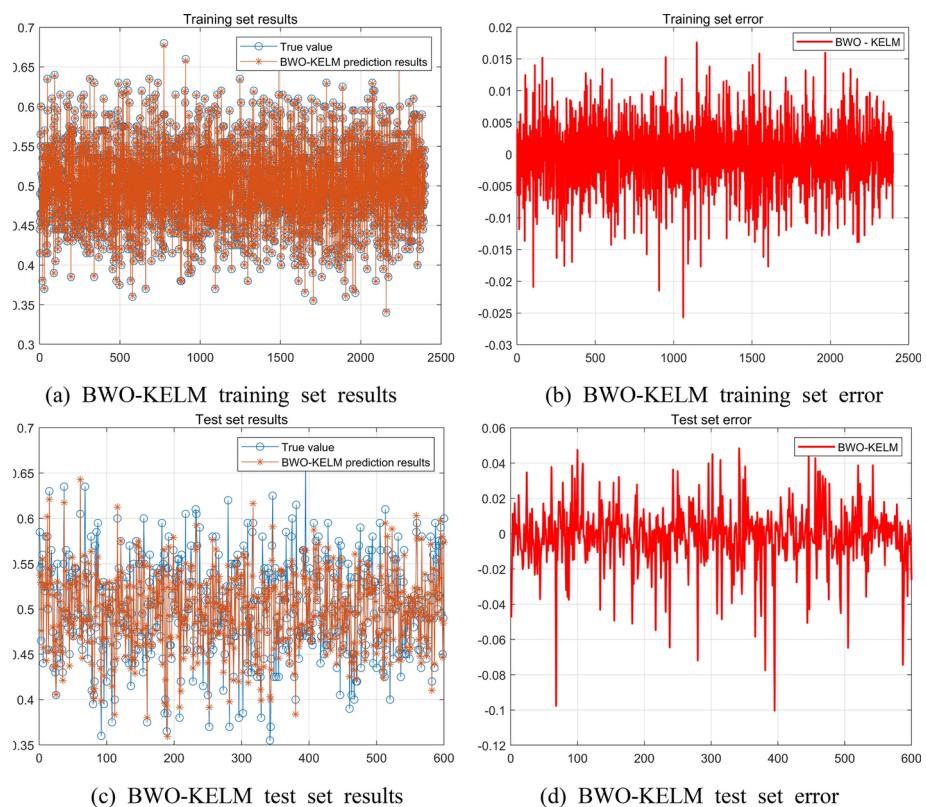
**Fig. 10** Prediction results of WOA-KELM



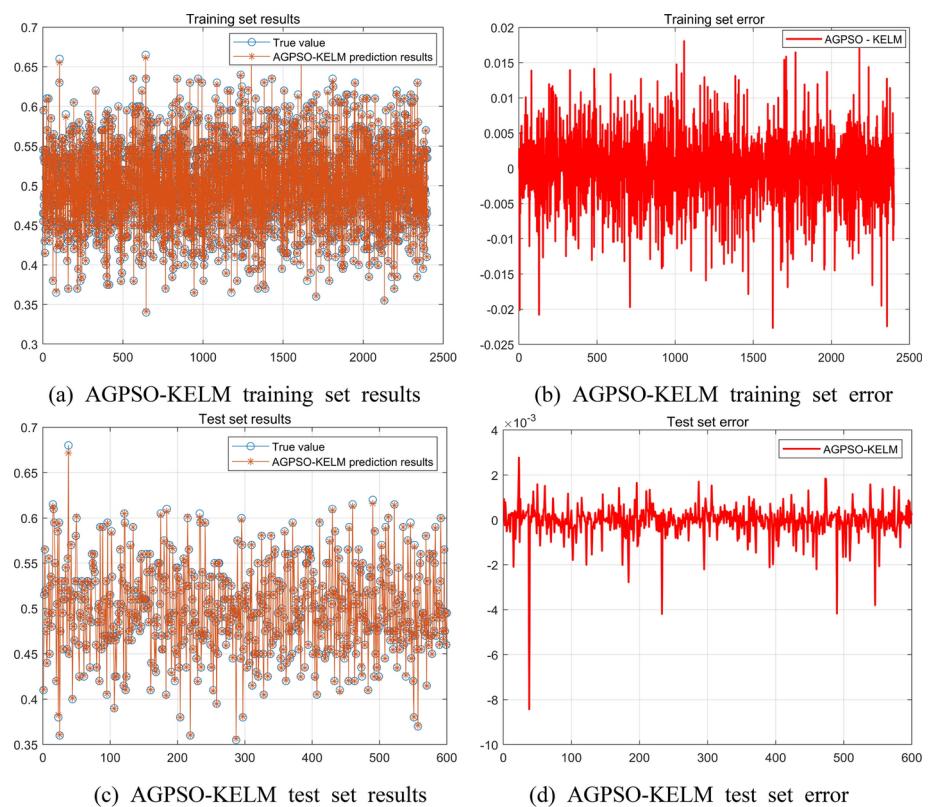
**Fig. 11** Prediction results of NRBO-KELM



**Fig. 12** Prediction results of IPSO-KELM



**Fig. 13** Prediction results of BWO-KELM



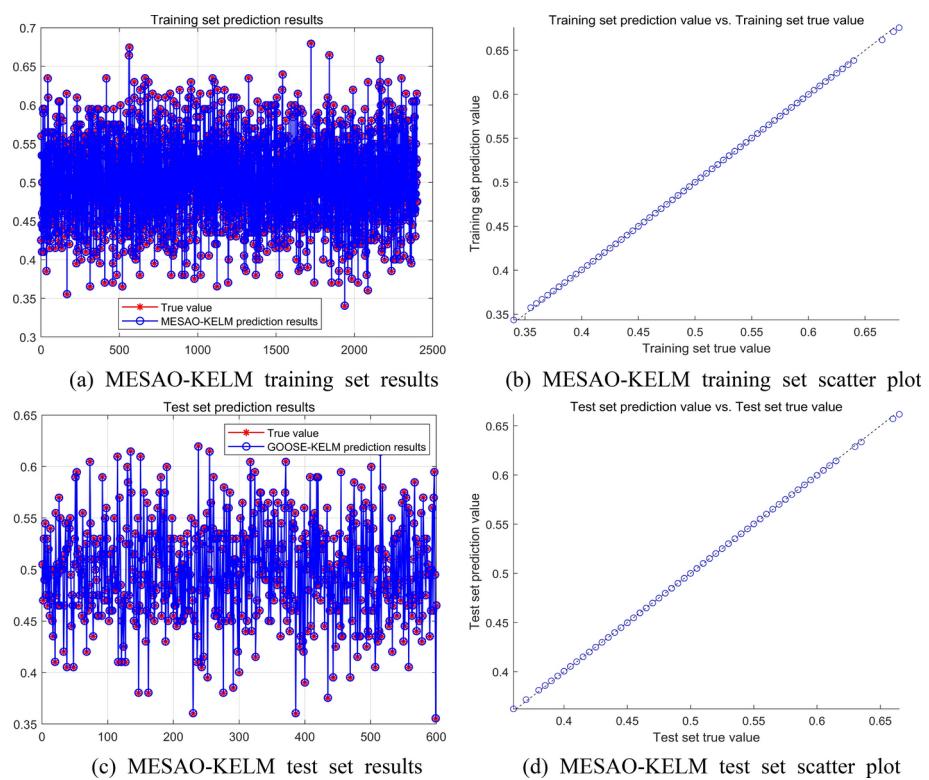
**Fig. 14** Prediction results of AGPSO-KELM

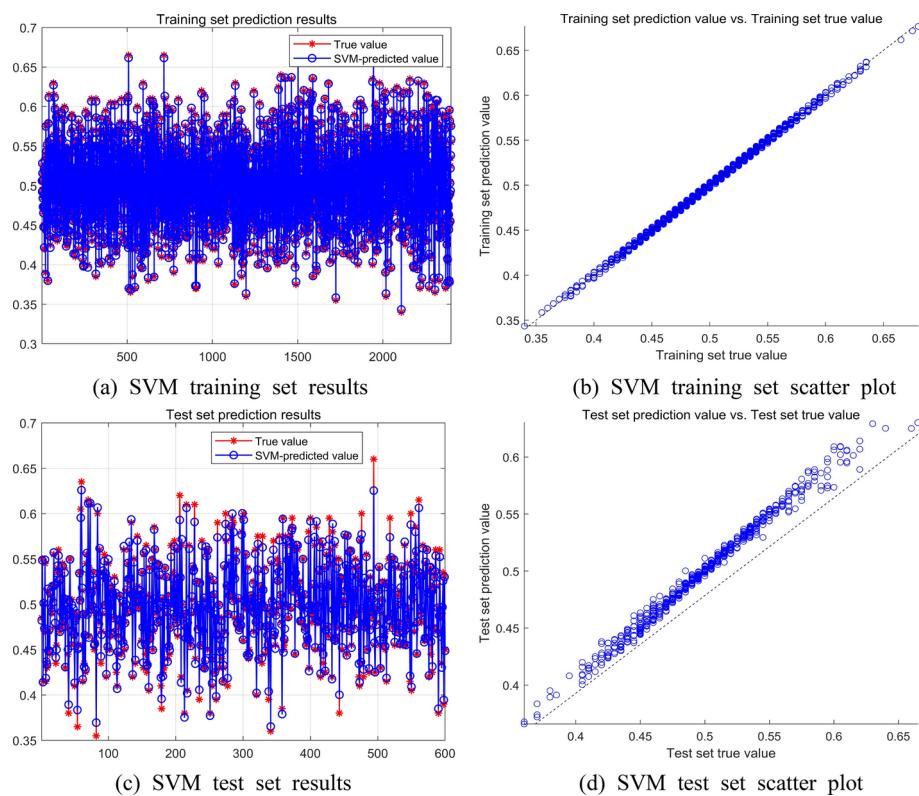
**Table 8** Results of optimizing KELM using different algorithms

Data Set	Index	MESAO-KELM	WOA-KELM	NRBO-KELM	IPOSO-KELM	BWO-KELM	AGPSO-KELM
Training set	MSE	2.2983E-09	3.9034E-07	3.6168E-07	3.6238E-07	2.6215E-07	3.7720E-07
	MAE	3.0447E-05	4.1617E-04	4.0310E-04	3.9788E-04	3.6357E-04	4.1047E-04
	RMSE	4.7941E-05	6.2477E-04	6.0140E-04	6.0198E-04	5.1201E-04	6.1417E-04
	MAPE	0.006123%	0.084600%	0.081801%	0.080526%	0.073273%	0.082943%
Test set	MSE	2.1785E-09	4.2902E-07	5.7032E-07	5.3508E-07	2.5912E-04	3.6590E-07
	MAE	2.9504E-05	4.4205E-04	4.5563E-04	4.6275E-04	1.0931E-02	3.9843E-04
	RMSE	4.6674E-05	6.5499E-04	7.5519E-04	7.3149E-04	1.6097E-02	6.0490E-04
	MAPE	0.005882%	0.089714%	0.090791%	0.092021%	2.236100%	0.080859%
Data Set	Index	SO-KELM	SCSO-KELM	RUN-KELM	PPSO-KELM	GOOSE-KELM	
	MSE	2.5265E-07	3.5680E-07	2.5297E-07	3.5566E-07	3.7454E-07	
	MAE	3.6020E-04	4.2902E-04	3.5943E-04	3.9855E-04	4.0723E-04	
	RMSE	5.0265E-04	5.9733E-04	5.0296E-04	5.9637E-04	6.1199E-04	
Test set	MAPE	0.073250%	0.086633%	0.073038%	0.080484%	0.082295%	
	MSE	2.2076E-04	2.2477E-04	3.0879E-04	5.9935E-07	5.1014E-07	
	MAE	1.0334E-02	9.9055E-03	1.1466E-02	4.4915E-04	4.4321E-04	
	RMSE	1.4858E-02	1.4992E-02	1.7573E-02	7.7418E-04	7.1424E-04	
	MAPE	2.062900%	2.017300%	2.318500%	0.089294%	0.089146%	

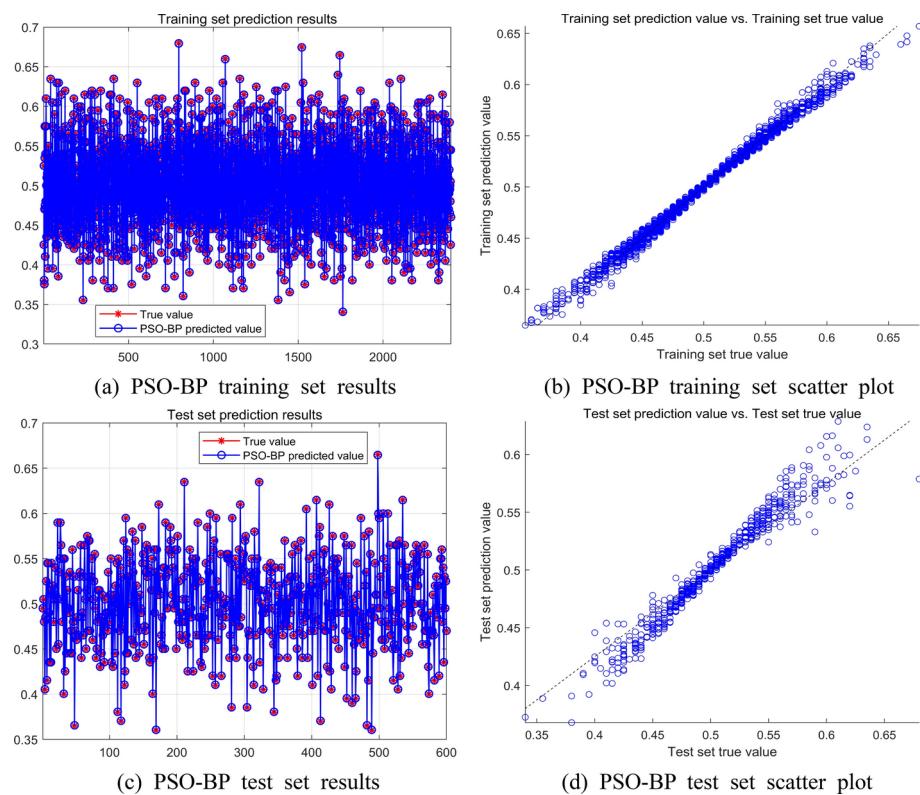
**Table 9** Parameter settings for different regression prediction models

Model	Parameter	Value
SVM	Penalty factor	4
	Radial basis function parameters	0.8
PSO-BP	Number of hidden layer nodes	5
	Training frequency	1000
	Target error	$1 \times 10^{-6}$
	Learning rate	0.01
	Learning factor $C1, C2$	$C1 = C2 = 4.494$
RF	Number of decision trees	100
	Minimum number of leaves	5
ELM	Number of hidden layer nodes	50
	Activation function	Sigmoid
CNN	Methods for adjusting learning rates	Adam gradient descent method
	Maximum number of training sessions	500
	Initial learning rate	0.001
	Regular parameter L2	0.0001
	Learning rate decline factor	0.1

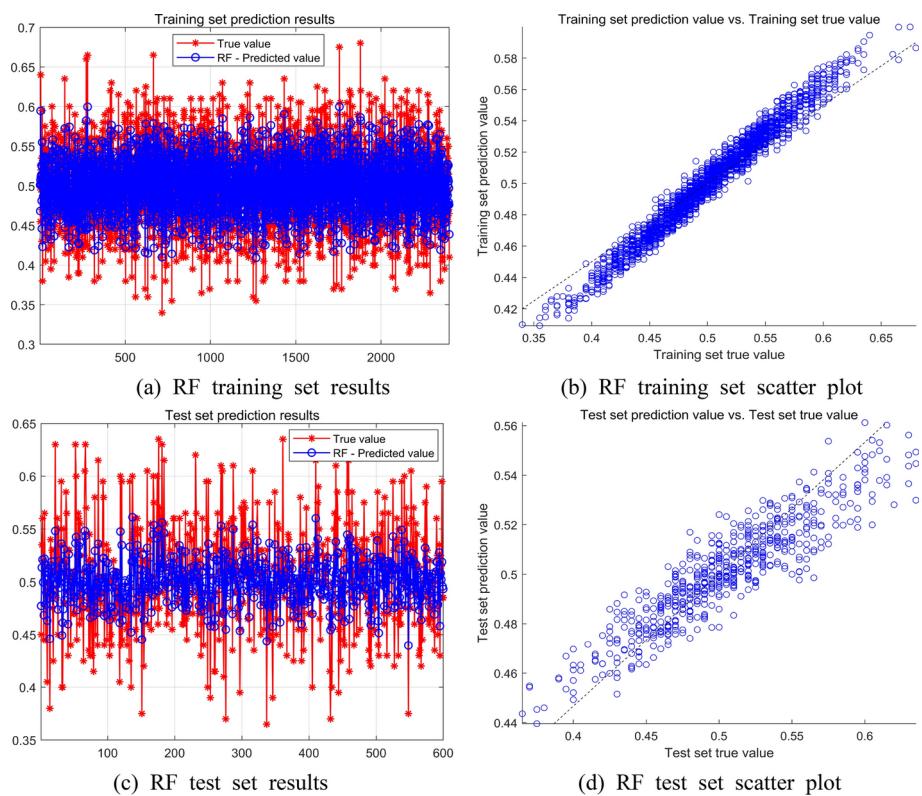
**Fig. 15** Prediction results of MESAO-KELM



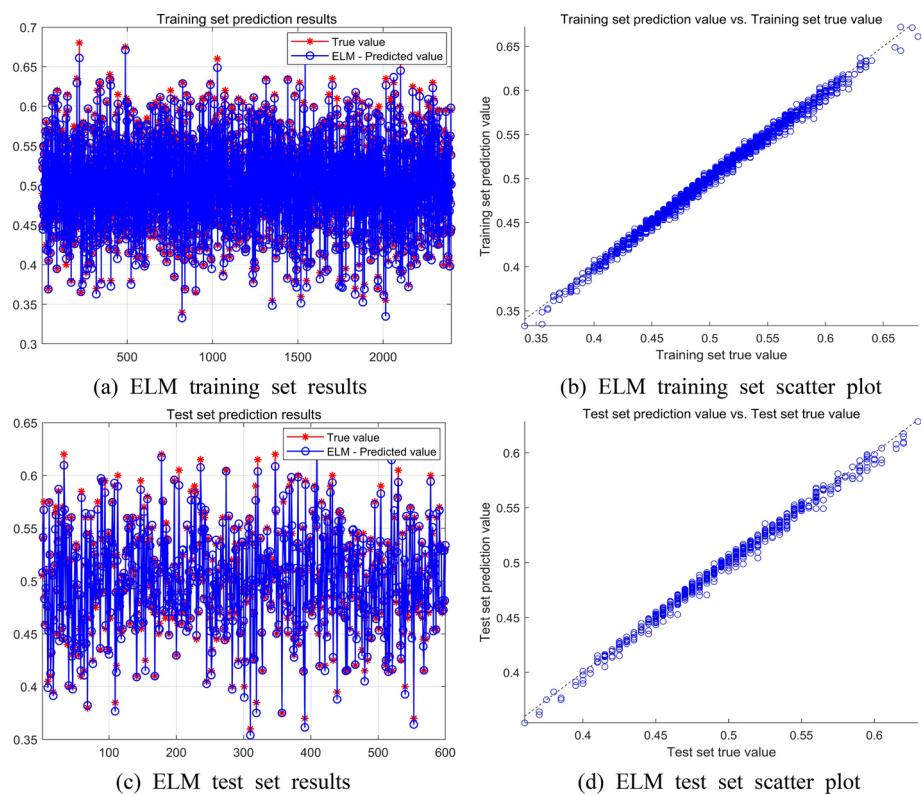
**Fig. 16** SVM prediction results



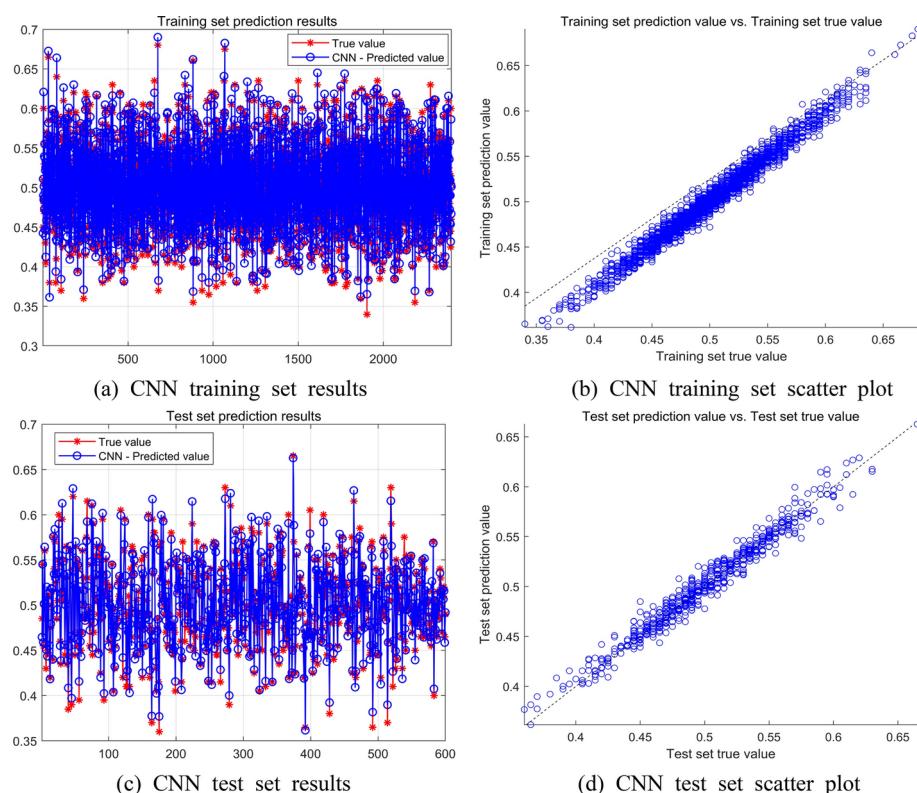
**Fig. 17** Prediction results of PSO-BP



**Fig. 18** RF prediction results



**Fig. 19** Prediction results of ELM



**Fig. 20** Prediction results of CNN

**Table 10** Comparison of MESAO-KELM and different regression prediction models

Data Set	Index	MESAO-KELM	SVM	PSO-BP	RF	ELM	CNN
Training set	MSE	2.2983E-09	5.0695E-05	5.0133E-08	3.5630E-04	2.7631E-05	6.4923E-05
	MAE	3.0447E-05	2.1002E-03	1.7119E-04	1.4638E-02	4.0302E-03	6.3823E-03
	RMSE	4.7941E-05	2.4208E-03	2.2390E-04	1.8876E-02	5.2565E-03	8.0575E-03
	MAPE	0.006123%	0.422980%	0.036196%	2.960100%	0.800860%	1.284300%
Test set	MSE	2.1785E-09	1.8045E-04	4.9779E-08	1.0111E-03	2.9791E-05	8.4994E-05
	MAE	2.9504E-05	3.1239E-03	1.7034E-04	2.4818E-02	4.1841E-03	7.2091E-03
	RMSE	4.6674E-05	4.5673E-03	2.2311E-04	3.1797E-02	5.4581E-03	9.2192E-03
	MAPE	0.005882%	0.630380%	0.036224%	5.017000%	0.837260%	1.436900%

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1007/s10462-025-11192-z>.

**Acknowledgements** This work is supported by the National Natural Science Foundation of China (Grants No. 52375264).

**Author contribution** L.C., G.H. and Y.Z. wrote the main manuscript text and L.C. prepared all Figs. 1–3. All authors reviewed the manuscript.

**Data availability** All data generated or analysed during this study are included in this published article.

## Declarations

**Conflict of Interests** The authors declare no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

## References

- Abbott M, Bathurst J, Connell P (1986a) An introduction to the European hydrologic system-system hydrologue European, SHE, a: history of a physically-based, distributed modeling system. *J Hydrol* 87:45–59
- Abbott M, Bathurst J, Cunge J (1986b) An introduction to the European Hydrologic System-Système Hydrologique Européen, SHE, 2: structure of a physically-based, distributed modeling system. *J Hydrol* 87(1–2):61–77
- Abdel-Basset M, Mohamed R, Abouhawwash M (2024) Crested porcupine optimizer: a new nature-inspired metaheuristic. *Knowl-Based Syst* 284:111257
- Ahmadianfar I, Heidari AA, Gandomi AH, Chu XF, Chen HL (2021) RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method. *Expert Syst Appl* 181(115079):0957–4174
- Akgungor AP, Korkmaz E (2022) Bezier Search Differential Evolution algorithm based estimationmodels of delay parameter k for signalized intersections. *Concurr Comput Pract Exp* 13:34
- Alrowaily MA, Alruwaili O, Alghamdi M, Alshammeri M, Alahmari M, Abbas G (2024) Application of extreme machine learning for smart agricultural robots to reduce manoeuvering adaptability errors. *Alex Eng J* 109:655–668
- Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, 2007 IEEE congress on evolutionary computation, IEEE. pp 4661–4667
- Awad NH, Ali MZ, Suganthan PN (2017) Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems, 2017 IEEE Congress on Evolutionary Computation (CEC)
- Azizi M, Aickelin U, Khorshidi AH (2023) Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization. *Sci Rep* 13:226
- Biswas S, Saha D, De S (2021) Improving differential evolution through Bayesian hyperparameter optimization, 2021 IEEE Congress on Evolutionary Computation (CEC)
- Chen X, Tianfield H, Du W (2016) Biogeography-based optimization with covariance matrix based migration. *Appl Soft Comput* 45:71–85
- Chen RZ (2024) High standard construction of digital twin Beijiang supporting basin flood control joint optimization scheduling. *Water Resour Dev Res* 1–7
- Cheng MY, Sholeh MN (2023) Optical microscope algorithm: a new metaheuristic inspired by microscope magnification for solving engineering optimization problems. *Knowl-Based Syst* 279:110939
- Cherkauer KA, Bowling LC, Lettenmaier DP (2003) Variable infiltration capacity cold land process model updates. *Global Planet Change* 38(1–2):151–159
- Dehghani M, Trojovský P (2023) Osprey optimization algorithm: a new bio-inspired metaheuristic algorithm for solving engineering optimization problems. *Front Mech Eng* 8:1126450
- Dehghani M, Montazeri Z, Trojovská E, Trojovský P (2023) Coati optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems. *Knowl-Based Syst* 259:110011

- Deng LY, Liu SY (2023) Snow ablation optimizer: a novel metaheuristic technique for numerical optimization and engineering design. *Expert Syst Appl* 225:120069
- Fu S, Li K, Huang H (2024) Red-billed blue magpie optimizer: a novel metaheuristic algorithm for 2D/3D UAV path planning and engineering design problems. *Artif Intell Rev*. <https://doi.org/10.1007/s10462-024-10716-3>
- Ghasemi M, Akbari E, Rahimnejad A (2018) Phasor particle swarm optimization: a simple and efficient variant of PSO. *Soft Computing*
- Hamad RK, Rashid TA (2024) GOOSE algorithm: a powerful optimization tool for real-world engineering challenges and beyond. *Evolv Syst* 1–26
- Han YM, Han LK, Shi XW, Li J, Huang XY, Hu X, Chu C, Geng ZQ (2024) Novel CNN-based transformer integrating Boruta algorithm for production prediction modeling and energy saving of industrial processes. *Exp Syst Appl Part A* 255:124447
- Hashim FA, Hussien AG (2022) Snake optimizer: a novel meta-heuristic optimization algorithm. *Knowl-Based Syst* 242:108320
- Hazarika BB, Gupta D (2020) Modelling and forecasting of COVID-19 spread using wavelet-coupled random vector functional link networks. *Appl Soft Comput* 96:106626
- Huang GB, Zhou HM, Ding XJ, Zhang R (2012) Extreme learning machine for regression and multiclass classification. *IEEE Transact Syst Man Cybernet Part B (Cybernetics)* 42(2):513e29
- Karkinli AE (2023) Detection of object boundary from point cloud by using multi-population based differential evolution algorithm. *Neural Comput Applic* 35:5193–5206
- Kashan AH (2014) League championship algorithm (LCA): an algorithm for global optimization inspired by sport championships. *Appl Soft Comput* 16:171–200
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks, IEEE, vol 4, pp 1942–1948
- Kramer O (2017) Genetic algorithms. In: *Genetic Algorithm Essentials*. Springer. pp 11–19
- Li YL (2000) Basic principles and applications of water tank model. *Gansu Water Conserv Hydropower Technol* 4:229–232
- Li F, Cai XW, Gao L (2019) Ensemble of surrogates assisted particle swarm optimization of medium scale expensive problems. *Appl Soft Comput* 74:291–305
- Li WZ, Liu CS, Xu YY, Niu CJ, Li RX, Li M, Hu CH, Tian L (2024a) An interpretable hybrid deep learning model for flood forecasting based on Transformer and LSTM. *J Hydrol Reg Stud* 54:101873
- Li B, Xue SK, Fu YH, Tang YD, Zhao YP (2024b) Kernel adapted extreme learning machine for cross-domain fault diagnosis of aero-engines. *Aerosp Sci Technol* 146:108970
- Lian JB, Hui GH, Ma L, Zhu T, Wu XC, Heidari AA, Chen Y, Chen HL (2024) Parrot optimizer: algorithm and applications to medical problems. *Comput Biol Med* 172:108064
- Liao YX, Wang ZL, Chen XH, Lai CG (2023) Fast simulation and prediction of urban pluvial floods using a deep convolutional neural network model. *J Hydrol* 624:129945
- Linsley RK, Crawford NH (1960) Computation of a synthetic stream-flow record on a digital computer. *Int Ass Sci Hydrol Publ* 51:526–538
- Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
- Mirjalili S, Mirjalili SM, Lewis A (2014a) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
- Mirjalili S, Lewis A, Sadiq AS (2014b) Autonomous particles groups for particle swarm optimization. *Arab J Sci Eng* 39(6):4683–4697
- Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Comput Appl* 27:495–513
- Mohamed AB, Mohamed R, Abdel Azeem SA, Jameel M, Abouhawwash M (2023) Kepler optimization algorithm: a new metaheuristic algorithm inspired by Kepler's laws of planetary motion. *Knowl Based Syst* 268:110454
- Mohamed AW, Hadi AA, Fattouh AM, Jambi KM (2017) L-SHADE with semi parameter adaptation approach for solving CEC 2017 benchmark problems, IEEE Congress on Evolutionary Computation (CEC-2017)
- Moradian S, AghaKouchak A, Gharbia S, Broderick C, Olbert AI (2024) Forecasting of compound ocean-fluvial floods using machine learning. *J Environ Manage* 364:121295
- Quan R, Liang WL, Wang JH, Li XR, Chang YF (2024) An enhanced fault diagnosis method for fuel cell systems using a kernel extreme learning machine optimized with improved sparrow search algorithm. *Int J Hydrog Energy* 50(Part A):1184–1196
- Rao RV, Savsani VJ, Vakharia DP (2012) Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Inf Sci* 183(1):1–15

- Riazi A, Saraeian S (2023) Sustainable production using a hybrid IPSO optimized SVM-based technique: fashion industry. *Sustain Comput Inform Syst* 37:100838
- Seyyedabbasi A, Kiani F (2022) Sand cat swarm optimization: a nature-inspired algorithm to solve global optimization problems. *Eng Comput* 39:1–25
- Sharma R, Goel T, Tanveer M, Dwivedi S, Murugan R (2021) FAF-DRVFL: Fuzzy activation function based deep random vector functional links network for early diagnosis of Alzheimer disease. *Appl Soft Comput* 106:107371
- Sowmya R, Premkumar M, Jangir P (2024) Newton-Raphson-based optimizer: a new population-based metaheuristic algorithm for continuous optimization problems. *Eng Appl Artif Intell* 128:107532
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* 11(4):341–359
- Su H, Zhao D, Heidari AA, Liu L, Zhang XQ, Mafarja M, Chen HL (2023) RIME: A physics-based optimization. *Neurocomputing* 532:183–214
- Su KH, Da WH, Li M, Li H, Wei J (2024) Research on a drilling rate of penetration prediction model based on the improved chaos whale optimization and back propagation algorithm. *Geoenergy Sci Eng* 240:213017
- Tian ZR, Gai M (2024) Football team training algorithm: a novel sport-inspired meta-heuristic optimization algorithm for global optimization. *Expert Syst Appl* 245:123088
- Trojovská E, Dehghani M, Trojovský P (2022) Zebra optimization algorithm: a new bio-inspired optimization algorithm for solving optimization algorithm. *IEEE Access* 10:49445–49473
- Trojovský P, Dehghani M (2023) Subtraction-average-based optimizer: a new swarm-inspired metaheuristic algorithm for solving optimization problems. *Biomimetics* 8(2):149
- Wang YG, Zhao KX, Hao Y, Yao YL (2024) Short-term wind power prediction using a novel model based on butterfly optimization algorithm-variational mode decomposition-long short-term memory. *Appl Energy* 366:123313
- Wang QM, Li C, Zhang S, Zhou C, Zhou YP (2025) Physics-informed extreme learning machine framework for solving linear elasticity mechanics problems. *Int J Solids Struct* 309:113157
- Wen K (2004) Flood management—a strategy for harmonious coexistence between humans and nature. *Water Resour Plan Design* 01:11–14
- Wu CD (2024) Catenary components state detection method based on the dimension reduction-kernel extreme learning machine. *Infrared Phys Technol* 136:105079
- Wu X, Yang X, Huang J (2024) A remaining useful life prediction algorithm incorporating real-time and integrated model for hidden actuator degradation. *ISA Trans* 151:243–257
- Xu JG, Dai JJ, Li C (2024) New situation of flood control in the Taihu Lake lake basin and countermeasures. *China Water Conserv* 1–6
- Xue JK, Shen B (2023) Dung beetle optimizer: a new meta-heuristic algorithm for global optimization. *J Supercomput* 79(7):7305–7336
- Yu F, Tong L, Xia XW (2022) Adjustable driving force based particle swarm optimization algorithm. *Inf Sci* 609:60–78
- Yuan HJ, Wang M, Li JJ, Zhang DQ, Ikram RMA, Su J, Zhou SQ, Wang YK, Zhang QF (2024) Matrix scenario-based urban flooding damage prediction via convolutional neural network. *J Environ Manage* 349:119470
- Zhang JP (2018) The harm caused by floods and waterlogging disasters. *Life Dis* 8:100123
- Zhang QS, Tsang ECC, He Q, Guo YT (2023) Ensemble of kernel extreme learning machine based elimination optimization for multi-label classification. *Knowl-Based Syst* 278:110817
- Zhao SJ, Zhang TR, Cai L, Yang RH (2024) Triangulation topology aggregation optimizer: a novel mathematics-based meta-heuristic algorithm for continuous optimization and engineering applications. *Expert Syst Appl Part B* 238:121744
- Zhong C, Li G, Meng Z (2022) Beluga whale optimization: a novel nature-inspired metaheuristic algorithm. *Knowl-Based Syst* 251:109215
- Zhou T, Wei YJ, Jie YX, Zhang YY (2024) Prediction intervals for concrete face sandy gravel dam settlement using Kalman filter-based kernel extreme learning machine. *Measurement* 236:115094

## Authors and Affiliations

**Lele Cui<sup>1</sup> · Gang Hu<sup>1</sup> · Yaolin Zhu<sup>2</sup>**

✉ Yaolin Zhu  
20070802@xpu.edu.cn

<sup>1</sup> Department of Applied Mathematics, Xi'an University of Technology, Xi'an 710054, PR China

<sup>2</sup> School of Electronics and Information, Xi'an Polytechnic University, Xi'an, China