IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 October 2022, accepted 28 October 2022, date of publication 7 November 2022, date of current version 16 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3220620

RESEARCH ARTICLE

A Dynamic Neural Network Optimization Model for Heavy Metal Content Prediction in Farmland Soil

KUN CAO[1], CONG ZHANG[2], LIANGLIANG LI[1], AND SHUAIFENG LI[1]
[1]School of Mathematics and Computer, Wuhan Polytechnic University, Wuhan 430048, China
[2]School of Electrical and Electronic Engineering, Wuhan Polytechnic University, Wuhan 430048, China

Corresponding author: Cong Zhang (hb_wh_zc@163.com)

This work was supported in part by the Natural Science Foundation of Hubei Province under Grant 2015CFA061, Grant 2018CFB408, and Grant 2020CFB761; in part by the Hubei Provincial Major Science and Technology Special Projects under Grant 2018ABA099; and in part by the National Natural Science Foundation of China under Grant 61272278.

ABSTRACT To improve the accuracy of soil heavy metal content prediction, this study proposes a dynamic neural network optimization model (DNNOM). The model is based on a radial basis function neural network (RBFNN). The weights and bias of the output layer of the RBFNN were generated using an adaptive dynamic genetic optimization algorithm (ADGOA), and the center point of the hidden layer of the RBFNN was determined using an efficient density peak clustering algorithm (EDPC). An adaptive variance measure (AVM) was then used to generate the width vector of RBFNN hidden layer. The model was applied to the predict soil heavy metal content in six new urban areas in Wuhan. Through comparison with support vector machine(SVM), light gradient boosting machine(LightGBM), RBFNN, and genetic algorithm optimizes the radial basis function neural network(GA-RBFNN), the experimental results demonstrate that the DNNOM is closer to the real value than the other four models, and the four error indicator values are also significantly lower than those of the other comparison models, which have higher prediction accuracy. Especially when compared with RBFNN, the MAPE and SMAPE of DNNOM decreased by 3.98% and 3.9%, respectively.

INDEX TERMS Dynamic neural network optimization model, soil heavy metal content prediction, radial basis function neural network, adaptive dynamic genetic optimization algorithm.

I. INTRODUCTION

Currently, soil pollution in China is not optimistic, particularly heavy metal pollution, which has become increasingly serious [1]. These heavy metals cannot be decomposed by microorganisms in the soil; therefore, they accumulate in the soil and eventually change its properties [2]. In addition, these heavy metals are absorbed by natural organisms and enter the human body through circulation, ultimately posing a significant threat to human health [3]. Investigation of heavy metal content in soils in polluted areas is an indispensable part of soil heavy metal pollution control. However, owing to the limitations of manpower, material, and financial resources, it is difficult to carry out detailed detection of heavy metal content in the soil of polluted areas. This problem can be effectively solved by using known data on soil heavy metal content to predict soil heavy metals in unknown areas.

In recent years, with the gradual deepening of research on artificial neural networks, artificial neural networks are more and more applied to soil data prediction. Cao et al. [4] proposed a parallel bird swarm algorithm (PBSA) to solve the parameter optimization problem of wavelet neural network based on wavelet neural network (WNN), so as to improve the accuracy of WNN model in predicting soil heavy metal content; Wang et al. [5] used convolutional neural network (CNN) to extract effective internal feature data from complex spectral data to predict the moisture content in soil. Li et al. [6] proposed a prediction method for the spatial variation of soil organic matter based on the RBFNN, and its prediction error was less than that of the Kriging method.

The associate editor coordinating the review of this manuscript and approving it for publication was Christian Pilato.

Hao et al. [7] proposed a convolutional neural network based on ensemble empirical mode decomposition (EEMD-CNN) for short-term soil temperature prediction, and the results showed that this model could successfully provide accurate and reliable short-term soil temperature prediction to help improve productivity. Yin et al. [8] proposed a GANN model based on a genetic algorithm and a neural network to predict the heavy metal content in soil, and the results showed that the prediction accuracy was better than that of the traditional interpolation method. Dutta et al. [9] proposed fuzzy cognitive maps (FCMs) based on recurrent neural networks (RNN) to identify and classify seven different types of soils to improve crop yield.

Compared to traditional prediction methods, artificial neural networks have a higher prediction accuracy for nonlinear problems [10], [11]. The RBFNN has been widely used in soil data prediction owing to its strong nonlinear fitting ability, ability to approximate any continuous function, and higher accuracy than traditional linear regression prediction; however, there are also some problems in the training process of the RBFNN [12], [13], [14]. The center point and width vector of the hidden layer in RBFNN are randomly generated by the K-means clustering algorithm, but the K-means clustering algorithm has high sensitivity to outliers, which easily causes the generated center point to be too far from the real center point. The weights and bias of the output layer are randomly generated, and different initial weights and biases lead to different training results, which easily causes the training results to fall into local optima.

Various solutions have been developed to address these issues. To improve the stability of the RBFNN prediction results, Cao et al. [15] proposed a deep composite model (DCM), which used the particle swarm optimization algorithm based on self-adaptive learning (SLPSO) to generate the weights and bias of the RBFNN to improve the prediction accuracy of the model for soil heavy metal content. Cheng et al. [16] introduced a genetic algorithm (GA) into RBFNN to generate the weights and bias of the output layer to prevent poor prediction performance of the model caused by poor initial weights and bias. Xiong et al. [17] used particle swarm optimization (PSO) and discrete particle swarm optimization (DPSO) to optimize the structure and parameters of an RBFNN. The experimental results show that the model has a high prediction accuracy for stock price time series. In addition, some experts and scholars have improved the generation method for the center points of the hidden layer of the RBFNN. Hu et al. [18] used a clustering method based on feature vectors in the feature space to identify the center points of the RBFNN, which can effectively improve the prediction accuracy of the RBFNN. According to the principle that the intra-class distance between samples in the same cluster should be smaller than the inter-class distance between clusters, Zhu et al. [19] adopted an improved algorithm to dynamically adjust the randomly selected initial cluster centers. The experimental results showed that the improved initial cluster centers were highly representative and that the accuracy of the K-means clustering algorithm was improved.

To solve the problems of RBFNN in prediction, researchers have proposed many solutions, but the actual prediction results are still not ideal. Therefore, this study proposes a dynamic neural network optimization model (DNNOM) to solve the problem of that low prediction accuracy of soil heavy metal content. The model adopts an adaptive dynamic genetic optimization algorithm (ADGOA) to solve the problem of randomly generating parameters of the RBFNN output layer and adopts an efficient density peak clustering algorithm (EDPC) to determine the center point of the RBFNN hidden layer, which reduces the influence of outliers. Finally, the effects of the scaling factor and data distribution were considered. An adaptive variance measure (AVM) was used to generate the width vector of the RBFNN hidden layer. Through a comparison experiment, it was proven that the model has a better prediction effect.

II. BASIC THEORY

A. RADIAL BASIS FUNCTION NEURAL NETWORK

The RBFNN is a three-layer feedforward neural network with a single hidden layer based on the function approximation proposed in the late 1980s [20],[21]. Because of its strong nonline arapproximation ability and simple structure, it is widely used in data prediction, data mining, and pattern classification. In addition to the traditional neural network processing information, the RBFNN uses a radial basis function to perform nonlinear mapping of the input data in its hidden layer and transmits the data to the next layer through linear calculations [22]. The structure of an RBFNN is shown in Fig. 1.

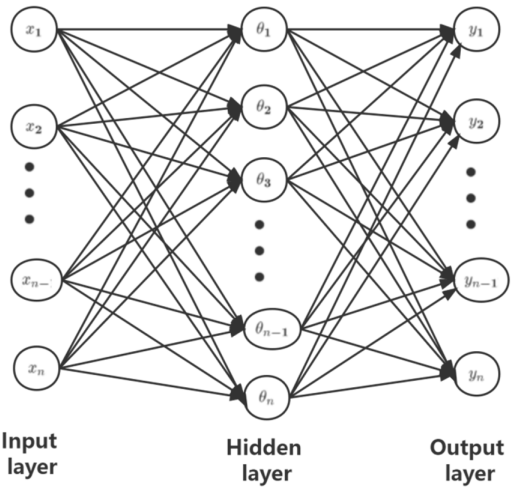

FIGURE 1. The structure of RBFNN.

The RBFNN processes input data in two parts. The two components were supervised and unsupervised. In supervised learning, the weight parameters of the hidden and output layers were determined by training the sample data set. In this

process, error functions are used to calculate the gradient value of each parameter, and gradient descent methods such as stochastic gradient descent (SGD) are used to continuously modify the parameters. Taking the weight of the output layer as an example, the updated formula is as follows:

$$w_t = w_{t-1} - \mu \frac{\partial E}{\partial w_{t-1}} \tag{1}$$

where μ denotes the learning rate, E denotes the error function, w denotes the weight of the output layer.

The center point of the RBFNN can change; therefore the density of the data sample distribution should be considered when selecting the center point. Several centers where the data sample distribution is dense can be appropriately selected. When the data samples are sparsely distributed, fewer central points can be selected. If the data sample is evenly distributed, the center point should be evenly selected. In summary, the selected center point should be representative of the entire dataset.

In unsupervised learning, self-organizing learning, such as the K-means clustering algorithm, is used to determine its position to obtain the central point of the radial basis function of the hidden layer. The width vector of the radial basis function is then obtained by processing the central point information of the radial basis function, and the calculation formula is as follows:

$$\sigma_{\mathrm{j}} = \frac{\mathrm{c_{max}}}{\sqrt{2\mathrm{h}}} \tag{2}$$

where h is the number of data center points and $\mathrm{c_{max}}$ is the maximum distance between the selected centers.

After determining the width vector of the radial basis function, relevant calculations were performed on the input data that passed through the hidden and output layers. The output of the input data sample x_i at the j node of the hidden layer is calculated as follows:

$$\emptyset\left(x_i, j\right) = exp\left(-\frac{1}{2\sigma_j^2} x_i - c_j\right) \tag{3}$$

where c_j and σ_j are the center point and width vector of the j node in the hidden layer, respectively.

The output of the input data sample x_i at the m node of the output layer was calculated as follows:

$$y_m = \varphi\left(\omega_m \emptyset\left(x_i, j\right)\right) \tag{4}$$

where φ is the activation function and ω_m is the weight of this node.

B. GENETIC ALGORITHM

Genetic Algorithm [23], [24] (GA) originated from the computer simulation of biological systems and borrowed from the theory of biological evolution and heredity to some extent. It is an imitation of natural evolution rules developed by a stochastic search algorithm, with different target problem solution as the populations of different individuals to generate the population and then to encode the population and calculate the fitness, selection, crossover, and mutation operations to find the optimal individual to complete the target problem solving [25], [26]. The main steps of the algorithm are as follows:

1. The relevant parameters of the GA were set according to the objective problem of obtaining different solutions.

2. Several candidate solutions were randomly generated as population individuals according to the candidate solutions of the target problem, and their fitness values were calculated. The individual with the highest fitness value is regarded as the historical optimal individual.

3. Encoding operation: the individuals of the population are first initialized, and then the individuals of the population are encoded in the form of array. The encoding method is typically binary or real coding, and the fitness value of each individual in the population is calculated.

4. Selection operation: the population of individuals with relatively high fitness values is selected according to the fitness value of the population, using the roulette method for the selection operation. In this method, a population of individuals was selected. Generally, when the fitness value of individuals is higher, the probability of individuals being selected is higher, and the corresponding probability of being inherited by the next generation is higher.

5. Crossover operation: gene segments are exchanged between selected population individuals according to the crossover probability, which is usually a fixed value.

6. According to the mutation probability, it is usually a fixed value.

7. Individuals that have been selected, crossed, and mutated are considered as the new population individuals and decoded. Subsequently, the fitness values of the individuals in the new population were calculated to update the historical optimal individuals.

8. Repeat steps 3-7. When the termination condition is met or the maximum number of iterations is reached, the iteration is terminated, and the optimal individual is output.

We provide a summary of GA in the following Algorithm 1.

Algorithm 1 GA

1: Initialization of population;
2: The individuals of the population are coded;
3: Calculate the fitness value of the individual;
4: while (stop condition in not reached) do
5: Selection operation;
6: Crossover operation;
7: Mutation operation;
8: Calculate the new fitness value;
9: end while

C. EFFICIENT ENSITY PEAK CLUSTERING ALGORITHM

The efficient density peak clustering algorithm (EDPC) [27], [28] is based on the density peak clustering algorithm, and a

kernel function is used to focus on the influence of neighboring data points. Only the nearest neighbor information of data points is used to estimate the density difference of data points, which simplies the calculation process and reduces the time complexity of the algorithm. Second, the local dynamic scale calculation method based on the nearest neighbor information is used to reduce the influence of different density clusters on the local data density to distinguish outliers (noise points) from normal points more easily.

1) LOCAL DENSITY DEFINITION

The data density estimation method was adopted based on the data sample points. The mean value of the influence of k nearest neighbors of data point p on its density is called the local density of data point p, and its formula is as follows:

$$\rho_p = \frac{1}{k} \sum_{x_i \epsilon KNN(p)} exp\left(-\frac{d(p, x_i)}{\delta_k(x_i)}\right) \tag{5}$$

where ρ_p is the local density of data point p; $d(p, x_i)$ is the distance between data point p and data point x_i; k is the number of neighboring points; $\delta_k(x_i)$ is the euclidean distance from data point x_i to its k-nearest neighbor, and $KNN(p)$ is the set of k nearest neighbors of a data point p.

2) DEFINITION OF k-NEAREST NEIGHBOR SET

The k-distance neighborhood of a data point p includes all data whose distance to some data points o is not greater than $\delta_k(p)$, and the formula is as follows:

$$KNN(p) = \{o \mid o \epsilon D, dist(p, o) \leq \delta_k(p)\} \tag{6}$$

where D is the data set, and the k distance $\delta_k(p)$ of data point p is the distance $dist(p, o)$ between data point p and data point o in data set D.

3) DEFINITION OF OUTLIERS

The outlier degree of data point p represents the anomaly degree of p. The larger the outlier degree, the more likely the point is to be an outlier. The formula is as follows:

$$M(p) = 1 - \rho_p \tag{7}$$

According to equations (5) and (7), the outlier degree is a positive number of no more than one, and the larger the outlier degree $M(p)$ is, the more likely the point is to be an outlier. Can better explain the degree of outliers in the data.

III. ADAPTIVE DYNAMIC GENETIC OPTIMIZATION ALGORITHM

Genetic algorithm optimizes the radial basis function neural network, which is a classical optimization method. The GA takes the parameters in the RBFNN as the population individuals and performs iterative optimization on the individuals in the population. After the optimization process, the individual with the best fitness in the population was assigned to the RBFNN as the initial value of its parameters and then trained. Thus, the problem of unstable network performance caused by the randomness of traditional parameter initialization is avoided, and the initial parameter values are more in line with the network requirements to achieve better training results. Therefore, we optimized the genetic algorithm to achieve better training results.

A. ADAPTIVE PROBABILITY ADJUSTMENT STRATEGY

In the traditional crossover and mutation operation of genetic algorithm, the crossover and mutation probabilities are generally fixed values. The fixed values determine whether the selected individuals participate in crossover and mutation operations. However, excellent individuals with relatively high fitness values in the population are also selected to participate in crossover and mutation operations, which often leads to a decrease in the fitness value of the individual after the crossover mutation operation compared with that before the crossover mutation operation. Therefore, the algorithm can easily obtain locally optimal solutions. Aiming at the above problems, this paper proposes a compound adaptive probability adjustment method (CAPAM) based on sigmoid and cosine functions to improve the crossover and mutation operators.

To maintain the excellent individuals with higher fitness as much as possible, the individuals with lower fitness can participate in crossover and mutation operations as much as possible. Crossover and mutation probabilities should be kept at a high value around the average fitness value, and slow transformation should be carried out, to allow individuals with low fitness values to have high crossover and mutation probabilities as far as possible and allow individuals with low fitness values to participate in crossover and mutation operations as far as possible. In the vicinity of the maximum fitness value, crossover and mutation probabilities should be kept at a slightly lower value than zero, so that the excellent population individuals can be preserved while avoiding falling into the local optimum.

The sigmoid function has two properties:

1. The range of the function is [0, 1], which can be used to more easily construct the probability adjustment formula for crossover and mutation.

2. It has a smoother bottom and top, and meets the requirements of crossover probability and mutation probability more than other functions.

The sigmoid function was formulated as follows:

$$\varphi(x) = \frac{1}{1 + e^{-x}} \tag{8}$$

The independent variable x in $\varphi(x)$ is replaced by the cosine part to form the composition function as follows:

$$\varphi(f) = \frac{1}{1 + exp\left(2Acos\left(\frac{f - f_{avg}}{f_{max} - f_{avg}}\pi\right)\right)} \tag{9}$$

According to the properties of the sigmoid function, when $x \geq 9.903438$, $\varphi(x)$ approaches 1; when $x \leq -9.903438$,

$\varphi(x)$ approaches 0. To ensure that the range of $\varphi(x)$ is within the interval [0,1], A = 9.903438.

Therefore, the CAPAM proposed in this study is as follows:

$$P_c = \begin{cases} P_{cmax} - \dfrac{P_{cmax} - P_{cmin}}{1 + \exp\left(2A\cos\left(\frac{f' - f_{avg}}{f_{max} - f_{avg}}\pi\right)\right)} & f' \geq f_{avg} \\ P_{cmax} & f' < f_{avg} \end{cases} \tag{10}$$

$$P_m = \begin{cases} P_{mmax} - \dfrac{P_{mmax} - P_{mmin}}{1 + exp\left(2A\cos\left(\frac{f - f_{avg}}{f_{max} - f_{avg}}\pi\right)\right)} & f \geq f_{avg} \\ P_{mmax} & f < f_{avg} \end{cases} \tag{11}$$

where P_c is the crossover probability, ranging from 0.5 to 0.9; P_{cmax} is the maximum crossover rate; P_{cmin} is the minimum crossover rate; f' is the larger fitness value of the two individuals involved in the crossover operation; f_{avg} is the average fitness value of the entire population; P_m is the mutation probability, ranging from 0.01 to 0.1; P_{mmax} is the maximum crossover rate; P_{mmin} is the minimum crossover rate; f is the fitness value of the mutants participating in the mutation operation.

As can be seen from the CAPAM function image in Fig. 2, the crossover and mutation probabilities have higher values near f_{avg}, and the crossover and mutation probabilities of individuals in $[f_{avg}, (f_{avg} + f_{max})/2]$ interval in the population are improved, the convergence speed of the entire population is accelerated, and the crossover and mutation probabilities have lower values near f_{max}, which also reduces the crossover and mutation probabilities of individuals with fitness in $[(f_{avg} + f_{max})/2, f_{max}]$ interval, so that the excellent individuals with high fitness in the population can be preserved while avoiding falling into the local optimal solution.

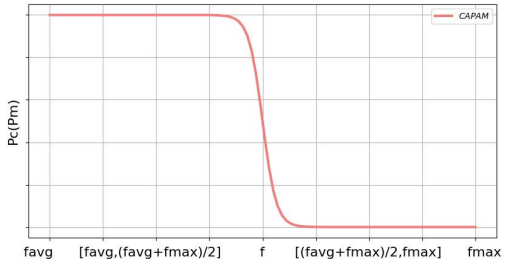

FIGURE 2. The CAPAM function image.

B. SIMULATED ANNEALING HEATING RULE

Traditional genetic algorithms tend to have the shortcoming of low local search accuracy when solving optimization problems. The introduction of a simulated annealing operation can compensate for this shortcoming of genetic algorithms. The v individual is searched at temperature T_k, where k is the number of cooling times. The cooling equation is as follows:

$$T_{k+1} = \alpha T_k \tag{12}$$

where α is the cooling coefficient,which is generally set as 0.99.

Owing to the large temperature drop at the beginning of the simulated annealing algorithm, an incomplete search may occur for some temperatures, which can easily cause the algorithm to fall into the local optimal solution. Therefore, adding a heating strategy can activate the acceptance probability of each state to adjust the current state and avoid falling into a local optimum.

We provide a summary of simulated annealing heating rule in the following Algorithm 2.

Algorithm 2 Simulated Annealing Heating Rule

1: for each individual do:
1: Select the v individual with the highest fitness value as the initial solution;
2: Let the current solution $S = v$;
3: Swapping randomly selected individuals produces a new solution $S' = v'$;
4: Calculate the increment of fitness value $dE = f\left(v'\right) - f(v)$;
5: if $dE > 0$ then
6: $\quad p = exp\left(-\frac{dE}{T_k}\right)$;
7: else
8: $\quad p = 1$;
9: end for

C. THE FRAMEWORK OF ADGOA

When ADGOA was used to generate the weights and bias of the RBFNN, they were in the form of an array of real numbers in the RBFNN, and the shapes were 8×1 and 1×1, respectively. Therefore, the individual form of the population generated by real number encoding should also be in the form of a real number array. and It is a large array combined with two small arrays with shapes of 8×1 and 1×1 respectively, and the small array is used as the basic unit for genetic operation. By encoding the weights and bias in the RBFNN, the loss of important gene segments during the evolutionary process can be avoided. We provide a summary of adaptive dynamic genetic optimization algorithm (ADGOA) in the following Algorithm 3.

Algorithm 3 ADGOA

1: Initialization of population;
2: The individuals of the population are coded;
3: Calculate the fitness values of the individual, f_{max} and f_{min};
4: while (stop condition in not reached) do
5: Selection operation;
6: use equations (10) to (11) to genarate P_c and P_m;
7: use Algorithm 2 to update p;
8: Calculate the new fitness value, f_{max} and f_{min};
9: end while

IV. ADAPTIVE RADIAL BASIS NEURAL NETWORK

A. DYNAMIC CENTER POINT SELECTION STRATEGY

The center point of the traditional RBFNN hidden layer basis function is generated by K-means clustering algorithm. K-means clustering algorithm is an unsupervised learning method. The algorithm steps are as follows:

1. In the whole data set, k sample points were randomly selected as the initial clustering centers of k subsets;
2. The euclidean distances from the remaining sample points to the k subset cluster centers were calculated, and the nearest cluster was assigned according to the minimum principle.
3. Based on the clustering results, the average value of the samples in each cluster was used as a new clustering center.
4. Repeat steps 2 and 3;
5. Finally, k cluster centers were obtained.

The selection of clustering center points of the K-means clustering algorithm mainly depends on the initial cluster center points randomly generated in the front, and the randomly selected initial cluster center points often lead to the instability of the algorithm.The K-means clustering algorithm can not ignore the sensitivity to outliers, and it is often easy to make the cluster center point and the real center point deviation. However, if the K-means clustering algorithm chooses outliers and edge points as the initial center points of the next clustering, it can easily degrade the performance and effect of the clustering, thus rendering the entire clustering algorithm unstable.

EDPC was introduced into the hidden layer of RBFNN to generate the initial cluster center points, which can effectively solved the problem of the K-means clustering algorithm randomly generating the initial cluster center points. Unlike the K-means algorithm based on segmentation, EDPC uses the closeness degree of sample data to cluster, classifies according to the closeness degree of the connection between sample points, and classifies closely connected sample points into a class until all sample points in the sample data are traversed. An EDPC based on data sample density optimization was introduced into the K-means clustering algorithm to measure the density of each sample in the data sample. The k sample points with the largest density were used to replace the k sample points randomly selected by the corresponding K-means clustering algorithm as the initial clustering center points of the k subsets. Thus, the sensitivity of the K-means algorithm to the initial clustering center point can be reduced.

EDPC adopts a local density formula to estimate the data density of data sample points to distinguish normal data points from outlier data points. Formula is as follows:

$$M(p) = \frac{1}{k} \sum_{x_i \epsilon KNN(p)} \exp\left(-\frac{d(p, x_i)}{\delta_k(x_i)}\right) \tag{13}$$

where $M(p)$ is the density of the data sample points,k is the number of neighboring points, $d(p, x_i)$ is the euclidean distance from the data sample point p to x_i, $\delta_k(x_i)$ is the euclidean distance from the data sample point x_i to its k neighboring points, and $KNN(p)$ is the k neighbor set of the data sample point p, that is, the outlier degree of the data sample point. For points in the same cluster, assuming that the k neighbors of p belong to the same class, $\delta_k(x_i) > d(p, x_i)$ and the value of $M(p)$ will be relatively large. In other words, in the k neighbor set of p, the closer the sample data points are to the center, the greater is the density of the sample data points.

B. ADAPTIVE VARIANCE MEASURE

The clustering center point $u = \{u_1, u_2, \ldots, u_m\}$, and the corresponding scaling factor is determined according to the data density distribution of each cluster center point; thus the width of each cluster center point can conform to the spatial distribution of its data samples. Because of the different distribution spaces of the cluster sample data, the distance or the same width of two adjacent cluster center points cannot reflect the density of cluster center points. Therefore, the selection of the width should comprehensively consider the density of the cluster data samples and the distance between the centers of each cluster.

The average distance between the center points of each cluster was taken as the base distance:

$$meanD(u_i) = \frac{\sum_{i \neq j} dist(u_i, u_j)}{k - 1} \tag{14}$$

where $meanD(u_i)$ is the average distance between the center points of each cluster, k is the number of neighboring points, $dist(u_i, u_j)$ are european distances.

The variance represents the density of the sample data distribution. Each cluster was regarded as a data set and the variance of each cluster was obtained as follows:

$$S_i = \frac{1}{size(C_i)} \sum_{x \in C_i} dist(x, u_i)^2 \tag{15}$$

where S_i is the variance of each cluster, $size(C_i)$ is the number of samples belonging to cluster center point u_i, $dist(x, u_i)$ is the european distance, and C_i is the subsample.

The scaling factor of the center point width is:

$$\varepsilon_i = S_i \Big/ \left(\frac{1}{k} \sum_{i=1}^{k} S_i\right) \tag{16}$$

Therefore, we can obtain the width of each center point according to equations (14), (15) and (16):

$$\sigma_i = \varepsilon_i meanD(u_i) \tag{17}$$

When the cluster data samples are densely distributed, the cluster variance S_i is relatively small, and the corresponding scaling factor ε_i was also smaller, thereby reducing the width of the center point. When the cluster data samples are sparsely distributed, the clustering variance S_i is relatively large, and the corresponding scaling factor ε_i is also large; therefore, the width of the center point also increases appropriately.

V. EXPERIMENTS AND ANALYSIS

A. DYNAMIC NEURAL NETWORK OPTIMIZATION MODEL

The DNNOM firstly uses an adaptive dynamic genetic optimization algorithm (ADGOA) to generate the weights and bias of RBFNN output layer. Subsequently, an efficient density peak clustering algorithm (EDPC) was used to determine the center point of the hidden layer of the RBFNN. Finally, the adaptive variance metric (AVM) was used to generate the width vector of the hidden layer of the RBFNN considering the influence of the scaling factor and data distribution. Through a comparison experiment, it was proven that the model had a better prediction effect.

The steps of DNNOM are briefly described in Algorithm 4, and the DNNOM framework is shown in Fig. 3.

Algorithm 4 DNNOM

1: Set the structure of RBFNN;
2: Use EDPC to find the center point of RBFNN hidden layer;
3: Use AVM to calculate the width vector of RBFNN hidden layer;
4: Generate the weight and bias of RBFNN output layer by ADGOA;
5: while (stop condition in not reached) do
6: Calculate the output of RBFNN;
7: Calculate gradient value;
9: end while

B. VALIDATION OF ADAPTIVE DYNAMIC GENETIC OPTIMIZATION ALGORITHM

In this experiment, we compared ADGOA, particle swarm optimization (PSO), bird swarm algorithm(BSA), genetic algorithm(GA) and new adaptive genetic algorithm (NAGA) [29]. The GA, PSO, and BSA are swarm intelligence optimization algorithms that attempt to simulate the adaptability of individual populations on the basis of natural characteristics. They all adopt certain transformation rules to solve the problem through the search space. PSO and BSA are classical optimization algorithms that are widely used in various fields and have a certain representativeness. NAGA is a classical improved algorithm in genetic algorithm. By comparing the above four algorithms, the effectiveness of ADGOA can be better verified. Therefore, we choose 12 groups of test functions for the performance test experiment. The first six are unimodal functions and the last six are multimodal functions [30]. The global optimum is 0.

1) Schwefels P2.22 (domain: [−10,10])

$$f_1(x) = \sum_{j=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i| \tag{18}$$

2) Step (domain: [−100,100])

$$f_2(x) = \sum_{i=1}^{n} (x_i + 0.5)^2 \tag{19}$$

3) Zaharov (domain: [−5,10])

$$f_3(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5 i x_i\right)^2 + \left(\sum_{i=1}^{n} 0.5 i x_i\right)^4 \tag{20}$$

4) Rosenbrock's (domain: [−10,10])

$$f_4(x) = \sum_{i=1}^{n-1} \left\{ 100 \left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2 \right\} \tag{21}$$

5) Sum of Different Powers (domain: [−1,1])

$$f_5(x) = \sum_{i=1}^{n} |x_i|^{i+1} \tag{22}$$

6) Sphere (domain: [−100,100])

$$f_6(x) = \sum_{i=1}^{n} x_i^2 \tag{23}$$

7) Ackley (domain: [−32,32])

$$f_7(x) = 20 + e - 20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos\left(2\pi x_i\right)\right) \tag{24}$$

8) Rastrigin (domain: [−5.12,5.12])

$$f_8(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right] \tag{25}$$

9) Schwefels (domain: [−500,500])

$$f_9(x) = 418.9829n - \sum_{j=1}^{n} x_i \sin\left(\sqrt{|x_i|}\right) \tag{26}$$

10) Dixon-Price (domain: [−10,10])

$$f_{10}(x) = (x_1 - 1)^2 + \sum_{i=2}^{n} i\left(2x_i^2 - x_{i-1}\right)^2 \tag{27}$$

11) Griewank (domain: [−600,600])

$$f_{11}(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{28}$$

12) Levy (domain: [−10,10])

$$\begin{aligned} f_{12}(x) = {} & sin^2(\pi\omega_1) \\ & + \sum_{i=1}^{n-1} (\omega_1 - 1)^2 \left[1 + 10sin^2(\pi\omega_i + 1)\right] \\ & + (\omega_n - 1)^2 \left[1 + sin^2(2\pi\omega_n)\right] \\ & Where\ \omega_i = 1 + (x_i - 1)/4, \\ & for\ all\ i = 1, \cdots, n \end{aligned} \tag{29}$$

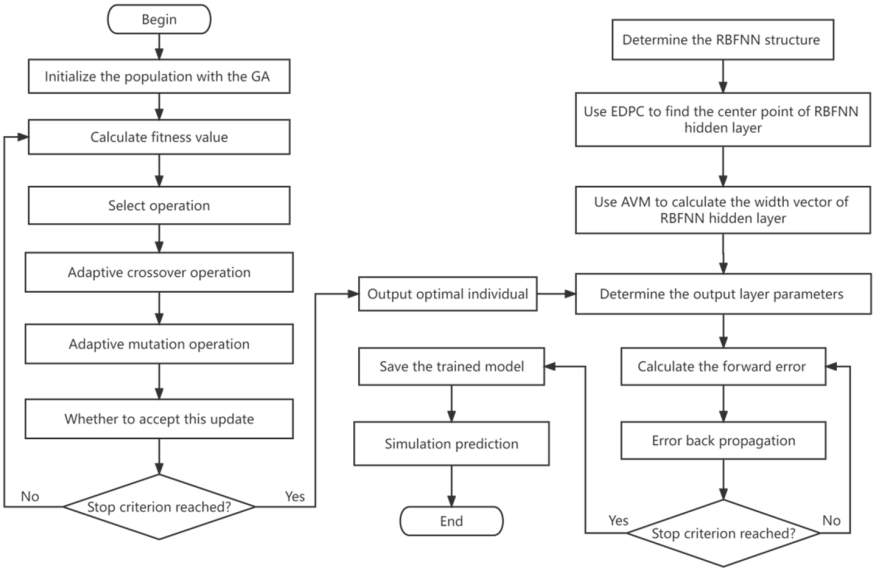

FIGURE 3. The framework of DNNOM.

The parameter settings for the experiments are listed in Table 1. To ensure the same experimental conditions, the 12 groups of test functions under each algorithm were iterated 100 times. The changes in the optimal fitness values of the five algorithms during the iteration process of the first six unimodal test functions are shown in Fig. 4, and the fitness mean values and standard deviations after the iteration are listed in Table 2. The change in the optimal fitness value during the iteration process of the last six multimodal test functions is shown in Fig. 5, and the fitness mean value and standard deviation after iteration are listed in Table 3.

TABLE 1. Parameter setting.

Algorithm	Parameters
PSO	$w = 0.9, c_1 = c_2 = 2$
GA	$P_c = 0.85, P_m = 0.075$
BSA	$FQ = 3, FL \in [0.5,0.9]$, $P \in [0.8,1]$, $C = S = 1.5, a_1 = a_2 = 1$
NAGA	$P_c = 0.85, P_m = 0.075$
ADGOA	$P_c = 0.85, P_m = 0.075$

As can be seen from Table 2 and 3, the standard deviation of BSA is lower than those of the other four algorithms for f_4 and f_7. Under other test functions, the average and standard deviation of NAGA and ADGOA were lower than those of the other three algorithms, indicating that the convergence accuracy of NAGA and ADGOA was better than that of the other three algorithms. Under f_1, f_2 and f_8, the standard deviation of NAGA is lower than that of ADGOA. Under f_5 and f_{11}, the mean value of NAGA was lower than that of ADGOA. Except for these two cases, the standard deviation and mean value of the ADGOA were lower than those of the NAGA for the other test functions. The results show that the convergence accuracy of ADGOA is better than that of the other four algorithms.

As shown in Fig. 4 and 5, the optimal adaptive value curve representing NAGA and ADGOA is always lower than that represented by PSO, GA, and BSA, regardless of the test function used. Moreover, the curves represented by NAGA and ADGOA always reached their lowest points before the other three. This indicates that the convergence speeds of the NAGA and ADGOA are better than those of the PSO, GA, and BSA algorithms. Under f_1, f_2, f_4 and f_{10},the NAGA and ADGOA curves are very close to each other, most of them coincide with each other, and there is no obvious difference. Under the other test functions, the curve representing ADGOA was consistently lower than that representing NAGA was. This shows that the convergence speed of ADGOA is better than that of NAGA under these test functions. Therefore, the convergence speed of the ADGOA was better than those of the other four algorithms.

C. SOIL HEAVY METAL CONTENT PREDICTION

The RBFNN has powerful nonlinear input and output mapping capabilities. When making a prediction, the relationship between the data can be mapped effectively so that the known data information can be used to predict the heavy metal contents in the soil. The higher the correlation between input variables and output variables of the neural network, the better the prediction effect of the neural network. Therefore, when using RBFNN to predict soil heavy metal content, it is necessary to collect external factors related to soil heavy metals, such as the location of sampling points and the heavy metal content in the soil.

Heavy metal content data of farmland soil in six new urban areas of Wuhan,Hubei Province,China, were selected as the experimental data. The data were obtained by the Wuhan

TABLE 2. Result comparison under $f_1 f_2 f_3 f_4 f_5$ and f_6.

Function	Algorithm	Mean value	Standard deviation	Function	Algorithm	Mean value	Standard deviation
f_1	PSO	3.52E+02	4.36E+03	f_4	PSO	3.68E+01	3.15E+00
	GA	5.56E+02	6.58E+02		GA	3.11E+03	3.55E+03
	BSA	3.23E+03	4.29E+02		BSA	2.98E+02	**2.48E+02**
	NAGA	3.21E+01	**3.05E-01**		NAGA	2.17E+03	3.28E+04
	ADGOA	**2.98E+00**	3.05E+02		ADGOA	**2.15E+04**	2.56E+00
f_2	PSO	4.28E+04	5.08E+03	f_5	PSO	5.88E+01	5.03E+04
	GA	6.56E+03	5.25E+02		GA	5.62E+00	4.89E+03
	BSA	3.89E+02	3.78E+02		BSA	5.03E+01	5.56E+00
	NAGA	3.14E+01	**3.05E-05**		NAGA	**4.56E-04**	4.05E+04
	ADGOA	**2.89E+04**	2.58E+03		ADGOA	3.54E+04	**3.78E+03**
f_3	PSO	3.58E+01	3.54E+03	f_6	PSO	4.89E+00	4.51E+02
	GA	4.31E+01	3.48E+03		GA	4.45E+02	4.56E+03
	BSA	4.35E+02	3.15E+00		BSA	4.02E+03	3.78E+00
	NAGA	3.55E+03	3.17E+01		NAGA	3.99E+05	3.56E+01
	ADGOA	**2.89E+02**	**3.12E+01**		ADGOA	**3.45E+01**	**3.12E+04**

TABLE 3. Result comparison under $f_7 f_8 f_9 f_{10} f_{11}$ and f_{12}.

Function	Algorithm	Mean value	Standard deviation	Function	Algorithm	Mean value	Standard deviation
f_7	PSO	6.32E+02	5.16E+01	f_{10}	PSO	5.68E+01	6.85E+00
	GA	4.53E+03	7.68E+03		GA	7.25E+05	5.55E+01
	BSA	3.16E+01	**4.18E+02**		BSA	5.02E+02	4.28E+02
	NAGA	3.01E+01	5.35E+04		NAGA	3.87E+03	3.28E+04
	ADGOA	**2.98E+00**	5.05E+02		ADGOA	**3.55E+04**	**2.56E+00**
f_8	PSO	3.18E+04	6.15E+02	f_{11}	PSO	6.18E+01	7.33E+04
	GA	5.65E+02	4.45E+02		GA	5.23E+05	5.88E+03
	BSA	4.89E+02	3.89E+02		BSA	5.03E+01	6.23E+00
	NAGA	3.14E+01	**3.05E-05**		NAGA	**5.36E-04**	4.05E+04
	ADGOA	**2.89E+04**	2.58E+03		ADGOA	4.78E+03	**4.18E+03**
f_9	PSO	5.58E+01	6.45E+03	f_{12}	PSO	5.58E+05	6.35E+02
	GA	5.71E+01	5.38E+03		GA	5.35E+02	6.76E+03
	BSA	5.25E+02	4.25E+00		BSA	5.02E+03	5.68E+00
	NAGA	4.15E+03	4.05E+01		NAGA	3.79E+05	4.78E+01
	ADGOA	**3.78E+02**	**3.25E+01**		ADGOA	**3.25E+01**	**4.15E+04**

TABLE 4. Contents of heavy metals in farmalnd soils in the suburbs of Wuhan.

Heavy metal	Min(ug/g)	Max(ug/g)	Mean(ug/g)	Standard deviation (ug/g)
Ni	3.32	77.67	28.22	12.04
Cd	0.01	4.94	0.21	0.39
Hg	0.01	2.37	0.14	0.17
Cu	2.16	159.36	26.21	14.06
Zn	15.16	293.73	71.17	29.29
Pb	1.96	83.30	19.46	8.60
Cr	11.13	171.21	57.47	24.66
As	0.24	82.07	10.15	6.00

TABLE 5. The pearson coefficient calculation result.

Heavy metal	Pearson coefficient
Ni	0.665215
Cd	0.323458
Hg	0.041006
Cu	0.147539
Zn	0.422606
Pb	0.635627
Cr	0.234578

Academy of Agricultural Sciences in strict accordance with the "Soil Environment Monitoring Technical Specifications," according to the land area and crop distribution in each region, combined with soil type and GPS positioning sampling. The dataset contained the longitude, latitude, altitude, functional area and eight different soil heavy metal contents. The functional area represents the specific crop type, such as rice or wheat. The eight heavy metals in the soil were As, Cd, Cr, Cu, Ni, Pb, Zn and Hg. Table 4 carries out statistical

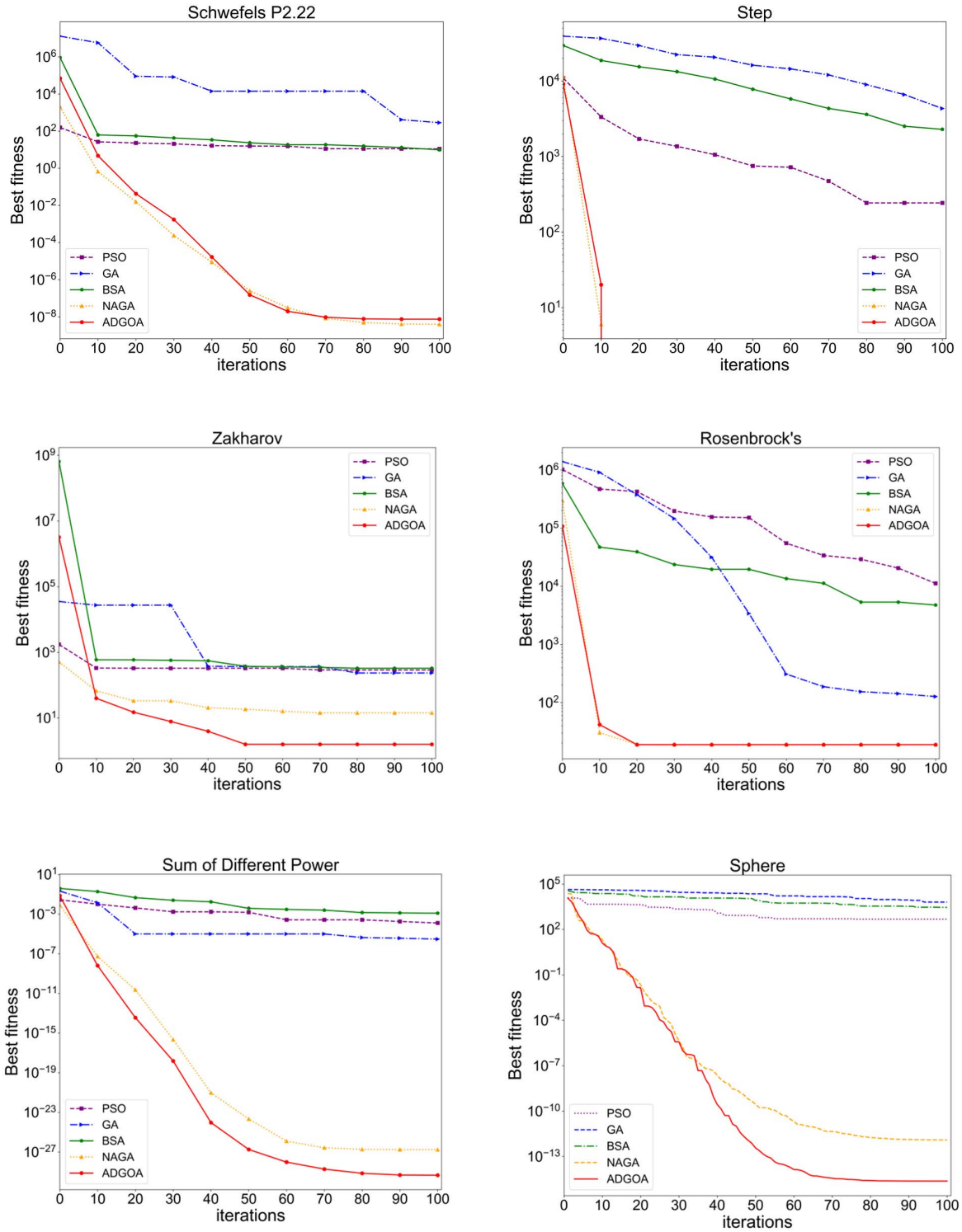

FIGURE 4. The change of the best fitness value under $f_1 f_2 f_3 f_4 f_5$ and f_6.

analysis of the heavy metal content in the data set. Because of the spatial differentiation of the enrichment degree of heavy metals in the soil within the region and the difference in the content of heavy metals in the soil in different functional regions, the location of the sampling points and the functional regions in which they are located are correlated with the

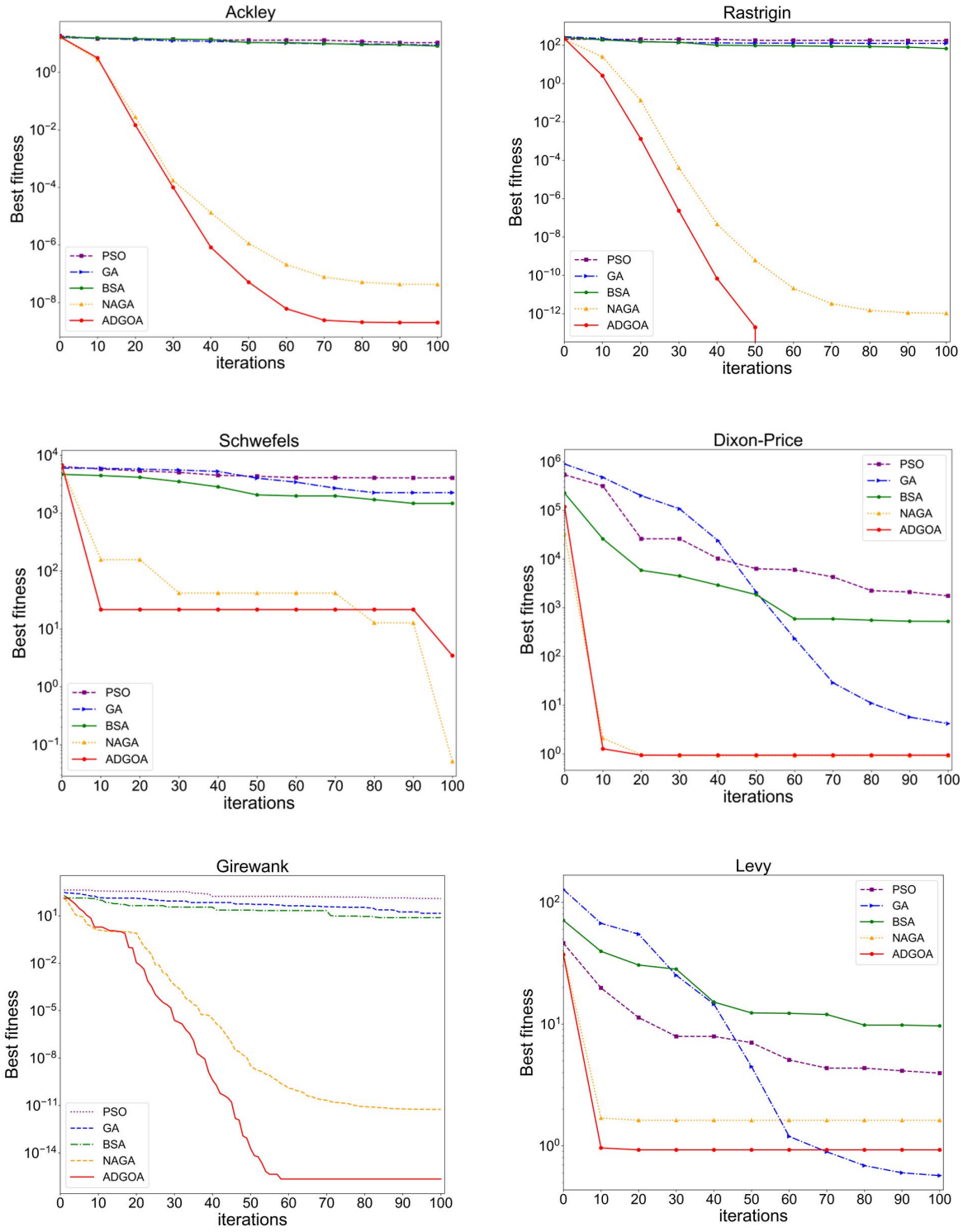

FIGURE 5. The change of the best fitness value under $f_7 f_8 f_9 f_{10} f_{11}$ and f_{12}.

content of heavy metal in the soil to a certain extent. In addition to the location of the sampling points, there was also a correlation between the contents of different heavy metals. Therefore, to obtain a better prediction effect, different heavy metal contents and location information were selected as the input variables of the model. In this data prediction experiment, heavy metal As was selected as the prediction data, the correlation between other indicators and heavy metal As was compared, and the Pearson coefficient was used as the measurement index. The calculation results are listed

TABLE 6. Loss statistics of different nodes.

Number of nodes	RBFNN	GA-RBFNN	DNNOM
2	6.3440	9.3523	6.4526
3	6.3853	9.7009	6.2813
4	6.8087	7.0773	7.1254
5	6.4756	7.3600	6.8524
6	5.7177	7.1211	5.8975
7	**5.2183**	5.5311	5.7856
8	6.4851	**5.4097**	**5.1345**
9	5.5315	7.4312	5.9856
10	6.5798	7.3711	6.1348
11	7.1578	6.2433	6.3596
12	6.5321	5.9441	6.5823

in Table 5. In the calculation results of the Pearson coefficient, the top four heavy metals were Cd, Ni, Pb and Zn. Therefore, the longitude, dimension, altitude, functional area, and contents of Cd, Ni, Pb and Zn were taken as the input of the model to predict the content of As. After sorting the soil heavy metal information, 500 sets of data were selected as training data, and 25 sets of data were used as testing data. Eight indices including longitude, dimension, altitude, functional area, and the content of four heavy metals (Cd, Ni, Pb and Zn) were used as input variables, and the content of heavy metals As was used as the output variable.

First, the data were normalized using the following equation:

$$PN = \frac{P - P_{min}}{P_{max} - P_{min}} \tag{30}$$

where PN is the normalized data, P is the original data, P_{max} and P_{min} are the maximum and minimum values of the original data respectively.

All experiments in this study were performed on a Windows 10 computer with an Intel Core I5 processor, PyCharm 2020 Professional programming software, and the Python programming language.

In terms of parameter setting, the number of nodes each layer of the various neural network models should be determined first. According to the experimental data, the input feature dimension is four, and the output feature dimension is one; thus, the input layer node of the neural network model can be determined as four, and the output layer node is one.

To set the hidden layer nodes of the RBFNN, GA-RBFNN and DNNOM models, the number of hidden the layer nodes are determined using the following formula:

$$h = \sqrt{m+n} + a \tag{31}$$

$$h = log_2 n \tag{32}$$

where h is the number of nodes in the hidden layer, m is the number of nodes in the output layer, n is the number of nodes in the input layer, and a is a constant between zero and ten. The number of hidden layer nodes was calculated from 2 to 12. The following equation was used to compare the loss value of 100 iterations for different hidden layer nodes, and the number of hidden layer nodes was determined based on the loss value. Formula is as follows:

$$loss = \sum (y - \hat{y})^2 \tag{33}$$

where $loss$ is the loss value after 100 iterations, y is the true value, and $\hat{y}$ is the predicted value.

Table 6 shows that the RBFNN model had the minimum loss value when the number of hidden layer nodes was seven. The GA-RBFNN and DNNOM models exhibited the lowest loss value when the number of hidden laminations was eight. In addition, the training times of the RBFNN, GA-RBFNN and DNNOM were all set to 500, and the learning rate was set to 0.01, which is common.

TABLE 7. Parameter setting.

Model	Parameters
RBFNN	learning rate=0.01,ite=500
GA-RBFNN	learning rate=0.01,ite=500, $P_c = 0.85, P_m = 0.075$
LightGBM	learningrate=0.01,ite=500
SVM	c=1,gamma=0.1
DNNOM	learning rate=0.01,ite=500 $P_c = 0.85, P_m = 0.075$

To verify the effectiveness of DNNOM, a comparative experiment was conducted on the prediction of soil heavy metal As content with the model constructed by RBFNN, and the genetic algorithm optimizes the radial basis function neural network(GA-RBFNN). Classical models commonly used for data prediction, such as support vector machine (SVM) and light gradient boosting machine(LightGBM), have been used to predict and compare the content of heavy metal As in soil. The parameter settings for the models are shown in Table 7.

By observing and comparing the RBFNN and GA-RBFNN in Fig. 6, it was found that the trends of the predicted values of RBFNN and GA-RBFNN were mostly close to or even coincide; however, at some points, the predicted values of

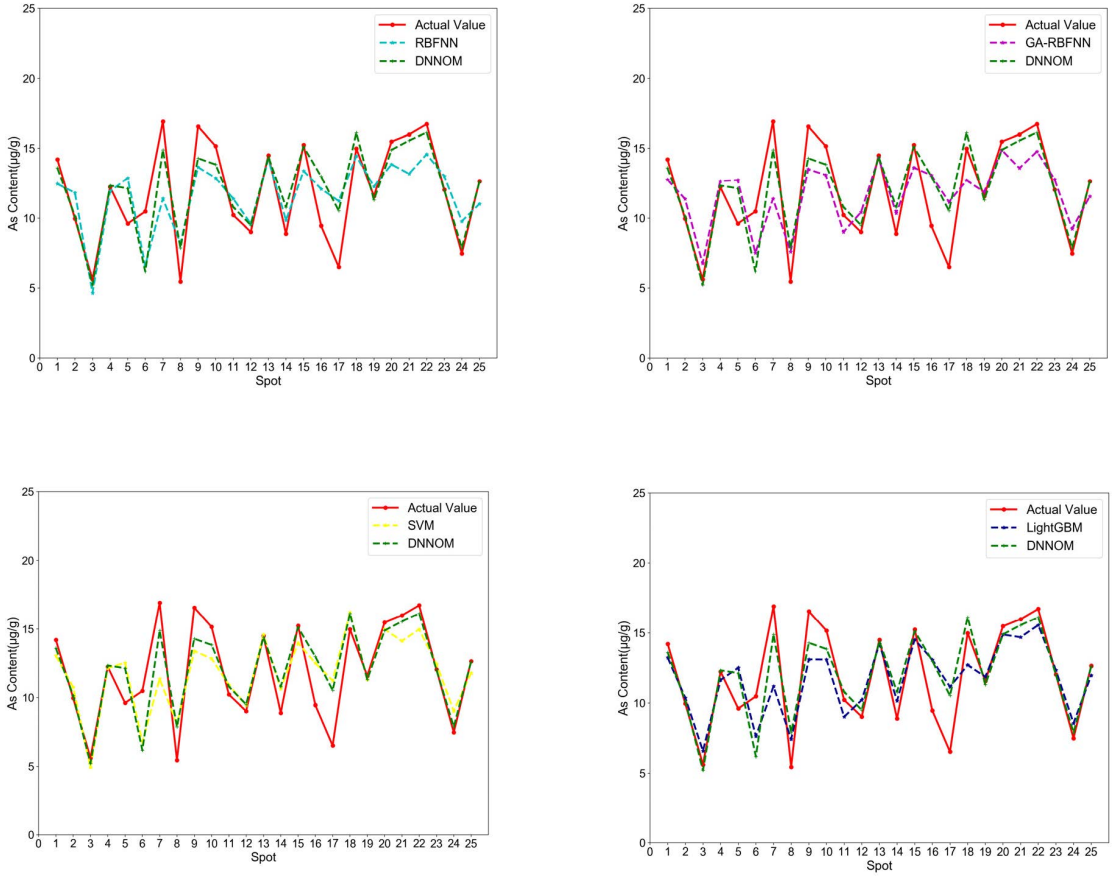

FIGURE 6. The prediction results of RBFNN, GA-RBFNN, SVM, LightGBM and DNNOM.

the GA-RBFNN were closer to the real values than those of RBFNN. This indicates that the prediction accuracy of the GA-RBFNN is higher than that of the RBFNN. By observing and comparing SVM, LightGBM and DNNOM, it was found that the predicted values of the three models were closer to the real values than those of RBFNN and GA-RBFNN, indicating that the prediction accuracies of the three models were higher than those of the RBFNN and GA-RBFNN. At the same time, it can be observed that the predicted value of DNNOM is closer to the real value than that of SVM and LightGBM at some points, indicating that the predicted value of DNNOM is closer to the real value than that of SVM and LightGBM at some points, indicating that the predicted value of DNNOM is closer to the real value than that of SVM and LgihtGBM.

To clarify the difference between the predicted and real values of the five models, we first calculated the difference between the predicted and real values of the five models, calculated the proportion of the travel value to the real value, and finally counted the points between the different proportions. The calculation formula is as follows:

$$p_i = \frac{|y_i - \hat{y}_i|}{y_i} \tag{34}$$

where y_i is the true value, $\hat{y}_i$ is the predicted value, and p_i is the proportion of difference between the predicted value. The statistical results are shown in Fig. 7.

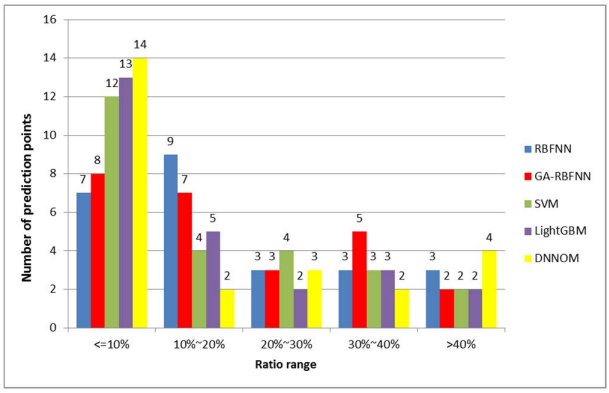

FIGURE 7. The distribution of the number of prediction points under different ratio range.

In Fig. 7, the predicted values of SVM, LightGBM, and DNNOM were more concentrated in the interval of less than 10%, whereas the predicted values of RBFNN and GA-RBFNN were more concentrated in the interval of 10% −20%, indicating that SVM, LightGBM, and DNNOM were closer to the real values than RBFNN and GA-RBFNN. In the interval of less than 10%, DNNOM had more predicted points than the SVM and LightGBM. Although SVM and LightGBM have more points in the interval between 10%

TABLE 8. The calculated value of error indicators.

Models	MAE(ug/g)	MSE(ug/g)	MAPE(%)	SMAPE(%)	R^2
RBFNN	2.011	6.125	19.70%	4.75%	0.459
GA-RBFNN	1.856	5.689	19.08%	4.50%	0.498
SVM	1.745	5.218	17.45%	4.35%	0.580
LightGBM	1.613	4.287	16.35%	3.98%	0.645
DNNOM	1.513	3.786	15.82%	3.73%	0.726

and 20%, the predicted value of DNNOM is closer to the real value than those of SVM and LightGBM. The results showed that when DNNOM was used to predict the heavy metal As content, the predicted value was closer to the real value than the other four models.

In addition to the above comparison between the predicted and real values, and the statistical distribution of the proportion of the difference between the predicted and real values in the real value, the prediction effect of the DNNOM model was evaluated using five indices: mean absolute error (MAE), mean square error (MSE), mean absolute percentage error (MAPE), symmetric mean absolute percentage error (SMAPE) and coefficient of determination (R^2). The MAE, MSE, MAPE, and SMAPE are commonly used in regression models to quantify the error between predicted and actual values. The smaller the value, the smaller the error, and the higher the prediction accuracy of the model. R^2 is used to reflect the proportion of the total variation of the dependent variable that can be explained by the independent variable through the regression relationship, that is, the degree of fit between the predicted value and the actual values. Its value is between 0 and 1. The larger the value, the higher is the degree of explanation of the independent variable for the dependent variable, the greater is the degree of fitting, and the better is the model performance. The formula used is as follows:

$$MAE = \frac{1}{n}\left(\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|\right) \tag{35}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{i}\left(y_i - \hat{y}_i\right)^2 \tag{36}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{37}$$

$$SMAPE = \frac{100\%}{n}\sum_{i=1}^{n}\frac{\left|y_i - \hat{y}_i\right|}{\frac{(|y_i| + |\hat{y}_i|)}{2}} \tag{38}$$

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

where y_i is the true value, $\hat{y}_i$ is the predicted value, and $\bar{y}$ is the average value. The calculation results of the five models for different error calculation formulas are listed in Table 8.

As shown in Table 8, comparing RBFNN,GA-RBFNN and DNNOM, we can see that the four error indices of DNNOM are significantly lower than those of RBFNN and GA-RBFNN. This indicates that the prediction performance of the RBFNN optimized by ADGOA and EDPC is significantly better than that of the traditional RBFNN and GA-RBFNN. Comparing SVM, LightGBM and DNNOM, we can see that the four error indices of DNNOM are lower than those of SVM and LightGBM, indicating that the prediction performance of DNNOM is better than that of SVM and LightGBM. In addition, the prediction of heavy metal As by DNNOM produced the maximum value of R^2 (0.726), which showed the highest fitting degree, indicating that the performance of DNNOM was better than other models.Therefore, in the prediction of soil heavy metal content, DNNOM proposed in this paper has better prediction performance than the other four models, and can be used as an effective method to accurately predict soil heavy metal content.

VI. CONCLUSION AND FUTURE WORK

Currently, an increasing amount of soil is being polluted by heavy metals, which directly affect human health. Therefore, strengthening the management and utilization of soil has become a priority. The prediction accuracy of current methods for soil heavy metal content prediction is generally low. The DNNOM proposed in this paper introduces ADGOA to solve the problem of random generation of output layer parameters based on RBFNN, and EDPC and AVM to solve the problem of random selection of hidden layer center points and width vectors. The experimental results show that the prediction performance of DNNOM is better than that of the other models, and the prediction accuracy of the soil heavy metal content can be higher, which can provide a new method for predicting soil heavy metal content. However, DNNOM may also be affected by the amount of input data, which may lead to long running times. In future work, we will further adjust the structure of DNNOM to reduce its running time and apply it to other fields.

REFERENCES

[1] M. Chodak, M. Niklinska, and F. Beese, "Near-infrared spectroscopy for analysis of chemical and microbiological properties of forest soil organic horizons in a heavy-metal-polluted area," *Biol. Fertility Soils*, vol. 44, no. 1, pp. 171–180, Sep. 2007, doi: 10.1007/s00374-007-0192-z.

[2] K. Chander, J. Dyckmans, R. Joergensen, B. Meyer, and M. Raubuch, "Different sources of heavy metals and their long-term effects on soil microbial properties," *Biol. Fertility Soils*, vol. 34, no. 4, pp. 241–247, Sep. 2001, doi: 10.1007/s003740100406.

[3] Z. Li, Z. Ma, T. J. van der Kuijp, Z. Yuan, and L. Huang, "A review of soil heavy metal pollution from mines in China: Pollution and health risk assessment," *Sci. Total Environ.*, vols. 468–469, pp. 843–853, Jan. 2014, doi: 10.1016/j.scitotenv.2013.08.090.

[4] W. Cao and C. Zhang, "A collaborative compound neural network model for soil heavy metal content prediction," *IEEE Access*, vol. 8, pp. 129497–129509, 2020, doi: 10.1109/ACCESS.2020.3009248.
[5] C. Wang, X.-H. Wu, L.-Q. Li, Y.-S. Wang, and Z.-W. Li, "Convolutional neural network application in prediction of soil moisture content," *Spectrosc. Spectral Anal.*, vol. 38, no. 1, pp. 36–41, Jan. 2018.
[6] Q. Li, C. Q. Wang, T. X. Yue, B. Li, and J. Yang, "Method for spatial variety of soil organic matter based on radial basis function neural network," *Trans. Chin. Soc. Agric. Eng.*, vol. 26, no. 1, pp. 87–93, Jan. 2010.
[7] H. Hao, F. Yu, and Q. Li, "Soil temperature prediction using convolutional neural network based on ensemble empirical mode decomposition," *IEEE Access*, vol. 9, pp. 4084–4096, 2021, doi: 10.1109/ACCESS.2020.3048028.
[8] G. Yin, X. Chen, H. Zhu, Z. Chen, C. Su, Z. He, J. Qiu, and T. Wang, "A novel interpolation method to predict soil heavy metals based on a genetic algorithm and neural network model," *Sci. Total Environ.*, vol. 825, Jun. 2022, Art. no. 153948, doi: 10.1016/j.scitotenv.2022.153948.
[9] A. Dutta, Y. Albagory, M. Al Faraj, M. Alsanea, and A. Sait, "Cat swarm with fuzzy cognitive maps for automated soil classification," *Comput. Syst. Sci. Eng.*, vol. 44, no. 2, pp. 1419–1432, Jul. 2022.
[10] S. Zhang, H. Jiang, Y. Yin, W. Xiao, and B. Zhao, "The prediction of the gas utilization ratio based on TS fuzzy neural network and particle swarm optimization," *Sensors*, vol. 18, no. 2, p. 625, 2018.
[11] L. Tian, H. Liu, F. Luan, and Y. Gao, "QSPR study on the prediction of ionization potential of various organic compounds by heuristic method and radial basis function neural network," in *Proc. 7th Int. Conf. Natural Comput.*, Jul. 2011, pp. 199–202, doi: 10.1109/ICNC.2011.6022125.
[12] A. Alexandridis, E. Chondrodima, and H. Sarimveis, "Radial basis function network training using a nonsymmetric partition of the input space and particle swarm optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 2, pp. 219–230, Feb. 2013, doi: 10.1109/TNNLS.2012.2227794.
[13] F. Fernández-Navarro, C. Hervás-Martínez, and P. A. Gutierrez, "Generalised Gaussian radial basis function neural networks," *Soft Comput.*, vol. 17, no. 3, pp. 519–533, Mar. 2013, doi: 10.1007/s00500-012-0923-4.
[14] Y. Zhai, L. Zuo, and E. Zhang, "Algorithm for structure design of RBF neural network based on parameter optimization," *J. Northeast. Univ.*, vol. 41, no. 2, pp. 176–181, Feb. 2020.
[15] W. Cao and C. Zhang, "Data prediction of soil heavy metal content by deep composite model," *J. Soils Sediments*, vol. 21, no. 1, pp. 487–498, Jan. 2021, doi: 10.1007/s11368-020-02793-y.
[16] S. Cheng, J. Miao, and S. Wu, "Use of metamodeling optimal approach promotes the performance of proton exchange membrane fuel cell (PEMFC)," *Appl. Energy*, vol. 105, pp. 161–169, May 2013, doi: 10.1016/j.apenergy.2013.01.001.
[17] T. Xiong, Y. Bao, Z. Hu, and R. Chiong, "Forecasting interval time series using a fully complex-valued RBF neural network with DPSO and PSO algorithms," *Inf. Sci.*, vol. 305, pp. 77–92, Jun. 2015.
[18] Y. Hu, J. J. You, J. N. K. Liu, and T. He, "An eigenvector based center selection for fast training scheme of RBFNN," *Inf. Sci.*, vol. 428, pp. 62–75, Feb. 2018, doi: 10.1016/j.ins.2017.08.092.
[19] M. Zhu, W. Wang, and J. Huang, "Improved initial cluster center selection in K-means clustering," *Eng. Computations*, vol. 31, no. 8, pp. 1661–1667, Oct. 2014, doi: 10.1108/ec-11-2012-0288.
[20] M. Han and J. Xi, "Efficient clustering of radial basis perceptron neural network for pattern recognition," *Pattern Recognit.*, vol. 37, no. 10, pp. 2059–2067, 2004, doi: 10.1016/j.patcog.2004.02.014.
[21] G. S. Babu and S. Suresh, "Sequential projection-based metacognitive learning in a radial basis function network for classification problems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 2, pp. 194–206, Feb. 2013.
[22] S. Oh, W. Kim, W. Pedrycz, and K. Seo, "Fuzzy radial basis function neural networks with information granulation and its parallel genetic optimization," *Fuzzy Sets Sys*, vol. 237, pp. 96–117, Feb. 2014.
[23] F. Li, L. Xu, C. Jin, and H. Wang, "Structure of multi-stage composite genetic algorithm (MSC-GA) and its performance," *Expert. Syst. Appl.*, vol. 38, no. 7, pp. 8929–8937, Jul. 2011, doi: 10.1016/j.eswa.2011.01.110.
[24] L. Wang, D. Fang, and W. Sheng, "Combination of genetic algorithm (GA) and fast Fourier transform (FFT) for synthesis of arrays," *Microw. Opt. Techn. Lett.*, vol. 37, no. 1, pp. 56–59, Apr. 2003.
[25] P. Lopez-Garcia, E. Onieva, E. Osaba, A. D. Masegosa, and A. Perallos, "A hybrid method for short-term traffic congestion forecasting using genetic algorithms and cross entropy," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 557–569, Feb. 2016, doi: 10.1109/TITS.2015.2491365.
[26] D.-Y. Oh and J. B. Gray, "GA-ensemble: A genetic algorithm for robust ensembles," *Comput. Statist.*, vol. 28, no. 5, pp. 2333–2347, Oct. 2013, doi: 10.1007/s00180-013-0409-6.
[27] X. Xu, S. Ding, and Z. Shi, "An improved density peaks clustering algorithm with fast finding cluster centers," *Knowl.-Based Syst.*, vol. 158, pp. 65–74, Oct. 2018, doi: 10.1016/j.knosys.2018.05.034.
[28] C. Wu, J. Lee, T. Isokawa, J. Yao, and Y. Xia, "Efficient clustering method based on density peaks with symmetric neighborhood relationship," *IEEE Access*, vol. 7, pp. 60684–60696, 2019, doi: 10.1109/ACCESS.2019.2912332.
[29] C. Yan, M. Li, and W. Liu, "Application of improved genetic algorithm in function optimization," *J. Inf. Sci. Eng.*, vol. 35, no. 6, pp. 1299–1309, Nov. 2019, doi: 10.6688/JISE.201911_35(6).0008.
[30] W. Cao and C. Zhang, "An effective parallel integrated neural network system for industrial data prediction," *Appl. Soft Comput.*, vol. 107, Aug. 2021, Art. no. 107397, doi: 10.1016/j.asoc.2021.107397.

KUN CAO received the B.E. degree in software engineering from Wuhan Polytechnic University, Wuhan, China, in 2020, where he is currently pursuing the M.S. degree in computer technology. His research interest includes artificial intelligence technology and its application.

CONG ZHANG received the bachelor's degree in automation engineering from the Huazhong University of Science and Technology, in 1993, the master's degree in computer application technology from the Wuhan University of Technology, in 1999, and the Ph.D. degree in computer application technology from Wuhan University, in 2010. He is currently a Professor with the School of Electrical and Electronic Engineering, Wuhan Polytechnic University. His research interests include multimedia signal processing, multimedia communication system theory and application, and pattern recognition.

LIANGLIANG LI received the B.E. degree in software engineering from Wuhan Polytechnic University, Wuhan, China, in 2020, where he is currently pursuing the M.S. degree in computer technology. His research interest includes artificial intelligence technology and its application.

SHUAIFENG LI received the B.E. degree in software engineering from Wuhan Business University, Wuhan, China, in 2020. He is currently pursuing the M.S. degree in computer technology with Wuhan Polytechnic University, Wuhan. His research interest includes artificial intelligence technology and its application.