Hindawi
International Journal of Antennas and Propagation
Volume 2021, Article ID 5586851, 13 pages
https://doi.org/10.1155/2021/5586851

Research Article

Radar Working State Recognition Based on Improved HPSO-BP

Huiqin Li, Yanling Li, Xuemei Wang, Zhe Xu, and Xinli Yin

Xi'an Research Institute of High Technology, Xi'an 710025, China

Correspondence should be addressed to Yanling Li; lyling998@163.com

Received 8 January 2021; Revised 16 March 2021; Accepted 22 March 2021; Published 9 April 2021

Academic Editor: Ana Alejos

In this paper, a recognition model based on the improved hybrid particle swarm optimisation (HPSO) optimised backpropagation network (BP) is proposed to improve the efficiency of radar working state recognition. First, the model improves the HPSO algorithm through the nonlinear decreasing inertia weight by adding the deceleration factor and asynchronous learning factor. Then, the BP neural network's initial weights and thresholds are optimised to overcome the shortcomings of slow convergence rate and falling into local optima. In the simulation experiment, improved HPSO-BP recognition models were established based on the datasets for three radar types, and these models were subsequently compared to other recognition models. The results reveal that the improved HPSO-BP recognition model has better prediction accuracy and convergence rate. The recognition accuracy of different radar types exceeded 97%, which demonstrates the feasibility and generalisation of the model applied to radar working state recognition.

1. Introduction

Radar working state recognition is used in developing models of the reconnaissance pulse signal characteristics and estimating the internal working state of a radar with prior knowledge. The rapid and accurate recognition of the radar working state is important for determining the radar threat level, evaluating the radar interference effect, and realising the interference decision.

When the radar is in a different working state, the signal parameter characteristic exhibits an obvious change. The radar working state recognition can be summarised as a pattern-classification problem. The existing research methods can be summarised as follows: methods based on statistical decisions, methods based on ambiguity decisions, methods based on syntactic structures, and methods based on artificial intelligence [1]. Various studies [2, 3] have solved the radar working pattern matching problem based on the Dempster–Shafer (D–S) evidence theory fusion method. However, for multifunctional radars with complex signals, the calculation of information fusion exhibits exponential growth. Other studies [4, 5] have introduced the recognition algorithm of the fuzzy function, which objectively reflects the samples' real attributes, but the selection of the ambiguity function was not theoretical. Some studies [6–8] have proposed a syntactic model for accurately extracting radar words from an intercepted radar pulse train for multifunctional radar state recognition. However, these studies did not consider the influence of unstable factors, such as inaccurate prior information and poor signal data, which resulted in poor fault tolerance and generalisation ability. Thus, the improvement of the efficiency of radar working state recognition requires further investigation.

The backpropagation (BP) neural network has been widely used [9, 10] owing to its good self-learning and adaptive ability and has been applied to radar working state recognition. Some studies [11, 12] have used neural networks to establish the corresponding relationship between the signal characteristic parameters and the radar working state to form a recognition database and then identify unknown working states. However, the BP neural network has high requirements regarding the dataset quality and is sensitive to the initial network weights and thresholds of the gradient descent algorithm. Moreover, the BP can easily fall into local optima and has slow convergence speed. Considering the abovementioned problems, related studies [13, 14] have used swarm intelligence optimisation algorithms to optimise the BP neural network parameters. Since

Kennedy and Ebarhant [15] proposed particle swarm optimisation (PSO) based on the influence of bird predation behaviour by group movement, numerous studies have increasingly analysed the phenomenon of different biological populations and were inspired by it to propose various metaheuristics. For example, Mirjalili [16] proposed a moth search algorithm (MSA) based on the way that moths spiral around a light source at night. Wang et al. [17] proposed monarch butterfly optimisation (MBO) inspired by the behaviour of monarch butterflies with seasonal changes. Heidari et al. [18] proposed Harris hawk optimisation (HHO) based on Harris' surprise predation mechanism. Li et al. [19] proposed the slime mould algorithm (SMA) based on the behaviour and morphological changes of slime mould in the spread and foraging process. The PSO method finds the optimal solution through collaboration and information sharing between the individuals in the group and has the characteristics of simple principles, fewer parameters, and strong global search capabilities. Therefore, this method has been widely used in function optimisation [20], neural network training [21], fuzzy control systems [22], and other fields, and the algorithm is considered to be relatively mature. However, PSO may easily converge prematurely. Similar to other swarm intelligence optimisation algorithms, PSO has the disadvantages of slow convergence speed, low optimisation accuracy, and ease of falling into local optima [23] when solving complex optimisation problems.

Considering the problems of the traditional PSO algorithm, this paper proposes the improved HPSO algorithm based on the following three aspects:

(1) Crossover and mutation operations in the genetic algorithm (GA) are introduced to update the particles, the hybrid population information is used to enhance the population diversity, and the algorithm's convergence speed and accuracy are improved

(2) Improved nonlinear decreasing inertia weights are used to balance the development and exploration of the algorithm

(3) An asynchronous learning factor update strategy is used to achieve stronger global search capabilities and faster convergence

This study considered the multifunctional phased array radar to investigate the quick and accurate identification of the radar's working state. A radar working state recognition model based on the improved hybrid PSO BP neural network (HPSO-BP) is proposed, using the improved HPSO global optimisation ability to optimise the structural parameters of the BP network and improve the recognition ability of the BP neural network. In this study, three types of radars were selected to construct the working state recognition models. The simulation results were compared to the results obtained by the BP, GA-BP, and PSO-BP to verify the accuracy, timeliness, and generalisation of the improved algorithm.

The rest of this paper is organised as follows. A detailed introduction to the HPSO algorithm and a description of the relevant optimisation technique are presented in Section 2. The radar working state recognition model is presented in Section 3. The simulation experiment and analysis of the results are presented in Section 4. Finally, the conclusions drawn from this study are discussed in Section 5.

2. Improved HPSO Algorithm

2.1. HPSO Algorithm. When the population converges, the similarity of the particles increases; therefore, the traditional PSO algorithm cannot easily jump out of a locally optimal solution. The HPSO algorithm abandons the method wherein the traditional PSO algorithm updates the particles by tracking the extremum and learns from chromosomal crossover and mutation operations in the GA. By combining the characteristics of the GA's global optimisation, the HPSO algorithm searches for the optimal solution through particle swarm crossover and mutation. The steps of the algorithm are summarised as follows:

Step 1. Initialise the particle velocity, position, and parameters, and set the population size k and maximum number of iterations T.

Step 2. Calculate the fitness value, individual extremum, and group extremum.

Step 3. Equations (1) and (2) are used to update the particle velocity and position, respectively:

$$\begin{aligned} V_i(t+1) = \omega V_i(t) + c_1 r_1 \left(P_i(t) - X_i(t)\right) \\ + c_2 r_2 \left(P_g(t) - X_i(t)\right), \end{aligned} \tag{1}$$

$$X_i(t+1) = X_i(t) + V_i(t+1), \tag{2}$$

where $i \in k$, r_1 and r_2 are random numbers between [0, 1], t is the current iteration number, V_i represents the particle velocity, P_i represents the individual extreme value, P_g represents the group extreme value, X_i represents the particle position, ω is the inertia factor, c_1 is the individual learning factor, and c_2 is the group learning factor.

Step 4. Equations (3)–(5) are used to perform cross and mutation operations on particles X_m and X_n:

$$\begin{cases} X_m(t'+1) = (1-\alpha) * X_m(t') + \alpha * X_n(t') \\ X_n(t'+1) = (1-\alpha) * X_n(t') + \alpha * X_m(t') \end{cases}, \tag{3}$$

$$X_{mn} = \begin{cases} X_{mn} + (X_{mn} - X_{\max}) * f(g), & r_1 \geq 0.5 \\ X_{mn} + (X_{\min} - X_{mn}) * f(g), & r_1 < 0.5 \end{cases}, \tag{4}$$

$$f(g) = r_2 * \left(1 - \frac{t}{T}\right)^2, \tag{5}$$

where α is the crossover probability, X_{mn} is the new particle of variation, and $X_{\max}$ and $X_{\min}$ are the upper and lower bounds of the particle values, respectively.

Step 5. Update fitness value, individual extremum, and group extremum.

Step 6 (*Interiteration Optimisation*). When the maximum number of iterations is reached, the optimal value should be the output. Otherwise, Steps 3–5 are repeated until the maximum number is reached.

In the HPSO algorithm, the selection of parameters, such as the inertia weight and learning factor, affects the optimisation ability of the algorithm. Therefore, these parameters must be improved.

2.2. Improvement of Inertia Weight. The inertia weight ω reflects the ability of inheriting the velocity of the previous particles. As the value of ω increases, the global search is improved. As the value of ω decreases, the local search is improved. In the traditional PSO algorithm, the fixed ω influences the algorithm's convergence. The linear regression of ω results in the algorithm falling into local optima and requires many iterations. A previous study [24] proposed a nonlinearly decreasing ω to improve the PSO algorithm and search efficiency. However, in the early stage of the algorithm's iterations, ω decreases too fast, which results in local convergence instead of the globally optimal solution. Accordingly, this study added two deceleration factors to ω so as to stably maintain a larger value in the early iteration for global optimisation and rapidly maintain a smaller value in the late iteration for algorithm convergence. Therefore, the convergence ability is improved and the global optimal solution is ensured. The improved nonlinear decreasing inertia weight is expressed as follows:

$$\omega = \omega_{\min} + (\omega_{\max} - \omega_{\min})\exp\left(-a\left(\frac{t}{T}\right)^{b}\right), \tag{6}$$

where $\omega_{\max}$ is the initial inertia weight, $\omega_{\min}$ is the inertia weight when the iteration reaches the maximum number, $\omega_{\max} = 0.9$ and $\omega_{\min} = 0.4$, and a and b are deceleration factors.

2.3. Improvement of Learning Factors. The learning factors c_1 and c_2 determine the maximum length of the particle's flight toward the optimal direction of the individual or group. A larger c_1 will result in the particles tending to search in their own neighbourhood, while a larger c_2 will result in the particles searching within the groups' scope. To ensure that the algorithm maintains an effective balance between global search and local search, this study used an asynchronous learning factor, as expressed in equation (7). In the early stage of the search, c_1 is larger and c_2 is smaller, which improves the self-learning ability of the HPSO. In the later stage of the search, c_1 is smaller and c_2 is larger, which makes the HPSO algorithm quickly approach the global optimal solution:

$$\begin{cases} C_1 = C_{1,start} + \left(C_{1,end} - C_{1,start}\right) * \dfrac{t}{T} \\ \\ C_2 = C_{2,start} + \left(C_{2,end} - C_{2,start}\right) * \dfrac{t}{T} \end{cases}, \tag{7}$$

where $C_{1,start} = C_{2,end} = 2.5$, $C_{1,end} = C_{2,start} = 0.5$, t is the current number of iterations, and T is the maximum number of iterations.

3. Recognition of Radar Working State

3.1. Structure of Radar Working State Recognition Model. This study combined the three-layer BP neural network with the improved HPSO algorithm. The structure of the radar working state recognition model based on the improved HPSO-BP is shown in Figure 1.

Generally, the modelling steps of the improved HPSO-BP are as follows:

(1) Extract the characteristic signal parameters: the characteristics of the radar signal intercepted by the reconnaissance plane include the carrier frequency, pulse repetition period, and pulse width. The ranges of the characteristic parameters corresponding to each working state of the radar are different.

(2) Generate and test the dataset: a random number generation method is used to generate the dataset. To reflect the characteristics of the radar signal parameters, the uniformity and independence of the dataset must be tested. After normalisation, the dataset is divided into the training and test sets.

(3) Train the improved HPSO-BP neural network model: after building the neural network model, the characteristic parameters of the training set are used as the model's input, and the working state of the radar is used as the output of the model to realise supervised autonomous learning. Then, the model performance is tested using the test set.

(4) Radar working state recognition: the trained neural network model is used as the knowledge base for the radar working state recognition. With this model, the radar working state can be assessed, and the radar's threat level can be estimated in real time.

3.2. Recognition Procedure Based on Improved HPSO-BP. To prevent the BP neural network from falling into local minima and improve the prediction accuracy and convergence rate of the network, this study used the global search capability of the improved HPSO algorithm to find the network's optimal initial weight and threshold such that the BP neural network can achieve better recognition performance. The algorithm's step flow is illustrated in Figure 2.

The specific steps of the algorithm are as follows:

Step 1. During the data preprocessing, the dataset is generated using the random number generation method. After verifying that the dataset has uniform distribution characteristics, the input parameters are

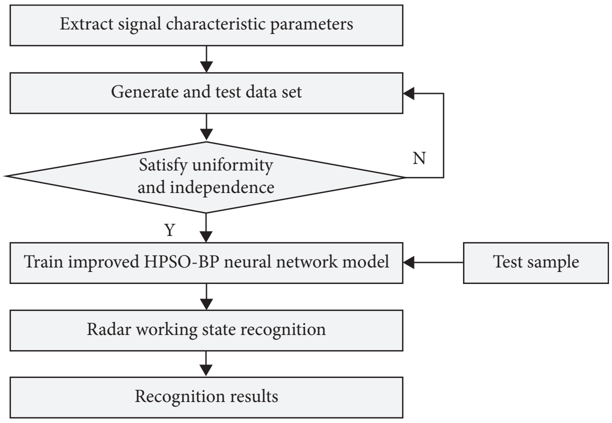

Extract signal characteristic parameters
Generate and test data set
Satisfy uniformity and independence
N
Y
Train improved HPSO-BP neural network model
Test sample
Radar working state recognition
Recognition results

Figure 1: Structure of radar working state recognition model based on improved HPSO-BP.

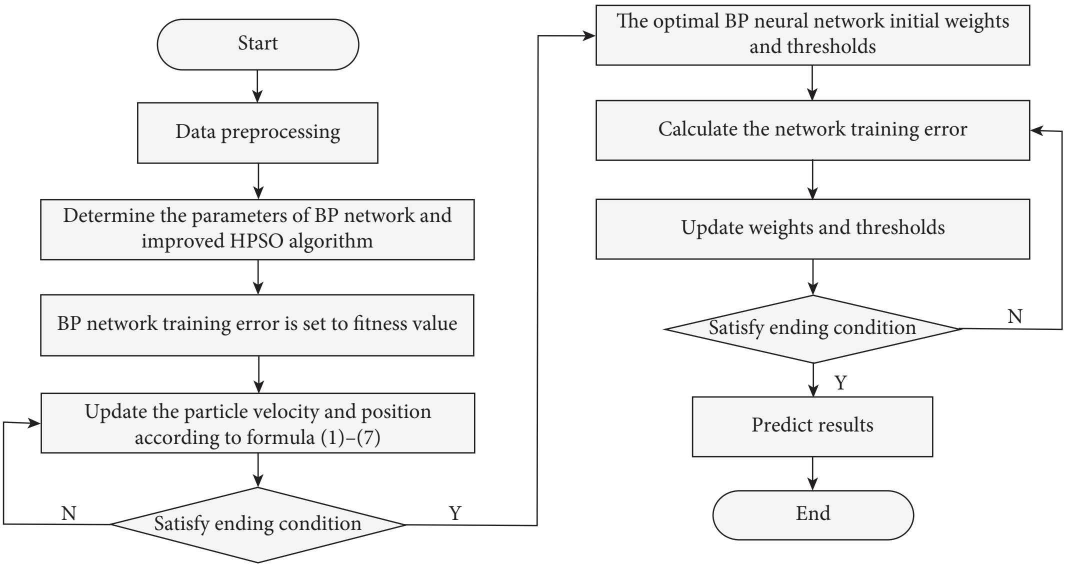

Start
Data preprocessing
Determine the parameters of BP network and improved HPSO algorithm
BP network training error is set to fitness value
Update the particle velocity and position according to formula (1)–(7)
N
Satisfy ending condition
Y
The optimal BP neural network initial weights and thresholds
Calculate the network training error
Update weights and thresholds
Satisfy ending condition
N
Y
Predict results
End

Figure 2: Flow of improved HPSO-BP algorithm.

normalised in the interval of [0, 1]. The dataset is divided into a training set and a test set.

Step 2. The structure of the BP network and the PSO parameters are determined. The structure of the BP network is determined by the dimensions of the input feature vector and the expected output dimension, and the calculation formula is expressed by equation (8). The BP network structure determines the particle dimension, and the calculation formula is expressed by equation (9):

$$m = \sqrt{n + l} + \alpha, \quad \alpha \in [1, 10], \tag{8}$$

$$D = n * m + m * l + m + l, \tag{9}$$

where n is the number of nodes in the input layer, l is the number of nodes in the output layer, m is the number of nodes in the hidden layer, and D is the particle dimension.

Step 3. The network training error is set as the fitness value of the particle:

$$f = \frac{1}{2}\sum_{k=1}^{N} (d_k - o_k)^2, \tag{10}$$

where d_k is the expected output of the network, o_k is the actual output, and N is the set size. As the fitness value decreases, the particle performance is improved.

Step 4. The particle position is updated according to equations (1)–(7) and the particle with the smallest error in each generation is considered as the current optimal particle.

Step 5. When the maximum number of the particle swarm iterations is reached, the output optimal particle position is used as the network's initial weight and threshold value. Otherwise, steps 4–5 are repeated.

```
(1)  Begin
(2)  For ($i = 1\,to\,N$) // $N$ is the particle population size
(3)    Initialise velocity $V_i$ and position $X_i$ for particle $i$;
(4)    Calculate the fitness value $fit$ of particle $i$ and set $P_i = X_i$; //$P_i$ is the individual extreme value
(5)  End for $i$
(6)  $P_g = \min\{P_i\}$;//$P_g$ is the group extreme value
(7)  For ($t = 1\,to\,T$) // $T$ is the maximum number of iterations
(8)    Improve $\omega$, $c_1$, and $c_2$ according to equations (6) and (7);
(9)    For ($i = 1\,to\,N$)
(10)     Update $V_i$ and $X_i$ according to equations (1) and (2);
(11)     Apply the crossover and mutation operator to particle $i$ according to equations (3)–(5);
(12)     Calculate the fitness value $fit$ of particle $i$;
(13)     If ($fit(X_i) < fit(P_i)$)
(14)       $P_i = X_i$;
(15)     End if
(16)     If ($fit(P_i) < fit(P_g)$)
(17)       $P_g = P_i$;
(18)     End if
(19)   End for $i$
(20) End for $t$
(21) Determine the optimal weight and threshold of the BP neural network according to $P_g$;
(22) While (the training target error is not reached)
(23)     Train the BP neural network;
(24) End while
(25) Predict the results on the test set;
(26) End
```

Algorithm 1: Pseudocode of improved HPSO-BP algorithm.

Step 6. The neural network is trained until the training target error is achieved, and the test set's result is predicted.

Combined with the implementation steps of the improved HPSO-BP algorithm, the specific pseudocode is summarised in Algorithm 1

According to the pseudocode of the algorithm, we gradually analysed the time complexity of the algorithm as described below:

(1) In steps 2–5, the population is randomly initialised, and the fitness is calculated. The population size is N, the solution dimension is D, and the time complexity is $O(N * D)$.

(2) In steps 7–20, there are two loop levels. The loop times are T and N, respectively. In the loop, the time complexity is mainly affected by steps 10–12. In step 10, the time complexity of updating the particle's D-dimensional velocity vector and position vector is $O(D)$. In step 11, the worst case is that all individuals participate in crossover and mutation, and the time complexity is $O(N)$. In step 12, the time complexity of the individual fitness calculation is $O(D)$. The sum of the worst time complexity of the operations from steps 10 to 12 is $O(D + N + D)$, and the time complexity magnitude is $O(D + N)$. Therefore, the total time complexity of steps 7–20 is $O(T * N * (D + N))$.

(3) In steps 22–24, according to the pseudocode analysis of the BP algorithm in the literature [25], the training time complexity of the BP neural network is $O(K * D)$, where K is the number of samples.

Therefore, the time complexity of the entire optimisation algorithm is $O(N * D) + O(T * N * (D + N)) + O(K * D) = O(T * N^2 + T * N * D + N * D + K * D)$, and the time complexity magnitude is $O(T * N^2 + T * N * D + K * D)$. Hence, the time complexity of the improved HPSO-BP algorithm is related to the total number of particles, number of training samples, dimensionality of the solution dimension (BP network structure), and number of iterations of the algorithm and is mainly affected by the total number of particles.

4. Simulation Experiment and Analysis of Results

This experiment considered an airborne-phased array radar in air-air combat mode [26] as an example and used the improved HPSO-BP neural network model to identify its working state. The results were compared to the simulation results obtained using the BP, GA-BP, and PSO-BP models in terms of recognition accuracy and convergence. Additionally, the working state parameters of radars A and B obtained from the literature [27] were used to verify the model's generalisation performance.

4.1. Generation and Verification of Datasets. In a complex electromagnetic environment, the radar signal parameters typically exhibit obvious randomness. Owing to the radar's military-level sensitivity, it is currently difficult to obtain real data for the radar operating parameters. Some studies [26, 27] have randomly generated data based on a table of radar operating parameter ranges, but have not tested the randomness of the data. Uniformity and independence are two important indicators for testing the randomness of data. In this study, classical random number generation methods, such as the square method, linear congruence method, and Mersenne rotation algorithm, were used to generate the dataset. The dataset's uniformity and independence were tested to select the dataset generation method used in this experiment. Table 1 lists the radar operating parameter ranges obtained from the literature [26].

4.1.1. Uniformity Test. In this study, the chi-square test was used to assess the uniformity. Under the same significance level, as the chi-square value decreased, the significance probability increased, and the uniformity improved. By considering the repetition variable of the S1 state in Table 1 as an example, the 1000 randomly generated sample data were normalised and divided into ten groups. The uniformity test results are presented in Table 2.

As presented in Table 2, when the standard significance level α was set to 0.05, because the probability of the progressive significance was greater than 0.05, there was no significant difference between the actual and theoretical distributions, which indicates that the data generated by each method were uniformly distributed in each interval.

The chi-square value calculated by the Mersenne Twister algorithm was minimum, while the progressive significance was maximum, which indicates that the uniformity was optimal.

4.1.2. Independence Test. In this study, the correlation coefficient test was used to perform the independence test. As the correlation coefficient decreased, the significance probability increased, and the independence of the data was enhanced. The autocorrelation test results obtained using the SPSS software are shown in Figure 3.

As shown in Figure 3, when the degree of freedom was greater than 13, the significance probability of the random numbers generated by the linear congruence algorithm was less than 0.05, which indicates that the random numbers are correlated. In contrast, the significance probability of the random numbers generated by the other two algorithms was greater than 0.05, which indicates that the random numbers generated by the Mersenne Twister algorithm and middle-square algorithm have a stable distribution and satisfy the independence condition.

Combined with the two abovementioned testing methods, the random numbers generated by the Mersenne Twister algorithm have higher quality and satisfy the uniform distribution conditions, compared with the other two algorithms.

4.2. Model Evaluation Index. To quantitatively evaluate the prediction accuracy of the model, this study adopted the recognition accuracy rate (ACC) and determination coefficient (R^2) as the evaluation indices. The convergence rate was evaluated based on the number of convergent steps. With a larger ACC, the result was more accurate. Moreover, R^2 was in the range of [0, 1], and the performance of the model improved as R^2 approached 1. With fewer convergent steps, the performance of the model also improved. The specific calculation formulas of the relevant indices are as follows:

$$\mathrm{ACC} = \frac{\sum_{m=1}^{N} s_m}{k}, \tag{11}$$

$$R^2 = \frac{\left(k\sum_{i=1}^{k} d_i o_i - \sum_{i=1}^{k} d_i \sum_{i=1}^{k} o_i\right)^2}{\left(k\sum_{i=1}^{k} d_i^2 - \left(\sum_{i=1}^{k} d_i\right)^2\right) - \left(k\sum_{i=1}^{k} o_i^2 - \left(\sum_{i=1}^{k} o_i\right)^2\right)}, \tag{12}$$

where k is the number of samples, N is the number of sample categories, s_m is the number of correctly identified samples of class m, d_i is the actual output of the sample, and i and o_i are the ideal output of sample i.

4.3. Model Parameters' Settings

4.3.1. Parameters' Setting for BP Network Structure. There are five characteristic variables in the airborne-phased array radar, and the number of input layer nodes was set to 5. The output results corresponding to the six states are represented by values 1–6, and the number of output layer nodes is set to 1. The results obtained using the trial and error method demonstrate that the error was minimised when the number of hidden layer nodes was eight; therefore, the network structure was 5-8-1, as shown in Figure 4. The main parameters of the BP network were set as follows:

(i) Activation function of hidden layer: Tansig

(ii) Activation function of output layer: Purelin

(iii) Training function of neural network: Trainlm

(iv) Maximum iteration step of neural network: 1000

(v) Training target error of neural network: 0.001

(vi) Learning rate factor of neural network: 0.01

To avoid the shock of network training, the selection of sample categories should be balanced, the number of samples in each category should be approximately equal, and cross input is required for the samples belonging to different categories. This study adopted the Mersenne Twister algorithm to randomly generate 100 samples for each radar working state, and each sample had the five abovementioned characteristic variables.

4.3.2. Parameters' Settings for Improved HPSO

(1) Comparison of Inertia Weight of Improved PSO. According to the improved formula of the inertia weight

Table 1: Radar operating parameter ranges.

State	Re-frequency (KHz)	Pulse width (ms)	Duty ratio (%)	Pulse pressure ratio	Instantaneous bandwidth (MHz)
S1	2–20	1–200	0.1–10	1–16384	0.2–100
S2	2–300	0.1–60	0.01–50	1–16384	0.2–500
S3	0.5–5	1–50	1–10	1–13	0.1–1
S4	0–20	0.1–0.5	0.1–1	1–5	1–10
S5	10–300	1–20	1–33	1–16	0.1–1
S6	6–20	1–20	1–25	5–526	1–10

Table 2: Comparison of uniformity test indices of three methods.

Test variables	Mersenne Twister algorithm	Middle-square algorithm	Linear congruence algorithm
Chi-square value	5.320	11.500	9.100
Progressive significant	0.806	0.243	0.428

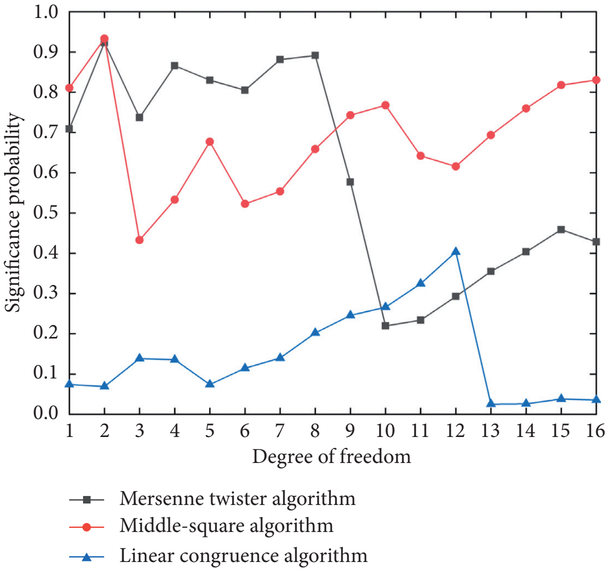


Figure 3: Significance level test of three methods.

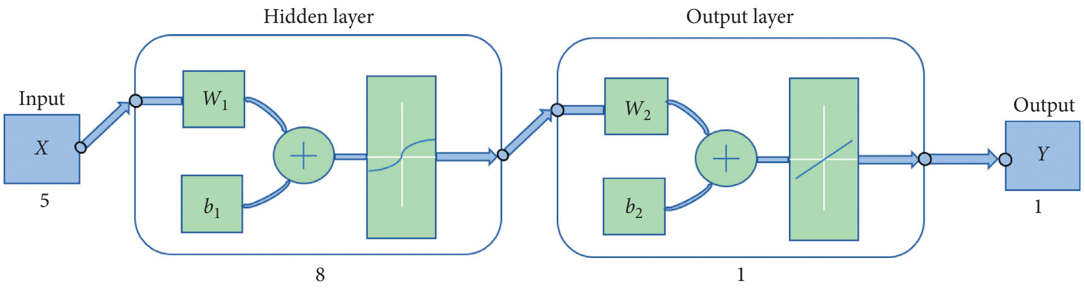


Figure 4: Neural network structure.

presented in Section 2.2, when the deceleration factor a was 8 and b was 3, the proposed improved nonlinearly decreasing ω, the nonlinearly decreasing ω [20], and linearly decreasing ω are shown in Figure 5.

From Figure 5, the following conclusions can be drawn:

(i) The linearly decreasing ω can easily fall into local optima

(ii) In the early phase of iteration, the nonlinearly decreasing ω maintains a large value search, but the decline rate is too fast and may easily lead to local convergence without a global optimal search.

(iii) The deceleration factor decelerates the decreasing rate of the improved nonlinear ω in the early stage of iteration, which results in maintaining a large global

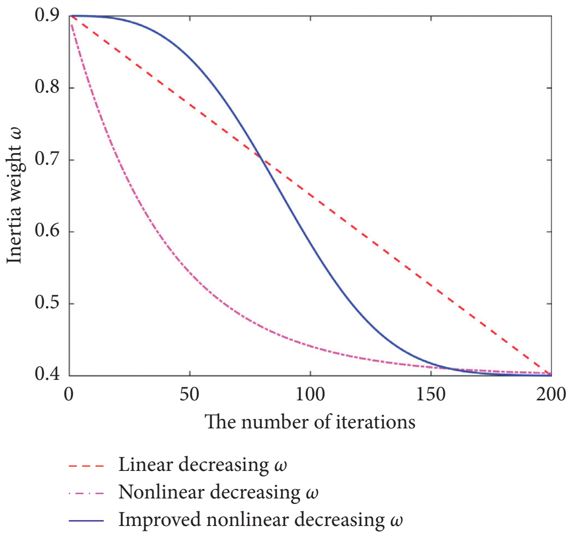


Figure 5: Comparison curve of inertia weight of three PSO algorithms.

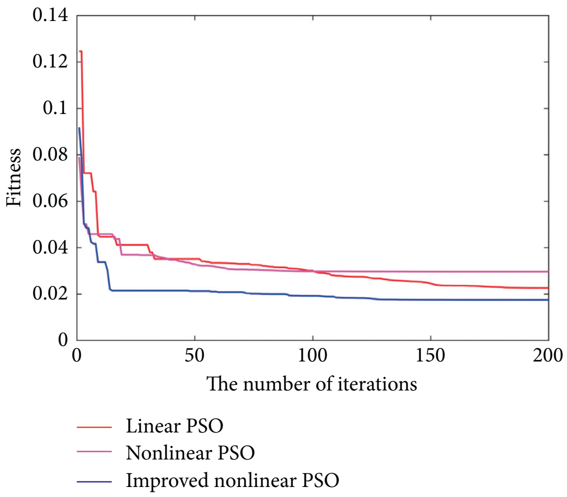


Figure 6: Comparison fitness curve of three PSO algorithms.

search value. Additionally, its rapid decrease in the late stage of iteration results in maintaining a stable value for local search and is more conducive to convergence.

Figure 6 compares the relationship between the fitness values and iteration times of the particles under different inertia weight methods. Compared with the other two methods, the number of iterations of the improved nonlinear PSO that is required to reach the optimal fitness is significantly reduced to avoid falling into local optima, and the optimal fitness of the proposed method is smaller. Therefore, the setting of the deceleration factor is reasonable. Furthermore, this demonstrates that the improved PSO algorithm can overcome the shortcomings of the slow iteration of particles and falling into local optima. When the number of PSO iterations is 20, the fitness curve of the IPSO-BP model converges, and the algorithm searches for the optimal value. If the search continues, the time cost of the search increases.

(2) Comparison of Learning Factor of Improved PSO. The value of the PSO learning factor affects the efficiency of information transfer between the individual particles and the group and ultimately determines the particle change speed and convergence effect. In most cases, this was considered to be constant. To prevent the basic PSO algorithm from falling into local optima and appearing premature during the optimisation, a shrinkage factor was introduced into equation (1), according to the literature [28], to eliminate the speed boundary limit and ensure the bounded and convergence characteristics of the PSO algorithm. The shrinkage factor calculation formula is expressed as follows:

$$\begin{aligned} V_i(t+1) = \varphi * \left(\omega V_i(t) + c_1 r_1 \left(P_i(t) - X_i(t)\right)\right) \\ + c_2 r_2 \left(P_g(t) - X_i(t)\right). \end{aligned} \tag{13}$$

The shrinkage factor calculation formula is expressed as follows:

$$\varphi = \frac{2}{\left|2 - c - \sqrt{c^2 - 4c}\right|}, \quad c = c_1 + c_2,\ c > 4, \tag{14}$$

where φ is the contraction factor and c is the learning factor.

To compare the advantages of using asynchronous learning factors for improving the PSO algorithm, this study adopted three learning factors, namely, the asynchronous learning factor, synchronous learning factor, and shrinkage factor. Under the same change trend of the inertia factor, the relationship between the fitness values of different learning factors and the number of iterations was established. As shown in Figure 7, the introduction of the shrinkage factor still has the problem of easily falling into local optima and requiring more iterations. The optimal fitness value achieved by the synchronous learning factor of the improved PSO was significantly lower compared with that of the shrinkage factor of the improved PSO. However, the required number of iterations was still high. The dynamic adaptive change of the asynchronous learning factor satisfies the requirements of fast optimisation and has a low number of evolutions. Hence, the parameter optimisation problems, such as falling into local optima and having low recognition accuracy, are avoided. After the optimal BP is obtained, the algorithm's recognition accuracy and search efficiency are improved.

In this study, the mean square error between the actual output and the BP neural network's ideal output was selected as the fitness function. The total number of network weights and thresholds were considered as the PSO dimension.

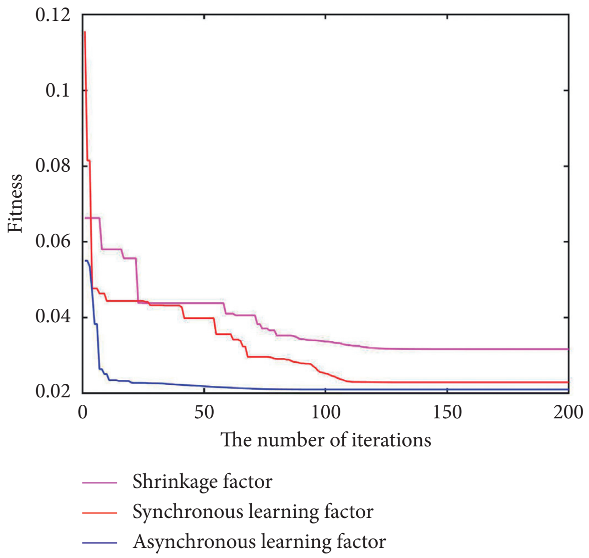


Figure 7: Comparison of fitness curve of three PSO algorithms.

According to the common value range of the other parameters, the improved operating parameters of the HPSO algorithm were set as presented in Table 3.

4.4. Analysis of Results

4.4.1. Comparison of Prediction Accuracy. In this study, the neural network toolbox in MATLAB was used in the simulation experiment. After the normalisation process, the dataset was used as the model input. In each state, 80% of the 100 samples were randomly used as training samples and the remaining 20% were used as test samples. To facilitate the comparative analysis of the model, the training sets were used to train the BP, GA-BP, PSO-BP, and improved HPSO-BP neural network, while the test sets were used to test the trained network. The comparison of the prediction accuracy was evaluated by considering the recognition accuracy and determination coefficient. The network structures of the four models were identical. The simulation program for each model was run in MATLAB, and the results are shown in Figure 8.

To further verify each model's prediction performance, the calculation results of each prediction model's evaluation indices are presented in Table 4.

As shown in Figure 8, the four models can realise the correct recognition of the radar working state. The overall recognition result of the improved HPSO-BP model is the most similar to the actual value, followed by the PSO-BP model, GA-BP model, and BP model with the worst recognition performance, respectively. This indicates that the PSO can overcome the shortcoming of the BP model's low recognition accuracy. However, owing to the limitations of PSO, the recognition results for some mutations have larger errors compared with the actual values. The improved HPSO-BP algorithm improves this shortcoming. The recognition accuracy and R^2 were 0.975 and 0.986, respectively, both of which are higher compared with those of the BP, PSO-BP, and GA-BP models. This indicates that the improved HPSO-BP model has the highest degree of fit and best recognition accuracy among the four models. Compared with PSO-BP, the recognition accuracy and R^2 increased by 8.3% and 4.6%, respectively, which further demonstrates that the proposed algorithm improves the model's performance compared with the traditional PSO algorithm.

4.4.2. Convergence Comparison. To evaluate the convergence, the number of convergent steps was considered. The test results are shown in Figure 9.

By comparing the error curves of the four network models during training, it can be seen that the mean square error decreased as the number of training steps increased. The following conclusions are drawn:

(i) The traditional BP neural network model tends to fall into local optima, and the convergence speed is slow.

(ii) The GA-BP model improves the convergence rate, but the parameter optimisation time is still long owing to the introduction of the GA's coding, selection, crossover, mutation, and other operations.

(iii) The PSO-BP model algorithm is simple and has fast convergence speed, but the PSO algorithm can easily fall into local optima.

(iv) Owing to the global search optimisation ability of PSO, the improved HPSO-BP network model tends to converge at step 26 with the best performance of 0.006375. The convergence speed is significantly higher compared with that of the other three algorithms.

The improved PSO algorithm is feasible and superior when applied to the optimisation of the BP neural network weights and threshold parameters. Therefore, the improved HPSO-BP algorithm is more efficient in identifying the radar working state.

Table 3: Parameter settings of improved HPSO algorithm.

Population size	Iterations	Particle dimension	Speed range	Position scope	Crossover probability
50	100	57	[-1, 1]	[-5, 5]	0.7

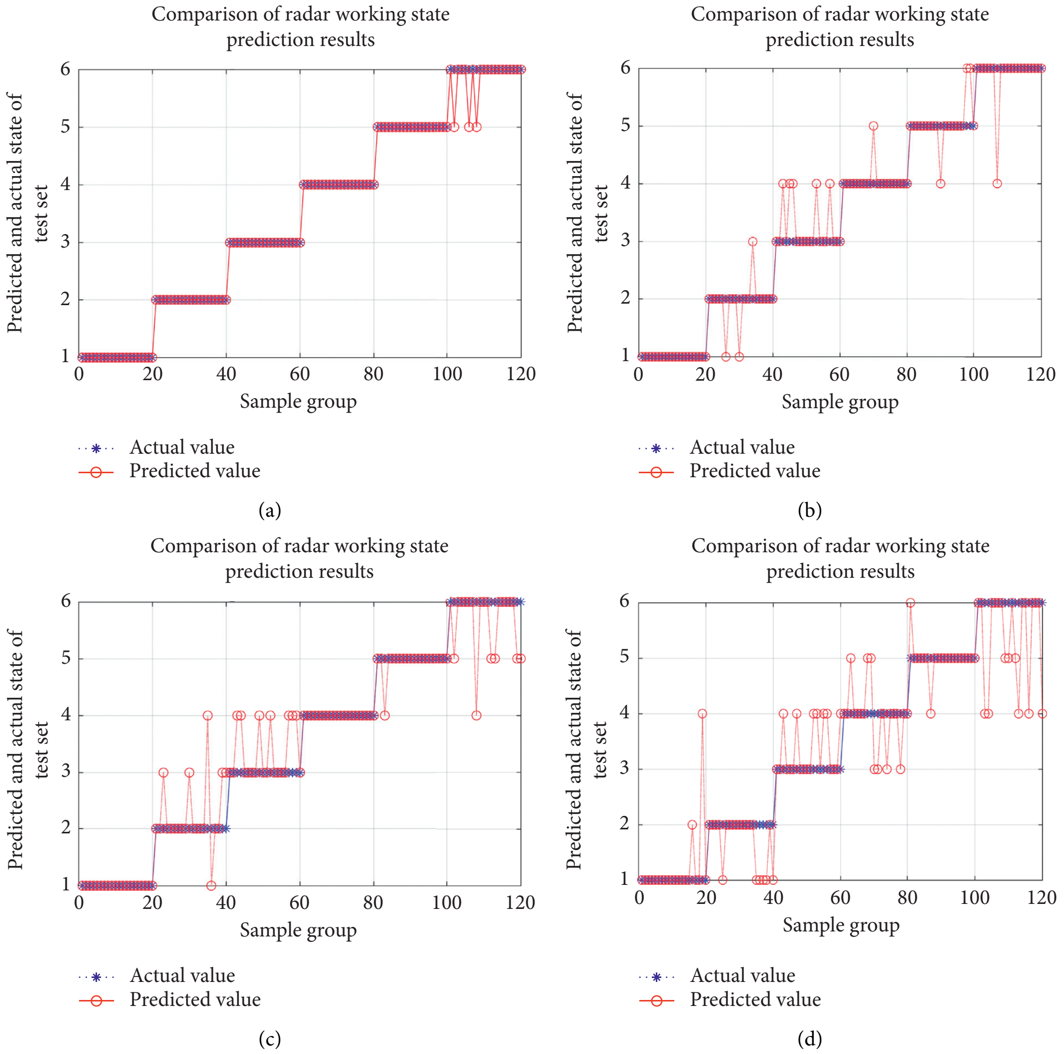


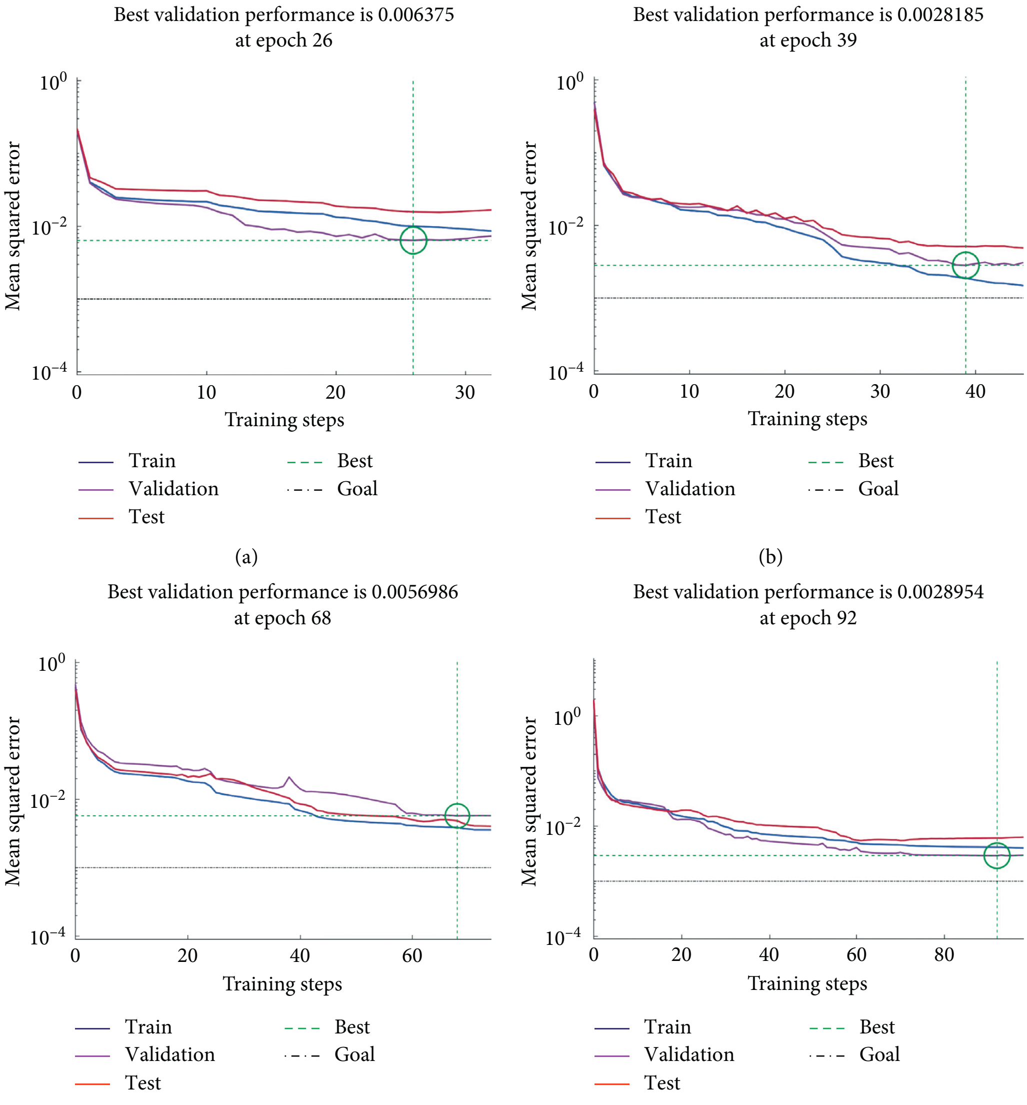
Figure 8: Comparison of prediction results obtained by various models of the airborne-phased array radar. (a) Improved HPSO-BP; (b) PSO-BP; (c) GA-BP; (d) BP.

Table 4: Comparison of evaluation indices of various models of the airborne-phased array radar.

Model	ACC	R^2
Improved HPSO-BP	0.975	0.986
PSO-BP	0.892	0.940
GA-BP	0.833	0.906
BP	0.733	0.873

4.4.3. Model Generalization Performance Verification. To test the generalisation performance of the improved HPSO-BP radar working state recognition model, we selected the operating state parameter ranges of the model A and model B radars [27] as presented in Table 5.

First, according to the Mersenne Twister algorithm, 100 samples were randomly generated for each working state of each radar type; 80% of the samples were randomly selected as the training set, and 20% of the samples were selected as the test set. Therefore, 80 test samples were used for each



(a)

(b)

(c)

(d)

Figure 9: Comparison of each model's convergence. (a) Improved HPSO-BP; (b) PSO-BP; (c) GA-BP; (d) BP.

Table 5: Radar A and B operating parameter ranges.

Radar type	State	Re-frequency (GHz)	Pulse re-interval (μs)	Pulse width (μs)
A	S1	9.7–9.9	3–10	1–3
	S2	9.7–9.9	3–125	0.1–20
	S3	9.5–9.8	50–100	0.1–0.5
	S4	9.5–9.8	50–500	1–200
B	S1	9.15–9.45	500–2000	1–200
	S2	9.15–9.45	50–500	0.1–10
	S3	8.5–10.6	125–1000	0.1–10
	S4	8.5–10.6	100–1000	3–60

radar. Secondly, the number of the input layer nodes of the network was set to 3, the number of output layer nodes was set to 1, and the number of hidden layer nodes was set to 7. Hence, the network structure was 3-7-1, as shown in Figure 10. The other parameters were set using the method described in Section 4.3.

Table 6 compares the proposed model's evaluation indices to those of the BP, GA-BP, and PSO-BP models.

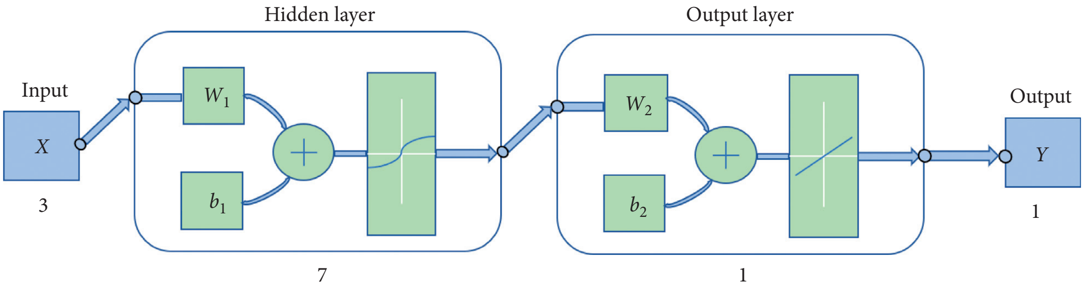


Figure 10: Neural network structure.

Table 6: Performance comparison of each model.

Radar type	Model	ACC	R^2	Convergence steps
A	Improved HPSO-BP	0.994	0.988	18
	PSO-BP	0.943	0.953	35
	GA-BP	0.906	0.940	67
	BP	0.751	0.852	96
B	Improved HPSO-BP	0.972	0.981	20
	PSO-BP	0.901	0.942	42
	GA-BP	0.892	0.937	73
	BP	0.785	0.877	105

As presented in Table 6, the accuracy of the improved HPSO-BP model was increased by up to 24.3%, and the improved HPSO-BP model had the highest recognition accuracy and minimum number of convergence steps, followed by the PSO-BP model, GA-BP model, and BP model with the worst performance, respectively. Because the data quality of different radar models is different, there may exist interference from abnormal data such as random noise; therefore, the accuracy improvement is also different. However, the indices of the radar working state recognition model based on the improved HPSO-BP are superior to those of other recognition models.

Based on the three abovementioned types of radar recognition results, it is demonstrated that the proposed method has good generalisation ability in the classification and recognition of the radar working state.

5. Conclusions

A model based on the improved HPSO-BP is proposed to identify the radar working state. The conclusions drawn from this study are as follows:

(1) By introducing cross variation based on PSO, an improved nonlinear decreasing inertia weight strategy, and an asynchronous learning factor, the improved HPSO algorithm achieved better global optimisation and particle searching speed and avoided falling into local optima.

(2) By combining the global optimisation capability of the improved HPSO algorithm with the recognition capability of the BP neural network, the corresponding relationship between the radar signal parameters and the working mode was established, and the working state recognition model of the radar was built.

(3) The proposed model overcomes the shortcomings of the traditional BP neural network, such as the slow convergence rate and low recognition accuracy. Compared with the standard BP network, GA-BP network, and PSO-BP network, the proposed model has higher accuracy and faster convergence rate. Additionally, the proposed model has higher recognition rate for different radar types and better generalisation.

This study has various limitations. For example, the BP algorithm is a supervised learning algorithm, and the recognition of newly arrived category samples lacks adaptability. In an actual combat environment, the radar is an enemy and its working state is unknown. Therefore, the scope of future work is to investigate an unsupervised incremental target state recognition method and give full play to the cognitive electronic countermeasure system's adaptability and timeliness.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China Youth Program (61703411).

References

[1] Y. D. Zhao and W. Xu, “A method of real-time electronic attack effectiveness-evaluation based on state transition of radar,” *Electronic Information Warfare Technology*, vol. 31, no. 3, pp. 42–46, 2016, in Chinese.

[2] S. X. Yuan, S. J. Lu, S. L. Wang et al., “Modified communication emitter recognition method based on D-S theory,” in *Proceedings of the IEEE International Conference on Signal Processing*, pp. 1–4, Kuala Lumpur, Malaysia, October 2015.

[3] W. Jiang, Y. Cao, L. Yang, and Z. He, “A time-space domain information fusion method for specific emitter identification based on dempster-shafer evidence theory,” *Sensors*, vol. 17, no. 9, pp. 1972–1991, 2017.

[4] M. Z. Zhu, X. L. Zhang, Y. Qi et al., “Compressed sensing mask feature in time-frequency domain for civil flight radar emitter recognition,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2146–2150, Calgary, Canada, April 2018.

[5] Y. T. Zhai, S. J. Fang, Z. X. Cheng et al., “Radar emitter recognition based on ambiguity function contour plot,” in *Proceedings of the 7th IEEE International Conference on Computer Science and Network Technology*, pp. 232–235, Dalian, China, October 2019.

[6] J. Ou, Y. G. Chen, F. Zhao et al., “Novel method for radar word extraction in the syntactic model of multi-function radar,” in *Proceedings of the 18th IEEE International Radar Symposium*, pp. 1–7, Prague, Czech Republic, June 2017.

[7] J. Ou, F. Zhao, X. F. Ai et al., “Data-driven method of reverse modelling for multi-function radar,” in *Proceedings of the 25th IEEE European Signal Processing Conference*, pp. 1947–1951, Kos, Greece, August 2017.

[8] J. Ou, Y. G. Chen, F. Zhao et al., “Research on extension of hierarchical structure for multi-function radar signals,” in *Progress in Electromagnetics Research Symposium-Spring*, pp. 2612–2616, St Petersburg, Russia, May 2017.

[9] H. Liu, K. Guo, Z. Zhang et al., “High-power LED photo-electrothermal analysis based on backpropagation artificial neural networks,” *IEEE Transactions on Electron Devices*, vol. 64, no. 7, pp. 2867–2873, 2017.

[10] L. F. Tang, Z. H. Xu, and V. Bala, “Contactor modeling technology based on an artificial neural network,” *IEEE Transactions on Magnetics*, vol. 54, no. 2, pp. 1–8, 2018.

[11] L. Ding, S. Wang, F. Wang, and W. Zhang, “Specific emitter identification via convolutional neural networks,” *IEEE Communications Letters*, vol. 22, no. 12, pp. 2591–2594, 2018.

[12] J. Matuszewski and K. Sikorska-Lukasiewicz, “Neural network application for emitter recognition,” in *Proceedings of the 18th International Radar Symposium*, pp. 1–8, Prague, Czech Republic, June 2017.

[13] C. Hou, X. Yu, Y. Cao, C. Lai, and Y. Cao, “Prediction of synchronous closing time of permanent magnetic actuator for vacuum circuit breaker based on PSO-BP,” *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 24, no. 6, pp. 3321–3326, 2017.

[14] H. Ji, X. Chen, K. Zhang et al., “Construct drawing man-hour forecasting based on GA-BP in chemical equipment design process,” in *Proceedings of 22nd International Conference on Automation and Computing (ICAC)*, pp. 1–4, IEEE, Colchester, UK, September 2016.

[15] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, UK, November 1995.

[16] S. Mirjalili, “Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm,” *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.

[17] G.-G. Wang, S. Deb, and Z. Cui, “Monarch butterfly optimization,” *Neural Computing and Applications*, vol. 31, no. 7, pp. 1995–2014, 2019.

[18] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, “Harris hawks optimization: algorithm and applications,” *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.

[19] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, “Slime mould algorithm: a new method for stochastic optimization,” *Future Generation Computer Systems*, vol. 111, no. 1, pp. 300–323, 2020.

[20] S. U. Khan, S. Yang, L. Wang et al., “A modified particle swarm optimization algorithm for global optimizations of inverse problems,” *IEEE Transactions on Magnetics*, vol. 52, no. 3, pp. 1–4, 2015.

[21] Q. Liu, K. Ding, B. Wu, and Q. Xie, “Frequency diverse array target localization based on IPSO-BP,” *International Journal of Antennas and Propagation*, vol. 2020, Article ID 2501731, 8 pages, 2020.

[22] S.-C. Wang and Y.-H. Liu, “A PSO-based fuzzy-controlled searching for the optimal charge pattern of Li-ion batteries,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 5, pp. 2983–2993, 2015.

[23] Y.-J. Won, K. H. Lee, and J.-H. Lee, “Performance improvement of spaceborne SAR using antenna pattern synthesis based on quantum-behaved particle swarm optimization,” *International Journal of Antennas and Propagation*, vol. 2017, Article ID 6928970, 12 pages, 2017.

[24] Y. Z. Ma, Q. Q. Wang, R. S. Wang et al., “Optical fiber perimeter signal recognition based on SVD and MPSO-SVM,” *System Engineering and Electronic Technology*, vol. 42, no. 8, pp. 1652–1661, 2020, in Chinese.

[25] Z. H. Zhou, *Machine Learning*, pp. 101–104, Tsinghua University Press, Beijing, China, 2018, in Chinese.

[26] S. Y. Cui, *Research on Target State Recognition Technology in Cognitive Electronic Warfare*, pp. 25-26, Beijing Institute of Technology, Beijing, China, 2016, in Chinese.

[27] Z. Li, *Radar Radiation Source and Working State Recognition*, pp. 35-36, Xidian University, Xi’an, China, 2019, in Chinese.

[28] C. Wang, J. H. Han, and Q. Ji, “Research on optimization design of gear transmission based on improved particle swarm algorithm,” *Mechanical and Electrical Engineering*, vol. 38, no. 2, pp. 239–244, 2021, in Chinese.