

An Improved BPNN Prediction Method Based on Multi-Strategy Sparrow Search Algorithm

Xiangyan Tang^{1,2}, Dengfang Feng^{2,*}, KeQiu Li¹, Jingxin Liu², Jinyang Song³ and Victor S. Sheng⁴

¹College of Intelligence and Computing, Tianjin University, Tianjin, 300350, China

²School of Computer Science and Technology, Hainan University, Haikou, 570228, China

³School of Cyberspace Security (School of Cryptology), Hainan University, Haikou, 570228, China

⁴Department of Computer Science Texas Tech University TX, 79409, USA

*Corresponding Author: Dengfang Feng. Email: fdf521x@yeah.net

Received: 14 April 2022; Accepted: 12 June 2022

Abstract: Data prediction can improve the science of decision-making by making predictions about what happens in daily life based on natural law trends. Back propagation (BP) neural network is a widely used prediction method. To reduce its probability of falling into local optimum and improve the prediction accuracy, we propose an improved BP neural network prediction method based on a multi-strategy sparrow search algorithm (MSSA). The weights and thresholds of the BP neural network are optimized using the sparrow search algorithm (SSA). Three strategies are designed to improve the SSA to enhance its optimization-seeking ability, leading to the MSSA-BP prediction model. The MSSA algorithm was tested with nine different types of benchmark functions to verify the optimization performance of the algorithm. Two different datasets were selected for comparison experiments on three groups of models. Under the same conditions, the mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE) of the prediction results of MSSA-BP were significantly reduced, and the convergence speed was significantly improved. MSSA-BP can effectively improve the prediction accuracy and has certain application value.

Keywords: Prediction; parrow search algorithm; back propagation; neural network

1 Introduction

In the era of big data, the field of artificial intelligence is making a splash, [1–8]. Accurate prediction plays a crucial role in our modern life, where the research of prediction methods based on machine learning, especially neural networks, has become increasingly popular, [9–12]. The back propagation (BP) neural network, a simple structured basic model, generally includes an input, implicit, and output layer. The BP neural network can simulate any complex nonlinear relationship by nonlinear elements in the three-layer structure. The method has been widely used in various fields because of its sound data processing and nonlinear mapping capabilities, [13–15]. Although BP neural



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

networks have many advantages, they tend to fall into local optimum, have low learning efficiency, and converge slowly, [16].

In recent years, much research has been conducted on improving the convergence speed of traditional BP neural networks and avoiding convergence to local optimum. Many optimization methods have been proposed. Among them, intelligent optimization algorithms abstracted by simulating the evolutionary process or foraging behavior of biological populations, [17–20], such as genetic algorithm (GA), ant colony optimization (ACO) and particle swarm optimization (PSO), have been widely used to solve optimization problems due to the advantages of simple implementation and easy scalability. Optimizing BP neural networks using intelligent algorithms has become a research hotspot. Researchers applied standard PSO algorithms to BP neural networks to effectively reduce learning time and improve computational accuracy, [21,22]. Li et al. conducted virtual simulation experiments on the short-term power generation of photovoltaic power plants through three sets of models of BP, GA-BP, and PSO-BP. They verified that GA-BP and PSO-BP could effectively reduce errors, [23]. Mohamad et al. used PSO to optimize BP neural networks for predicting uniaxial compression strength (UCS) of rocks and conducted simulation experiments with laboratory datasets to demonstrate that PSO-BP models have good predictive performance, [24]. Zhu et al. used a GA to optimize BP neural network to obtain the GA-BP model to predict the risk coefficient of rainfall-induced landslides. The results showed that the GA-BP model could predict the landslide risk coefficient of large areas more effectively by testing 100 landslide data in Sichuan Province, China, [25]. Hu et al. proposed a prediction model using ACO optimized BP neural network to predict the production increase effect of oil field development accurately. The experimental results showed that the model effectively predicted the production increase effect of oil fields, [26]. Li et al. proposed a battery state of charge estimation based on gray wolf optimization (GWO) and BP neural network for the problem of inaccurate battery state of charge (SOC) estimation of lithium batteries model, which has a higher SOC estimation accuracy and a minor relative error compared with the traditional BP neural network, [27]. Wen et al. developed a novel BP neural network model based on PSO to forecast the carbon dioxide emissions and made some improvements to PSO to improve the accuracy of the forecast. Finally, the validity of the model was verified using panel data of the Chinese commercial sector from 1997 to 2017, [28].

However, these standard intelligent optimization algorithms decline in species diversity in the late iterations and tend to fall into local optima, which may fail to search for optimal weights and thresholds when guiding BP neural networks to adjust network parameters, thus failing to achieve the best prediction.

2 Sparrow Search Algorithm and its Improvement

In 2020, a novel sparrow search algorithm (SSA) was proposed by Xue et al. [29]. The authors conducted comparative experiments with 19 sets of functions and demonstrated that SSA has high search accuracy and fast convergence. However, like other population intelligence optimization algorithms, SSA still suffers from the problem of reduced population diversity and tends to fall into local optimum when it iterates to a later stage. Based on the standard SSA algorithm, we design a dynamic discoverer strategy that can adjust the proportion of discoverers according to the number of iterations. We also introduce adaptive t-distribution changes to improve the algorithm's global exploitation capability in the early stage and local search capability in the later stage. Meanwhile, after the sparrow search is completed, we employ a random wandering strategy to perturb the sparrow population and prevent the algorithm from falling into local optimum.

2.1 Standard Sparrow Search Algorithm

The sparrow search algorithm simulates the behavior of sparrows to find food and resist being predated. The roles of sparrows can be divided into three categories: discoverers, followers, and scouts. Finders are adaptable, able to search for food in a large area and guide the food direction for followers. To increase the success of predation, followers will follow the discoverers to forage, while some followers will watch the discoverers and compete with them for food or forage around them. When the sparrow population recognizes the danger, it immediately goes on alert and updates its position.

Assuming a population of N sparrows foraging in the m -dimensional search space, the position of the i -th sparrow can be expressed as X_i , Where $i = 1, 2, \dots, N$.

First, the discoverers in the population update the position by Eq. (1):

$$x_{im}^{r+1} = \begin{cases} x_{im}^r \cdot \exp\left(\frac{-i}{\alpha \cdot T}\right) & R_2 < ST \\ x_{im}^r + Q \cdot L & R_2 \geq ST \end{cases} \quad (1)$$

where r represents the current number of iterations, T represents the maximum number of iterations, α is a locally distributed random number in the range of (0,1), Q is a random number that obeys a normal distribution, and L represents a matrix with 1 row and m columns and all elements are 1. R_2 represents the warning value, $R_2 \in [0, 1]$, ST stands for the safety value, $ST \in [0.5, 1]$, x_{im}^r stands for the position of the i th sparrow in the m -th dimension at the r -th iteration.

Except the discoverers, all the sparrows are followers, and their positions are updated by Eq. (2):

$$x_{im}^{r+1} = \begin{cases} Q \cdot \exp\left(\frac{x_w^r - x_{im}^r}{i^2}\right) & i > \frac{n}{2} \\ x_b^{r+1} + |x_{im}^r - x_b^{r+1}| \cdot A^T(AA^T)^{-1} & i \leq \frac{n}{2} \end{cases} \quad (2)$$

where x_w^r represents the worst position of the sparrow in the current dimension at the r -th iteration, x_b^{r+1} represents the optimal position of the sparrow in the current dimension at the $r + 1$ -th iteration, and A represents a 1-row and m -column matrix with elements of 1 or -1 .

In addition, the sparrow population will randomly generate scouts, generally accounting for 10% to 20% of the entire population, and their locations are updated by Eq. (3):

$$x_{im}^{r+1} = \begin{cases} x_b^r + \beta (x_{im}^r - x_b^r) & f_i \neq f_g \\ x_{im}^r + K \cdot \left(\frac{x_{im}^r - x_w^r}{|f_i - f_w| + \gamma}\right) & f_i = f_g \end{cases} \quad (3)$$

where β is a normally distributed random number with a mean value of 0 and a variance of 1, which represents the step size control parameter, K is a random number between $[-1,1]$, f_i represents the fitness of the i -th sparrow, f_g, f_w represents the optimal fitness and the worst fitness of the current sparrow population, and γ is a very small constant to avoid the situation where the denominator is 0 when $f_i=f_g$.

From the algorithmic process of SSA, it can be seen that each update of sparrow position is based on the position of the last sparrow, which will lead to the algorithm may fall into local optimum when the population size decreases in the late iteration, and the optimal weights and thresholds cannot be

searched when guiding the BP neural network to adjust the network parameters, thus affecting the prediction accuracy of the model.

2.2 Multi-strategy Improved Sparrow Search Algorithm

In this section, we design a dynamic discoverers strategy, adopt an adaptive t-distribution policy and a random wandering policy to improve the performance of SSA, and the framework of multi-strategy sparrow search algorithm (MSSA) is Algorithm 1.

2.2.1 Dynamic Discoverers Strategy

The proportion of discoverers affects the search capability of the algorithm. Since the proportion of discoverers in the standard SSA is a fixed value, it does not adapt well to the changes in the iterative process. Therefore, as shown in Eq. (4), we design a dynamic discoverer strategy, which can dynamically adjust the proportion of discoverers according to the number of iterations. A large number of discoverers at the early stage of operation effectively improves the overall development ability of the algorithm at the beginning of the operation, and a large number of followers at the later stage can improve the local optimization ability of the algorithm.

$$PD = PD_{start} - \frac{|PD_{start} - PD_{end}|}{T} \times r \quad (4)$$

where PD is the proportion of discoverers, PD_{start} and PD_{end} are PD ' initial and final values, respectively. The final number of discoverers is determined by multiplying the population size and PD and rounding.

2.2.2 Adaptive T-Distribution Strategy

The t-distribution is also called the student distribution, [30], and its curve shape is related to the size of the degree of freedom parameter n . When $n = 1$, the t-distribution is a Cauchy distribution, and $n \rightarrow \infty$, the t-distribution is a Gaussian distribution. The adaptive t-distribution combines the characteristics of the Cauchy and Gaussian distributions, so we assign the current number of iterations to its degree of freedom parameter n . In the initial stage of the algorithm operation, the t-distribution approximates the Cauchy variant because of the small value of n . In the later stage of the algorithm operation, the t-distribution approximates the Gaussian distribution because of the significant value of n , which is conducive to enhancing the algorithm's merit-seeking ability.

In this paper, the adaptive t-distribution is used to vary the sparrow position and update it by Eq. (5):

$$x'_i = x_i + x_i \times t(r) \quad (5)$$

where x'_i is the sparrow position after variation, x_i is the position of the i -th sparrow individual, and $t(r)$ is the t-distribution with the current number of iterations as the degree of freedom parameter.

2.2.3 Random Wandering Strategy

In SSA, sparrows are based on the position of the previous generation and update the position by updating the equation, which makes the algorithm easily fall into the local optimum during the iteration. We introduce the random wandering strategy to improve the searchability of the sparrow population by using random wandering to perturb the sparrow population after the sparrow search is completed, [31]. At the beginning of the beginning iteration, the boundaries of random wandering

are more significant, which improves global searchability. After many iterations, the boundaries of wandering become smaller, improving the local searchability.

The process of the random walk can be expressed mathematically as Eq. (6):

$$Y(t) = [0, \text{cussum}(2r(t_1) - 1), \dots, \text{cussum}(2r(t_n) - 1)] \quad (6)$$

Among them, $Y(t)$ is the set of steps of random wandering, cussum is the cumulative sum of calculations, t is the number of steps of random wandering (this article takes the maximum number of iterations), and $r(t)$ is a random function defined as Eq. (7), where rand is a random number of $[0,1]$.

$$r(t) = \begin{cases} 1, & \text{rand} > 0.5 \\ 0, & \text{rand} \leq 0.5 \end{cases} \quad (7)$$

Eq. (6) cannot be used directly to update the position of the sparrow due to the presence of boundaries in the feasible domain. According to Eq. (8), normalization is required to ensure that the random wandering is within the feasible range.

$$X_m^r = \frac{(X_m^r - a_m) \times (d_m^r - c_m^r)}{b_m - a_m} + c_m^r \quad (8)$$

where a_m is the minimum random walk of the m -dimensional variable; b_m is the maximum random walk of the m -dimensional variable; c_m^r is the minimum of the m -dimensional variable at the r -th iteration; d_m^r is the m -th dimension variable in the r th iteration the maximum value of r iterations.

Algorithm 1: The Framework of the Multi-strategy Sparrow Search Algorithm (MSSA)

Input:

T : the maximum iterations
 PD_{start} : the start ratio of discoverers
 PD_{end} : the end ratio of discoverers
 SD : the proportion of sparrows that perceive danger
 R_2 : the alarm value
 N : the total number of sparrows
 Initialize the population of sparrows and set the parameters.

Output: X_b, f_g .

- 1: **if** ($r < T$)
- 2: Update the PD by Eq. (4);
- 3: $PD_{Number} = \text{round}(PD \times N)$;
- 4: Calculate the fitness and record the best and worst individuals.
- 5: **for** $i = 1:PD_{Number}$
- 6: Update the sparrow position by Eq. (1);
- 7: **end for**
- 8: **for** $i = (PD_{Number} + 1):N$
- 9: Update the sparrow position by Eq. (2);
- 10: **end for**
- 11: **for** $i = 1:SD$
- 12: Update the sparrow position by Eq. (3);
- 13: If the new location is better than the previous one, update it;

(Continued)

Algorithm 1: Continued

```

14: end for
15: for  $i = 1:N$ 
16:   Update the sparrow position by Eq. (5);
17:   If the new location is better than the previous one, update it;
18: end for
19: Update the position of the optimal sparrow by Eqs. (6)–(8);
20: If the new location is better than the previous one, update it;
21:  $r = r + 1$ 
22: end if
23: return  $X_b, f_g$ .
```

3 Multi-strategy Sparrow Search Algorithm Optimizes BP Neural Network

Based on the proposed algorithm, this paper designs the MSSA-BP model. In this part, we conduct simulation experiments on nine benchmark functions to verify the algorithm's search performance and select the offshore wind farm dataset in the western Gulf of Mexico and the air quality dataset in Beijing, China, for the SSA-BP, MSSA-BP model, and DPSO-BP model to verify the effectiveness of the models.

Firstly, we determine the three-layer structure of the BP neural network, as shown in Fig. 1, and determine the number of input layer neurons and output layer neurons according to the number of inputs and outputs. Secondly, the range of the number of neurons in the hidden layer is determined according to the empirical Eq. (9). The optimal number of neurons in the hidden layer is finally determined by comparing the MSE of the predicted and actual values of the training set corresponding to different numbers of neurons in the hidden layer.

$$h = \sqrt{n_1 + n_2} + v \quad (9)$$

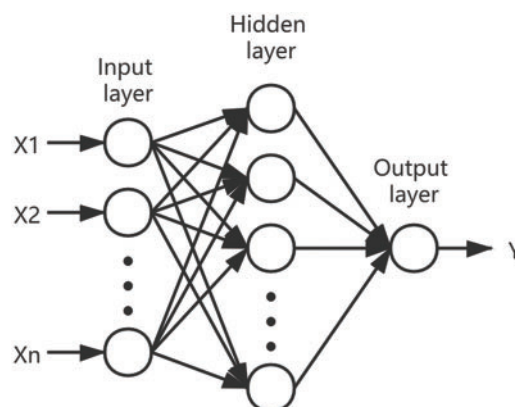


Figure 1: The topological structure of the three-layer BP neural network model

The MSE generated by each network training set is used as an approximate fitness function to calculate the fitness value, and the MSE is as in Eq. (10).

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y'_i - Y_i)^2 \quad (10)$$

In Eq. (8), Y_i and Y'_i are the target and predicted values, respectively. The smaller the MSE, the more accurate the model.

Fig. 2 shows the running process of MSSA in the BP neural network.

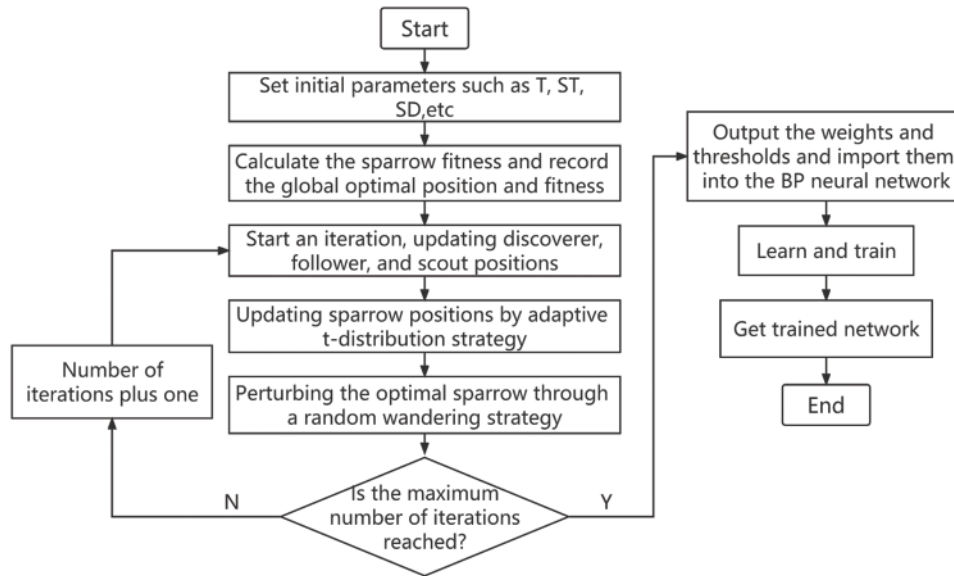


Figure 2: MSSA-BP flow chart

Step 1: Set initial parameters: population size, the maximum number of iterations, warning value ST, the proportion of scouts SD, etc.

Step 2: Initialize the population.

Step 3: The fitness f_i of each sparrow is calculated by the fitness function and then sorted, select the current optimal fitness f_g and its corresponding position X_b , and the current worst fitness f_w and its corresponding position X_w .

Step 4: Determine the current PD value according to Eq. (4), select the sparrow with high adaptability as the discoverers, and the rest as followers, and update the positions of the discoverers and followers according to Eqs. (1) and (2). Randomly select a part of sparrows from the sparrow population for reconnaissance and early warning, and update their positions according to Eq. (3).

Step 5: After one iteration is completed, the fitness value f_i of each sparrow is recalculated. According to Eq. (4), the sparrow is mutated. If the sparrow is better than the one before the mutation, the previous sparrow will be replaced by the mutated sparrow. Otherwise, it will remain unchanged.

Step 6: According to the current state of the sparrow population, update the optimal position X_b and its fitness f_g experienced by the entire population, as well as the worst position X_w and its fitness f_w .

Step 7: Perturbation of the optimal sparrow employing a random wandering strategy. If the scrambled sparrow is better than the previous one, replace the previous sparrow with the scrambled one and update f_g . Otherwise, leave it unchanged.

Step 8: Determine if the algorithm has reached the maximum number of iterations. If the condition is met, the loop ends, and the optimization result is output; otherwise, return to step 5.

Step 9: The obtained optimal weights and thresholds are assigned to the BP neural network for training and learning.

4 Experiment and Analysis

4.1 Algorithm Performance Comparison Analysis

In order to verify the performance of the MSSA algorithm, we selected nine benchmark functions for testing, as shown in Tab. 1, where F1~F3 are high-dimensional single-peak functions, F4~F6 are high-dimensional multi-peak functions, and F7~F9 are low-dimensional functions, to thoroughly investigate the merit-seeking ability of MSSA through different types of benchmark functions.

Table 1: Benchmark functions

Function	Dimension	Range	Optimal value
$F_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10, 10]$	0
$F_3 = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	$[-100, 100]$	0
$F_4(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	0
$F_5(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30	$[-32, 32]$	0
$F_6(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0
$F_7(x) = \sum_{i=1}^{11} \left(a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right)^2$	4	$[-5, 5]$	0.000307
$F_8(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2\right)$	3	$[0, 1]$	-3.86
$F_9(x) = -\sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.5363

We tested the MSSA, SSA, and GWO algorithms under Intel(R) Core(TM) i5-8500 CPU @3.00 GHz, 8.00 GB of RAM, Windows 10 and Matlab R2018b environment, and the parameters of each algorithm were set in Tab. 2. The population size N for each algorithm was 30 and the maximum number of iterations $T = 100$.

Table 2: Algorithm parameter value

Algorithm	Parameter
GWO	$a_{max} = 2, a_{min} = 0$
SSA	$PD = 0.2, ST = 0.8, SD = 0.2$
MSSA	$PD_{start} = 0.8, PD_{end} = 0.2, ST = 0.8, SD = 0.2$

In order to avoid chance and test the performance of the algorithm more accurately, we run each group of experiments independently ten times, and finally, take the optimal value, the average value, and the standard deviation as the evaluation index, where the optimal value and the average value reflect the algorithm's ability to find the best and accuracy, and the standard deviation reflects the robustness of the algorithm. The experimental results are shown in [Tab. 3](#), and the algorithm's convergence is shown in [Fig. 3](#).

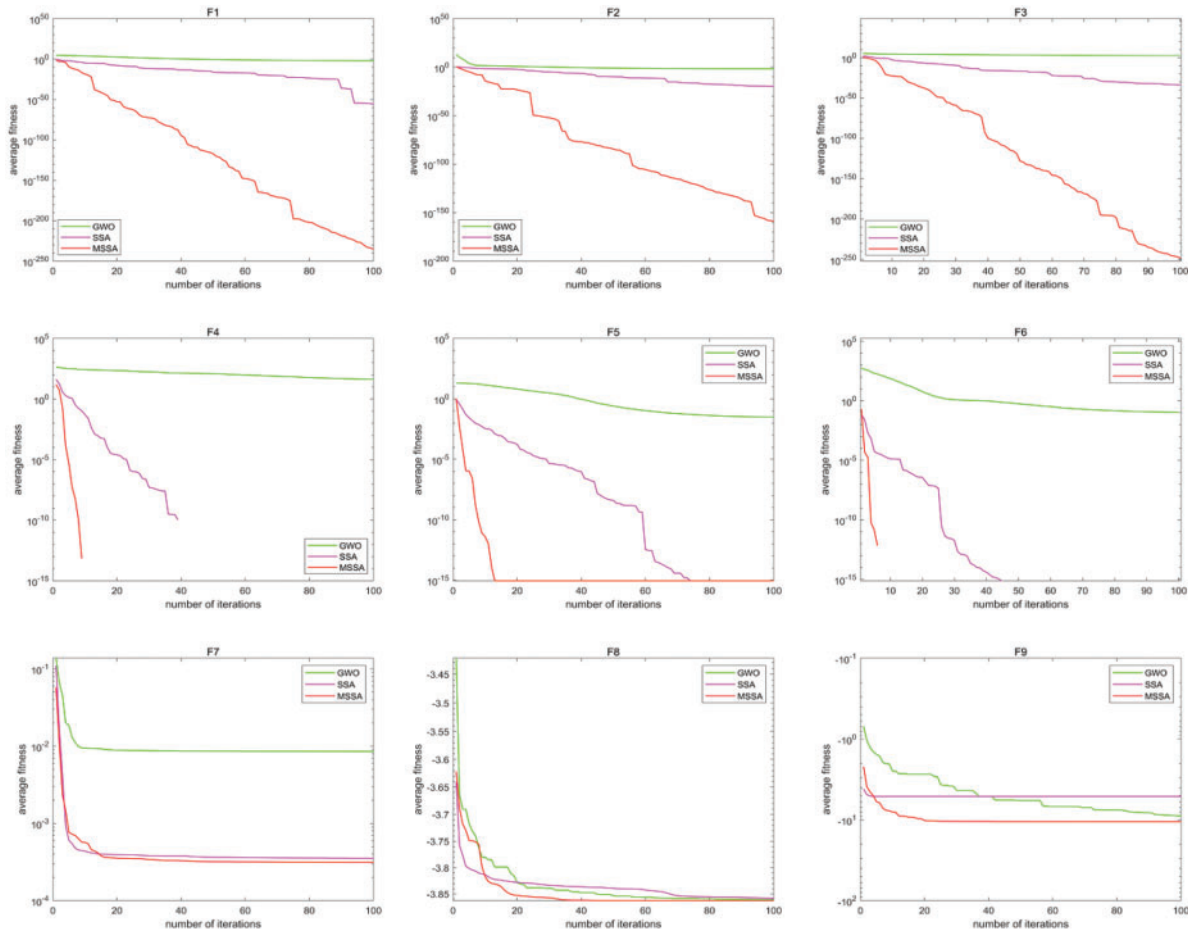
Table 3: Comparison of test results of three algorithms

Function		GWO	SSA	MSSA
F1	Best	2.88E-03	0.00E+00	0.00E+00
	Avg	1.03E-02	2.70E-56	4.52E-236
	Std	8.03E-03	5.40E-56	0.00E+00
F2	Best	1.58E-02	7.89E-127	3.19E-254
	Avg	2.12E-02	1.92E-20	9.98E-160
	Std	4.08E-03	3.84E-20	0.00E+00
F3	Best	7.90E+01	3.24E-84	0.00E+00
	Avg	2.52E+02	1.91E-34	4.95E-249
	Std	1.94E+02	3.81E-34	0.00E+00
F4	Best	2.66E+01	0.00E+00	0.00E+00
	Avg	4.33E+01	0.00E+00	0.00E+00
	Std	1.33E+01	0.00E+00	0.00E+00
F5	Best	2.19E-02	8.88E-16	8.88E-16
	Avg	3.22E-02	8.88E-16	8.88E-16
	Std	1.01E-02	0.00E+00	0.00E+00
F6	Best	1.02E-02	0.00E+00	0.00E+00
	Avg	1.09E-01	0.00E+00	0.00E+00
	Std	8.77E-02	0.00E+00	0.00E+00
F7	Best	4.97E-04	3.10E-04	3.08E-04
	Avg	8.54E-03	3.52E-04	3.13E-04
	Std	9.65E-03	4.21E-05	7.25E-06
F8	Best	-3.86E+00	-3.86E+00	-3.86E+00
	Avg	-3.86E+00	-3.86E+00	-3.86E+00
	Std	2.30E-03	5.83E-03	9.23E-07

(Continued)

Table 3: Continued

Function		GWO	SSA	MSSA
F9	Best	$-1.05E+01$	$-5.13E+00$	$-1.05E+01$
	Avg	$-8.96E+00$	$-5.13E+00$	$-1.05E+01$
	Std	$3.05E+00$	$4.85E-10$	$5.38E-10$

**Figure 3:** Convergence graph of the benchmark functions

According to the results in Fig. 3 and Tab. 3, for the high-dimensional single-peaked functions F1, F2, and F3, MSSA finds the optimal values and has multiple orders of magnitude improvement in mean and standard deviation compared with other algorithms. The convergence speed of MSSA is also significantly better than other algorithms, which indicates that MSSA has better speed, accuracy, and robustness in finding the optimal values. For the high-dimensional multi-peaked functions F4, F5 and F6, both SSA and MSSA can converge to the best accuracy. However, MSSA can obtain the optimal solution within 20 iterations, effectively avoiding falling into a local optimum and outperforming the other algorithms. For the low-dimensional functions F7, F8 and F9, MSSA slightly improves

the optimality-seeking accuracy. However, the but the robustness of this algorithm is significantly improved compared with GWO, and the standard deviation is slightly improved compared with SSA.

In summary, MSSA outperforms other intelligent algorithms in the search performance on high-dimensional single-peaked functions, high-dimensional multi-peaked functions, and low-dimensional functions, especially in solving high-dimensional functions with a considerable improvement. The dynamic discoverers strategy, adaptive t-distribution, and random wandering strategy effectively enhance the global and local optimization-seeking ability, and to a certain extent, prevent the algorithm from falling into local optimum during operation, which makes MSSA have excellent performance in the speed, accuracy, and robustness of optimization-seeking.

4.2 Comparative Analysis of Model Performance

This study uses two datasets to compare the three models, SSA-BP, MSSA-BP, and DPSO-BP: an offshore wind farm dataset located in the western Gulf of Mexico and an air quality dataset from Beijing, China. In the wind farm dataset, considering that wind power generation is influenced by wind and has little dependence on factors such as humidity and temperature, wind direction and wind speed are chosen as input variables for the model, and power generation (MW) is used as the prediction target. We took 1009 consecutive data samples from Dec. 25–Dec. 31, 2012, with 10-min intervals for each sample, 709 data were randomly selected for training, with the remaining 300 data used for testing. In the air quality dataset, we selected six input variables, including 24-h average delicate particulate matter (PM_{2.5}), inhalable particulate matter (PM₁₀), ozone (O₃), NO₂, CO, and SO₂ [32–36], air quality index (AQI) as the prediction target. We collected the air quality dataset of Beijing from 2018 to 2019. There were 685 data after excluding invalid data, of which 485 data were randomly selected for training, and the remaining 200 were used for testing.

In order to evaluate the prediction performance of the model, three error metrics were chosen to analyze and evaluate the model in this study: RMSE, MAE, and MAPE. To avoid chance and verify the performance of the model more accurately, we ran each group of experiments ten times independently and finally took the average value as the experimental data.

For both datasets, Fig. 4 shows that the MSSA-BP model has the best initial adaptation and is basically ahead of SSA-BP and DPSO-BP in convergence speed and accuracy, The average best adaptation achieved in the end is still better than theirs. As shown in Tab. 4, the average MAE and RMSE of MSSA-BP are lower than those of SSA-BP and DPSO-BP. The average MAPE of MSSA-BP is 0.72% lower than that of SSA-BP and 0.61% lower than that of DPSO-BP in the wind farm data set, and the average MAPE of MSSA-BP is 0.72% lower than that of SSA-BP in the air quality data set. BP by 1.30% and by 0.82% compared to DPSO-BP.

In summary, we conclude the following:

- (1) The performance of MSSA is significantly improved compared with SSA, and the dynamic discoverers strategy, adaptive t-distribution, and random wandering strategy are effective in improving the performance of SSA.
- (2) We propose to use MSSA to optimize the weights and thresholds of BP neural networks, and the designed MSSA-BP model can effectively improve the prediction performance.
- (3) Through benchmark function testing and model validation, MSSA remains stable during the iterative process, proving that MSSA has strong robustness.

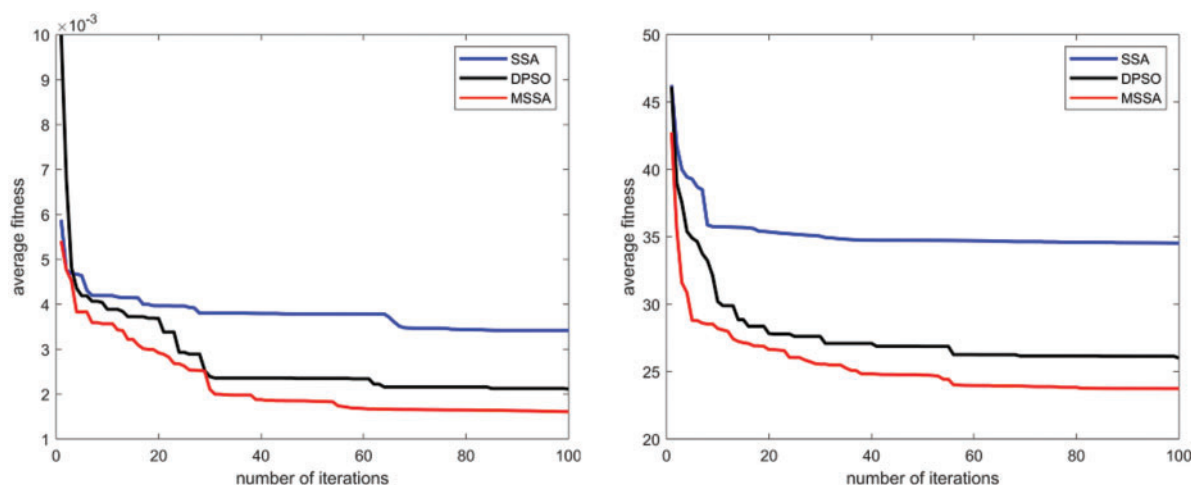


Figure 4: Fitness curve (Left: Wind farm data sets; Right: Air quality data sets)

Table 4: Experimental results of three sets of models

Dataset	Index	SSA-BP	DPSO-BP	MSSA-BP
Wind farm dataset	MAE	0.0487	0.0382	0.0326
	RMSE	0.0665	0.049	0.0426
	MAPE	1.76%	1.65%	1.04%
Air quality dataset	MAE	5.063	4.582	4.106
	RMSE	9.326	7.914	7.529
	MAPE	6.74%	6.26%	5.44%

5 Conclusion

This research proposes a BP neural network prediction method based on a multi-strategy improved sparrow search algorithm. Based on the standard SSA, we design a dynamic discoverers strategy, which uses adaptive t-distribution to mutate the sparrow, and uses a random walk strategy to sparrow perturbs and improve the algorithm's performance. The MSSA-BP prediction model was also designed to improve the prediction performance based on the MSSA algorithm. The superiority-seeking ability of MSSA is demonstrated by testing nine different types of benchmark functions. The results of simulation experiments with two data sets show that the average MAE, average RMSE, and average MAPE of MSSA-BP are better than the comparison model, proving that MSSA-BP has better prediction accuracy and robustness. The contributions of our research to engineering practice are as follows:

- (1) The MSSA proposed in this paper has the advantages of fast convergence speed, high convergence accuracy and good stability.
- (2) In this paper, we tested different benchmark functions and proved that MSSA has strong optimization ability and can be applied to more scenarios with special application value.
- (3) The comparative experiments of the two data sets demonstrate that the model proposed in this paper has better prediction performance and wider applicability.

Of course, MSSA-BP still has some shortcomings, and its more complex structure leads to slower running time than SSA-BP, which is an area for improvement. Meanwhile, the intelligent algorithm inevitably encounters the optimal localization problem due to NP theory, which remains a challenge. Our future research may incorporate engineering practice problems.

Funding Statement: This work was supported by the National Natural Science Foundation of China (Grant No. 62162024 and 62162022), Key Projects in Hainan Province (Grant ZDYF2021GXJS003 and Grant ZDYF2020040), the Major science and technology project of Hainan Province (Grant No. ZDKJ2020012).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] X. R. Zhang, X. Sun, W. Sun, T. Xu and P. P. Wang, "Deformation expression of soft tissue based on BP neural network," *Intelligent Automation & Soft Computing*, vol. 32, no. 2, pp. 1041–1053, 2022.
- [2] X. R. Zhang, X. Sun, X. M. Sun, W. Sun and S. K. Jha, "Robust reversible audio watermarking scheme for telemedicine and privacy protection," *Computers Materials & Continua*, vol. 71, no. 2, pp. 3035–3050, 2022.
- [3] D. Zhang, J. Hu, F. Li, X. Ding, A. K. Sangaiah *et al.*, "Small object detection via precise region-based fully convolutional networks," *Computers Materials & Continua*, vol. 69, no. 2, pp. 1503–1517, 2021.
- [4] T. Yang, C. Wang, T. Zhou, Z. Cai, K. Wu *et al.*, "Leveraging active decremental TTL measuring for flexible and efficient NAT identification," *Computers Materials & Continua*, vol. 70, no. 3, pp. 5179–5198, 2022.
- [5] J. R. Cheng, J. X. Liu, X. B. Xu, D. W. Xia, L. Liu *et al.*, "A review of Chinese named entity recognition," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 6, pp. 2012–2030, 2021.
- [6] F. F. Lei, J. R. Cheng, Y. Yang, X. T. Tang, V. S. Sheng *et al.*, "Improving heterogeneous network knowledge transfer based on the principle of generative adversarial," *Electronics*, vol. 10, no. 13, pp. 1525, 2021.
- [7] X. Y. Tang, W. X. Tu, K. Q. Li and J. R. Cheng, "DFNet: An IoT-perceptive dual feature fusion network for general real-time semantic segmentation," *Information Sciences*, vol. 565, pp. 326–343, 2021.
- [8] J. R. Cheng, Y. Yang, X. Y. Tang, N. X. Xiong, Y. Zhang *et al.*, "Generative adversarial networks: A literature review," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 12, pp. 4625–4647, 2020.
- [9] X. Shao, "Accurate multi-site daily-ahead multi-step pm2.5 concentrations forecasting using space-shared cnn-lstm," *Computers Materials & Continua*, vol. 70, no. 3, pp. 5143–5160, 2022.
- [10] S. H. Lee and H. Yoe, "Predicting net income for cultivation plan consultation," *Journal of Information and Communication Convergence Engineering*, vol. 18, no. 3, pp. 167–175, 2020.
- [11] P. Wadhwa, A. Tripathi, P. Singh, M. Diwakar and N. Kumar, "Predicting the time period of extension of lockdown due to increase in rate of COVID-19 cases in India using machine learning," *Materials Today: Proceedings*, vol. 37, pp. 2617–2622, 2021.
- [12] H. Kwon, K. C. Oh, Y. G. Chung, H. Cho and J. Kim, "Development of machine learning model for predicting distillation column temperature," *Applied Chemistry for Engineering*, vol. 31, no. 5, pp. 520–525, 2020.
- [13] Z. Li, and X. Zhao, "BP artificial neural network based wave front correction for sensor-less free space optics communication," *Optics Communications*, vol. 385, pp. 219–228, 2017.
- [14] M. Zhang, "Application of BP neural network in acoustic wave measurement system," *Modern Physics Letters B*, vol. 31, pp. 19–21, 2017.
- [15] J. D. Pan, J. N. Liang, T. F. Sun, H. Wang, G. Xie *et al.*, "Optimization of rotor position observer with BP neural network," in *2019 22nd Int. Conf. on Electrical Machines and Systems (ICEMS)*, Harbin, China, pp. 1–6, 2019.

- [16] G. T. Ren, "Application of neural network algorithm combined with bee colony algorithm in English course recommendation," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 5307646, 2021.
- [17] A. Naik and S. C. Satapathy, "A comparative study of social group optimization with a few recent optimization algorithms," *Complex & Intelligent Systems*, vol. 7, no. 1, pp. 249–295, 2021.
- [18] A. Tharwat and W. Schenck, "A conceptual and practical comparison of PSO-style optimization algorithms," *Expert Systems with Applications*, vol. 167, pp. 114430, 2021.
- [19] W. Li, G. G. Wang and A. H. Gandomi, "A survey of learning-based intelligent optimization algorithms," *Archives of Computational Methods in Engineering*, vol. 28, no. 5, pp. 3781–3799, 2021.
- [20] C. Feng, "Group intelligent optimization algorithm and its evaluation," *Agro Food Industry Hi-Tech*, vol. 28, no. 1, pp. 1084–1088, 2017.
- [21] J. Sang, "Research on pump fault diagnosis based on PSO-BP neural network algorithm," in *2019 IEEE 8th Joint Int. Information Technology and Artificial Intelligence Conf. (ITAIC)*, Chongqing, China, pp. 1748–1752, 2019.
- [22] H. Gong, E. Zhang and J. Yao, "BP neural network optimized by PSO algorithm on ammunition storage reliability prediction," *2017 Chinese Automation Congress (CAC)*, Jinan, China, pp. 692–696, 2017.
- [23] Y. Q. Li, L. Zhou, P. Q. Gao, B. Yang, Y. M. Han *et al.*, "Short-term power generation forecasting of photovoltaic plant based on PSO-BP and GA-BP neural networks," in *Frontiers in Energy Research*, vol. 9, pp. 824691, 2022.
- [24] E. T. Mohamad, D. J. Armaghani, E. Momeni, A. H. Yazdavar and M. Ebrahimi, "Rock strength estimation: A PSO-based BP approach. *Neural Computing and Applications*, vol. 30, no. 5, pp. 1635–1646, 2018.
- [25] C. H. Zhu, J. J. Zhang, Y. Liu, D. H. Ma, M. F. Li *et al.*, "Comparison of GA-BP and PSO-BP neural network models with initial BP model for rainfall-induced landslides risk assessment in regional scale: A case study in sichuan, China," *Natural Hazards*, vol. 100, no. 1, pp. 173–204, 2020.
- [26] H. T. Hu, J. Wu and X. Guan, "Research on ACO-BP based prediction method of the oilfield production stimulation results," in *2020 IEEE 10th Int. Conf. on Electronics Information and Emergency Communication (ICEIEC)*, Beijing, China, pp. 240–243, 2020.
- [27] Z. W. Li, D. Liu, F. Lu, X. D. Heng, Y. D. Guo *et al.*, "Research on SOC estimation of lithium battery based on GWO-BP neural network," in *2020 15th IEEE Conf. on Industrial Electronics and Applications (ICIEA)*, Kristiansand, Norway, pp. 506–510, 2020.
- [28] L. Wen and X. Y. Yuan, "Forecasting CO2 emissions in chinas commercial department, through BP neural network based on random forest and PSO," *Science of the Total Environment*, vol. 718, pp. 137194, 2020.
- [29] J. K. Xue and B. Shen, "A novel swarm intelligence optimization approach: Sparrow search algorithm," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22–34, 2020.
- [30] S. Samal, A. Sarangi and S. K. Sarangi. "Weighted particle swarm optimization with T-distribution in machine learning applications." *Intelligent and Cloud Computing*, Springer, Singapore, vol. 153, pp. 299–307, 2021.
- [31] J. Shen and Y. Li, "The random wander ant particle swarm optimization and random benchmarks," in *2011 Fourth Int. Joint Conf. on Computational Sciences and Optimization*, IEEE, Kunming and Lijiang, Yunnan, China, pp. 200–204, 2011.
- [32] H. L. Zhu and J. L. Hu, "Air quality forecasting using SVR with quasi-linear kernel," in *2019 Int. Conf. on Computer, Information and Telecommunication Systems (CITS)*, Beijing, China, pp. 1–5, 2019.
- [33] K. Gu, J. F. Qiao and W. S. Lin, "Recurrent air quality predictor based on meteorology-and pollution-related factors," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 3946–3955, 2018.
- [34] K. T. Han and L. W. Ruan, "Effects of indoor plants on air quality: A systematic review," *Environmental Science and Pollution Research*, vol. 27, no. 14, pp. 16019–16051, 2020.
- [35] Y. D. Mei, L. Gao, J. W. Zhang and J. H. Wang, "Valuing urban air quality: A hedonic price analysis in Beijing, China," *Environmental Science and Pollution Research*, vol. 27, no. 2, pp. 1373–1385, 2020.
- [36] M. L. Carvour, A. E. Hughes, N. Fann and R. W. Haley, "Estimating the health and economic impacts of changes in local air quality," *American Journal of Public Health*, vol. 108, no. S2, pp. S151–S157, 2018.