ISSN 2616-5775 Vol. 7, Issue 5: 143-151, DOI: 10.25236/AJCIS.2024.070519

Improved BP neural network prediction model based on particle swarm algorithm

Yiwei Huang[1], Xiang Wei[2,*]

[1]*School of Big Data, Qingdao Huanghai University, Qingdao, 266420, China*
[2]*School of Information Engineering, Wuhan Business University, Wuhan, 430056, China*
[*]*Corresponding author: 15271284279@163.com*

Abstract: *In the context of the development of artificial intelligence, machine learning is one of the most important techniques in the field of artificial intelligence. At the same time, more and more practitioners have begun to try to utilize predictive models in machine learning to solve problems in practice. However, as water pollution has become a global environmental problem, traditional water quality prediction methods are difficult to cope with the complex and dynamically changing water environment. This project is dedicated to combining Particle Swarm optimization (PSO) and Back Propagation Neural Network (BPNN) by optimizing the weights and thresholds of the BP neural network, as well as dynamically tuning the parameters to optimize the PSO, and outputting a coupled PSO-BP model, thus overcoming the limitations of traditional water quality assessment methods. It effectively improves the global search ability of the model and avoids falling into local optimization. This method can also be widely used in the fields of environmental monitoring, financial forecasting, medical diagnosis, etc., demonstrating the prospect of wide application of artificial intelligence in solving practical problems.*

Keywords: *BP Neural Network, Particle Swarm Optimization Coupled Model, Water Quality Prediction*

1. Introduction

The advancement of artificial intelligence (AI) technologies has positioned various intelligent and machine learning algorithms at the forefront of numerous domains. Leveraging AI, we gain deeper insights and predictive capabilities regarding water quality changes, enabling proactive measures to enhance water conditions and safeguard our ecosystems. This project harnesses the synergy between PSO and BPNN to refine a water quality prediction model. By utilizing PSO for optimizing the BPNN's weight thresholds, we address challenges such as susceptibility to local optima and slow convergence rates when tackling complex issues. Furthermore, we explore the impact of PSO parameter adjustments on model performance to identify the optimal network parameter configurations. Our goal is to markedly boost the accuracy and efficiency of water quality predictions, offering robust solutions for environmental monitoring and resource management challenges.

Most of the swarm intelligence algorithms are generated by simulating the biological behaviors in nature, searching for the optimal solution in the solution space through the combination of exploration phase as well as development phase, which has been widely used in the field of optimization since its emergence and has achieved good results; and PSO, which is one of the earliest swarm intelligence algorithms proposed, has played a key role in the development of intelligent algorithms and provided a good demonstration for the subsequent research of swarm intelligence algorithms, and based on this, a number of swarm intelligence algorithms are proposed. Based on this, many swarm intelligence algorithms have been proposed and continuously improved in performance: Ant Colony Algorithm (ACO) [1], Artificial Bee Colony Algorithm (ABC) [2], Cuckoo Search Algorithm (CS) [3], Bat Algorithm (BA) [4], Whale Algorithm (WOA) [5], Grey Wolf Algorithm (GWO) [6], Grasshopper Algorithm (GOA) [7], Harris Hawk Algorithm (HHO) [8] and so on. The essence of the Particle Swarm Optimization (PSO) algorithm is to simulate the foraging behavior of birds, consider each bird in the space as a particle that is a candidate for a solution, and find the optimal value of an individual and then find the optimal solution of the group by tracking the particle flights (movement) [9]. The basic idea of PSO algorithm is to make use of the information sharing of each individual in the flock to make the group position evolve from disorder to order in the solution space [10], so as to obtain the optimal solution of the problem.

As one of the most widely used techniques in the 21st century, machine learning is highly favored by scholars. Unlike traditional statistical methods, various algorithms of machine learning show excellent

ISSN 2616-5775 Vol. 7, Issue 5: 143-151, DOI: 10.25236/AJCIS.2024.070519

performance in problems such as classification and prediction, with advantages such as simplicity and ease of use and excellent results. Among many machine learning algorithms, BPNN has become one of the important algorithms in the field of artificial intelligence due to its early development. BPNN is a widely used network model with excellent nonlinear mapping, generalization, and fault-tolerance capabilities [11]. However, the performance of traditional machine learning algorithms is often affected by many parameters, for example, BPNNs have challenges in the selection of initial weights and thresholds, which can easily fall into local optimal solutions, leading to a decrease in prediction accuracy [12]. Therefore, many researchers have optimized and improved the performance of BP neural networks with the help of swarm intelligence algorithms and applied them in various fields. For example, Zhang et al. used Grey Wolf Algorithm (GWO) [13] to optimize the BP neural network to predict the short-term traffic flow; Wang Yudong et al. used the improved Fruit Fly Algorithm (FOA) [14] to optimize the BP neural network to construct a financial crisis early warning model to verify the superiority of the improved algorithm; Ebrahimi et al. used ABCBPNN optimization algorithm to rock blast crushing prediction [15]; and Ghosh et al. used GABPNN algorithm to test materials [16]. It can be seen that how to optimize the key parameters of BP neural network to improve its prediction and classification performance is also one of the hot topics in current research.

In essence, water quality prediction is a critical area closely linked to environmental conservation and public health, attracting substantial research interest. By proposing a PSO-BP algorithm for optimizing weight and threshold parameters more efficiently, this project aims to develop a highly accurate water quality prediction model, contributing to environmental protection efforts.

2. Method

2.1 BP neural network

BP neural network is a widely used supervised learning algorithm for multilayer feedforward neural networks [17]. Since it was proposed by Rumelhart, Hinton, and Williams in 1986, it has become a powerful tool for solving complex nonlinear problems, and has especially excelled in the fields of pattern recognition, speech recognition, and image processing. BP neural network learns iteratively and continuously adjusts the weights and biases in the network in order to minimize the difference between the network outputs and the actual labels.

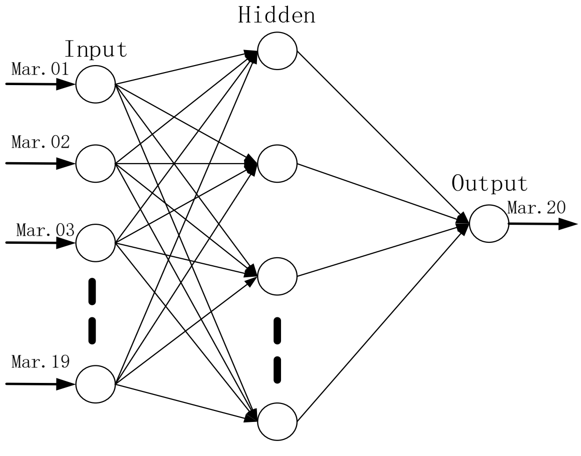


Figure 1: Neural network structure

Figure 1 illustrates the fundamental architecture of a neural network, comprising three distinct layers: the input layer, the hidden layer, and the output layer. The dimensions of the input and output layers are directly determined by the nature of the problem being addressed and the desired outcome, respectively. However, the optimal size of the hidden layer is not straightforwardly defined by any specific criteria and thus requires manual tuning to ascertain the most effective node count. The BP neural network operates through two principal phases: forward propagation and backward propagation. This dual-phase approach renders the BP neural network capable of adaptively modifying weights and biases, thereby enhancing the model's accuracy beyond the capabilities of a standard neural network setup. Through this structured learning process, divided into forward and backward propagation stages, the BP neural network adeptly fine-tunes its parameters to optimize performance.

ISSN 2616-5775 Vol. 7, Issue 5: 143-151, DOI: 10.25236/AJCIS.2024.070519

2.1.1 Basic step

Step 1. For the initialization of weights and biases within the network, it's common practice to start with small, randomly selected values.

Step 2. Forward propagation: input samples are passed forward through the network to compute the output of each layer.

$$\alpha_j^l = \sigma\left(\sum_i w_{ij}^l a_i^{l-1} + b_j^l\right) \tag{1}$$

Where a_j^l is the activation value of the jth node in the lth layer, w_{ij}^l denotes the weight from the ith node in the $l-1$th layer to the jth node in the lth layer, b_j^l denotes the queue value of the$j$th node in the lth layer, and σ denotes the activation function.

Step 3. Calculation error: the error between the predicted value and the true value is calculated at the output layer.

$$\delta_j^L = \frac{\partial C}{\partial a_j^L}\sigma'\left(z_j^L\right) \tag{2}$$

where δ_j^L is the error of the jth node of the output layer, C is the loss function, a_j^L is the activation value of the jth node of the output layer, and $\sigma'(Z_j^L)$ is the derivative of the activation function corresponding to the weighted input z_j^L of the jth node of the output layer.

$$\delta_j^l = \left(\sum_k w_{kj}^{l+1}\delta_k^{l+1}\right)\sigma'\left(z_j^l\right) \tag{3}$$

where δ_j^l is the error of the$j$th node of the lth layer and z_j^l is the weighted input of the jth node of the lth layer.

Step 4. Backpropagating the error: the partial derivatives of the error with respect to each weight are computed using the chain rule, and the error is backpropagated layer by layer from the output layer to the input layer, updating the weights and biases.

$$\frac{\partial C}{\partial w_{ij}^l} = \delta_j^l a_i^{l-1} \tag{4}$$

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l \tag{5}$$

Update the weights and bias using gradient descent:

$$w_{ij}^l = w_{ij}^l - \eta\frac{\partial C}{\partial w_{ij}^l} \tag{6}$$

$$b_j^l = b_j^l - \eta\frac{\partial C}{\partial b_j^l} \tag{7}$$

Where η is the learning rate, which controls the step size of each parameter update.

Step 5. Update cyclically: Continue repeating steps 2 to 4 until the predefined stopping criteria are fulfilled, such as achieving a specific number of iterations or minimizing the error to a satisfactory threshold.

2.2 Particle Swarm Optimization (PSO)

The Particle Swarm Optimization (PSO)[9] technique models the collective behavior observed in natural flocks or swarms, such as birds searching for food, to tackle optimization challenges. Introduced by Eberhart and Kennedy in 1995, PSO draws upon the concept of swarm intelligence. Its simplicity, ease of implementation, and effective global search capabilities make it a popular choice for a broad range of optimization tasks. These include function optimization, neural network optimization, machine learning applications, and various engineering problems. At its heart, PSO views the optimization issue as a search operation within a multidimensional space, where each potential solution is termed a "particle." By iteratively adjusting their positions and velocities, these particles collaborate and share information to navigate towards the globally optimal solution.

2.2.1 Basic step

Step 1. Initialization: Generate a swarm of particles, each representing a point in the search space.

Each particle has a randomly initialized position and velocity.

Step 2. Evaluation: calculates the fitness value for each particle, i.e., evaluates the performance of each particle or the quality of the solution.

Step 3. Update Individual Optimum: for each particle, if the current position is better than the previously recorded Individual Optimum, update the Individual Optimum position for that particle.

Step 4. Update global optimum: find the optimal one from the individual optimal positions of all particles and update it to the global optimal position.

Step 5. Update speed and location:

Speed update formula：

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot rand_1(\,) \cdot \left(pbest_i - x_i(t)\right) + c_2 \cdot rand_2(\,) \cdot \left(gbest - x_i(t)\right) \quad (8)$$

Position update formula:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (9)$$

Where,$v_i(t)$ is the velocity of the particle at time $t$and $x_i(t)$is the position of the particle at time t.$w$is the inertia weight, which controls the continuity of the particle's velocity and helps to balance the global and local search. c_1 and c_2 are the individual learning factor and the social learning factor, which indicate the degree to which the particles are affected by the individual's historical optimal solution and the group's historical optimal solution, respectively. $rand_1()$and $rand_2()$are the random numbers in the range of [0,1] for the algorithm to introduce randomness and increase diversity.$pbest_i$is the particle's historical optimal position.$gbest$is the global optimal position.

Step 6. Termination conditions: repeat steps 2 through 5 until the termination conditions are met (e.g., the maximum number of iterations is reached or the quality of the solution meets specific criteria).

2.2.2 Algorithmic improvements

The success of Particle Swarm Optimization (PSO) hinges on the precise tuning of its parameters: inertia weight (w), personal learning coefficient (c_1), and social learning coefficient (c_2). These settings influence the exploration-exploitation balance, with the inertia weight affecting the exploration scope and the learning coefficients adjusting the search towards personal bests and the global optimum. Improper configurations can lead to inefficient search strategies, either converging prematurely on local optima or wandering aimlessly in the solution space.

This study proposes an adaptive dynamic tuning strategy for PSO parameters to enhance algorithm performance. It employs a threshold-based mechanism (improvement_threshold) to dynamically modulate w, c_1, and c_2 in response to the observed progress towards the global optimum. When significant improvements occur, the strategy decreases w and c_1 to foster global exploration while increasing c_2 to encourage information sharing and collective movement towards the global best. If improvements are lacking, it shifts focus towards local search by increasing w and c_1 and reducing c_2.

This adaptive approach mitigates the need for manual parameter setting in PSO, overcoming challenges of static parameterization that can lead to premature convergence or excessive exploration. By adjusting search strategies in real-time based on performance feedback, it significantly enhances the algorithm's efficiency and broadens its effectiveness across diverse problem landscapes.

2.3 PSO-BP coupled model

The coupling idea of PSO-BP in this project is to combine the particle swarm algorithm (PSO) with the back propagation neural network (BPNN) in order to optimize the BPNN weights and threshold determination. Meanwhile, the introduction of adaptive dynamic parameter tuning to optimize the PSO improves its ability to jump out of the local optimal solution, which further improves the accuracy of the coupled model. In training BPNN, the gradient descent method is used for weight and bias updating, and although it is a commonly adopted method, it has a significant limitation: the method is prone to trap the network in local minima. This problem stems from the mechanism by which the gradient descent method relies on the gradient of the loss function to guide the parameter updates. In this project, the PSO algorithm is used to determine the weights and biases of the network to optimize the parameter selection with its global search capability. The PSO algorithm can effectively avoid the local minimum problem and enhance the global search capability of parameter optimization. Combining the PSO algorithm in the

ISSN 2616-5775 Vol. 7, Issue 5: 143-151, DOI: 10.25236/AJCIS.2024.070519

parameter optimization process of BPNN, it can explore in a wider parameter space in order to find better weights and bias configurations, which can improve the efficiency of network training and the performance of the model. This method not only solves the local minimum problem faced by the gradient descent method, but also provides a more effective and flexible strategy for solving complex nonlinear problems.

2.3.1 Model construction

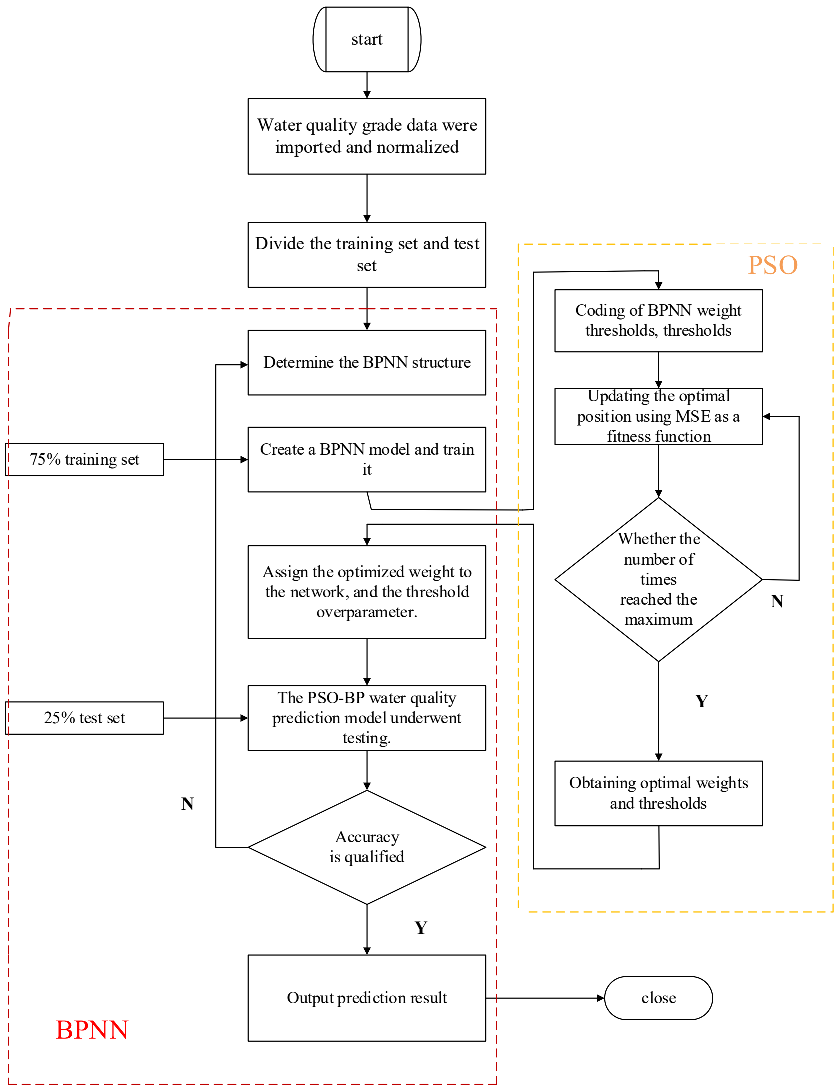


Figure 2: Flowchart of PSO-BP coupling model

As shown in Figure. 2, the steps of the PSO-BP coupled model are:

Step 1. The collected preprocessed data is divided into 25% as a test set to check the prediction accuracy of the model, and the rest is used as a training set to train the network.

Step 2. According to the empirical formula to determine the optimal number of nodes in the hidden layer, and accordingly construct the standard BP neural network to process the data, and get the results of the evaluation system of the standard BP neural network.

Step 3. The PSO algorithm is introduced to obtain the optimal weights and bias setting values for the BP neural network after dynamic tuning optimization.

Step 4. BP neural network construction with optimal weights and bias setting values, output coupled BP neural network evaluation system results.

ISSN 2616-5775 Vol. 7, Issue 5: 143-151, DOI: 10.25236/AJCIS.2024.070519

3. Results

3.1 Data sources

The experimental data for the water quality prediction model came from the national real-time data dissemination system for automatic surface water quality monitoring, collected by Qingyue Data (https://data.epmap.org/page/index), and the monitoring data of the corresponding state-controlled surface water sections in the Yangtze River Basin of Wuhan City, Hubei Province, were specifically selected for the period from January 1, 2023 to January 1, 2024 The final data involve cross sections of Pouring Water, Axe Lake, Hanjiang River, Jinshui, Lianshui, Liangzi Lake, Name Water, Tongshun River, input water, and Yangtze River cross sections, and the initial cross section name, data volume, and the amount of data in the normal state of the site are shown in the chart, which is in the state of maintenance of the site record data are missing in the subsequent data processing process needs to be excluded.

Table 1: Description of data

Name of section	data volume	Number of site normal data	Name of section	data volume	Number of site normal data
Daoshui	3429	3354	Liangzi Lake	9056	8796
Axe Lake	2276	2178	Sheshui	2007	1768
Hanjiang	2273	2143	Tongshun River	3749	3288
Jinshui	1629	1590	Yunshui	3401	3225
Jushui	3011	2861	Yangtze River	4448	4088
aggregate	35279	33291			

3.2 Data preprocessing

The water quality modelling indicator system is given in Table 1 and Table 2:

Table 2: Model Indicator System

Indicator dimension	Indicator name	Content of the indicators	Indicator name	Content of the indicators
Water Quality Prediction Modeling Indicator System	x_1	Water Quality Category	x_7	Total Phosphorus
	x_2	Water Temperature	x_8	Total Nitrogen
	x_3	PH	x_9	Conductivity
	x_4	Dissolved Oxygen	x_{10}	Turbidity
	x_5	HDI	x_{11}	Chlorophyll
	x_6	Ammoniacal	x_{12}	Algae Density

Table 3: Data pre-processed forms

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
3	7.53	8.083	9.835	4.484	0.0307	0.0267	0.829	314.401	4.782	0.00335	946082.2
2	6.96	7.69	10.283	2.533	0.025	0.0225	0.7	197.669	6.505	0.00354	585104.8
3	7.26	7.64	11.136	2.704	0.025	0.0268	1.118	185.02	31.882	0.00347	3531876
3	7.55	7.22	10.788	3.207	0.025	0.0305	0.932	192.69	28.601	0.00282	2041310
......	......	......	......	......	......	......	......	......	......	......	......

In accordance with regulatory guidelines, surface water quality is classified into five levels. Any quality falling below the fifth level is referred to as below-V quality. Specific benchmarks are set for each quality parameter, exemplified by the potassium permanganate values for the five levels: 2, 4, 6, 10, and 15, respectively. A value exceeding 15 signifies a below-V quality rating. To ensure data integrity, entries recorded during maintenance of monitoring stations and certain instances of incomplete data were eliminated. Additionally, a statistical method, the three-standard deviation rule, was utilized to identify and remove anomalies. Due to variations in the scale of data and for ease of analysis, normalization techniques were applied. Consequently, water quality ratings were represented numerically, with below-V quality specifically marked as "6". The original and processed datasets are compiled in Table 3.

ISSN 2616-5775 Vol. 7, Issue 5: 143-151, DOI: 10.25236/AJCIS.2024.070519

3.3 Comparison of model parameter settings and prediction results

3.3.1 Model Parameter Setting

Guided by the established water quality monitoring criteria, eleven specific indicators were meticulously selected to serve as inputs for the BP neural network, thereby determining the number of nodes in the input layer (m = 11). These indicators collectively inform the network about the water quality level, culminating in a singular output node (n = 1) that represents the assessed water quality grade. The configuration of nodes within the hidden layer adheres to a refined empirical formula, encapsulated as (10), where a ranges from 0 to 10:

$$h = \sqrt{m+n} + a \tag{10}$$

By fine-tuning the value of parameter a to ascertain the optimal number of nodes in the hidden layer, rigorous testing on the training set revealed that a configuration of 7 nodes in the hidden layer achieves the lowest mean square error (MSE) of 0.017211 on the test set, marking the most effective setup. This optimal arrangement requires a total of 84 ($m \times h + h \times n$) weights and 8 (h + n) biases, paving the way for optimization of the network's parameters. We divided our dataset, reserving 25% for testing and the remainder for training, employing cross-validation to ensure the robustness of our findings.

In setting up the BP neural network, we established initial parameters, including the number of hidden layers and a target for convergence error, among others. Specifically, we configured the BP neural network training to run 1000 iterations, with a learning rate of 0.01 and a training goal error of 0.00001. We updated the display frequency to every 25 iterations, implemented a momentum factor of 0.01, and set a minimum performance gradient of 1e-6. The system was designed to tolerate up to 6 consecutive training failures before termination.

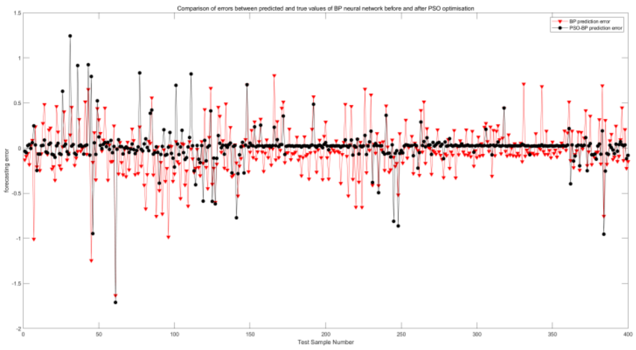
For the PSO algorithm, aimed at further refining the neural network's parameter optimization, we initiated with a population size of 10 and a maximum of 50 evolutionary generations. The dimensionality of the problem was established at 92, accounting for the total number of network parameters to be optimized, allowing each to vary within a range of [-3, 3]. The PSO was configured with an individual learning factor (c_1) of 1.0, a social learning factor (c_2) of 2.0, and an inertia weight (w) of 0.9. Moreover, we set an improvement threshold of 0.01 to assess the optimization process's effectiveness.

This comprehensive methodology underscores our dedication to enhancing the predictive accuracy of our neural network model. Through a blend of conventional and evolutionary optimization strategies, we ensure the model's suitability for addressing the intricacies of water quality prediction.

3.3.2 Comparison of forecast results

Table 4: Experimental results

Algorithmic model	Average absolute error(MAE)	Mean square error(MSE)	Rms error(RMSE)	Mean absolute percentage error(MAPE)
BPNN	0.19234	0.077035	0.27755	7.5301%
PSO-BP	0.10088	0.047083	0.21699	3.8648%



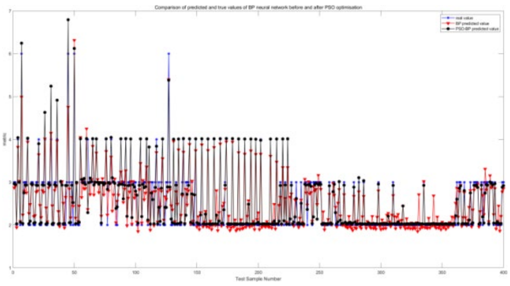


Figure 3: Visualization of results

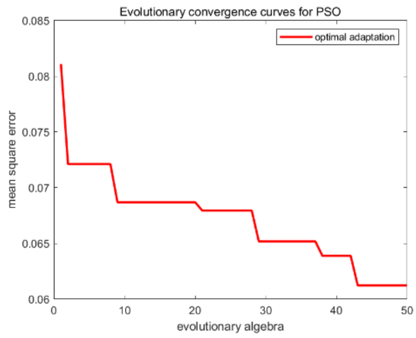


Figure 4: Convergence curve

The results of the evaluation system of BPNN and coupled PSO-BP models are shown in Table 4, which show that MAE, MSE, RMSE, and MAPE are significantly improved, in which the average absolute percentage error is optimized from 7.5301% to 3.8648%. Figure 3 shows the comparison between the predicted and true values of the BP neural network before and after PSO optimization (left) and the error comparison between the predicted and true values of the BP neural network before and after PSO optimization (right), respectively, and it can be seen that the optimized BP neural network of the PSO is closer to the true value, and the error is also more converged to 0.Tosum up, the coupled PSO-BP model improves the prediction accuracy for water quality.

The convergence curve of PSO algorithm is shown in Figure 4. According to the above experimental results, the PSO algorithm after dynamic parametric optimization converges completely around 40 generations, and the overall trend of global search followed by local search is presented, and the parameters can be adjusted in time to make it jump out of the local optimum in the face of the local optimum solution, which avoids converging to the local optimum solution too early.

4. Conclusion

In this research, we innovated within the domain of water quality forecasting by integrating a PSO-BP hybrid model. This model synergizes the strengths of PSO and BPNN to tackle the inherent challenges of neural network-based predictions, particularly those related to optimization of network parameters such as weights and biases. The primary innovation lies in applying PSO to dynamically refine these parameters, thereby circumventing the limitations of traditional gradient descent methods, which are prone to premature convergence and trapping in local optima.

To further enhance the model's efficacy, we introduced an adaptive mechanism for tuning the PSO's intrinsic parameters (w, c_1, c_2), tailored to improve its search capabilities. This adaptive tuning is predicated on threshold values, facilitating a more nuanced and effective search process. This methodological enhancement not only elevates the precision of our water quality forecasts but also endows the PSO with a robustness against common pitfalls like premature convergence.

The experimental validation of our PSO-BP hybrid model demonstrates a marked improvement in predictive accuracy over standard BPNN. This is quantitatively reflected in significant reductions across a suite of error metrics, including mean absolute error, mean squared error, root mean square error, and mean absolute percentage error. Such improvements underscore the utility of our proposed optimizations in refining the model's predictive performance.

Moreover, the adaptive parameter tuning not only bolsters the PSO algorithm's efficiency but also enriches our understanding of its operational dynamics, particularly in how it navigates the optimization

ISSN 2616-5775 Vol. 7, Issue 5: 143-151, DOI: 10.25236/AJCIS.2024.070519

landscape to avoid common obstacles like local optima.

This study, while focused on the specific application of water quality prediction, unveils a broader potential for the PSO-BP hybrid model across various scientific and engineering disciplines. The challenges and shortcomings inherent in traditional neural network applications can potentially be mitigated through this hybrid approach, opening new avenues for research and application. The versatility and improved performance of the PSO-BP model herald promising opportunities for its adoption in diverse fields, ranging from environmental science to engineering and beyond, where predictive modeling plays a critical role. This cross-disciplinary potential invites further exploration and development, aimed at harnessing the full capabilities of hybrid neural network models in complex problem-solving scenarios.

References

[1] Dorigo M, Stützle T. Ant colony optimization: overview and recent advances[J]. Handbook of metaheuristics, 2019: 311-351.

[2] Xiang Wanli, An Meiqing, He Ruichun, et al. Artificial bee colony algorithm based on search ability balance [J]. Computer Engineering and Applications, 2014, 50 (23): 51-55. DOI: 10.3778/j. issn.1002-8331.1305-0076

[3] Lei B , Tao H L , Xu X G .Railway freight volume prediction based on grey neural network with improved particle swarm optimization[J].Journal of Computer Applications, 2012, 32(10):2948-2951.DOI:10.3724/SP.J.1087.2012.02948.

[4] Mirjalili S, Lewis A. The whale optimization algorithm[J]. Advances in engineering software, 2016, 95: 51-67.

[5] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in engineering software, 2014, 69: 46-61.

[6] Saremi S, Mirjalili S, Lewis A. Grasshopper Optimization algorithm: theory and application[J]. Advances in engineering software, 2017, 105: 30-47.

[7] Heidari A A, Mirjalili S, Faris H, et al. Harris hawks optimization: Algorithm and applications[J]. Future generation computer systems, 2019, 97: 849-872.

[8] Su H S , Zhang P J .Research of Distribution Network Reactive Power Optimization Based on Improved Cloud Particle Swarm Optimization BP Neural Network[J].International Journal of Computer and Electrical Engineering, 2013, 5(2):164-168.DOI:10.7763/IJCEE.2013.V5.687.

[9] Wang Wenjing, Li Gang. Optimal resource allocation for database based on adaptive particle swarm [J]. Control Engineering, 2017, 24(7): 1519-1524.

[10] Bei Q, Xuanmin L. A Subgrade Settlement Prediction Model Based on Improved Particle Swarm Neural Network[C]//CIPAE 2020: 2020 International Conference on Computers, Information Processing and Advanced Education.2020.DOI:10.1145/3419635.3419677.

[11] Zhang Yanfan, Wei Hua, Ge Jizhi, et al. Prediction of edible ink viscosity based on genetic algorithm optimized BP neural network[J]. Packaging Engineering, 2021, 42(19): 49-54.

[12] Zhang W. S.,et al. BP neural network model for short-time traffic flow forecasting based on transformed grey wolf optimizer algorithm[J]. Journal of Transportation Systems Engineering and Information Technology, 2020, 20(2): 196.

[13] Wang Yudong,Wang Di,Wang Shanshan. Comparison of financial crisis early warning models based on PSO-BP and FOA-BP neural networks[J]. Statistics and Decision Making, 2018, 34(15):177-179.

[14] Ebrahimi E, Monjezi M, Khalesi M R, et al. Prediction and optimization of back-break and rock fragmentation using an artificial neural network and a bee colony algorithm[J]. Bulletin of Engineering Geology and the Environment, 2016, 75(1): 27-36.

[15] Ghosh S, Dubey A K, Das A K. Numerical Inspection of Heterogeneity in Materials using 2D Heat-Conduction and Hybrid GA-tuned Neural-Network[J]. Applied Artificial Intelligence, 2020, 34(2): 125-154.

[16] Wang Gaofei, Deng Lizhi, Tian Jinxin, et al. Bilateral multi-attribute negotiation solution based on BPNN-GA algorithm [J]. Journal of Wuhan University of Technology: Information and Management Engineering Edition, 2011, 33 (2): 5. DOI: 10.3963/j.issn.1007-144X.2011.02-038

[17] Zhu Jiahao, Zheng Wei, Yang Fengyu, et al. Software quality prediction based on ant colony algorithm optimized backpropagation neural network [J]. Computer Applications, 2023, 43 (11): 3568-3573. DOI: 10.11772/j.issn.1001-9081.2022101600