RESEARCH ON NODE NETWORK TRANSMISSION CAPACITY PREDICTION MODEL FOR LARGE SCALE REMOTE SENSING DATA COLLECTION

Lu Bai 1, Xiaogang Liu 2, Man Zhao 3, Zhibao Wang 2, 4*, Guiying Shi 2

1 School of Computing, Ulster University, Belfast BT15 1ED, UK
2 School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China
3 School of Communication and Electronic Engineering, Qiqihaer University, Qiqihaer 161003, China
4 Bohai-Rim Energy Research Institute, Northeast Petroleum University, Qinhuangdao 066004, China

KEY WORDS: PSO, BP, prediction model, remote sensing data, network transmission capacity.

ABSTRACT:

In recent years, the use of remote sensing technology has grown exponentially in various industries such as agriculture, forestry, and urban planning. Remote sensing data collection systems rely on a network of nodes to collect and transmit data. The transmission capacity of these node networks is a critical factor in the performance and efficiency of the entire system. However, accurately predicting the transmission capacity of a node network can be a challenging task. To carry out large scale open remote sensing data collection, it is necessary to predict the network transmission capacity of nodes in the face of the difference in the execution speed of each node for various tasks. It is necessary to predict the network transmission capacity of nodes. In this research, we propose a node network transmission capacity prediction model for large scale remote sensing data collection using a combination of Particle Swarm Optimization (PSO) and Backpropagation (BP) algorithms. The proposed PSO-BP model aims to accurately predict the transmission capacity of a node network in a remote sensing data collection system. The model is tested and evaluated using a large-scale dataset and the results show that the proposed model outperforms existing models in terms of prediction accuracy. This work contributes to the field of remote sensing data collection by providing a reliable and efficient method for predicting the transmission capacity of node networks.

1. INTRODUCTION

Space remote sensing technology is entering the era of industrial applications, in which Landsat, Sentinel and other public interest satellite remote sensing data play an important role (Hemati et al., 2021; Phiri et al., 2020; Segarra et al., 2020). With the development of remote sensing technologies and advancement of deep learning-based algorithms, significant progress has been achieved in recent years in many remote sensing tasks, for example, object detection from remote sensing images (Cheng and Han, 2016; Deng et al., 2018; Li et al., 2020; Qian et al., 2020; Song et al., 2020; Wang et al., 2021; Zhang et al., 2019; Zhu et al., 2021), remote sensing change detection (Hecheltjen et al., 2014; Jensen and Im, 2007; Khelifi and Mignotte, 2020; Shi et al., 2022; Q. Wang et al., 2018; Zhang et al., 2021) and remote sensing big data (Chi et al., 2016; Deren et al., 2014; Liu et al., 2018; Ma et al., 2021, 2015; Yu et al., 2021a, 2021b). Remote sensing data user groups such as individuals, small and medium-sized enterprises are gradually increasing. In the process of data application, satellites are generally publicly released on the Internet for global users, which is limited by the service capability of the data source itself and the acquisition capability of data user nodes. Due to the constraints of data source’s own service capability and data user node acquisition capability, the problems of low efficiency of large-scale public remote sensing data acquisition and low utilization rate of user acquisition nodes still exist (Lee et al., 2011). The problems such as low efficiency of large-scale public remote sensing data collection and low utilization rate of user collection nodes still exist (Wang et al., 2022).

However, there are few studies on network node transmission capacity prediction. To carry out large scale open remote sensing data collection in crowdsourcing mode, it is necessary to predict the network transmission capacity of nodes considering the difference in the execution speed of each node for various tasks. The traditional Backpropagation (BP) algorithm uses the inertia weights between the input and hidden layers and the inertia weights between the hidden layer and the output layer to establish the prediction model. The inertia weights between the hidden layer and the output layer are randomly initialized, and the learning factors are randomly generated constants. The algorithm has a strong sensitivity to these coefficients during the training process, and it is easy to fall into long iteration time and local extremes. This study analyses the possible factors affecting the transmission speed of the node network at the levels of data source, transmission medium and receiving terminal of open remote sensing data collection. A range of controllable factors were selected to establish a prediction model of network transmission capacity. For the traditional BP neural network algorithm, the inertia weights of network algorithm, it is easy to fall into local minima and other shortcomings. The Particle Swarm Optimization (PSO) algorithm with good robustness and global search capability is combined with the BP neural network algorithm to propose an improved PSO-BP algorithm. The improved algorithm includes the introduction of gradient, momentum factor, and inertia weight adjustment function. To further improve the accuracy of the algorithm in the iteration, the accuracy dynamic adjustment function is introduced, and the learning factor is also introduced into dynamic adjustment function. Finally, a comparison experiment between the
improved PSO-BP algorithm and the traditional BP algorithm is conducted on a real data set collected from Sentinel 2.

2. NODE NETWORK TRANSMISSION CAPACITY PREDICTION MODEL

2.1 Analysis of factors affecting the transmission capacity of node networks

This paper considers the data transmission in large-scale public remote sensing data collection under the crowdsourcing model. During the collection process, data is transmitted from the public remote sensing data source through multiple transmission media to the collection node. The data transmission process mainly involves “two sources and one medium”, which are the data source, the transmission medium, and the receiving terminal. The paper analyses the characteristics of the three stages of factors affecting node network transmission capacity, and Figure 1 shows the factors affecting node network transmission capacity.

![Diagram: Factors affecting node network transmission capacity](image)

**Figure 1.** The factors that affect the transmission ability of nodes in a network

2.1.1 Constraint analysis based on the data source side

The main factors that affect the transmission capacity of a node network at the data source are related to the physical configuration and server parameter settings of the data source server.

In order to ensure the long-term stability and security of the data source, the number of concurrent accesses, IPs, access frequencies, access bandwidth speeds, allowed access time periods, and allowed access areas of users are usually restricted based on the physical configuration of the server node.

Through long-term collection, it has been found that when a single user frequently accesses the USGS data source for a long period of time, the user’s order production speed is relatively slow, and the data download speed in the morning is much faster than that in the afternoon. The data source is closed for maintenance every Wednesday in Beijing time. User information must be provided when downloading, and the download speed is the fastest in the early morning in Beijing time.

2.1.2 Constraint analysis based on transmission medium

In the long-term practice process, it is found that the acquisition nodes located in different geographical locations also have significant differences. Therefore, it is guessed that when using the network medium for data transmission, the total network sites in different spatial regions sources differ, and the service capacity of configured network bandwidth varies from one spatial region to another. Similarly, in the same spatial region under the same network configuration environment, different physical facilities such as routers, cables, switches, similarly affect the data in transmission speed in the network medium.

To verify the effect of network topology on the transmission speed of data in the network medium, relevant experiments on relay service nodes were conducted. First, taking the public data source as the source end and the relay node as the receiving end, the relay service node host directly collected a certain amount of remote sensing data from the above-mentioned public data source, and stored it in the local physical hard disk of the relay service node. Then, taking the relay node as the source end and the various collection nodes as the receiving end, other collection nodes download data from the relay node to the local. The experimental results show that the speed of using the relay node to collect public remote sensing data is indeed much higher than that of not using the relay node, but due to the high cost of using the relay node, it is not suitable for large-scale long-term public remote sensing data collection.

2.1.3 Constraint analysis based on the acquisition terminal

Among the many factors that affect the network transmission capability of the acquisition terminal, the node’s own network bandwidth configuration plays a dominant role. Having a higher network bandwidth configuration is more beneficial for network data transmission. During the work process of the collecting terminal, when other applications occupy a large amount of network bandwidth, it seriously affects the node’s ability to collect data using the network bandwidth. When the terminal uses third-party software to limit the network bandwidth of the collection client, it also affects the efficiency of data network transmission. The number of threads, task type, start time, and task length set by the terminal collection client also affect the network transmission speed of the node, and these factors are controllable factors of the collection terminal, so these controllable parameters can be used as input for subsequent network transmission ability prediction models. In addition, factors such as disk read and write speed, network card, and other physical hardware facilities on the node also affect the network transmission ability of the node.

2.2 Model parameter feature selection

To establish a model for predicting node network transmission capabilities and accurately predict the network transmission capabilities of crowd-sourced collection nodes, the factors that
affect the data in the network transmission environment were analysed in the previous section from the three stages of data source, transmission medium, and receiving terminal. Many of these factors are uncertain and not controlled by the receiving terminal, making it difficult to discover their intrinsic patterns. Therefore, they cannot be considered in the node network transmission capability prediction model. Considering the need for publicly available remote sensing data collection, and from the perspective of being long-term, stable, inexpensive, and practical, many uncontrollable factors are discarded. Without considering many data source-side factors and physical factors, a selection of controllable factors is made from the perspective of dynamic adjustments to the client application parameters of the collection terminal. Using machine learning theory, a model for predicting node network transmission capabilities is established. The model is trained using the node’s actual historical collection records, enabling the prediction of the node’s network transmission speed. Table 1 shows the selected features that affect network transmission capabilities.

<table>
<thead>
<tr>
<th>Feature Source</th>
<th>Feature Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data source</td>
<td>Number of concurrencies</td>
</tr>
<tr>
<td></td>
<td>Access frequency</td>
</tr>
<tr>
<td></td>
<td>Access time</td>
</tr>
<tr>
<td></td>
<td>Order submission time</td>
</tr>
<tr>
<td></td>
<td>Order capacity</td>
</tr>
<tr>
<td></td>
<td>Whether it is a working day</td>
</tr>
<tr>
<td>Transmission medium</td>
<td>Space area</td>
</tr>
<tr>
<td></td>
<td>Networking method</td>
</tr>
<tr>
<td></td>
<td>Router model</td>
</tr>
<tr>
<td></td>
<td>Switch models</td>
</tr>
<tr>
<td></td>
<td>Network cable type</td>
</tr>
<tr>
<td>Receiving terminal</td>
<td>Network bandwidth</td>
</tr>
<tr>
<td></td>
<td>Network card type</td>
</tr>
<tr>
<td></td>
<td>Hard drive transfer rate</td>
</tr>
<tr>
<td></td>
<td>Number of task threads</td>
</tr>
<tr>
<td></td>
<td>Task progress</td>
</tr>
<tr>
<td></td>
<td>Task length</td>
</tr>
<tr>
<td></td>
<td>Collection terminal time: hour</td>
</tr>
<tr>
<td></td>
<td>Collection terminal time: minute</td>
</tr>
<tr>
<td></td>
<td>Collection terminal time: second</td>
</tr>
<tr>
<td></td>
<td>Whether it is a working day</td>
</tr>
</tbody>
</table>

Table 1. Selected features affecting network transmission capability.

3. METHODS

3.1 PSO

PSO is an algorithm with advantages of simplicity, fast convergence and easy implementation, and is widely used in scheduling optimization, data mining, model training and other aspects of swarm intelligence optimization algorithms (D. Wang et al., 2018).

The position of the i-th particle in the population in a D-dimensional search space consisting of N particles is denoted as $X_i = (x_{i1}, x_{i2}, ..., x_{iD})$, $i = 1, 2, ..., N$ which are the possible solutions of the problem. The velocity of the particle also consists of a D-dimensional vector, denoted as $V_i = (v_{i1}, v_{i2}, ..., v_{iD})$, $i = 1, 2, ..., N$. It determines the direction and distance of the particle moving in the population. In addition, the movement of each particle component of the velocity is bounded by the maximum limiting velocity $V_m$.

$$V_i = \begin{cases} \frac{v_{i1}}{v_{i1}} + c_1 r_1 |p_{i1} - x_{i1}| + c_2 r_2 |p_g - x_{i1}| \\ x_{i1}^{t+1} = x_{i1}^t + v_{i1}^t, i = 1, 2, ..., N \end{cases}$$

In Equation (1), the number of current iterations of the particle is denoted by $k$, and the inertia weight coefficient is denoted by $w$. The larger the value of inertia weight, the stronger the global search ability of the particle. $c_1$ and $c_2$ denote the learning factors. $c_1$ describes the influence of the particle by the individual extremes, so that the particle has global search ability and avoids getting into local solutions. $c_2$ represents the influence of global optimum on the particle. $r_1$ and $r_2$ are random numbers between (0,1). $w$, $c_1$ and $c_2$ are the three constants that jointly determine the spatial search ability of the particle. The position of the optimal fitness value calculated by the particle during the iteration process is represented by the individual extreme value, which are expressed as $p_i = \left( p_{i1}, p_{i2}, ..., p_{iD} \right)$, $i = 1, 2, ..., N$. The position of the optimal fitness value computed by all particles in the population during the iteration is denoted by the global extremum $p_g$.

$$p_g = \left( p_{g1}, p_{g2}, ..., p_{gD} \right)$$

3.2 PSO-BP Algorithm

In traditional BP neural network algorithms, the weights and thresholds are randomly generated, so the algorithm has randomness and unreliability. This paper proposes the PSO-BP neural network algorithm. First, the particles of PSO algorithm are initialised as the initialization weights of BP neural network algorithm, then iteration of PSO algorithm starts. Each iteration of PSO algorithm is followed by an execution of BP neural network algorithm. The iteration termination condition of BP neural network algorithm is determined by the prediction accuracy, which is dynamically adjusted as the number of iterations of PSO algorithm increases. The training termination condition of BP neural network algorithm becomes stricter in the later stage of PSO algorithm. After each iteration of BP neural network algorithm, the absolute value of the error is used to evaluate the fitness of each particle of PSO algorithm. When the PSO algorithm satisfies its iteration termination condition, the PSO-BP algorithm also ends.

The parameters involved in the algorithm are the population size $N$, the dimensionality of the particles $D$, the number of iterations of the particles $M$, the inertia weight $w$, the learning factors $c_1$ and $c_2$, the maximum displacement $x_m$, and the maximum velocity $v_m$. The BP neural network algorithm uses a three-layer network topology. The value of each particle in the PSO algorithm represents the inertia weight in the BP neural network algorithm. The dimensionality of the particles $D$ can be expressed as:

$$D = D_{in} \cdot D_h + D_h \cdot D_{out}.$$
In equation (2), the $D_{in}$, $D_{hn}$, and $D_{out}$ denote the number of neurons in the input layer, hidden layer, and output layer of the BP neural network, respectively.

In the process of predicting the transmission capacity of the nodes, six factors including the number of download task threads, task progress, task length, acquisition terminal time (in hours), acquisition terminal time (in minutes), and data source time (in hours) are used as the six neurons in the input layer. The number of neurons in the output layer is 1, and the number of neurons in the hidden layer is 10. The dimension $D$ is 70, which indicates there are 70 inertia weights. For each particle's position $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, $x_{ish}$ to $x_{iD}$ denote the inertia weights from the input layer to the hidden layer and $x_{ish}$ to $x_{ish, 0}$ denote the inertia weights from the hidden layer to the output layer.

### 3.2.1 Improvement of inertia weights: Inertia weight ($w$) in Particle Swarm Optimization (PSO) represents the ability of a particle to inherit its current velocity from its previous velocity. A smaller value of $w$ is beneficial for local search, while a larger value is beneficial for global search. To balance the global and local search abilities of particles in PSO, the inertia weight is improved by modifying the inertia weight using the following equation (3)

$$w = \frac{\left( w_{start} - w_{end} \right) \cdot \cos \left( \frac{k}{M} \right) + \left( w_{start} + w_{end} \right)}{2}$$  \hspace{1cm} (3)

In Equation (3), the initial and termination values of inertia weights are denoted by $w_{start}$ and $w_{end}$, respectively, and $w_{start} \leq w_{end}$. The current number of particle iterations $t$ is $k$, and the maximum number of iterations is denoted by $M$. Due to the changing characteristics of the cosine function, the inertia weight ($w$) decreases slowly during the initial and final iterations. Therefore, the algorithm has a longer global optimization time at the beginning of the iterations, effectively reducing the risk of getting trapped in a local optimum. Additionally, in the later stages of the iterations, the algorithm can perform local search with a finer granularity, allowing for more incremental and precise adjustments.

### 3.2.2 Improvement of learning factors: In PSO algorithm, the learning factors $c_1$ and $c_2$ are the acceleration factors for the individual and global best values of a particle, respectively. Similarly, to enable the particles to have a good global search ability in the early stages of the iterations and to improve the precision and convergence speed of the particles in the later stages, the values of $c_1$ and $c_2$ are adjusted dynamically. This approach maintains the diversity of the population in the early stages of the search and improves the search performance in the later stages. The formula for calculating $c_1$ and $c_2$ is as follows.

$$c_1 = c_{start} - \frac{c_{start} - c_{end}}{M - k} (M - k)$$

$$c_2 = 4 - c_1$$  \hspace{1cm} (4)

In Equation (4), the initial and termination values of the learning factor $c_1$ are denoted by $c_{start}$ and $c_{end}$, respectively, and $0 \leq c_{start} \leq c_{end} \leq 4$, $M$ is the maximum number of iterations, and $k$ is the current number of iterations.

### 3.2.3 Adaptation function: After initializing the neural network, the absolute value of the actual output result and the predicted result of the download force of the node at the next moment as the fitness function of the PSO algorithm:

$$F(X, x_i) = |y_i - y_i|, i = 1, 2, \ldots, N$$  \hspace{1cm} (5)

In Equation (5), $X$ is the input to the neural network which is the six input metrics mentioned earlier. $y_i$ and $y_i$ are the node the actual and predicted download speed. In the iterative process, the variation of velocity and displacement of the particles uses equation (1). The inertia weights and learning factors are calculated using in equations (3) and (4), respectively.

### 3.2.4 Data normalisation: To improve the accuracy of the training and reduce the errors caused by the difference in the magnitude of the input values, the input data is normalized to the range of (0, 1). This is done to make the input neurons more sensitive and to reduce the errors caused by the difference in the scale of the factors. The equation for normalization is as follows:

$$X_{i, norm} = \frac{X_i - X_{i, min}}{X_{i, max} - X_{i, min}}$$  \hspace{1cm} (6)

In equation (6), the normalized result of input neuron $i$ is denoted as $X_{i, norm}$. $X_{i, min}$ is the minimum value of the input neuron, and $X_{i, max}$ is the maximum value of the input neuron.

### 3.2.5 Data inverse normalisation: To facilitate the comparison of predicted results with actual results and to make it easier to calculate the fitness value of each particle, it is necessary to reverse the normalization process of the particle’s predicted results. The equation for reverse normalization is as follows:

$$Y_{i, norm} = Y_i \cdot \left( Y_{i, max} - Y_{i, min} \right) + Y_{i, min}$$  \hspace{1cm} (7)

In Equation (7), $Y_{i, norm}$ represents the result of the i-th dimension data after de-normalization, $Y_i$ represents the result of the i-th dimension data before de-normalization, that is, the un-normalized result after prediction, $Y_{i, max}$ represents the maximum value of the i-th dimension data in the actual results, and $Y_{i, min}$ represents the minimum value of the i-th dimension data in the actual results.

### 3.2.6 Accuracy dynamic adjustment function: To achieve the goal of low precision (low expected error) in the early stages of BP neural network training and high precision (high expected error) in the later stages, this paper introduces a precision dynamic adjustment function to dynamically adjust the error precision of the BP algorithm during the training process. The expression for the precision dynamic adjustment function is as follows:

$$accuracy = a - b \cdot k$$  \hspace{1cm} (8)

In Equation (8), $a$ is the initial accuracy which usually is 1, $b$ is the accuracy adjustment factor, and $k$ is the current number of iterations.

The execution flow of the PSO-BP algorithm is shown in Figure 2.
The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-M-1-2023

Figure 2. PSO-BP algorithm flow chart

4. EXPERIMENT AND RESULTS

4.1 Experimental parameters setting

To validate the execution results of the above proposed PSO-BP algorithm, BP, and PSO-BP algorithms were evaluated to conduct the experiments with real data sets. In Table 2, the parameters of the above two algorithms are set. The number of neurons in the input layer is 6, the number of neurons in the hidden layer is 10, and the number of neurons in the output layer is 1. The sigmoid function is used for the excitation function of the hidden layer. The inertia weights from the input layer to the hidden layer and from the hidden layer to the output layer are adjusted by combining gradient descent and momentum factor.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP</td>
<td>LearningRate = 0.08, Momentum factor = 0.5, AvgTrainError &gt; 0.001</td>
</tr>
<tr>
<td>PSO-BP</td>
<td>N=50, Cmax=3.5, Cmin=2.5, wmax=1.0, wmin=0.5, tmax=200, tmax=2, XL, ρ ∈ [-0.5, 0.5], LearningRate = 0.08, Momentum factor = 0.5, Iter = 5</td>
</tr>
</tbody>
</table>

Table 2. Algorithm parameters setting

4.2 Experimental results analysis

To validate the execution results of the BP algorithm and the proposed PSO-BP algorithm, experiments have been done on real data sets. After cleaning and filtering the historical dataset, we collected more than 20,000 data files from a collection client of Sentinel 2 data collection. Around 20,000 data files were selected as the training set, and more than 2900 data files were selected as the test set. Table 3 lists the some of the experimental results of the tests using the two algorithms mentioned above.

<table>
<thead>
<tr>
<th>No</th>
<th>Actual Value</th>
<th>BP Predicted Value</th>
<th>Error Rate</th>
<th>PSO-BP Predicted Value</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>976</td>
<td>677</td>
<td>0.3064</td>
<td>835</td>
<td>0.1445</td>
</tr>
<tr>
<td>2</td>
<td>840</td>
<td>854</td>
<td>0.0167</td>
<td>927</td>
<td>0.1036</td>
</tr>
<tr>
<td>3</td>
<td>832</td>
<td>886</td>
<td>0.0649</td>
<td>895</td>
<td>0.0757</td>
</tr>
<tr>
<td>4</td>
<td>816</td>
<td>484</td>
<td>0.0392</td>
<td>916</td>
<td>0.1225</td>
</tr>
<tr>
<td>5</td>
<td>848</td>
<td>859</td>
<td>0.0310</td>
<td>899</td>
<td>0.0601</td>
</tr>
<tr>
<td>6</td>
<td>840</td>
<td>867</td>
<td>0.0321</td>
<td>916</td>
<td>0.0905</td>
</tr>
<tr>
<td>7</td>
<td>920</td>
<td>833</td>
<td>0.0946</td>
<td>905</td>
<td>0.0163</td>
</tr>
<tr>
<td>8</td>
<td>944</td>
<td>842</td>
<td>0.1081</td>
<td>894</td>
<td>0.0530</td>
</tr>
<tr>
<td>9</td>
<td>920</td>
<td>842</td>
<td>0.0848</td>
<td>909</td>
<td>0.0120</td>
</tr>
<tr>
<td>10</td>
<td>840</td>
<td>799</td>
<td>0.0488</td>
<td>887</td>
<td>0.0560</td>
</tr>
<tr>
<td>11</td>
<td>824</td>
<td>834</td>
<td>0.0121</td>
<td>872</td>
<td>0.0583</td>
</tr>
<tr>
<td>12</td>
<td>912</td>
<td>843</td>
<td>0.0757</td>
<td>889</td>
<td>0.0252</td>
</tr>
<tr>
<td>13</td>
<td>976</td>
<td>777</td>
<td>0.2039</td>
<td>876</td>
<td>0.1025</td>
</tr>
<tr>
<td>14</td>
<td>846</td>
<td>833</td>
<td>0.0359</td>
<td>853</td>
<td>0.0127</td>
</tr>
<tr>
<td>15</td>
<td>808</td>
<td>813</td>
<td>0.0062</td>
<td>873</td>
<td>0.0804</td>
</tr>
<tr>
<td>16</td>
<td>808</td>
<td>798</td>
<td>0.0124</td>
<td>857</td>
<td>0.0606</td>
</tr>
<tr>
<td>17</td>
<td>912</td>
<td>867</td>
<td>0.0493</td>
<td>926</td>
<td>0.0154</td>
</tr>
<tr>
<td>18</td>
<td>880</td>
<td>888</td>
<td>0.0091</td>
<td>932</td>
<td>0.0591</td>
</tr>
<tr>
<td>19</td>
<td>832</td>
<td>848</td>
<td>0.0192</td>
<td>922</td>
<td>0.1082</td>
</tr>
<tr>
<td>20</td>
<td>856</td>
<td>858</td>
<td>0.0023</td>
<td>903</td>
<td>0.0549</td>
</tr>
<tr>
<td>21</td>
<td>920</td>
<td>842</td>
<td>0.0848</td>
<td>903</td>
<td>0.0185</td>
</tr>
<tr>
<td>22</td>
<td>872</td>
<td>876</td>
<td>0.0046</td>
<td>904</td>
<td>0.0367</td>
</tr>
<tr>
<td>23</td>
<td>824</td>
<td>850</td>
<td>0.0316</td>
<td>890</td>
<td>0.0801</td>
</tr>
<tr>
<td>24</td>
<td>832</td>
<td>853</td>
<td>0.0252</td>
<td>879</td>
<td>0.0565</td>
</tr>
<tr>
<td>25</td>
<td>832</td>
<td>819</td>
<td>0.0156</td>
<td>885</td>
<td>0.0637</td>
</tr>
<tr>
<td>26</td>
<td>808</td>
<td>845</td>
<td>0.0458</td>
<td>866</td>
<td>0.0718</td>
</tr>
<tr>
<td>27</td>
<td>896</td>
<td>840</td>
<td>0.0625</td>
<td>861</td>
<td>0.0391</td>
</tr>
<tr>
<td>28</td>
<td>808</td>
<td>833</td>
<td>0.0309</td>
<td>916</td>
<td>0.1337</td>
</tr>
<tr>
<td>29</td>
<td>816</td>
<td>802</td>
<td>0.0172</td>
<td>901</td>
<td>0.1042</td>
</tr>
<tr>
<td>30</td>
<td>928</td>
<td>838</td>
<td>0.0970</td>
<td>900</td>
<td>0.0302</td>
</tr>
<tr>
<td>31</td>
<td>816</td>
<td>791</td>
<td>0.0306</td>
<td>871</td>
<td>0.0674</td>
</tr>
<tr>
<td>32</td>
<td>984</td>
<td>824</td>
<td>0.1626</td>
<td>889</td>
<td>0.0965</td>
</tr>
<tr>
<td>33</td>
<td>840</td>
<td>919</td>
<td>0.0940</td>
<td>942</td>
<td>0.1214</td>
</tr>
<tr>
<td>34</td>
<td>920</td>
<td>909</td>
<td>0.0120</td>
<td>928</td>
<td>0.0087</td>
</tr>
<tr>
<td>35</td>
<td>856</td>
<td>921</td>
<td>0.0759</td>
<td>927</td>
<td>0.0829</td>
</tr>
<tr>
<td>36</td>
<td>888</td>
<td>907</td>
<td>0.0214</td>
<td>925</td>
<td>0.0417</td>
</tr>
<tr>
<td>37</td>
<td>920</td>
<td>906</td>
<td>0.0152</td>
<td>931</td>
<td>0.0120</td>
</tr>
<tr>
<td>38</td>
<td>936</td>
<td>898</td>
<td>0.0406</td>
<td>922</td>
<td>0.0150</td>
</tr>
<tr>
<td>39</td>
<td>832</td>
<td>905</td>
<td>0.0877</td>
<td>921</td>
<td>0.1070</td>
</tr>
<tr>
<td>40</td>
<td>888</td>
<td>905</td>
<td>0.0191</td>
<td>928</td>
<td>0.0450</td>
</tr>
<tr>
<td>41</td>
<td>816</td>
<td>906</td>
<td>0.1103</td>
<td>927</td>
<td>0.1360</td>
</tr>
<tr>
<td>42</td>
<td>808</td>
<td>909</td>
<td>0.1250</td>
<td>947</td>
<td>0.1720</td>
</tr>
<tr>
<td>43</td>
<td>936</td>
<td>913</td>
<td>0.0246</td>
<td>937</td>
<td>0.0011</td>
</tr>
<tr>
<td>44</td>
<td>824</td>
<td>899</td>
<td>0.0910</td>
<td>936</td>
<td>0.1359</td>
</tr>
<tr>
<td>45</td>
<td>920</td>
<td>900</td>
<td>0.0217</td>
<td>950</td>
<td>0.0326</td>
</tr>
<tr>
<td>46</td>
<td>808</td>
<td>882</td>
<td>0.0916</td>
<td>905</td>
<td>0.1200</td>
</tr>
<tr>
<td>47</td>
<td>968</td>
<td>876</td>
<td>0.0950</td>
<td>891</td>
<td>0.0795</td>
</tr>
<tr>
<td>48</td>
<td>832</td>
<td>754</td>
<td>0.0938</td>
<td>870</td>
<td>0.0457</td>
</tr>
<tr>
<td>49</td>
<td>824</td>
<td>771</td>
<td>0.0643</td>
<td>871</td>
<td>0.0570</td>
</tr>
<tr>
<td>50</td>
<td>848</td>
<td>765</td>
<td>0.0979</td>
<td>847</td>
<td>0.0012</td>
</tr>
</tbody>
</table>

Table 3. Selected prediction results

Figure 3 below shows the comparison of the error rates of the two algorithms for the data in Table 3. The prediction errors of the BP algorithm and the PSO-BP algorithm are very different, with the former having a large variance in the prediction results and the latter having a relatively flat prediction error. The errors of the two algorithms are mostly concentrated around 5%.
In Figure 4, the true value, and the prediction value from the two algorithms are shown. Prediction from PSO-BP algorithm is closer to the true value, while the error of the traditional BP algorithm is larger compared with PSO-BP algorithm.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Mean relative error</th>
<th>Root mean square error</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP</td>
<td>0.0925</td>
<td>0.1163</td>
</tr>
<tr>
<td>PSO-BP</td>
<td>0.0595</td>
<td>0.0703</td>
</tr>
</tbody>
</table>

Table 4. Evaluation of algorithm prediction results

From Table 4, the mean relative error and root mean square error of PSO-BP algorithm are smaller than those of BP algorithm, which indicates that the PSO-BP algorithm has better prediction results. Tables 5 show the number of test sets in different error rate ranges for each of the two algorithms. The proposed PSO-BP algorithm achieves better accuracy comparing with traditional BP algorithm.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Error &lt;= 5%</th>
<th>Error &lt;= 15%</th>
<th>Error &gt; 5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP</td>
<td>1004</td>
<td>2371</td>
<td>567</td>
</tr>
</tbody>
</table>

Table 5. Error statistics on test set

5. CONCLUSION

This paper proposed an improved PSO-BP algorithm by introducing the accuracy dynamic adjustment function and the learning factor adjustment function. Finally, a comparison experiment between the improved PSO-BP algorithm and the BP algorithm is conducted on the real data set to verify the feasibility of the prediction model and the improved PSO-BP algorithm.

ACKNOWLEDGEMENTS

This work was supported in part by TUOHAI special project 2020 from Bohai Rim Energy Research Institute of Northeast Petroleum University under Grant HBBZKX202002 and project of Excellent and Middle-aged Scientific Research Innovation Team of Northeast Petroleum University under grant KYCXTD201903.

REFERENCES


This contribution has been peer-reviewed. https://doi.org/10.5194/isprs-archives-XLVIII-M-1-2023-25-2023 | © Author(s) 2023. CC BY 4.0 License.


