

# A NEW BP NEURAL NETWORK FUSION ALGORITHM FOR MULTI-SOURCE REMOTE SENSING DATA ON GROUNDWATER

ZHANG, F.<sup>1,2\*</sup> – XUE, H. F.<sup>2</sup> – ZHANG, Y. H.<sup>1</sup>

<sup>1</sup>*Yulin University of School of Information Engineering of Shannxi Province, Yulin 719000, China*

<sup>2</sup>*China Aerospace Academy of Systems Science and Engineering, Beijing 100048, China*

*\*Corresponding author*

*e-mail: tfnew21@sina.com; phone: +86-136-4922-0169; fax: +86-091-2389-1364*

(Received 18<sup>th</sup> Mar 2019; accepted 17<sup>th</sup> May 2019)

**Abstract.** This paper aims to enhance the accuracy and reduce the cost of the fusion of multi-source remote sensing data. For this purpose, the existing multi-source remote sensing data fusion methods were reviewed in detail. Then, a new back propagation (BP) neural network (BPNN) fusion algorithm for the groundwater was put forward based on hybrid soft computing. Using the function approximation ability of BP neural network, it was combined with the Kalman filter to form an optimization method. The BP neural network was coupled with the particle swarm optimization (PSO) algorithm into the PSO-BPNN-EKF data fusion algorithm. On this basis, the least squares support vector machine (LSSVM) was introduced to create the LSSVM-PSO data fusion algorithm. Through simulation experiments, it is learned that the proposed algorithm can effectively fuse the multi-source remote sensing data on groundwater, especially in the case of big data. The research findings shed a new light on the fusion of remote sensing data collected by multiple sensors.

**Keywords:** *groundwater, Kalman filter, data fusion, particle swarm optimization, hybrid soft computing*

## Introduction

Back propagation (BP) neural networks (BPNNs) are neural network models based on the cross section of machine learning and neural networks. BP neural networks are computing systems vaguely inspired by the biological neural networks that constitute animal brains (Costantini et al., 1997). A BPNN is based on a collection of connected units or nodes called artificial neurons which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one artificial neuron to another. For any remote sensing platform, it is impossible to fully reflect the features of the ground target through the remote sensing data acquired by a single sensor. Facing multi-source spatial data, the geographic information system (GIS) data acquisition has become a popular choice. However, the resulting data resources differ in coordinate system, scale standard and storage format, making it difficult for data integration and sharing. The traditional data fusion algorithm requires a more accurate mathematical model of the object, which is not suitable for complex models, by using BP neural network algorithm to realize multi-sensor data fusion, the prior requirement of object is not high, and it has strong adaptive ability (Pohl and Van Genderen, 1998). Against this backdrop, the researchers on remote sensing have paid much attention to the removal of redundancy in data acquired by multiple sensors and improve data fusion and sharing. In light of this, this paper attempts to develop an effective multi-source remote sensing data fusion technology based on BP neural network, using BP neural network to fuse sensor data can improve the stability and accuracy of sensors.

Traditional neural network training algorithm is slow in convergence and easy to fall into local optimum. In recent years, swarm intelligence optimization algorithms such as particle swarm optimization (algorithm) have better global convergence performance and can be used to train neural network parameters and structures. This model mechanistically reflects the natural process but fails to reflect the interaction between man and nature. To solve the problem, the data fusion technology, a desirable tool for estimation and identification, can be introduced to the model, allowing the timely and reliable acquisition of multi-format information from multiple sources like experts and the multimedia. Currently, the key data fusion technologies include the estimation technique, fuzzy set theory, clustering analysis, template method, human-computer interaction, expert system, multimedia technology, neural network method, distributed database technology and parallel processing technology (Li et al., 2018; Neelapu et al., 2018; Bhoi, 2017; Li, 2017; Liu and Xu, 2017; Hu et al., 2017; Zhang et al., 2010). Coupled with geophysical and geological databases, these technologies can increase the constraint recognition of in remote sensing applications, allowing them to describe the spatial geometry of the target from different aspects.

For efficient and accurate multi-source remote sensing data on groundwater, this paper creates a BP neural network fusion algorithm. Specifically, the extended Kalman filter (EKF) was adopted to sample the prior information and coupled with the particle swarm optimization (PSO) model and BP neural network (BPNN) into the PSO-BPNN-EKF data fusion algorithm. On this basis, the least squares support vector machine (LSSVM) was introduced to create the LSSVM-PSO data fusion algorithms.

## **Materials and methods**

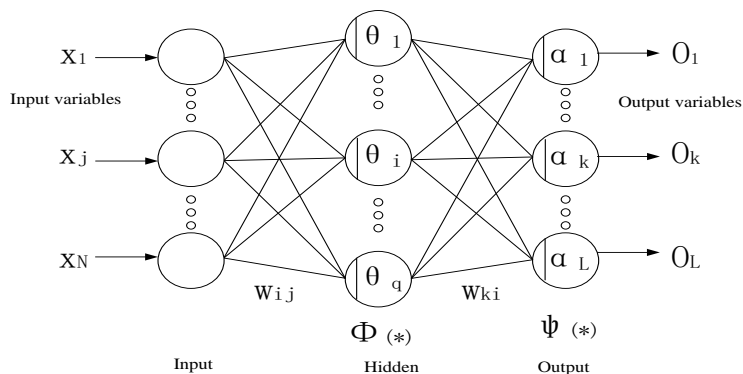
### ***Neural network data fusion methods***

#### ***Neural network model***

The typical three-layer neural network model is given in *Figure 1*, BPNN is a multi-layer feedforward network trained according to error back propagation algorithm. This algorithm can learn and store a great deal of mapping relations of input-output model, without the need to disclose in advance the mathematical equation that describes these mapping relations. The learning rule is to adopt the steepest descent method in which the back propagation is used to regulate the weight value and threshold value of the network to achieve the minimum error sum of square (Li et al., 2018). The structures of the BP neural networks are relatively mature in both network theory and performance. The distinctive advantage is that it has a strong nonlinear mapping capability and flexible network structure. In this paper, the BPNN is optimized using the PSO to speed up the convergence and avoid the local minimum trap.

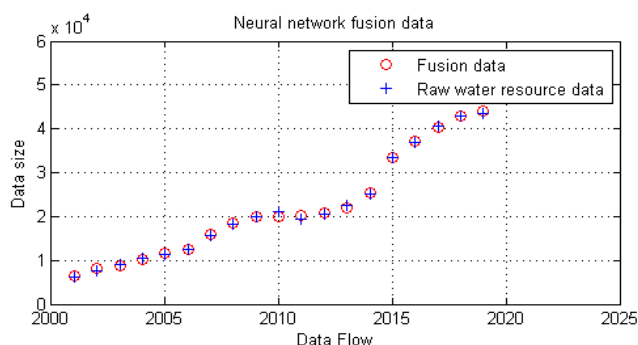
#### ***Optimization of neural network parameters***

The error correction algorithm is often used in the training process of feedforward neural networks. The most typical error correction algorithm is BP algorithm. The algorithm first generates a set of weights randomly, and then calculates the weight correction by gradient descent method until the training error reaches the target precision range. But the most obvious defect of this algorithm is slow convergence and sometimes even no convergence.



**Figure 1.** Structure diagram of three-layer neural network model

This paper used the optimization of neural network parameters translation method to build the training dataset. Using BP neural network's function approximation ability, BP neural network and Kalman filter are combined to form an estimator, which can make full use of sensor information from different noise pollution, improve estimation performance while maintaining estimation. The filtered computational structure is as simple as possible. Before network training, we must first normalize the data to ensure that the output of the network layer is not too small. What the initial value of the center vector  $p_i$  is determined by training samples, and where the  $p_i = (p_{i1}, p_{i2}, p_{i3}, p_{i4}, \dots)$ . If it belongs to the training sample set of the first kind of data change, It is represented as  $\{S_1, S_2, \dots, S_m\}$ , and the initial value of each element in the centre vector  $p_1$  is the average value of each element in the input vector of these samples. Firstly, take the logarithm of the sensor information from different noise pollution data which is denoted as is shown in *Figure 2*.



**Figure 2.** Neural network fusion data

### ***KF data fusion***

The traditional filtering algorithms mainly target the state variables of the system to be processed. Their performance depends on the system model, the input signal and the measured signal (Ryu and Huber, 2007). For these algorithms, form transformation is required to get the estimates of model parameters. In other words, the state equation and output equation need to be transformed from the standard form via bit filtering, with unknown parameters as the state variables. The transformation equation can be expressed as *Equation 1*:

$$\begin{cases} x = ax + 1 \\ y = z \end{cases} \Rightarrow \begin{cases} \begin{bmatrix} x \\ a \end{bmatrix} = \begin{bmatrix} ax \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ y = [1 \quad 0] \begin{bmatrix} z \\ a \end{bmatrix} \end{cases} \quad (\text{Eq.1})$$

In a dynamic environment, the Kalman filter is often employed thanks to its real-time integration of data. The filter can describe linear differential equations of a discrete-time system. Let  $x \in R^n$  be the state variables. Then, the state equation can be expressed as *Equation 2*:

$$x_k = f_k(x_{k-1}, \mu_{k-1}, \omega_{k-1}) + a_{k-1} \quad (\text{Eq.2})$$

The measured variables  $z \in R^m$  can be obtained as *Equation 3*:

$$z_k = h_k(x_k, v_k) + \beta_k \quad (\text{Eq.3})$$

In the above equations,  $a_{k-1}$  is the system noise at time  $k-1$ ;  $\beta_k$  is the measured noise;  $\omega_k$  and  $v_k$  are two random variables representing the excited noise and the observed noise, respectively;  $\mu_k$  and  $\omega_k$  are the noises of the driving function and zero-mean normalization, respectively;  $x_k$  and  $z_k$  are the nonlinear functions of the state variables and measured variables, respectively.

The EKF is an extension of the Kalman filter. The basic idea of the EKF is to convert nonlinear vector function  $f(\cdot)$  and nonlinear system model  $f(\cdot)$  into linear filter values. Following this train of thoughts, *Equations 2* and *3* can be expanded into a Taylor series (*Eq. 4*) to obtain the linear model of the system. Then, the basic equations of the Kalman filter can be applied to solve the nonlinear filtering problem.

$$\begin{cases} x_k = f_k(x_{k-1}, \mu_{k-1}, \omega_{k-1}) + \frac{\partial f(x)}{\partial x} \Big|_{x=x_{k-1} + \alpha_{k-1}} \\ z_k = h_1(x_k) + \frac{\partial f(x)}{\partial x} \Big|_{x=x_k + \beta_k} \end{cases} \quad (\text{Eq.4})$$

### **Particle swarm optimization algorithm**

The PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. It solves a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best-known position but is also guided toward the best-known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions (Quiroga et al., 2013). Below is a brief description of the standard PSO algorithm.

Suppose there are a  $d$ -dimensional search space and a population containing  $N_p$  particles. Let  $x_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{id}^t)$  be a  $d$ -dimensional vector representing the initial position of particle  $i$  of the  $t$ -th generation, let the vector  $v_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{id}^t)$  be the velocity of particle  $i$ , i.e., the rate of change in the position, and let

$p_i^t = (p_{i1}^t, p_{i2}^t, \dots, p_{id}^t)$  be the best known position of particle  $i$  of the  $t$ -th generation. Then, the velocity and position of particle  $i$  can be obtained by updating *Equations 5* and *6* below.

$$v_{ij}(t+1) = \omega \cdot v_{ij}(t) + c_1 \cdot rand1_{ij} \cdot (pbest_{ij}(t) - x_{ij}(t)) + c_2 \cdot (gbest_j(t) - x_{ij}(t)) \quad (\text{Eq.5})$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (\text{Eq.6})$$

where  $t$  is the iteration number;  $\omega$  is the inertia coefficient;  $c_1$  and  $c_2$  are acceleration coefficients;  $rand_1$  and  $rand_2$  two uniformly-distributed random independent numbers in  $[0, 1]$ . The values of  $\omega$ ,  $c_1$  and  $c_2$  value should be determined according to the specific problem.

The performance of the PSO can be greatly enhanced if  $\omega$  decreases linearly with the increase in the number of iterations as *Equation 7*:

$$\omega = \omega_{\min} + (iter_{\max} - iter) \times (\omega_{\max} - \omega_{\min}) / iter_{\max} \quad (\text{Eq.7})$$

where  $\omega_{\min}$  and  $\omega_{\max}$  are the maximum and minimum weighting factors, respectively;  $iter$  is the current iteration number;  $iter_{\max}$  is the maximum number of iterations. The flight speed  $v_i$  falls between the maximum and minimum weighting factors. This constraint condition ensures the convergence to the optimal solutions and improves the global search ability of the PSO.

### ***LSSVM data fusion methods***

The SVM (Jalalkamali et al., 2011) is a supervised learning model with associated learning algorithms that analyze data used for classification and regression analysis. It is known for its excellent generalization in the case of small samples. The SVM learning has been commonly used for the analysis on spatial data (Babaoğlu et al., 2010; Awan et al., 2013). The traditional SVM aims to solve convex quadratic optimization problems. However, the kernel matrix of the SVM has to occupy a large storage capacity, and the solution becomes less efficient when the sample size is excessively large. For this reason, the LSSVM (Zhang, 2011; Üstün et al., 2005) has been developed to improve the optimization effect. The new model replaces the traditional inequality constraints with an equality equation in the construction of the optimal objective function. Thus, the optimization process is transformed into the solution of a set of linear equations. In this way, the LSSVM achieves twice the efficiency of the traditional SVM. Despite the reduced cost, the LSSVM faces the loss of sparsity of the traditional method, owing to the use of  $\epsilon$ -insensitive loss function.

Suppose there is a sample containing  $n$  training sets  $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ , where  $x_k \in R^p$  the  $k$ -th input vector is. Assuming that the dataset is linearly separable, there must exist a linear classifier  $y(x) = \omega^T \varphi(x) + b$  in the input space. If an optimization problem is nonlinear and separable, a nonlinear function  $\varphi(\cdot)$  can be adopted to map the

raw data space into a high-dimensional feature space, before looking for the classification surface-dimensional feature space  $y(x) = \omega^T \varphi(x) + b$ .

To obtain the optimal high-dimensional plane, the inequality equation needs to be introduced to the traditional SVM as *Equation 8*:

$$\begin{aligned} \min_{w,b,\gamma} \quad & -\gamma \\ \text{s.t} \quad & y_i(\langle w, x_i \rangle + b) \geq \gamma, i = 1, \dots, \xi \\ & \|w\|^2 = 1 \end{aligned} \quad (\text{Eq.8})$$

where  $\gamma$  is the spacer;  $\xi$  is the number of training samples;  $x_i$  is the vector of the  $i$ -th training sample;  $w$  is the weight vector;  $b$  is the threshold;  $y_i$  are the marked samples ( $y_i = \begin{cases} 1 & x_i \in \omega_1 \\ -1 & x_i \in \omega_2 \end{cases}$ );  $\omega_i$  is the class of the  $i$ -th sample.

Then, the Lagrange function can be established as *Equation 9*:

$$\begin{aligned} L(w,b,\gamma,a,\lambda) = & \\ & -\gamma - \sum_{i=1}^{\xi} a_i [y_i(\langle w, x_i \rangle + b) - \gamma] \\ & + \lambda (\|w\|^2 - 1) \end{aligned} \quad (\text{Eq.9})$$

Unlike the traditional SVM, the LSSVM is built with equality constraint instead of the inequality constraint. Hence, the optimization problem can be expressed as *Equation 10*:

$$\begin{aligned} \min_{w,b,\xi} J(w,b,\xi) = & \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^N \xi_i^2 \\ \text{s.t} \quad & y_i [w^T \varphi(x_i) + b] = 1 - \xi_i, i = 1, 2, \dots, N \end{aligned} \quad (\text{Eq.10})$$

Then, the structure of the Lagrangian dual problem can be expressed as *Equation 11*:

$$L = J - \sum_{j=1}^N a_k \{ y_k [w^T \varphi(x_k) + b] + \xi_k - 1 \} \quad (\text{Eq.11})$$

According to the optimization conditions, the partial derivatives of  $w$ ,  $b$ ,  $\xi$  and  $a$  can be obtained as *Equation 12* and set to zero.

$$\begin{cases} w = \sum_{i=1}^N a_i y_i \varphi(x_i) \\ \sum_{i=1}^N a_i y_i = 0 \\ a_i = \gamma \xi_i \\ y_i [w^T \varphi(x_i) + b] - 1 + \xi_i = 0 \end{cases} \quad (\text{Eq.12})$$

Under the above conditions, the kernel function can be defined as  $k(x_i, y_i) = \psi(x_i)\psi(y_i)$ . Then, the optimization problem can be transformed into solving the linear equations as *Equation 13*.

$$\begin{bmatrix} 0 & 1 & \dots & 1 \\ 1 & k(x_1, x_1) & \dots & k(x_1, x_N) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & k(x_N, x_1) & \dots & k(x_N, x_N) + 1/c \end{bmatrix} \begin{bmatrix} b \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} 0 \\ y_1 \\ \vdots \\ y_N \end{bmatrix} \quad (\text{Eq.13})$$

The above linear equations can be solved by the LSSVM classifier as *Equation 14*:

$$f(x) = \text{sign}\left(\sum_{i=1}^N a_i y_i K(x, x_i) + b\right) \quad (\text{Eq.14})$$

The radial-basis function (RBF) can be selected as the kernel function, as *Equation 15*:

$$k(x_i, y_i) = \exp\left(\frac{-1\|x_i - y_i\|^2}{2\sigma^2}\right) \quad (\text{Eq.15})$$

Based on these parameters, the penalty factor  $C$  and kernel function parameters  $\sigma$  can be obtained through the common optimization procedure.

### ***BP neural network data fusion model***

Soft computing was proposed by Prof. Zadein in the 1990s to solve uncertainty problems through fuzzy and intelligent technologies. The method is capable of tackling one or more complex datasets in realistic environment. Unlike traditional hard computing, soft computing does not pursue the exact solution, considering the inaccuracies in real-world problems and the high cost of the traditional method (Zhang et al., 2013). Instead, it pursues the next best solution when it is impossible or extremely difficult to obtain the optimal solution.

So far, soft computing has been extensively applied to data fusion and classification. This technology can combine the member attributes of fusions sequence into different logic structures (i.e. series structure, mosaic structure and parallel structure), depending on the specific data. Here, the parallel structure is adopted for fusing the remote sensing data on groundwater.

### ***PSO-BPNN-EKF***

In this paper, the PSO-BPNN-EKF optimization algorithm is proposed by optimizing the EKF state equation with the PSO, the algorithm uses PSO instead of the parameters in BPNN training algorithm and optimizes BP parameters. Each particle is a vector representing a set of parameters. The process of finding the global optimum is the process of obtaining the optimal parameters. Before the optimization, the particle set converges to the high likelihood area, which is far away from the true state. This

problem is solved by the optimization, together with particle degeneration and in accuracy prediction. The PSO-EKF algorithm consists of the following steps:

(1) *Initialization*

Let  $N$  be the number of particles, and  $p(x_0)$  be the initial population. Set the initial values of parameters as  $\Delta x_0 = 0$ ,  $p_0 = C_{x0}$  and  $C_{x0} = 1.5$ .

(2) *Importance sampling*

Adjust the velocity and position of each particle  $v_k^i = \omega v_{k-1}^i + \gamma \text{rand}_i(p_{k-1}^g - x_{k-1}^i)$ , and update the state of the particle according to the EKF algorithm. In other words, estimate the state of the particle  $\hat{x}_k^i = x_k^i + \Delta x_k^i + v_k^i$  at time  $k$  according to the EKF importance sampling algorithm. Find the mean  $x_k^{-i}$  and variance  $P_{xk}^i$  of the particle set  $\{\hat{x}_k^i\}_{i=1}^N$ . Update the weight of the particles in the set  $x_k^i$  according to the importance of the density function  $(x_k^i | x_{k-1}^i, \omega_k) = N(\hat{x}_k^i, x_k^{-1}, P_{xk}^i)$ .

(3) *Recompiling*

Estimate the posterior probability of the target state at time  $k$   $x_k = \sum_{i=1}^N x_k^i \omega_k^i$ , and find the current global optimal solution  $p_k^g$ . For  $k = k + 1$ , return to Step (2) and estimate the posterior probability of the target state at the next time.

### **PSO-based LSSVM parameter optimization**

The LSSVM parameters  $\gamma$  and  $\sigma$  directly impact the prediction accuracy. Normally, the parameter space is search exhaustively to optimize these two parameters. However, it is difficult to determine the reasonable range of each parameter. To solve the difficulty, the LSSVM parameters were optimized by the PSO through the following steps (Yu et al., 2012; Anand et al., 2013).

(1) Initialize the PSO parameters, including population size, learning factor, maximum number of iterations, and the initial velocity and position of the particles.

(2) Predict the particle vector of each LSSVM learning sample, and obtain the current position, prediction error and fitness of each particle. Then, compare the current fitness of each particle with the best-known fitness. If the former is better, it should be selected as the optimal position of the particle.

(3) Compare the optimal position of each particle with the best known global position. If the former is better, it should be selected as the optimal global position.

(4) Calculate the inertia weight according to *Equation 7* and update the particle velocity and position according to *Equations 5* and *6*, respectively.

(5) Terminate the search process if the termination condition is satisfied; otherwise, return to Step (2) and start a new round of search.

## **Results**

The study area is Yulin Prefecture in northern China's Shaanxi Province. There are 12 river basins in the prefecture, with a perennial mean runoff of 1.9446 billion  $\text{m}^3$ . Under the reliabilities of 50%, 75% and 95%, the annual runoff is 1.847, 1.504 and



1.136 billion m<sup>3</sup>. Specifically, the perennial mean runoff of the 10 outflow basins stands at 1.8099 billion m<sup>3</sup>. Under the reliabilities of 50%, 75% and 95%, the annual runoff is 1.7194, 1.4034 and 1.0660 billion m<sup>3</sup>. The perennial mean runoff of the other 2 inflow basins amounts to 0.1347 billion m<sup>3</sup>. Under the reliabilities of 50%, 75% and 95%, the annual runoff is 0.1277, 0.1001 and 0.0699 billion m<sup>3</sup>. The central districts, including Shenmu County, Fugu County, Yuyang District and Hengshan County, have a perennial mean runoff of about 1.557 billion m<sup>3</sup>. Under the reliabilities of 50%, 75% and 95%, the annual runoff is 1.477, 1.201 and 0.907 billion m<sup>3</sup>. The peripheral districts like Jingbian County have a perennial mean runoff of about 0.388 billion m<sup>3</sup>. Under the reliabilities of 50%, 75% and 95%, the annual runoff is 0.370, 0.303 and 0.327 billion m<sup>3</sup>. The annual runoffs are recorded in *Table 1*. The water quality of major rivers is given in *Table 2*.

**Table 1.** Annual runoffs in Yulin Prefecture

River basin	Basin area (km <sup>2</sup> )	Average annual runoff (million m <sup>3</sup> )	Average runoff depth (mm)	Different frequencies annual runoff (million m <sup>3</sup> )		
				50%	75%	95%
Huangfuchuan	2 827	1. 546	54. 69	1.326	0.868	0.498
Shimizu River	321	0. 183	57. 01	0.160	0.111	0.067
Sichuan Gushan	261	0. 195	74. 71	0.171	0.112	0.064
Kuye River	4 629	3. 096	66. 88	2.876	2.146	1.409
Wudinghe River	9 396	4. 260	45. 34	4.133	3.478	2.708
Total	17434	9.28	298.63	8.666	6.715	4.746

**Table 2.** Groundwater quality in the study area

River	PH value	Dissolved oxygen (mg/L)	Oxygen consumption (mg/L)	Total hardness (Germany degrees)	Salinity (g/L)
Lu River (Hengshan)	7.6	2.3	5.4	14.0	0.92
Wudinghe River (Xiangshui)	7.6	3.7	2.4	10.1	0.77
Wudinghe River (Baijia Chuan)	8.2	6.3	1.4	12.1	0.53
Tuweihe (upstream)	7.4	4.3	2.1	8.26	0.23
Tuweihe (downstream)	7.9	3.2	2.9	8.27	0.20
Kuye River (upstream)	7.4	3.1	3.3	9.81	0.28
Kuye River (midstream)	7.9	6.1	4.1	11.36	0.32
Kuye River (downstream)	7.4	5.1	4.5	9.53	0.26
Jia Lu River	7.9	4.2	3.1	6.7	0.22
Bali River	7.3	5.5	3.2	8.8	0.25

To validate the proposed PSO-BPNN-EKF algorithm, the performance of the PSO-BPNN-EKF and PSO-SVM were contrasted with that of the KF, the EKF, the PSO-KF, the LSSVM-PSO and the BP- PSO-BPNN-EKF through 100 Monte-Carlo simulations with  $\delta = 2$  at the number of particles of 50, 100 and 150. The results of different algorithms are shown in *Table 3*.

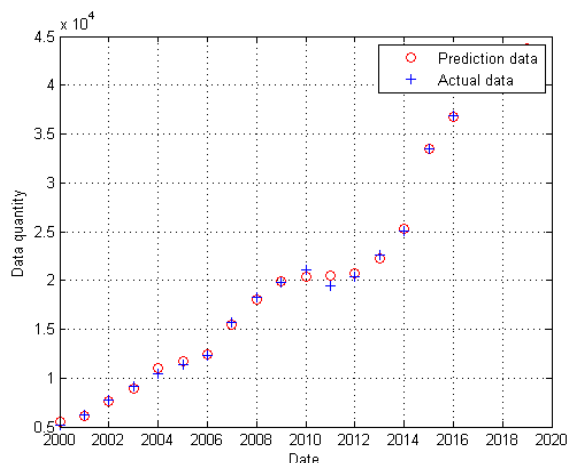
**Table 3.** Results of contrastive algorithms

Filter	PF	EPF	PSO-EPF	PSO-SVM	PSO-BPNN-EKF
N = 60 Variance	4.199	3.823	2.310	2.121	2.112
N = 100 Variance	3.523	3.123	2.132	2.092	2.093
N = 160 Variance	3.223	3.121	2.622	2.122	2.031
N = 260 Variance	2.133	1.987	1.232	1.154	1.131

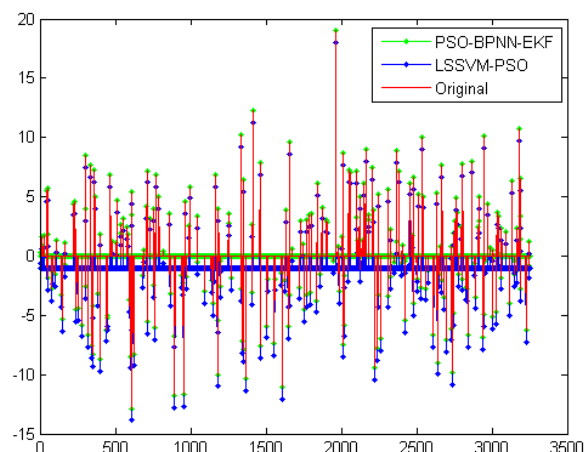
Considering the good nonlinear fitting effect of the RBF, the function was adopted as an improved kernel for the LSSVM. The selection of parameters is essential to the performance of the LSSVM model. In this paper, the parameters are determined by the improved PSO as: number of particles  $N = 25$ , the maximum number of iterations  $G_{max} = 100$ , the learning factors  $c_1 = 1.5$  and  $c_2 = 1.5$ , the inertia weight = 0.9. Then, the LSSVM-PSO model was simulated on the Matlab. The particle position was updated constantly until the termination condition was satisfied. In this way, the optimal RBF parameters  $\sigma$  was determined as 0.5, and the normalization parameter  $c$  was identified as 60. Next, the optimal parameters were adopted for the fusion of groundwater data by the LSSVM-PSO model. The results are presented in *Figure 3*. And then we compare the experimental results of data fusion between LSSVM-PSO and PSO-BPNN-EKF algorithm, as shown in *Figure 4*. In order to more clearly observe the change of the value of the sum of squares of errors, *Figure 5* has trained the variation of the sum of error squared sum of the PSO-BPNN-EKF algorithm.

### Discussion

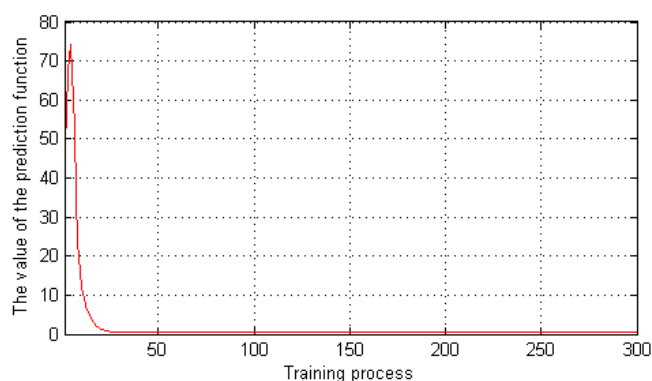
According to lab analysis, the groundwater quality in the central districts is as follows: the pH ranges in 7.6~8.3, the total hardness falls in 10.1~17.3 (°D), the salinity belongs to 0.33~0.92 g/L, and the poisonous contents like mercury, arsenic and chromium VI are minimal or non-detectable. Overall, the quality of the groundwater meets the *Standards for Drinking Water Quality* (GB5749-2006) and *Standards for Irrigation Water Quality* (GB 5084-2005). In the peripheral district, the groundwater in a few areas is not suitable for drinking or irrigation, as its hardness and salinity are higher than the specified ranges.



**Figure 3.** Comparison of neural network water resources data volume learning and testing



**Figure 4.** Comparison of data prediction results between LSSVM-PSO and PSO-BPNN-EKF algorithm



**Figure 5.** Variation of square sum of errors in training neural network

As shown in *Table 2*, the groundwater in the study area has simple chemical compositions. In the loess area, the groundwater is of the  $\text{HCO}_3\text{-CaMg}$  type with a salinity below 0.5 g/L; in the plain areas, the groundwater mainly belongs to the  $\text{ClSO}_4\text{-N}_8$  type with a salinity between 1.0 g/L and 18.46 g/L.

As shown in *Table 3*, when the number of particles  $N = 100$ , the PSO-SVM achieved a much better performance than the other algorithms, as evidenced by the extremely low error. This means the PSO optimization of EKF state equations increases the particle diversity and enhances the filtering accuracy. Overall, the proposed PSO-BPNN-EKF and PSO-SVM boasted the best performance, followed by the PSO-BPNN-EKF, the EKF and the LSSVM. The ranking demonstrates the effectiveness of the proposed algorithm.

## Conclusions

Real-time monitoring of groundwater resources with multiple sensors is a difficult dynamic control problem, due to the large scale and heterogeneity of multi-source data. The existing models cannot satisfactorily reduce the dimensions of the monitoring data. Therefore, this paper attempts to create an innovative approach for dynamic monitoring

of groundwater based on large data. For this purpose, a new data fusion algorithm was proposed, aiming to reduce the data size and energy consumption in the fusion process. Inspired by the superiority of artificial neural networks, the rough set theory and the PSO were employed to reduce the complexity of the proposed algorithm. Through several simulations, it is proved that the proposed algorithm can process data more efficiently than the BPNN and the PSO, two popular data processing methods. The excellent performance is attributable to the reduction of data table in the case of big data. The research findings shed new light on the fusion of remote sensing data collected by multiple sensors. Further studies are expected to understand the connotation of the problem of big data fusion, in the era of big data, the analysis and mining for the multi-source remote sensing data is a research field and which attracts much attention. To effectively learn the characteristics of massive, low-quality, heterogeneous, high-dimensional and fast-changing big data, there are still a series of problems and challenges. Our study provides a corresponding multi-source remote sensing data fusion algorithm for the incompleteness of multimodal data, real-time processing and multi-source data fusion.

**Acknowledgements.** This work is partially supported by the National Natural Science Foundation of China (Grant No. U1501253, No. 51651901); Natural Science Basic Research Plan in Shaanxi Province of China (2017NY-134, 2016KJXX-62, 2016NY141, 2017NY132); Funding Project for Department of Yulin University (16GK24) and Thanks for the help.

## REFERENCES

- [1] Anand, A, Galelli, S, Lakshminarayanan, S, Sundaramoorthy, S. (2013): Coordinating multiple model predictive controllers for the management of large-scale water systems. – *Journal of Hydroinformatics* 15(2): 293-305.
- [2] Awan, U. K., Tischbein, B., Martius, C. (2013): Combining hydrological modeling and GIS approaches to determine the spatial distribution of groundwater recharge in an arid irrigation scheme. – *Irrigation Science* 31(4): 793-806.
- [3] Babaoğlu, İ., Findik, O., Ülker, E. (2010): A comparison of feature selection models utilizing binary particle swarm optimization and genetic algorithm in determining coronary artery disease using support vector machine. – *Expert Systems with Applications* 37(4): 3177-3183.
- [4] Bhoi, A. K. (2017): Classification and clustering of Parkinson's and healthy control gait dynamics using LDA and K-means. – *International Journal Bioautomation* 21(1): 19-30.
- [5] Costantini, M., Farina, A., Zirilli, F. (1997): The fusion of different resolution SAR images. – *Proceedings of the IEEE* 85(1): 139-146.
- [6] Hu, T., Lv, J., Xie, Q. S., Sun, H., Yuan, Q. N. (2017): A novel human behaviour information coding method based on eye-tracking technology. – *Traitement du Signal* 34(3-4): 153-173.
- [7] Jalalkamali, A., Sedghi, H., Manshouri, M. (2011): Monthly groundwater level prediction using ANN and neuro-fuzzy models: a case study on Kerman plain, Iran. – *Journal of Hydroinformatics* 13(4): 867-876.
- [8] Li, B., Zhang, C., Han, C., Bai, B. X. (2018): Fingertip data fusion of Kinect v2 and leap motion in unity. – *Ingénierie des Systèmes d'Information* 23(6): 143-159.
- [9] Li, M. X., Liao, R. Q., Dong, Y. (2018): A new BP neural network model for the prediction problem of equally spaced time sequences and its application. – *NeuroQuantology* 16(6): 28-32.

- [10] Li, Z. J. (2017): Application of neural network technology in machining error recovery. – *Academic Journal of Manufacturing Engineering* 15(3): 6-11.
- [11] Liu, B. L., Xu, X. W. (2017): A power system active power network loss based calculation method on partial priority clustering algorithm. – *Review of Computer Engineering Studies* 4(1): 17-21.
- [12] Neelapu, R., Devi, G. L., Rao, K. S. (2018): Deep learning based conventional neural network architecture for medical image classification. – *Traitement du Signal* 35(2): 169-182.
- [13] Pohl, C., Van Genderen, J. L. (1998): Multisensor image fusion in remote sensing: concepts, methods and applications. – *International Journal of Remote Sensing* 19(5): 823-854.
- [14] Quiroga, V. M., Popescu, I., Solomatine, D. P., Bociort, L. (2013): Cloud and cluster computing in uncertainty analysis of integrated flood models. – *Journal of Hydroinformatics* 15(1): 55-70.
- [15] Ryu, H. R., Huber, M. (2007): A particle filter approach for multi-target tracking. – *IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 29 - November 2, 2007, San Diego, CA.
- [16] Üstün, B., Melssen, W. J., Oudenhuijzen, M., Buydens, L. M. C. (2005): Determination of optimal support vector regression parameters by genetic algorithms and simplex optimization. – *Analytical Chimica Acta* 544(1): 292-305.
- [17] Yu, F., Lu, W., Li, P., Xin, X., Li, J. (2012): Dynamic optimal control for groundwater optimization management with covariates. – *Journal of Hydroinformatics* 14(2): 386-394.
- [18] Zhang, F. (2011): Research on water-saving irrigation automatic control system based on internet of things. – *Proceedings of ICEICE 2011*: 2541-2544.
- [19] Zhang, F., Xue, H. F., Xu, D. S., Zhang, Y. H., You, F. (2013): Big data cleaning algorithms in cloud computing. – *International Journal of Online Engineering* 9(3): 77-81.
- [20] Zhang, K., Huang, H. P., Yang, H. T., Xie, Q. (2010): A transformer fault diagnosis method integrating improved genetic algorithm with least square support vector machine. – *Power System Technology* 34(2): 164-168.