Hydrological big data prediction based on similarity search and improved BP neural network

Dingsheng Wan[1], Yan Xiao[1], Pengcheng Zhang[1], Hareton Leung[2]
[1]College of Computer and Information, Hohai University, Nanjing, P.R.China 210098
[2]Department of Computing, Hong Kong Polytechnic University, HongKong, China
Email: pchzhang@hhu.edu.cn, hhu_xiaoyan@163.com

***Abstract*—Large amount of hydrological data set is a kind of big data, which has much hidden and potentially useful knowledge. Hydrological prediction is important for the state flood control and drought relief. How to forecast accurately and timely with hydrological big data becomes a big challenge. There are some forecasting techniques used widely. However, they are limited by their adaptability, the data volume and the data feature. The most important problems are the high time consumption, low accuracy and bad adaptability of prediction.**

In this paper, a new forecasting approach based on an integration of two tasks of data mining is put forward. This approach which is called S_LMDBP combines similarity search and Levenberg-Marquardt(LM) algorithm improved Double-hidden-layer Back Propagation(BP) neural network. A specialized data pretreatment including three parts is applied to process the hydrological data. The results of similarity search are then input into the improved BP neural network, which not only reduces dimensionality of training data without losing important patterns, but also improves the accuracy of the prediction. A set of experiments are conducted to validate the proposed approach. The data in the experiment is the daily water level data of three stations in Jiangxi province of China from 1950 to 2010. The experimental results demonstrate the real-timing, accuracy and robustness of our approach.

***Index Terms*—Hydrological prediction; Hydrological big data; Data mining; Similarity search; Improved BP neural network**

I. Introduction

With the rapid development of science and technology, there are large amounts of data generated in real life. These big data come from all areas of human life and most of them are stored in computer. For example, scientific measurement system stored huge information of observation; stock market accumulated long-term information of stock; hydrological data base stored abundant historical hydrological data. Hydrological data (water level data) may be recorded every half hour even every fifteen minutes. One observation station may generate 96 records in one day and 35136 records in one year. There are already over eighty thousands stations in China. These data are more abundant than what we can imagine. They are changed with time and stored as time series. Time series, that have large data scale and update very frequently, reflect the characteristic of the attribute value related with time [19].

If there are no effective techniques to understand and analyze these stored big data, these data cannot be fully utilized. Data Mining is proposed to discover useful knowledge from big data. It extracts hidden useful information and knowledge which people are interested in and which is unknown in advance from vast, incomplete, noisy, obscure and random data of actual application [13]. Time series data mining (TSDM) [10] analyzes local feature of time series by different techniques of data mining to mine inherent laws.

The task of data mining is to find data patterns. Generally it can be divided into two types: description and prediction [13]. The description task is to describe the characteristic of data normatively. The prediction task is to extrapolate and explore the phenomenon that may appear in the future based on the past and present relevant facts and data. There are many prediction approaches in hydrology. The traditional physically based hydrological models just suit the specific basin, whose flexibility is not satisfactory. The performance of grey system [8] is affected by the discrete degree of data. Support vector machine (SVM) [4] is limited by the size of data base.

However, most studies separate the two tasks of data mining. In fact, they can be combined to study. The description can be used in the task of prediction, especially the useful results generated by the description task, such as similarity results and association rules of historical data. This paper aims to combine two tasks of data mining in the area of hydrological time series. Firstly, we extract hydrological rules from historical data by description task. Secondly, we predict future water level by forecasting approaches based on the extracted rules.

In summary, the contributions of this paper are described as follows:

- A specialized data pretreatment that contains three parts (data selection, data cleaning and data conversion) is conducted to preprocess the wrong and noisy hydrological data.
- A new approach called S_LMDBP is proposed to forecast hydrological big data. Similarity search and BP neural network improved by LM algorithm that has double hidden layers are combined to forecast the water level.
- Four kinds of evaluation criteria are applied to evaluate the forecasting approaches.

The rest of this paper is organized as follows. Section 2 discusses some common techniques about prediction of hydrological time series. In Section 3, we introduce our approach that includes similarity measurement and BP neural network improved by Levenberg-Marquardt that has double hidden layers. We experimentally validate the new approach based

Pengcheng Zhang is Corresponding Author

on the daily water level data of real hydrological stations in Section 4. Section 5 contains our conclusions and suggestions for future work.

II. Related work

In hydrology, the most known prediction approaches are traditional physically based hydrological models, such as Xinanjiang Model [3]. However, these models can just be applied to specific watershed. They have a unique corresponding relationship with each other, and are demanding for hydrological knowledge that mastered by some hydrological experts. However, the approaches of prediction based on hydrological time series that is easy to understand are more popular. They pay more attention to the historical big data. The most widely used prediction approaches based on time series are described as follows.

A. Multiple linear regression (MLR)

MLR [5] utilizes linear relation to fit the relationship between multiple independent variable and dependent variable, then calculates the parameter of regression equation. After that, regression equation can be determined. Finally, this regression equation will be applied to forecast the dependent variable.

Many studies improve MLR in the part of calculating the parameter, such as Least Square Method, Gradient Descent, batch Gradient Descent, incremental Gradient Descent and so on. But MLR is more suitable for the prediction of linear sequence. When the time series are complex and nonlinear, such as water level data, its performance degrades.

B. Grey system

The grey system theory is an approach to deal with situations that are full of uncertain and insufficient information. The key feature of grey system is modeling with less data. Therefore, the grey system has no particular requirement and limit about experimental data, which provides a broad application field.

The principle of grey system is to deal with data indirectly by accumulating generation operator to discover the hidden rules of the data. GM(1, 1) is the primary method of grey system. But GM(1, 1) is basically designed to process unidirectional data and it is not applicable to omnidirectional data [15]. For example, it can forecast by water level or flow data in hydrology, but it cannot consider other factors simultaneously, such as rainfall. Li et al [15] proposed EGM(1, 1) to deal with omnidirectional data by adding improved inverse accumulating generation operator. However, the accuracy of prediction is affected by grey size of data which is affected by the discrete degree of data. If the discrete degree is greater, the grey size will be bigger. Then the accuracy of prediction will be lower.

C. Support vector machine (SVM)

SVM was originally proposed by Ma [4]. It is based on VC theory and structural risk principle of statistical learning theory [22]. SVM switches sample space to higher even infinite dimensional feature space through nonlinear transformation defined by kernel function. Then it finds nonlinear relationship between the input and output variables in this space. At first, SVM was used in pattern classification [18]. Later it was applied to nonlinear regression estimate. Vapnik [17] used it in the prediction of financial time series, which had a good performance. Hu [6] applied SVM to the prediction of flow in semi-arid and semi-humid area, which solved practical problems of nonlinear and small sample. However, for big sample set, the number of support vectors returned by SVM is large. The computing cost in classification decisions is high. So SVM is incapable of implementing large-scale training sample. When the number and randomness of time series is large, the results of prediction are unacceptable and unstable.

D. Artificial neural network (ANN)

Artificial neural network (ANN) [20] is a technique based on the operation of biological neural networks. It is the abstract and simulation of human brain and natural neural network. Because of its robustness and flexibility, ANN attracts a great deal of attention in the application of hydrological forecasting [2]. The applicability of ANN in Hydrology has been extensively evaluated by the American Society of Civil Engineers (ASCE) task committee on application of artificial neural networks in Hydrology [11]. These studies validate that ANN is an efficient alternative to traditional physically based hydrological models. It has a good application prospect.

Zhang [21] applied regularized RBF network model in the groundwater level prediction. They found that the forecasting results were good when the time series were small sample. But with the increasing of the sample scale, the structure of the model would be very large, which meant it was not suit for the training and forecasting of vast water level data. BP neural network was used by Mohanty [16] to predict groundwater level in a river island of Eastern India. Carcano et al [7] modelled daily streamflows by BP that has two hidden layers, which showed two-hidden-layer BP might enhance the nonlinear mapping ability between input and output.

III. Similarity search and LMDBP

These existing approaches of hydrological prediction have a common problem of superabundant previous historical data. How to reduce dimensionality and fully reserve the patterns that are useful to forecast becomes a key point. In this paper, an approach combining similarity search and LM algorithm improved double-hidden-layer BP neural network (S_LMDBP) is put forward. The following are the abbreviations that will be used in the next section.

Predicting day. The day that will be predicted is called as the predicting day.

Matching sequence. The water levels of several days before the predicting day are a sequence that will be matched. We call it the matching sequence.

Searching sequence. The daily water levels of the previous years and the water levels before the predicting day in the same

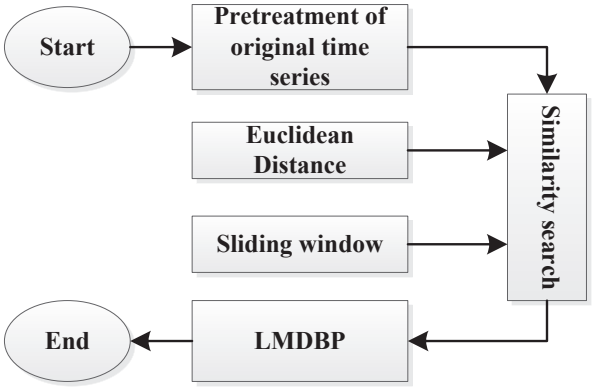


Fig. 1: The overview of S_LMDBP

Fig. 2: The flow chart of data pretreatment

year are as a sequence that will be searched. We call it the searching sequence.

Figure 1 gives an overview of S_LMDBP. Firstly, we preprocess the hydrological data to fill up the missing data, wipe off the wrong data and normalize the data. Secondly, similarity search is used to seek the similar sequences with the matching sequence in the previous historical years. Lastly, the similar sequences and the data of the corresponding later day are used as the training set to build LMDBP model.

A. Data pretreatment

In real life, one data or consecutive data are missing or wrong in hydrological water level database, which is caused by many factors, such as equipment failure, human fault and so on. Such data with low quality may affect the results of similarity search to some extent, and then influence the accuracy of prediction. So it is necessary to preprocess water level data. The processes of pretreatment contain data selection, data cleaning and data conversion, as shown in Figure 2.

Data selection: The selection of experimental data is of great significance to a successful study. In this paper, the daily average water level data of a real observation station are appropriate.

Data cleaning: The abnormal data of water level mainly include two kinds: wrong values and missing values. The wrong data can be wiped out by the criterion of 3σ. Almost all of the correct data (99.74%) are in the interval $(\mu-3\sigma, \mu+3\sigma)$, where μ is the mean value, and σ is the standard deviation. The data that are outside of the interval can be considered as the wrong data. The missing data come from missing a data or continuous data. If one data is lost, it can be replaced by the mean value of adjacent two data. If losing continuous data, the water level data in the same period of adjacent stations can be used as a reference.

Data conversion: Because of the noise and volatility of the time series, the similar series may present many kinds of transformations, such as timeline scaling, amplitude scaling, discontinuity, linear drift and so on. The normalization can eliminate the dimensional effects. Min-Max normalization is a linear transformation for the original data. Standard results are mapped into [0, 1]. The transfer function is shown as equation (1), in which max is the maximum of the data, and min is the minimum of the data.

$$x' = \frac{x - min}{max - min} \tag{1}$$

B. Similarity search

Agrawal et al [1] first published papers about the similarity search of time series in 1993. Similarity search was described as "Given certain time series, search similar series from large time series database". Similarity search is based on similarity measurement. And distance measurement is similar with similarity measurement in many fields. So similarity measurement is always replaced by distance measurement [19].

Minkowski Distance [9] defines a series of distances. There are two time series $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$. The Minkowski Distance between them is shown as the following equation.

$$L_p = D(X, Y) = [\sum_{i=1}^{n}(x_i - y_i)^p]^{1/p} \tag{2}$$

With the changes of the values of p, the definition of Minkowski Distance is different. If p=1, L_1 is called as Manhattan Distance; If p=2, L_2 is Euclidean Distance; If p=∞, L_∞ is Chebyshev Distance.

Euclidean Distance is the most widely used distance measurement of time series, which is simple and easy to calculate. The lengths of the two time series must be same when calculating the Euclidean Distance.

The following is the formula of Euclidean Distance between two given time series $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$.

$$D(X, Y) = [\sum_{i=1}^{n}(x_i - y_i)^2]^{1/2} \tag{3}$$

Euclidean Distance complies with the following three conditions [14]:

(i) *Non-negativity:* $D(X, Y) \geq 0$; Only when $X = Y, D(X, Y) = 0$.

(ii) *Symmetry:* $D(X, Y) = D(Y, X)$.

(iii) *Triangle Inequality:* $D(X, Z) \leq D(X, Y) + D(Y, Z)$.

When calculating Euclidean Distance between two time series, the corresponding elements of series are required to be aligned for comparison. If a certain degree of deviation appears in the time series, Euclidean Distance can also calculate the distance accurately and then determine whether the two time series are similar or not.

Sliding window is applied to search the sequence similar with the matching sequence more comprehensively from the searching sequences. Euclidean Distance requires that the two

time series are isometric. This feature is suitable for the demand that the input length of BP network is equal, which means that the input sequence has the same feature dimension. Therefore, this paper adopts fixed sliding window.

Specific searching process is shown in Algorithm 1. Line 2 to 6 obtain the matching sequence and normalize them. In lines 7-9, the searching sequences are got as templates. The Euclidean Distance between the matching sequence and the searching sequence in each sliding window is obtained from Line 10 to 17. In lines 18-23, the p minimum distances are calculated as the results of similarity search $Seq = (s_1, s_2, ..., s_p)$ that are the training set of the next LMDBP. There is only one ***for*** loop in this algorithm, so the time complexity is O(n)(n is the length of the searching sequence).

Algorithm 1 Similarity search

Require: The predicting day: date; the text file of water level data (the searching sequences);
double distance(double[], double[]): Euclidean Distance between two time series;

Ensure: The p time series similar with the matching sequence: $Seq = (s_1, s_2, ..., s_i, ..., s_p)$;

1: Read the text file of water level data, save data as $X = (x_1, x_2, ..., x_i, ..., x_n)$;
2: $index \leftarrow date - firstDay$; //firstDay is the date of first day in the text file
3: **for** $i = 0$ *to* m **do**
4: $cursor[m - i] \leftarrow X[index - i - 1]$; //gain the m days before the predicting day as the matching sequence
5: **end for**
6: Normalize the cursor;
7: **for** $i = 0$ *to* $index - 1$ **do**
8: $template[i] \leftarrow X[i]$; //index-1 days before the predicting day are as searching sequences(templates)
9: **end for**
10: **for** $i = 0$ *to* $template.length$ **do**
11: **for** $j = 0$ *to* m **do**
12: $window[j] \leftarrow template[i + j]$; //initialize values of sliding window
13: **end for**
14: Normalize the window;
15: $result[i].value \leftarrow distance(cursor, window)$; //calculate Euclidean Distance between two time series
16: $result[i].data \leftarrow window; i++$;
17: **end for**
18: **for** $i = 0$ *to* $result.length$ **do**
19: sort result;
20: **end for**
21: **for** $i = 0$ *to* $Seq.length$ **do**
22: Seq[i] = result[i]; //the p minimum results are as the results of similarity search
23: **end for**
24: $cout \ll Seq$;

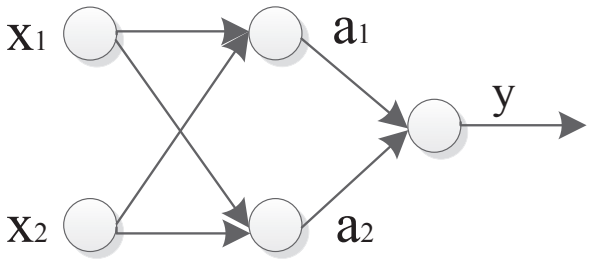


Fig. 3: The topological structure of a three-layer BP neural network

C. LM algorithm improved double-hidden-layer BP neural network (LMDBP)

(1) BP neural network

BP neural network is a multilayer feedforward neural network using the error back propagation algorithm. There are an input layer, an output layer and several hidden layers in BP. Each layer is interlinked. The neurons in each layer are not connected. The topological structure of a three-layer BP neural work is shown as Figure 3, in which x represents the input, a represents the results of hidden layer, and y represents the output.

There are two parts in the training process of BP neural network: forward process and reverse process. The forward process obtains output through input signal. The reverse process adjusts the weights and thresholds of each layer according to the error between output signal and expected output. The implementation procedure of BP contains four parts [12]: input, network training, network testing and output. The specific training steps are as follows:

Step 1: Initialize the network. Determine the number of layers and the number of neurons in each layer. Random numbers are distributed to the weights and thresholds of each layer.

Step 2: Obtain output signal. According to the initialized BP neural network, calculate output signal through input signal and distributive weights and thresholds.

Step 3: Calculate the error of output layer. Calculate the error between the gained output and expected output.

Step 4: Train the weights and thresholds of output unit. Amend the weights and thresholds between the hidden layer and the output layer by the error of output layer and the output of each neuron in the hidden layer.

Step 5: Calculate the errors of hidden layer. Calculate the errors of hidden layer according to the values of each neuron in the hidden layer and the weights of output unit.

Step 6: Train the weights and thresholds of hidden layer. Amend the weights and thresholds of the hidden layer using the error of each neuron in the hidden layer and the input signal.

Step 7: Judge whether the algorithm is over or not, according to the overall error or the number of iteration. If it does not finish, go to Step 2 to continue training. Otherwise, produce the output.

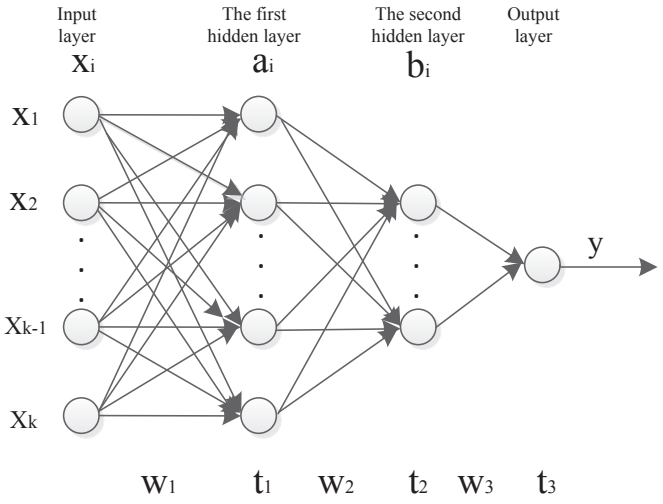


Fig. 4: The structure chart of a double-hidden-layer BP neural network

(2) Double-hidden-layer BP neural network

In this paper, a double-hidden-layer BP with multi-input and mono-output is built. The input layer includes k nodes that are used to input k signals. The first hidden layer contains m neurons that are applied to complete the space weighted aggregation and excitation output of input signals. There are n neurons in the second hidden layer that improve the ability of non-linear mapping between input and output. The output layer has a node to accomplish the output of system. The structure chart is shown as Figure 4.

The corresponding weights w and thresholds t are shown as Equation (4).

$$w_1 = \begin{bmatrix} w_1(1,1) & w_1(1,2) & ... & w_1(1,m) \\ w_1(2,1) & w_1(2,2) & ... & w_1(2,m) \\ ... & ... & ... & ... \\ w_1(k,1) & w_1(k,2) & ... & w_1(k,m) \end{bmatrix} t_1 = \begin{bmatrix} t_1(1) \\ t_1(2) \\ ... \\ t_1(m) \end{bmatrix}$$

$$w_2 = \begin{bmatrix} w_2(1,1) & w_2(1,2) & ... & w_2(1,n) \\ w_2(2,1) & w_2(2,2) & ... & w_2(2,n) \\ ... & ... & ... & ... \\ w_2(m,1) & w_2(m,2) & ... & w_2(m,n) \end{bmatrix} t_2 = \begin{bmatrix} t_2(1) \\ t_2(2) \\ ... \\ t_2(n) \end{bmatrix}$$

$$w_3 = \begin{bmatrix} w_3(1,1) \\ w_3(2,1) \\ ... \\ w_3(n,1) \end{bmatrix} \quad t_3 \tag{4}$$

The following is the relationship between each layer.

$$a_i = f(\sum_{j=1}^{k} w_1(j,i)x_j + t_1(i)), i = 1,2,...,m \tag{5}$$

$$b_i = f(\sum_{j=1}^{m} w_2(j,i)a_j + t_2(i)), i = 1,2,...,n \tag{6}$$

$$y = f(\sum_{j=1}^{n} w_3(j,1)b_j + t_3) \tag{7}$$

In the above-mentioned equations, f is the transfer function, which is usually the function of tansig.

(3) The improvement of BP neural network

BP adjusts its weights according to the principle of error gradient descent. When the error falls slowly, the number of iterations will increase, which will affect the convergence speed. What's more, it is easy to fall into local minimum. So some optimized algorithms are necessary. There are many mature optimization approaches, such as steepest descent method, Newton method, LM algorithm and so on.

Steepest descent method is used to solve a function's minimum point in BP network, which starts from a point, iterate along the negative gradient direction and then find the minimum point. However, the convergence speed of steepest descent method is slow. The basic idea of Newton method is to find the estimated value of the minimum point of f(x) through second order Taylor expansion of the objective function f(x) around minimum point. The convergence rate of Newton method is higher, but calculating second derivative and inverse matrix increases the calculation and memory capacity.

LM algorithm, which was proposed by Levenberg and Marquardt, is a combination of steepest descent method and Newton method. LM has not only high learning efficiency and fast convergence speed, but also high recognition rate. The error function E can be represented as:

$$E(W) = \frac{1}{2}\sum_{i=1}^{n_L} e_i^2(W) = \frac{1}{2}e^T(W)e(W) \tag{8}$$

$$W_{k+1} = W_k - [J^T(W_k)J(W_k) + \mu_k I]^{-1}J^T(W_k)e(W_k) \tag{9}$$

W is the vector of the weights and thresholds and e presents the error. When μ is giant, LM is steepest descent method; when μ is zero, LM is Newton method.

$$J^T(W) = \begin{bmatrix} \frac{\partial e_1(W)}{\partial W_{11}^1} & \frac{\partial e_1(W)}{\partial W_{21}^1} & \cdots & \frac{\partial e_1(W)}{\partial W_{n_1 n_L}^2} & \frac{\partial e_1(W)}{\partial \theta_1^1} & \cdots & \frac{\partial e_1(W)}{\partial \theta_{n_L}^2} \\ \frac{\partial e_2(W)}{\partial W_{11}^1} & \frac{\partial e_2(W)}{\partial W_{21}^1} & \cdots & \frac{\partial e_2(W)}{\partial W_{n_1 n_L}^2} & \frac{\partial e_2(W)}{\partial \theta_1^1} & \cdots & \frac{\partial e_2(W)}{\partial \theta_{n_L}^2} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ \frac{\partial e_{n_L}(W)}{\partial W_{11}^1} & \frac{\partial e_{n_L}(W)}{\partial W_{21}^1} & \cdots & \frac{\partial e_{n_L}(W)}{\partial W_{n_1 n_L}^2} & \frac{\partial e_{n_L}(W)}{\partial \theta_1^1} & \cdots & \frac{\partial e_{n_L}(W)}{\partial \theta_{n_L}^2} \end{bmatrix} \tag{10}$$

$J^T(W)$ denotes Jacobian matrix, which is the first derivative of error function.

The detailed computational process of LM is described as follows.

Step 1: Given the permissible error ε, the constants μ_0 and $\beta(0 < \beta < 1)$. Initialize the weights and thresholds W. Set $k = 0, \mu = \mu_0$;

Step 2: Calculate the output of BP and the error function $E(W_k)$;

Step 3: Compute Jacobian matrix $J(W_k)$;

Step 4: Count ΔW;

Step 5: If $E(W_k) < \varepsilon$, go to Step 7;

Step 6: Set $W_{k+1} = W_k + \Delta W$ as the new weights and thresholds. Calculate the error function $E(W_{k+1})$. If $E(W_{k+1}) < E(W_k)$, set $k = k+1, \mu = \mu\beta$, then go to Step 2. Otherwise, set $\mu = \mu/\beta$ then go to Step 4;

Step 7: The LM algorithm is over.

The advantage of LM is that the convergence speed is very fast when the number of network weights is less, which shortens the learning time. LM has good performance in practical application. It has higher convergence speed and fewer times of training. Therefore, this paper adopts LM to improve BP neural network.

IV. Experimental evaluation

In this section, a set of experiments are conducted to show the low time consumption and high accuracy of our approach (S_LMDBP) by comparing it with single BP neural network (BP) and double-hidden-layer BP neural network improved by LM (LMDBP). The experimental data are the daily water level data of Waizhou Station in Jiangxi Province of China from 1950 to 2010. Then this paper proves the good adaptability of S_LMDBP based on the daily water level data of Xingzi Station and Nanchang Station from 1950 to 2010. The experiments are designed to answer the following questions.

- REQ 1: How much time do the three forecasting approaches cost?
- REQ 2: What is the accuracy of the three forecasting approaches?
- REQ 3: How is the adaptability of the new approach S_LMDBP?

Experimental setup and parameter selection: The experimental environment is a Windows PC with Acer Intel Pentium G2030 CPU4G RAM. We use Java language to implement similarity search and Matlab to execute ANN. To answer the above first two questions, we predict the water level from May 1st to 31st in 2010 using the previous days from 1950 to 2010 of Waizhou Station. Then we predict the water level of Xingzi Station from September 1st to 30th in 2010 and Nanchang Station from January 1st to 31st in 2010 to answer the third question. The training sets of BP and LMDBP are the water level data before the predicting day from 1950 to 2010. The training set of S_LMDBP is the results of similarity search that include 1200 time series. The parameters of each forecasting model are presented as follows, which are determined by abundant experiments.

(i) Single BP neural network (BP): The input number of each training set is 15 and the output number is 1. The structure of neural network is 15-4-1. The transfer function between the input layer and the hidden layer is tansig. The transfer function between the hidden layer and the output layer is purelin.

(ii) LM improved double-hidden-layer BP neural network (LMDBP): The input number of each training set is 15 and the output number is 1. The structure of neural network is 15-15-1-1. The transfer function between the input layer and the first hidden layer is tansig. The transfer function between the first hidden layer and the second hidden layer is purelin. The transfer function between the second hidden layer and the output layer is purelin.

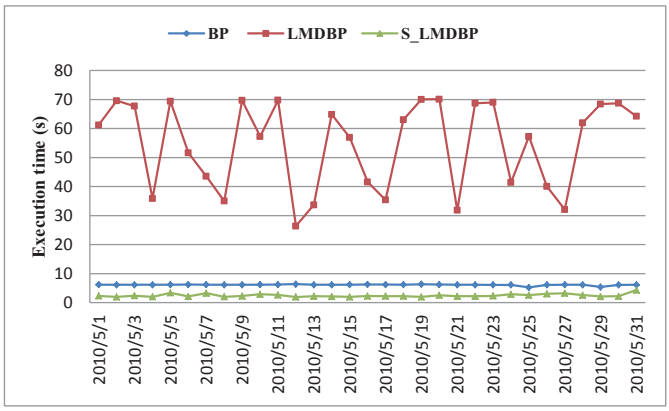


Fig. 5: The execution time of three forecasting approaches

(iii) Similarity search and LM improved double-hidden-layer BP neural network (S_LMDBP): The input number of each training set is 15 and the output number is 1. The structure of neural network is 15-15-1-1. All transfer functions are tansig.

Experimental results:

REQ 1: How much time do the three forecasting approaches cost?

Real-time is a key factor to evaluate the efficiency of prediction approaches. Even if the forecasting results are accurate, we cannot use them if they are delayed. Figure 5 gives the execution time of the three approaches. The execution time of S_LMDBP contains the time of similarity search and LMDBP, but as shown in Figure 5, the run-time of S_LMDBP is still shorter than the other two approaches. The main reason of this phenomenon is that the time complexity of similarity search is O(n) which is linear and the training set of S_LMDBP is reduced. It just has 1200 time series that are similar to the matching sequence, which is far less than the training set of BP and LMDBP. The more accurate the training set is, the less number of iterations will be. Then the time consumption will be lower.

REQ 2: What is the accuracy of the three forecasting approaches?

Because the initial weights and thresholds of neural network are generated randomly, the forecasting results are not stable. To better show the accuracy of our new approach, we train BP and LMDBP for five times to choose the best forecasting result as the final result, and train S_LMDBP for ten times to set the mean value of the ten results as the final forecasting result.

There are four kinds of evaluation criteria to estimate the accuracy of forecasting approaches.

(i) Mean Absolute Error

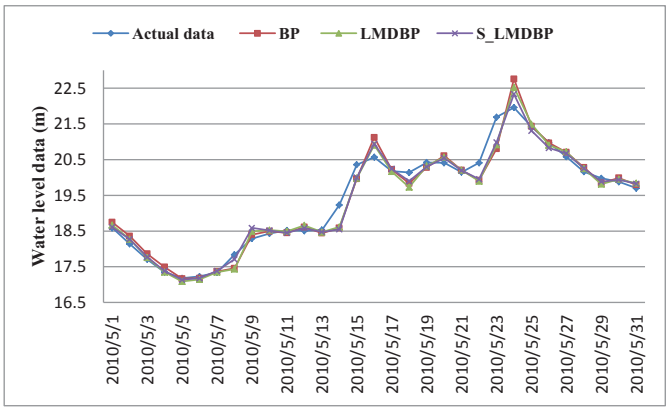


Fig. 6: The results of the three forecasting approaches

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widetilde{y}_i| \qquad i = 1, 2, ..., n \tag{11}$$

(ii) Mean Relative Error

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \widetilde{y}_i|}{y_i} \qquad i = 1, 2, ..., n \tag{12}$$

(iii) Root Mean Square Error

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widetilde{y}_i)^2} \qquad i = 1, 2, ..., n \tag{13}$$

y_i represents the actual water level and $\widetilde{y}_i$ represents the forecasting result.

(iv) Qualified rate of prediction

There are three kinds of errors: forecasting error ($|y_i - \widetilde{y}_i|$), measurement error ($y_i * 5\%$), and amplitude error ($|y_i - y_{i-1}| * 20\%$). y_i represents the water level of the predicting day and y_{i-1} represents the water level of the day before the predicting day. If the measurement error is larger than the amplitude error, the permissible error is the measurement error; otherwise it is the amplitude error. If the forecasting error is in the range of the permissible error, it means that the forecasting result is qualified. The qualified rate of prediction P is calculated by the following formula.

$$P = \frac{The\ number\ of\ qualified\ forecasting\ results}{The\ whole\ number\ of\ forecasting\ results} * 100\% \tag{14}$$

The prediction results of the three forecasting approaches are presented in Figure 6. The accuracy of the three forecasting approaches are shown as Table I. We also draw a boxplot of forecasting error in Figure 7 to help to visualize the prediction performance of the three approaches.

The lower the first three errors (MAE, MRE and RMSE) are, the better the prediction performs. As we can see in Table I, the three errors of S_LMDBP are lower than the other two approaches, which means S_LMDBP has a better forecasting performance. From Figure 6, the three approaches can predict

Prediction model	MAE	MRE	RMSE	P
BP	0.2154	0.0108	0.3143	100%
LMDBP	0.1947	0.0098	0.2780	100%
S_LMDBP	0.1735	0.0087	0.2484	100%

TABLE I: Accuracy of the three forecasting approaches

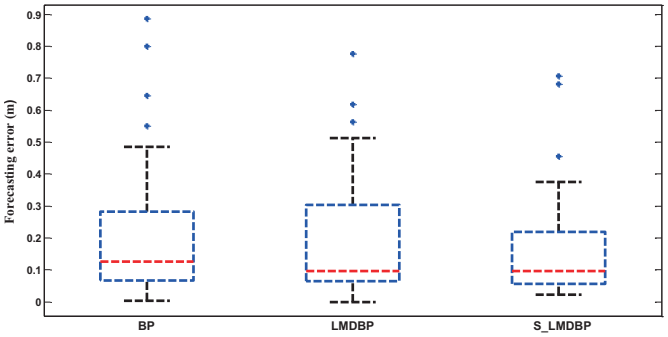


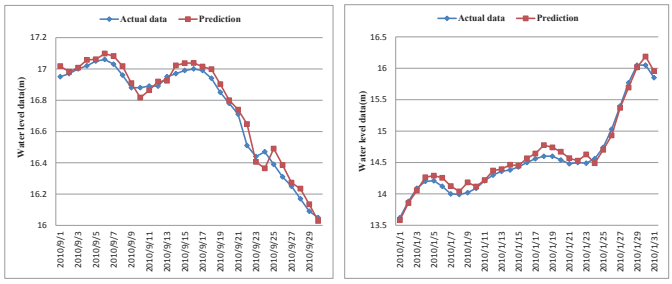
Fig. 7: The boxplot of forecasting error of three forecasting approaches

the overall trend of the water level. And the qualified rates P of the three approaches are all 100%, because we just predict a few days. On the other hand, this phenomenon verifies that ANN is an efficient technique for hydrological prediction. From the boxplot in Figure 7, the smaller the height of box is, the more concentrated is the forecasting error, which means this forecasting approach is more stable. S_LMDBP is more stable than BP and LMDBP. BP has four outliers. LMDBP and S_LMDBP just have three outliers. But the box's top edge of S_LMDBP is below LMDBP. And the maximum forecasting error of S_LMDBP is lower than LMDBP. Therefore, the new approach S_LMDBP is better.

REQ 3: How is the adaptability of the new approach S_LMDBP?

On top of the execution time and accuracy of the forecasting approach, the adaptability has been a focus of attention. A good prediction model should be stable, which means it can be applied in many stations. To demonstrate the adaptability of our approach S_LMDBP, another two stations are chosen to forecast. We predict the water level data of Xingzi Station from September 1st to 30th in 2010 and Nanchang Station from January 1st to 31st in 2010 using the previous days from 1950 to 2010. Figure 8 gives the forecasting results by S_LMDBP of the two stations. It reveals that S_LMDBP can still predict the general trend of the water level. To be more intuitive, Table II reflects the accuracy of S_LMDBP by calculating MAE, MRE, RMSE and P. We can see that the first three errors are low and the P is always 100%. The charts mentioned in the above verify the good adaptability of S_LMDBP.

In conclusion, from the above experiments, the new proposed forecasting approach S_LMDBP costs less time and has higher accuracy than BP and LMDBP. Its real-timing, accuracy and robustness have been validated.



(a)The prediction of Xingzi Station (b)The prediction of Nanchang Station

Fig. 8: The prediction of S_LMDBP

Station	MAE	MRE	RMSE	P
Xingzi	0.0461	0.0028	0.0545	100%
Nanchang	0.0776	0.0053	0.0903	100%

TABLE II: Accuracy of S_LMDBP

Threats to validity: Although experimental results reveal the real-time performance and accuracy of our approach, there are still some threats to validity.

Firstly, in ANN, the number of input, the number of neurons in the hidden layer and the transfer function between each layer are determined through abundant experiments. Wrong choice may lead to imprecise results.

Secondly, because the initial weights of neural network are generated randomly, the forecasting results of three approaches are not stable. Even though we take actions to relieve this phenomenon, some bad forecasting results may appear. However, through a large number of experiments, the performance of our new approach is always better than the other two approaches.

V. Conclusions and suggestions for future work

This paper proposes a new forcasting approach (S_LMDBP) for hydrological big data that combines similarity search and ANN. After conducting a specialized data pretreatment, the dimensionality of training set is reduced by similarity search, which removes noisy and redundant information. Finally, the water level of the predicting day is forecasted by LMDBP. A set of experiments are performed to validate that S_LMDBP has the lower time consumption and higher accuracy comparing to other approaches.

However, there are some issues remained to be further studied in the future. On one hand, a more simple and scientific approach to determine the parameters of LMDBP that conforms to the characteristic of hydrology and ANN need be researched. On the other hand, even though the forecasting results are satisfactory, there are several days whose predictions by the three approaches are not very good. Maybe we can add some other factors to help to forecast, such as the rainfall of the basin.

VI. Acknowledgements

The work is supported by the Special Scientific Research of Public Welfare Profession of China (Nos. 201501022), whose name is flood forecasting research of the small and medium-sized rivers based on data mining and driving, and the National Natural Science Foundation of China under Grant (Nos. 61202097 and 61370091)

References

[1] R. Agrawal, C. Faloutsos, and A. Swami, *Efficient similarity search in sequence databases*. Springer, 1993.

[2] S. Araghinejad, M. Azmi, and M. Kholghi, "Application of artificial neural network ensembles in probabilistic hydrological forecasting," *Journal of Hydrology*, vol. 407, no. 1, pp. 94–104, 2011.

[3] H. Bao, L. Zhao, Y. He, Z. Li, F. Wetterhall, H. Cloke, F. Pappenberger, and D. Manful, "Coupling ensemble weather predictions based on tigge database with grid-xinanjiang model for flood forecast," *Advances in Geosciences (ADGEO)*, vol. 29, pp. 61–67, 2011.

[4] M. Blatt, S. Wiseman, and E. Domany, "Superparamagnetic clustering of data," *Physical review letters*, vol. 76, no. 18, p. 3251, 1996.

[5] S. R. Bonellie, "Use of multiple linear regression and logistic regression models to investigate changes in birthweight for term singleton infants in scotland," *Journal of clinical nursing*, vol. 21, no. 19, pp. 2780–2788, 2012.

[6] H. Caihong, G. Jing, Z. Yeyu, M. Dong, and Z. Qingshan, "Applied research on support vector machine in hydrological forecast in semi-arid and semi-humid area," *Meteorological and Environmental Sciences*, vol. 2, p. 002, 2010.

[7] E. C. Carcano, P. Bartolini, M. Muselli, and L. Piroddi, "Jordan recurrent neural network versus ihacres in modelling daily streamflows," *Journal of hydrology*, vol. 362, no. 3, pp. 291–307, 2008.

[8] J. Deng, "The primary methods of grey system theory," *Huazhong University of Science and Technology Press, Wuhan*, 2005.

[9] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[10] P. Esling and C. Agon, "Time-series data mining," *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, p. 12, 2012.

[11] R. S. Govindaraju, "Artificial neural networks in hydrology. ii: hydrologic applications," *Journal of Hydrologic Engineering*, vol. 5, no. 2, pp. 124–137, 2000.

[12] Z. Guo, J. Wu, H. Lu, and J. Wang, "A case study on a hybrid wind speed forecasting method using bp neural network," *Knowledge-based systems*, vol. 24, no. 7, pp. 1048–1056, 2011.

[13] M. Kantardzic, *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.

[14] E. M. Knorr and R. T. Ng, "A unified notion of outliers: Properties and computation." in *KDD*, 1997, pp. 219–222.

[15] D. Li, C. Chang, W. Chen, and C. Chen, "An extended grey forecasting model for omnidirectional forecasting considering data gap difference," *Applied Mathematical Modelling*, vol. 35, no. 10, pp. 5051–5058, 2011.

[16] S. Mohanty, M. K. Jha, A. Kumar, and K. Sudheer, "Artificial neural network modeling for groundwater level forecasting in a river island of eastern india," *Water resources management*, vol. 24, no. 9, pp. 1845–1865, 2010.

[17] V. Vapnik, *The nature of statistical learning theory*. springer, 2000.

[18] V. Vapnik, S. E. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," *Advances in neural information processing systems*, pp. 281–287, 1997.

[19] D. Wan, Y. Xiao, P. Zhang, J. Feng, Y. Zhu, and Q. Liu, "Hydrological time series anomaly mining based on symbolization and distance measure," in *Big Data (BigData Congress), 2014 IEEE International Congress on*. IEEE, 2014, pp. 339–346.

[20] W. Wei and D. W. Watkins Jr, "Data mining methods for hydroclimatic forecasting," *Advances in Water Resources*, vol. 34, no. 11, pp. 1390–1400, 2011.

[21] Z. Yinqin, L. Junmin, H. Jian *et al.*, "Application of regularized rbf network model in the groundwater level prediction," *Journal of Northwest Agriculture and Forestry University*, 2011.

[22] H. Yoon, S. Jun, Y. Hyun, G. Bae, and K. Lee, "A comparative study of artificial neural networks and support vector machines for predicting groundwater levels in a coastal aquifer," *Journal of Hydrology*, vol. 396, no. 1, pp. 128–138, 2011.