Mid- to long-term runoff prediction by combining the deep belief network and partial least-squares regression
Zhaoxin Yue, Ping Ai, Chuansheng Xiong, Min Hong and Yanhong Song

ABSTRACT

Data representation and prediction model design play an important role in mid- to long-term runoff prediction. However, it is challenging to extract key factors that accurately characterize the changes in the runoff of a river basin because of the complex nature of the runoff process. In addition, the low accuracy is another problem for mid- to long-term runoff prediction. With an aim to solve these problems, two improvements are proposed in this paper. First, the partial mutual information (PMI)-based approach was employed for estimating the importance of various factors. Second, a deep learning architecture was introduced by using the deep belief network (DBN) with partial least-squares regression (PLSR), together denoted as PDBN, for mid- to long-term runoff prediction, which solves the problem of parameter optimization for the DBN using PLSR. The novelty of the proposed method lies in the key factor selection and a novel forecasting method for mid- to long-term runoff. Experimental results demonstrated that the proposed method can significantly improve the effect of mid- to long-term runoff prediction. Also, compared with the results obtained by current state-of-the-art prediction methods, i.e., DBN, backpropagation neural networks, and support vector machine models, our prediction results demonstrate the performance of the proposed method.

Key words | comprehensive runoff index, deep belief network, mid- to long-term runoff prediction, partial least-squares regression, partial mutual information

HIGHLIGHTS

- Constructing the comprehensive runoff index to perform the comprehensive characterization of runoff changes in the whole watershed.
- Applying the partial mutual information method for factor selection.
- The PDBN is used as a deep learning approach for the mid- to long-term runoff prediction.
- The proposed method can significantly improve the effect of the mid- to long-term runoff prediction.
INTRODUCTION

The rapid development of information technology, especially the theory of data science and soft computing techniques, has significantly influenced the development and application of hydrology (Chen & Han 2016; Chen & Wang 2018; Ma et al. 2018), such as drought prediction (Shamshirband et al. 2020), flood prediction and management (Wang et al. 2017; Fotovatikhah et al. 2018; Mosavi et al. 2018), river flow forecasting (Bai et al. 2016b; Yaseen et al. 2019a, 2019b), and runoff forecasting (Valipour 2015; Tan et al. 2018). Among them, the mid- to long-term runoff prediction is an important research direction in hydrological forecasting and hydroinformatics. In general, the amount of basic hydrological data necessary for hydrological forecasting is increasing, both in terms of the amount of data and the number of data modes, which provides informative factors for the mid- to long-term runoff prediction. However, the increasing amount of hydrological data can introduce redundant and noisy information to the prediction feature or factor, which may deteriorate the performance of the mid- to long-term runoff prediction. At this time, traditional methods face difficulty in adapting to hydrological data analysis and runoff prediction due to insufficient noise removal and representation learning capabilities.

Data representation plays an important role in mid- to long-term runoff prediction. However, it is challenging to extract key factors that accurately characterize the changes in the runoff of a river basin because of the complex nature of the runoff process, which seriously affects the speed and accuracy of runoff prediction. Over the past decades, input factors have also evolved from considering only previous rainfall and runoff to multiple factors including runoff, rainfall, climate, and others. How to select a suitable set of key factors from the above factors is still difficult in mid- to long-term runoff prediction. In order to select key factors, some methods have been applied in runoff prediction, mainly including prior knowledge method (Huang et al. 2017), correlation coefficient method (Li et al. 2015), principal component analysis (PCA; Li et al. 2012), and partial mutual information (PMI; Fernando et al. 2009). Among them, the prior knowledge method is seldom applied due to the large amount of calculation. The methods of correlation coefficient and PCA are commonly used to describe the linear relationship between the impact factors, but difficult to characterize the complex nonlinear relationship, which limits the applicable scope of these general methods. The PMI method is based on information entropy theory, which has the advantages of characterizing relationships of linear and nonlinear among factors, and has been widely used in many fields.

Prediction model design is another important research content in mid- to long-term runoff prediction. At present, some various advanced data-driven methods have been proposed to improve the effect of mid- to long-term runoff prediction, such as the wavelet transforms (Jiang et al. 2012).
support vector machines (Zhao et al. 2017), artificial neural networks (Alizadeh et al. 2017; Shoaib et al. 2018; Yaseen et al. 2019a, 2019b), and Bayesian statistic prediction models (Ma et al. 2013; Chu et al. 2017). Data-driven models have provided a useful tool for solving the watersheds with too complex and unknown runoff production. However, the mid- to long-term runoff prediction is low accuracy because of the highly nonlinear complex characteristics of mid- to long-term runoff process in the watershed. Moreover, an extremely high dimensional feature will be extracted if multiple factors are modeled, which significantly increases the scale of the prediction model and calculation complexity.

The important point for solving the above problems is to extract key factors that are closely related to the runoff in a river basin and design prediction model. Aiming to improve the mid- to long-term prediction of the runoff process, this study comprehensively introduces multiple hydrological factors to construct the prediction feature. Moreover, aiming to solve the problems caused by the highly nonlinear complex characteristics of mid- to long-term runoff forecasting in the watershed, and the shallow network is degenerated for the mid- to long-term runoff forecasting task, a deep belief network (DBN) combined with partial least-squares regression (PDBN) is established to learn the most representative key factors for mid- to long-term runoff prediction, which solves the problem of weight optimization for the DBN using PLSR. The main contribution of this study is three-fold:

1. Investigating the consistency of runoff at different hydrological stations for constructing the COM according to granulation computing to perform the comprehensive characterization of runoff changes in the whole watershed.
2. Generating key factors for runoff prediction on the basis of the PMI method combined with manual selection.
3. Applying to the mid- to long-term runoff prediction with the proposed PDBN based on the above key factors. The novelty of the proposed PDBN lies in: (i) an adaptive learning rate is introduced to accelerate the convergence rate in the unsupervised learning stage of the DBN and (ii) the PLSR is used to determine the optimal weights in the supervised learning stage to improve prediction accuracy.

A flowchart of our proposed method is shown in Figure 1.

The remainder of this paper is structured as follows: The introduction and review of literature are presented in this section. The next two sections describe the methodology and the proposed model PDBN used in this study, respectively. Moreover, the performance indices used to evaluate the models developed in this study is also reported in the section ‘Methodology’. The data used in the study are described in the section ‘Case Study’, and results of the different developed models are elaborated in the section ‘Case Study’. The conclusions are provided in the final section.

**METHODOLOGY**

The mid- to long-term runoff prediction method presented in this study consists of three parts: construction of COM, factor reduction using PMI, and DBN. In the following section, these parts will be elaborated in detail.

**Construction of COM**

The main influencing factors of the runoff change within the basin include rainfall, climate, and other factors. Unlike the previous research works, we investigate the consistency of runoff at different hydrological stations for constructing the COM according to granulation computing (Yao et al. 2013), to characterize the abundance and drought of runoff in the basin. According to the characteristics of mid- to long-term runoff forecasting, we choose the monthly scale as the time granularity. Specifically, the study includes two parts: the average runoff consistency analysis based on information from different hydrological stations and the construction of the COM.

**Average runoff consistency analysis based on information from different hydrological stations**

Firstly, the runoff consistency analysis considering information from different hydrological stations is conducted, and then, the correlation is used to calculate the correlation of the monthly mean runoff at different hydrological stations, thereby providing the basis for the construction of the COM for the entire basin.
Construction of the COM

To avoid the cumulative impact between the hydrological stations, we estimate the weights of the selected hydrological stations depending on their contribution and influence on the downstream hydrological stations. The COM is usually established on the monthly mean runoff at different hydrological stations (COM1). Specifically, the proposed COM is established on the controlling area of different hydrological stations (COM2).

Calculation steps of COM1. Assuming that the number of hydrological stations is \( n \), the average monthly runoff at the hydrological station \( i \) is \( C_i \), the mean monthly runoff at the hydrological station \( i \) in the month \( j \) is \( C_{ij} \), and the weight of hydrological station \( i \) can be expressed as:

\[
W^{(1)}_i = \frac{1}{\sum_{i=1}^{n} \frac{C_i}{\sum_{j=1}^{n} 1/ C_{ij}}}
\]

Therefore, the COM1 is established on the monthly mean runoff at different hydrological stations in the month \( j \) can be expressed as:

\[
C^{(1)}_j = \sum_{i=1}^{n} W^{(1)}_i C_{ij}
\]

It should be noted that label \( j \) is not successive, as hydrological stations collect the data with different time intervals.

Calculation steps of COM2. Assuming that the number of hydrological stations is \( n \), the percentage of the control area of \( i \)th hydrological station is \( S_i \), the mean monthly runoff at the hydrological station \( i \) in the month \( j \) is \( C_{ij} \), and the weight of hydrological station \( i \) can be expressed as:

\[
W^{(2)}_i = \frac{1}{\sum_{i=1}^{n} 1/ S_i}
\]

Therefore, the COM2 is established on the controlling area of different hydrological stations in the month \( j \) can be expressed as:

\[
C^{(2)}_j = \sum_{i=1}^{n} W^{(2)}_i C_{ij}
\]

It should be noted that label \( j \) is not successive, as hydrological stations collect the data with different time intervals.
**Factor reduction using PMI**

There is nonlinear correlation observed among hydrological and meteorological variables. In this case, MI can be used to select the input factors in runoff forecasting more effectively (He et al. 2011). However, while selecting input variables using MI, it should be taken into consideration that the coupling relationship between input variables affects the calculation results, which may lead to the mis-selection or omission of the input variables.

To solve the above problems, May et al. (May et al. 2008a; 2008b; Qian & Shu 2015) proposed the PMI methods to eliminate the relationship between variables through calculating conditional expectations, thereby ensuring the reliability and accuracy of variable selection.

**Partial mutual information (PMI)**

In the 1940s, Shannon employed the information entropy theory to measure the amount of information, and the information entropy of the COM can be expressed as follows:

$$H(C_i) = - \sum_{j=1}^{k} p_i \log p_j$$

(5)

where $H(C_i)$ represents the information entropy of variable $C_i$, $p_i$ denotes the probability of variable $C_i$ at different values, and $k$ represents the number of variables.

Assuming that the joint probability distribution of the COM $(C_i, C_j)$ is $p_{ij}(i = 1, 2, \ldots, k; j = 1, 2, \ldots, k)$, then the two-dimensional joint entropy can be expressed as:

$$H(C_i, C_j) = - \sum_{i,j=1}^{n} p_{ij} \log p_{ij}$$

(6)

Assuming that the marginal distribution values of $C_i$ and $C_j$ are $p_i$ and $p_j$, respectively, the conditional entropy of $C_i$ can be expressed as follows:

$$H(C_i | C_j) = - \sum_{i,j=1}^{k} p_{ij} \log \frac{p_{ij}}{p_j}$$

(7)

Consequently, the mutual information can be expressed as:

$$I(C_i, C_j) = H(C_i) - H(C_i | C_j)$$

(8)

Considering the third variable $C_l$, and taking the part of $I(C_i, C_j)$ that is not included in $C_l$, we define the PMI:

$$I(C_i, C_j|C_l) = H(C_i, C_j) + H(C_l) - H(C_i, C_j, C_l)$$

$$= \sum_{i,j,l=1}^{k} p_{il} \log \frac{p_{il} p_{jl}}{p_{il} p_{jl}}$$

(9)

Here, we select the Gaussian kernel function to estimate the sample probability density function.

**PMI-based factor selection**

When inputting multiple variables, there may be correlation presented among these variables. For example, if $X$ and $Z$ are input variables, and $Y$ is the output variable, then there may be correlation between input variables $X$ and $Z$, and the value of mutual information $I(Y,Z)$ may be greater than the actual value. Therefore, we can eliminate $X$ contained in variables $Y$ and $Z$ with conditional expectation, and the correlation between variables can be measured using PMI. After eliminating $x$, $Y$ is denoted as $u$, $Z$ is denoted as $v$, which can be expressed as:

$$m_Y(x) = E[y|x] = \frac{\sum_{i=1}^{n} y_i f(x)}{\sum_{i=1}^{n} f(x)}$$

(10)

$$u = Y - m_Y(x)$$

(11)

$$v = Z - m_Z(x)$$

(12)

PMI between $Y$ and $Z$ is converted into:

$$\text{PMI}(Z, Y) = I(v, u)$$

(13)

Specifically assuming that $C$ stands for the input variable set, $Y$ stands for the output variable (runoff forecasting results), $S$ stands for the optimal input variable set, and $C_i$ stands for the candidate variable corresponding to the maximum PMI, the PMI-based variable selection can be realized as per the following steps:

Step 1. Initialize $S$ as an empty set
Step 2. If $C$ is a null set, return
Step 3. Calculate the $u = Y - m_Y(S)$ by Equation (11)
Step 4. For each element of $C_{\tau} \in C$, calculate $v = C_{\tau} - m_{C_{\tau}}(S)$ by Equation (12)

Step 5. If PMI($Z, Y$) reaches the maximum, select $C_s$ as the candidate variable

Step 6. Calculate the Akaike Information Criterion (AIC). If the AIC is decreased, moving $C_s$ to the optimal input variable set $S$, and returning Step 2; Stop selection if AIC increases.

AIC is defined as follows:

$$AIC = n \log \left( \frac{1}{n} \sum_{i=1}^{n} u_i^2 \right) + 2p$$

(14)

where $u_i$ stands for the $Y$ regression residual calculated from the selected variables; $n$ stands for the number of samples, and $p$ stands for the number of selected variables. With the selection of independent variables, AIC keeps decreasing. When AIC reaches the minimum, the collection of optimal independent variables is finished.

Deep belief network

The DBN is constructed by sequential stacking of multiple restricted Boltzmann machines (RBMs) (Ackley et al. 1985). The learning includes two stages: unsupervised and supervised stages. The unsupervised stage is mainly used to pre-train the RBM, while in the supervised stage, the gradient-based error back propagation algorithm is used to fine tune the whole network. This method allows for particular advantages in learning the deep structure and has been successfully applied to pattern recognition, prediction, and other tasks. In addition to its advantages in weight optimization, the DBN can also be used as a probability generation model to overcome the problem of overfitting and underfitting of the model. A typical RBM is a random indirect graphical model, and a common DBN is a ubiquitous deep neural network (Figure 2).

Unsupervised contrastive divergence algorithm

Unsupervised learning based on the contrastive divergence (CD) algorithm (Hinton 2002) is based on obtaining the initial weight of the entire DBN by training each RBM layer by layer. In an RBM, assuming that $v$ represents the states of visual neurons, $h$ represents the states of hidden neurons, and the model parameter is $\theta = \{w, b, c\}$, so the probability distribution function $p(v, h; \theta)$ based on the energy model can be defined as:

$$p(v, h; \theta) = \frac{e^{-E(v, h; \theta)}}{Z}$$

(15)

where $Z$ is the normalization factor:

$$Z = \sum_{v, h} e^{-E(v, h; \theta)}$$

(16)
The edge distribution of the energy model $v$ is defined as

$$p(v; \theta) = \frac{1}{Z} \sum_h e^{-E(v, h; \theta)}$$ (17)

The energy function of the RBM is defined as follows:

$$E(v, h; \theta) = -\sum_{i=1}^{m} b_i v_i - \sum_{j=1}^{n} c_j h_j - \sum_{i=1}^{m} \sum_{j=1}^{n} v_i w_{ij}^R h_j$$ (18)

where the connection weight of the network is $w_{ij}^R$, the bias value of the visual layer is $b_i$, and the bias value of the hidden layer is $c_j$. Therefore, the conditional probability distribution can be described as:

$$p(h_j = 1/v; \theta) = \sigma \left(c_j + \sum_{i=1}^{m} v_i w_{ij}^R \right)$$ (19)

$$p(v_i = 1/h; \theta) = \sigma \left(b_i + \sum_{j=1}^{n} w_{ij}^R h_j \right)$$ (20)

where $\sigma$ is activation function, and $\sigma(x) = (1) / (1 + e^{-x})$.

For the convenience of operation, the states of visible neurons and hidden neurons are binary, the values of Equations (19) and (20) are usually determined as follows (taking the hidden neuron as an example):

$$h_j = \begin{cases} 1 & \text{if } p(h_j = 1/v) \geq \delta \\ 0 & \text{if } p(h_j = 1/v) < \delta \end{cases}$$ (21)

where $\delta$ is a constant ranging from 0.5 to 1.

Calculating the gradient of $\log p(v; \theta)$, and the renewal equation of RBM is as follows:

$$w_{ij}^R = w_{ij}^R + \eta \Delta w_{ij}^R$$ (22)

$$\Delta w_{ij}^R = E_{\text{data}}(v, h) - E_{\text{model}}(v, h)$$ (23)

where $\eta$ is the learning rate, $E_{\text{data}}(v, h)$ is the expectation of input samples, and $E_{\text{model}}(v, h)$ is the expectation of the distribution of the model. The approximate expectation values can be obtained by the Gibbs sampling method.

**Supervised learning**

Supervised learning fine-tunes the initial weights $w^R$ obtained by unsupervised learning based on the CD algorithm in the DBN and finally obtains the optimal parameters of the model. Taking the last hidden layer and the output layer as examples, if $y_s$ is the desired output and $y'_s$ is the actual output of the model, the cost function of the output layer can be defined as:

$$F(a) = \frac{1}{2} \sum_k (y_s - y'_s)^2$$ (24)

where $\alpha$ is the number of iterations and $K$ is the number of samples. The changed weights are as follows:

$$\Delta w_{\text{out}}(a) = -\frac{\partial F(a)}{\partial w_{\text{out}}(a)}$$ (25)

$$w_{\text{out}}(a + 1) = w_{\text{out}}(a) - \eta \frac{\partial F(a)}{\partial w_{\text{out}}(a)}$$ (26)

where $\eta$ is the learning rate.

Thus, the updated weight $w_{\text{out}}$ between the output layer and the last hidden layer can be obtained, and the weight $w = (w_{\text{out}}, w_1, w_{l-1}, \ldots, w_2, w_0)$ of the whole DBN model can be obtained by fine-tuning layer by layer.

**Evaluation criteria**

To estimate the performance of the PDBN model, some evaluation criteria are included in the evaluation, including mean absolute percentage error (MAPE), root-mean-square error (RMSE), deterministic coefficient (DC), the relative error (RE), and the running time (RT):

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{y'_t - \hat{y}_t}{y'_t} \right|$$ (27)

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (y'_t - \hat{y}_t)^2}$$ (28)

$$\text{DC} = 1 - \frac{\sum_{t=1}^{n} (y'_t - \hat{y}_t)^2}{\sum_{t=1}^{n} (y'_t - \hat{y}_t)^2}$$ (29)

$$\text{RE} = \left| \frac{y'_t - \hat{y}_t}{y'_t} \right| \times 100\%$$ (30)
where $y^t$ represents the observed value of the COM in the Yalong River Basin at $t$, $\hat{y}^t$ denotes the predicted value at $t$, and $n$ is the sample data. $\bar{y}$ represents the mean of the observed value of the COM in the Yalong River Basin.

The MAPE is a most frequently used statistics index and employed for examining the error between the predicted value and the observed value. The RMSE is used to measure the predicted value accuracy. A RMSE value of 0 indicates perfect match between the predicted value and the observed value. Likewise, the DC provides a measure of the capacity of the model to predict observed values. In general, high values of DC (up to 1) and small values for RMSE indicate a good model. The RE is used to reflect the reliability of the prediction model. In order to better estimate the calculation speed of the PDBN model, we add the running time (RT) index.

### Determining the network depth of the DBN

The suitability of parameter selection in the DBN is related to the prediction performance; therefore, the determination of network depth is the first problem to be solved. When the network depth of the DBN is large, more data features can be excavated, but the computational complexity is increased, and the performance is worse. At present, the exact methods for determining the network depth of a DBN include trial-and-error methods and parameter optimization methods. These methods search the most suitable network depth via many experiments. This study adopts the parameter optimization method by fixing the number of hidden nodes and the learning rate, and then determining the optimal network depth of the DBN by combining the particle swarm optimization (PSO) algorithm.

The common calculation method is applied to estimate the number of nodes in the hidden layer, and the formula is as follows:

$$l = \sqrt{m + n + \alpha}$$

(31)

In Equation (31), $l$ is the number of nodes in the hidden layer, $m$ is the number of input variables, $n$ is the number of output variables, and $\alpha$ is an integer ranging from 0 to 10.

In addition, because the existence of the normalization factor makes it difficult to realize the logarithmic similarity evaluation, this study adopts the reconstruction error method to evaluate the gap between the reconstructed data and the observed data. The smaller the reconstruction error is, the more stable the model is. The two theoretical bases of this evaluation method are as follows: first, the network depth of the DBN is proportional to the prediction accuracy of the model; second, after unsupervised training, the network parameters only need fine-tuning during the supervised training stage. The flow of the network error reconstruction is shown in Figure 3, and the calculation method of reconstruction error is as follows:

$$RE = \sum_{i=1}^{n} (X - S_i)^2$$

(32)

where $X$ is the $i$th batch matrix, $S_i$ is the value after the reconstruction of the $i$th batch matrix, and RE is the total error of the reconstruction.
It can be seen from Figure 3 that the DBN may judge whether the hidden layer can better depict the data characteristics of the visual layer. The DBN does this by reconstructing the input sample data and updating the hidden nodes by increasing the network depth of DBN and weight optimization, so that the reconstructed error is a minimum. Then, the label data is output.

**Optimization algorithm in unsupervised stage**

The CD calculation is based on Gibbs sampling. Gibbs sampling is used in Markov chain Monte Carlo statistical analysis and approximately extracts the sample sequence from a multivariable probability distribution under the condition that it is difficult to sample directly (Geng et al. 2018). The process of Gibbs sampling is shown in Figure 4. The main steps are as follows:

1. The Gibbs chain is initialized with the observation sample data \( v \) and obtaining the input vector \( v^0 \) of the visible layer.
2. Referring to Equations (19) and (20), \( h^{(t)} \) is sampled from \( P(h^{(t)} \mid v^{(t)}; \theta) \) and \( v^{(t+1)} \) from \( P(v^{(t+1)} \mid h^{(t)}; \theta) \) in turn.

Among them, \( t \) is the number of sampling steps, and the effect gets better as \( t+1 \) gets larger.

\((v^{(1)}, h^{(1)}) \) is acquired by sampling and a preliminary estimation of \( E_m(v, h) \), and using \((v^{(1)}, h^{(1)}) \) to estimate \( E_m(v, h) \) for the CD-1 algorithm. In the CD-1 algorithm, because all RBMs require many iterations and the direction of parameter update is different, if the learning rate remains unchanged, ‘premature’ phenomenon may appear in the algorithm or it may be difficult to converge, so selecting the adaptive learning rate according to the direction of the parameter update is essential to unsupervised learning.

Based on previous research results (Geng et al. 2018; Qiao et al. 2018; Saufi et al. 2019), this study designs an adaptive learning rate method according to the similarities and differences of parameter updating directions after two consecutive iterations in the training process of RBM. The principle is that when the parameter updating direction (positive or negative changes) is the same after two consecutive iterations, the learning rate will increase, or conversely, it will decrease. The adaptive updating mechanism of learning rate is as follows:

\[
\eta = \begin{cases} 
D\eta \times (\Delta w^R_{ji})^{(t)} \times (\Delta w^R_{ji})^{(t+1)} > 0 \\
d\eta \times (\Delta w^R_{ji})^{(t)} \times (\Delta w^R_{ji})^{(t+1)} < 0 
\end{cases}
\]

(33)

\[
(\Delta w^R_{ji})^{(t)} = v_i^{(t)} h_j^{(t)} - v_i^{(t+1)} h_j^{(t+1)}
\]

(34)

\[
(\Delta w^R_{ji})^{(t+1)} = v_i^{(t+1)} h_j^{(t+1)} - v_i^{(t+2)} h_j^{(t+2)}
\]

(35)

Among them, \( D \) is the enhancement coefficient of learning rate, \( d \) is the decrease coefficient of learning rate, and \( 0 < d < 1 < D \).

Based on the CD algorithm, the weights of the DBN are updated once every Gibbs sampling period, while the
binary sampling of the intermediate state is carried out twice, and the weights of each update are proportional to the state sampling, so \( D = 2\delta, d = \delta \). \( \delta \) is the probability threshold ranging from 0.5 to 1, and the general value is 0.7. As a result, the \( D \) values range from 1 to 2 when the general value is 1.4, and the \( d \) values range from 0.5 to 1 when the general value is 0.7.

**Optimization algorithm in the supervised stage**

PLSR is a many-to-many regression model that considers both independent variables and dependent variables. Therefore, it is a supervised learning model. Especially when facing multiple variables, multiple correlations, and a relatively insufficiently observed sample, the PLSR has the advantages that traditional classical regression analysis and other methods do not have (Qiao et al. 2018). Therefore, this study conducts supervised fine-tuning for the DBN via PLSR. Starting from the output layer, the output layer and the last hidden layer comprise the first group, and so on, until the first hidden layer and the input layer is the last group, and supervised fine-tuning is achieved. Taking the output layer and the last hidden layer as an example to establish the regression model, the specific steps include the following:

1. Assuming that the actual output, \( y_s \), of the model is a dependent variable, the dimension of the output matrix is \( p \), the state, \( h_0 \), of the last hidden layer is the independent variable, and the dimension of the state matrix is \( q \), then:

\[
Y = \begin{bmatrix}
y_{11} & \cdots & y_{1p} \\
\vdots & \ddots & \vdots \\
y_{N1} & \cdots & y_{Np}
\end{bmatrix} \quad (36)
\]

\[
X = \begin{bmatrix}
h_{11} & \cdots & h_{1q} \\
\vdots & \ddots & \vdots \\
h_{N1} & \cdots & h_{Nq}
\end{bmatrix} \quad (37)
\]

where \( Y \) is the observation matrix of the \( N \)-sample. \( X \) is the state observation matrix of the last hidden layer.

2. Assuming that the first pair of components \( u_1 \) and \( v_1 \) extracted from the two observation sample matrices, and \( u_1 \) is the linear combination of the dependent variable set \( (y_1, y_2, \ldots, y_p)^T \), and \( v_1 \) is the linear combination of independent variable set \( (h_1, h_2, \ldots, h_q)^T \), then we can define:

\[
u_1 = a_{11}y_{11} + \cdots + a_{1p}y_{1p} = a_1^T y_s \quad (38)
\]

\[
v_1 = \beta_{11}h_{11} + \cdots + \beta_{1q}h_{1q} = \beta_1^T h_1 \quad (39)
\]

Calculating the score vectors of the first pair of components, denoted as \( \hat{u}_1 \) and \( \hat{v}_1 \), then

\[
\hat{u}_1 = \begin{bmatrix}
y_{11} & \cdots & y_{1p}
\vdots & \ddots & \vdots \\
y_{N1} & \cdots & y_{Np}
\end{bmatrix} \begin{bmatrix}
a_{11} \\
a_{21} \\
\vdots \\
a_{p1}
\end{bmatrix} \quad (40)
\]

\[
\hat{v}_1 = \begin{bmatrix}
h_{11} & \cdots & h_{1q}
\vdots & \ddots & \vdots \\
h_{N1} & \cdots & h_{Nq}
\end{bmatrix} \begin{bmatrix}
\beta_{11} \\
\beta_{21} \\
\vdots \\
\beta_{q1}
\end{bmatrix} \quad (41)
\]

3. This step calculates the values of \( a_1 \) and \( \beta_1 \) when the correlation between \( \hat{u}_1 \) and \( \hat{v}_1 \) is strongest, and converts the inner product for calculating score vectors into a conditional extreme solution.

\[
\arg \max_{(a_1, \beta_1)} \langle \hat{u}_1, \hat{v}_1 \rangle \quad (42)
\]

Then, the score vectors \( \hat{u}_1 \) and \( \hat{v}_1 \) are obtained corresponding to the optimal parameters \( a_1 \) and \( \beta_1 \).

4. This step establishes the regression models of \( v_1 \) corresponding to \( y_1, y_2, \ldots, y_p \) and \( h_1, h_2, \ldots, h_q \), respectively, as

\[
\begin{align*}
X &= \hat{v}_1 \left( \frac{\hat{v}_1^T \hat{v}_1}{\|\hat{v}_1\|^2} \right)^T + X_1 \\
Y &= \hat{v}_1 \left( \frac{\hat{Y}^T \hat{v}_1}{\|\hat{v}_1\|^2} \right)^T + Y_1
\end{align*} \quad (43)
\]

where \( X_1 \) and \( Y_1 \) are the residual matrices of \( X \) and \( Y \), respectively.

5. The above steps are repeated using residual matrices \( X_1 \) and \( Y_1 \) instead of \( X \) and \( Y \), respectively.
6. Assuming that the rank of the observation matrix $X$ of the last hidden layer is $r$, the number of components is $r$ ($v_1, v_2, \ldots, v_r$), such that:

$$
\begin{align*}
X &= \eta_1 \left( \frac{X^T \hat{v}_1}{\|\hat{v}_1\|} \right)^r + \cdots + \eta_r \left( \frac{X^T \hat{v}_r}{\|\hat{v}_r\|} \right)^r + X,
Y &= \zeta_1 \left( \frac{Y^T \hat{v}_1}{\|\hat{v}_1\|} \right)^r + \cdots + \zeta_r \left( \frac{Y^T \hat{v}_r}{\|\hat{v}_r\|} \right)^r + Y,
Y_s &= v_1 \frac{Y^T \hat{v}_1}{\|\hat{v}_1\|} + \cdots + v_r \frac{Y^T \hat{v}_r}{\|\hat{v}_r\|}.
\end{align*}
$$

(44)

(45)

Substituting $v_k = \beta_k H_1 + \cdots + \beta_0 H_q$ ($k = 1, 2, \ldots, r$) into (38), and obtaining the PLSR equation between the desired output and the last hidden layer.

And then obtaining the PLSR equation between the desired output and the last hidden layer.

$$
y_i^c = w_1^{c0} H_1 + \cdots + w_p^{c0} H_q \quad (i = 1, 2, \ldots, p)
$$

(46)

Finally, this step obtains the weight $w_{out}$ between the output layer and the last hidden layer.

In general, the number of extracted components $c (c \leq r)$ is determined by cross-validation, that is, removing the $i$th ($i = 1, 2, \ldots, N$) state of the last hidden layer of the DBN network every time, and realizing modeling based on PLSR for the remaining $(n-1)$ states. Then, the regression equation including $c (c \leq r)$ components is obtained, and finally, the eliminated $i$th state is substituted into the fitted regression equation $y_i^c (j = 1, 2, \ldots, p)$, and the corresponding output $\hat{y}_{ij} (c)$ is obtained. This process repeats until the $n$th state is completed. The error square sum of the $j$th predicted output $y_j (i = 1, 2, \ldots, p)$ is defined as

$$
\text{ES}_j (c) = \sum_{i=1}^{N} (y_{ij} - \hat{y}_{ij} (c))^2
$$

(47)

where $y_{ij}$ is the expected output, then the error square sum of predicted output $y_s = (y_1, \ldots, y_p)^T$ is defined as

$$
\text{ES}(c) = \sum_{j=1}^{p} \text{ES}_j (c)
$$

(48)

After that, the regression equation including $c$ components is fitted by all the states of the last hidden layer of the DBN. Noting that the predicted output corresponding to the $i$th state is $\hat{y}_{ij} (c)$, then the error square sum of predicted output can be defined as

$$
\text{ESS}_i (c) = \sum_{j=1}^{N} (y_{ij} - \hat{y}_{ij} (c))^2
$$

(49)

The error square sum of predicted output $y_s = (y_1, \ldots, y_p)^T$ is defined as:

$$
\text{ESS}(c) = \sum_{j=1}^{p} \text{ESS}_j (c)
$$

(50)

According to the PLSR, there is:

$$
\begin{cases}
\text{ES}(c) > \text{ESS}(c) \\
\text{ESS}(c) < \text{ESS}(c-1)
\end{cases}
$$

(51)

The threshold $L$ is defined as:

$$
L = \frac{\text{ES}(c)}{\text{ESS}(c-1)}
$$

(52)

The value of $L$ is inversely proportional to that of $c$, so $c$ can be determined by choosing the appropriate $L$.

After the above steps, the fine-tuning process between the output layer and the last hidden layer is accomplished during the supervised learning stage. By analogy, the fine-tuning process of the whole DBN is finally completed, and the network weight $w = (w_{out}, w_1, w_{1-1}, \ldots, w_2, w_{in})$ is obtained.

In summary, the mid- to long-term runoff prediction by combining the DBN and PLSR can be divided into three steps: firstly, determining the number of hidden layer nodes and network learning rate, and adopting the PSO algorithm for searching the optimal network depth of the DBN in different depths; secondly, introducing the adaptive learning rate for improving the convergence speed of the DBN in the unsupervised training stage; and finally, adopting the PLSR to avoid easily falling into local optimum and to accelerate training speed in the supervised learning stage, so as to improve the fine-tuning accuracy of the DBN network.
CASE STUDY

Study area and data

The Yalong River (25°12′–34°9′N, 96°47′–102°42′E) is the largest tributary of the Jinsha River in the southern region of the Tibetan Plateau. The total basin area of the Yalong River is approximately 136,000 km², and the mainstream length is 1,571 km. The maximum altitude is greater than 6,000 m, and the minimum altitude is less than 1,000 m. The mean annual temperature is approximately 2.5 °C. The mean annual precipitation is approximately 600–1,800 mm decreasing from the south to the north. The average annual estuarine discharge is 1,860 m³/s. The major vegetation types are forests, shrubs, and meadows (Ye et al. 2017). The geography of the Yalong River is shown in Figure 5.

According to the time span characteristics of hydrometeorological data, the relevant data of the mid-long runoff prediction in the Yalong River Basin include: (1) 130 atmospheric circulation factors (selecting 21 teleconnection climatic factors) for the period from January 1951 to December 2012; (2) meteorological data such as pressure, temperature, humidity, precipitation, wind speed, and sunshine from January 1960 to September 2012; (3) runoff data of Lianghekou, Jinping, Guandi, and Ertan hydrological stations from January 1960 to December 2016. In this study, the dataset is 624 sets of samples from the period of January 1960 to December 2011, and it is divided into two parts: the training set (from January 1960 to December 2001) and the test set (from January 2002 to December 2011) (Figure 6).

Model development

BP neural networks

In this study, the BP neural networks are used as a competitive model for point forecasts. In BP neural networks, the number of hidden layer nodes is 12, and the training function is ‘tansig’, the learning function is ‘logsig’, the maximum training times are 600, the learning rate is 0.1, the model training adopts LM algorithm, the momentum factor is 0.9, and the expected error is 0.001.

Support vector machine

The least-squares support vector machine (LS-SVM) is used in this study. In SVM, the kernel function is ‘sigmoid’, and
regularization parameters and kernel function’s parameters for SVM are optimized using the grid-search algorithm with cross-validation. The LS-SVM toolbox is available on http://www.esat.kuleuven.be/sista/lssvmlab/.

**Results and discussion**

In this section, the proposed approach for mid- to long-term runoff prediction is applied, including COM selection result, factor selection result, and the PDBN model. The forecasting performances of PDBN are also compared with the application of the DBN, BP, and SVM to demonstrate the superiority using five common criteria.

**COM selection result**

After analyzing the consistency of the four hydrological stations (Lianghekou, Jinping, Guandi, and Ertan hydrological stations) in the Yalong River Basin, the results show that there is a high correlation among the average runoff of the four hydrological stations on a monthly scale (Figure 7). Therefore, we construct the COM (coarse-grained) based on the monthly mean runoff of the four hydrological stations (Figure 8) and the results of PCC, Kendall’s Tau-b, and Spearman correlation coefficient are shown in Tables 3–5, respectively. Results of the correlation analysis between the COM1 and COM2 on the monthly scale indicate that the correlation of COM2 is higher. Consequently, this study selects the COM2 as the comprehensive runoff index of the Yalong River Basin.

**Factor selection result**

*Alternative candidate predictive factors.* In this study, the main influencing factors of the runoff change include antecedent runoff, antecedent rainfall, and 21 teleconnection climate factors in the Yalong River Basin (Table 6). Lagged predictive factors from the current time step $t$ up to the previous 12 time steps (1 month) are considered sufficient for capturing key factors affecting the runoff. Here, the predictive factors from the current time $t$ to $t - 12$ are selected as potential input candidates. Consequently, the alternative candidate predictive factors include comprehensive runoff index $\{\text{com}(t - 1), \text{com}(t - 2), \ldots, \text{com}(t - 12)\}$, area rainfall $\{\text{rain}(t - 1), \text{rain}(t - 2), \ldots, \text{rain}(t - 12)\}$, and 21 teleconnection climatic factors $\{\text{atm}_1(t - 1), \text{atm}_1(t - 2), \ldots, \text{atm}_1(t - 12)\}$, $\{\text{atm}_2(t - 1), \text{atm}_2(t - 2), \ldots, \text{atm}_2(t - 12)\}$, $\ldots, \{\text{atm}_{21}(t - 1), \text{atm}_{21}(t - 2), \ldots, \text{atm}_{21}(t - 12)\}$. The total number of factors is 276 ($23 \times 12$).
Factor selection procedure and result. In this study, the PMI method is adopted to select all the alternative factors affecting runoff change according to the AIC. However, due to the significant influence of the autoregressive term on the factor selection, the method based on the AIC is not fully applicable to the factors selection in this study. In order to study the effects of other hydrological factors (such as rainfall, climate, and others), this study completes the factor selection on the basis of the PMI method combined with manual selection. The selection procedure includes:

1. The PMI method is adopted to sort all the candidate factors according to the correlation size. The partial results of the primary selection are shown in Table 7.
2. On the basis of the above primary results, the top 20 candidate factors are selected, in terms of correlation size (Table 8).
3. On the basis of the above results, the correlation size from the top of the list of variable factors (up to the top five) is manually selected from the antecedent COM, antecedent rainfall, and antecedent teleconnection climate factors of the Yalong River Basin. New comprehensive selection results are formed, as shown in Table 9. The finally selected results show that the factors affecting the change of runoff process in the Yalong River Basin are mainly related to the lag time of 1, 2, 3, 11, and 12 months of the COM, the lag time of 1, 7, and 12 months of the area rainfall, and the lag time of 1 month of the North African Subtropical High Area Index (20°W–60°E), 6 months of the Indian Ocean Warm Pool Area Index, 7 months of the Northern Hemisphere Polar Vortex Central Intensity Index (50–360), 8 months of the NINO 4 SSTA Index.

### Table 1

<table>
<thead>
<tr>
<th>Station</th>
<th>Lianghekou</th>
<th>Jinping</th>
<th>Guandi</th>
<th>Ertan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monthly mean</td>
<td>661.54</td>
<td>1208.65</td>
<td>1416.25</td>
<td>1617.32</td>
</tr>
<tr>
<td>Normalization</td>
<td>0.134906</td>
<td>0.246474</td>
<td>0.288808</td>
<td>0.329812</td>
</tr>
<tr>
<td>Runoff impact</td>
<td>7.412580</td>
<td>4.057225</td>
<td>3.462504</td>
<td>3.032031</td>
</tr>
<tr>
<td>Weight</td>
<td>0.412627</td>
<td>0.225849</td>
<td>0.192743</td>
<td>0.168781</td>
</tr>
</tbody>
</table>

### Table 2

<table>
<thead>
<tr>
<th>Station</th>
<th>Lianghekou</th>
<th>Jinping</th>
<th>Guandi</th>
<th>Ertan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage</td>
<td>51%</td>
<td>81%</td>
<td>89%</td>
<td>97%</td>
</tr>
<tr>
<td>Controlled area</td>
<td>1.960784</td>
<td>1.234568</td>
<td>1.123596</td>
<td>1.030928</td>
</tr>
<tr>
<td>Weight</td>
<td>0.366510</td>
<td>0.230766</td>
<td>0.210023</td>
<td>0.192701</td>
</tr>
</tbody>
</table>
and 1 month of the Indian Ocean Warm Pool Strength Index.

Determining the structure and parameter optimization of PDBN

Network depth of PDBN. According to Equation (31), it is assumed that the number of nodes in the hidden layer is 10. Therefore, on the basis of standardized processing of the dataset, the structure and parameters of the DBN are set as follows: the number of input variables is 13, the number of output variables is 1, it is assumed that the number of hidden nodes is 10, the learning rate is 0.01, and the number of iterations is 800. The network depth of the DBN is between 2 and 6, and the experimental results are shown in Table 10 and Figure 9.

As can be seen from Table 10 and Figure 9, the reconstruction errors at the bottom and top of the PDBN are
relatively stable in the early stage but changed greatly in the later stage as the network depth increased. The time complexity of model training also increased. When the network depth is less than or equal to 4, the comprehensive performance of the model is relatively stable. Since then, increasing the network depth leads to the fluctuation of PDBN and the increase of error accumulation.

Comprehensive comparison, when the network depth is 4 (the number of hidden layers is 3), the comprehensive performance of PDBN is the best, so this study selected 4 as the network depth of PDBN.

Structure of PDBN. According to Equation (31), the number of neurons in the hidden layer selected in this study is an integer ranging from 4 to 14. However, the number is small, and the generalization ability is weak.

In order to avoid the shortage of data feature description caused by selecting too few hidden nodes, this study

| Table 5 | Spearman analysis of COM and runoff at four hydrological stations |
| Lianghekou | Jinping | Guandi | Ertan | COM1 | COM2 |
| Lianghekou | Spearman | 1.000 | 0.976 | 0.971 | 0.953 | 0.985 | 0.983 |
| Jinping | Spearman | 0.976 | 1.000 | 0.994 | 0.984 | 0.997 | 0.997 |
| Guandi | Spearman | 0.971 | 0.994 | 1.000 | 0.987 | 0.996 | 0.996 |
| Ertan | Spearman | 0.953 | 0.984 | 0.987 | 1.000 | 0.987 | 0.988 |
| COM1 | Spearman | 0.985 | 0.997 | 0.996 | 0.987 | 1.000 | 1.000 |
| COM2 | Spearman | 0.983 | 0.997 | 0.996 | 0.988 | 1.000 | 1.000 |

| Table 6 | The name and definition of alternative candidate predictive factors |
| Predictive factors | Definition and interpretation | Predictive factors | Definition and interpretation |
| comprehensive runoff index (com) | Describing the abundance and drought of the monthly average runoff in the watershed based on the controlling area of four hydrological stations | atm11 | Number of landing typhoon on China |
| area rainfall (rain) | Adopting the Tyson Polygon Method to construct the rain factor based on 22 meteorological stations in the Yalong River Basin | atm12 | NINO 3 SST A Index |
| atm1 | North African Subtropical High Area Index (20 W – 60E) | atm13 | NINO 4 SST A Index |
| atm2 | Atlantic-European Polar Vortex Area Index (450 W – 60E) | atm14 | NINO B SST A Index |
| atm3 | Northern Hemisphere Polar Vortex Central Intensity Index (50 – 360) | atm15 | Indian Ocean Warm Pool Area Index |
| atm4 | Asian Zonal Circulation Index (IZ, 60E – 150E) | atm16 | Indian Ocean Warm Pool Strength Index |
| atm5 | East Asian Trough Intensity Index | atm17 | Western Pacific Warm Pool Strength Index |
| atm6 | West Pacific Pattern | atm18 | Atlantic Multi-decadal Oscillation Index |
| atm7 | East Atlantic/West Russia Pattern | atm19 | West Wind Drift Current SST Index |
| atm8 | Polar/Eurasia Pattern | atm20 | ENSO Modoki Index |
| atm9 | Mid-Eastern Pacific 200 mb Zonal Wind Index | atm21 | Tropic Indian Ocean Dipole Index |
| atm10 | East Pacific 850 mb Trade Wind Index |
increases the number of neurons in the hidden layer by 20, 30, 40, and 50 levels on the basis of the previous hidden layer node range. Finally, the training of PDBN at different hidden nodes is completed, and the MAPE, RMSE, DC, and RT are recorded for each experiment. Based on these, the optimal number of neurons in the hidden layer is determined. The experimental results are shown in Table 11.

Table 7 | Partial ranking results of the correlation size of input factors

<table>
<thead>
<tr>
<th>Iteration times</th>
<th>Selected factors (dimension)</th>
<th>PMI</th>
<th>AIC</th>
<th>Iteration times</th>
<th>Selected factors (dimension)</th>
<th>PMI</th>
<th>AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>253th</td>
<td>MI = 1.06327</td>
<td>7359.21</td>
<td>16</td>
<td>27th</td>
<td>0.242582</td>
<td>2297.18</td>
</tr>
<tr>
<td>1</td>
<td>276th</td>
<td>0.342728</td>
<td>7313.61</td>
<td>17</td>
<td>43th</td>
<td>0.242350</td>
<td>1759.57</td>
</tr>
<tr>
<td>2</td>
<td>259th</td>
<td>0.272634</td>
<td>7317.04</td>
<td>18</td>
<td>164th</td>
<td>0.245940</td>
<td>1140.58</td>
</tr>
<tr>
<td>3</td>
<td>1st</td>
<td>0.250970</td>
<td>7116.99</td>
<td>19</td>
<td>76th</td>
<td>0.249823</td>
<td>512.49</td>
</tr>
<tr>
<td>4</td>
<td>264th</td>
<td>0.281160</td>
<td>6850.16</td>
<td>20</td>
<td>118th</td>
<td>0.217033</td>
<td>309.10</td>
</tr>
<tr>
<td>5</td>
<td>265th</td>
<td>0.263007</td>
<td>6393.48</td>
<td>21</td>
<td>186th</td>
<td>0.189791</td>
<td>286.41</td>
</tr>
<tr>
<td>6</td>
<td>275th</td>
<td>0.201745</td>
<td>5937.04</td>
<td>22</td>
<td>104th</td>
<td>0.177744</td>
<td>83.84</td>
</tr>
<tr>
<td>7</td>
<td>266th</td>
<td>0.159973</td>
<td>5842.28</td>
<td>23</td>
<td>229th</td>
<td>0.157730</td>
<td>317.03</td>
</tr>
<tr>
<td>8</td>
<td>174th</td>
<td>0.198184</td>
<td>5408.06</td>
<td>24</td>
<td>74th</td>
<td>0.119725</td>
<td>1063.35</td>
</tr>
<tr>
<td>9</td>
<td>31th</td>
<td>0.165484</td>
<td>5202.29</td>
<td>25</td>
<td>241th</td>
<td>0.109808</td>
<td>1124.14</td>
</tr>
<tr>
<td>10</td>
<td>152th</td>
<td>0.189054</td>
<td>4894.24</td>
<td>26</td>
<td>1st</td>
<td>0.119725</td>
<td>1063.35</td>
</tr>
<tr>
<td>11</td>
<td>181th</td>
<td>0.227255</td>
<td>4579.80</td>
<td>27</td>
<td>183th</td>
<td>0.031665</td>
<td>1134.98</td>
</tr>
<tr>
<td>12</td>
<td>185th</td>
<td>0.221514</td>
<td>4408.21</td>
<td>28</td>
<td>97th</td>
<td>0.031665</td>
<td>1134.98</td>
</tr>
<tr>
<td>13</td>
<td>267th</td>
<td>0.237344</td>
<td>4275.77</td>
<td>29</td>
<td>205th</td>
<td>0.068223</td>
<td>1124.14</td>
</tr>
<tr>
<td>14</td>
<td>57th</td>
<td>0.219919</td>
<td>3717.66</td>
<td>30</td>
<td>169th</td>
<td>0.046273</td>
<td>1127.05</td>
</tr>
<tr>
<td>15</td>
<td>245th</td>
<td>0.245911</td>
<td>3045.74</td>
<td>31</td>
<td>182th</td>
<td>0.010511</td>
<td>1135.73</td>
</tr>
</tbody>
</table>

Table 8 | Candidate factors ranked in the top 20 based on PMI

<table>
<thead>
<tr>
<th>Sorting by PMI value</th>
<th>The candidate factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>rain(t – 1), 253th</td>
</tr>
<tr>
<td>2</td>
<td>com(t – 12), 276th</td>
</tr>
<tr>
<td>3</td>
<td>rain(t – 7), 259th</td>
</tr>
<tr>
<td>4</td>
<td>atm13(t – 1), 1st</td>
</tr>
<tr>
<td>5</td>
<td>rain(t – 12), 264th</td>
</tr>
<tr>
<td>6</td>
<td>com(t – 1), 265th</td>
</tr>
<tr>
<td>7</td>
<td>com(t – 11), 275th</td>
</tr>
<tr>
<td>8</td>
<td>com(t – 2), 266th</td>
</tr>
<tr>
<td>9</td>
<td>atm15(t – 6), 174th</td>
</tr>
<tr>
<td>10</td>
<td>atm3(t – 7), 31th</td>
</tr>
</tbody>
</table>

Table 9 | Reduction results of input factors based on PMI

<table>
<thead>
<tr>
<th>Input factors</th>
<th>Selected factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comprehensive runoff index</td>
<td>com(t – 12), com(t – 1), com(t – 11), com(t – 2), com(t – 3)</td>
</tr>
<tr>
<td>Rainfall factors</td>
<td>rain(t – 1), rain(t – 7), rain(t – 12)</td>
</tr>
<tr>
<td>Climate factors</td>
<td>atm13(t – 1), atm13(t – 6), atm13(t – 7), atm13(t – 8), atm13(t – 1)</td>
</tr>
</tbody>
</table>
As can be seen from Table 11, with the increase of neurons in the hidden layer, the RT increases. The MAPE and RMSE decrease at early stages and then begin to increase. With more than 20 neurons in the hidden layer, the performance of the PDBN decreases sharply. The performance change of PDBN is shown in Figure 10.

As can be seen from Figure 10, when the number neuron in the hidden layer is 12, the MAPE and RMSE corresponding to the PDBN are the smallest and the DC is the largest. This result indicates that there is a better fitting effect between the predicted value and the true value; however, the RT increases with an increasing number of neurons in the hidden layer. For a comprehensive comparison, when the number neurons in the hidden layer is 12, the PDBN is optimal (MAPE = 13.28%, RMSE = 0.1932, DC = 0.8846, t = 29.8 s). Therefore, the optimal structure of the

### Table 10 | Training results of PDBN with different network depths

<table>
<thead>
<tr>
<th>Network depths</th>
<th>Top level reconstruction error</th>
<th>Bottom reconstruction error</th>
<th>MAPE (%)</th>
<th>RMSE</th>
<th>DC</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4.38–4.95</td>
<td>8.33–9.11</td>
<td>17.61</td>
<td>0.1753</td>
<td>0.8376</td>
<td>11.72</td>
</tr>
<tr>
<td>3</td>
<td>4.68–5.07</td>
<td>9.09–9.52</td>
<td>21.34</td>
<td>0.2121</td>
<td>0.8126</td>
<td>18.44</td>
</tr>
<tr>
<td>4</td>
<td>4.24–4.89</td>
<td>8.13–8.98</td>
<td>15.25</td>
<td>0.1826</td>
<td>0.8413</td>
<td>25.86</td>
</tr>
<tr>
<td>5</td>
<td>5.32–5.51</td>
<td>9.29–9.55</td>
<td>20.25</td>
<td>0.2434</td>
<td>0.7787</td>
<td>32.55</td>
</tr>
<tr>
<td>6</td>
<td>5.98–6.37</td>
<td>9.77–9.95</td>
<td>32.88</td>
<td>0.2697</td>
<td>0.7423</td>
<td>39.29</td>
</tr>
</tbody>
</table>

### Figure 9 | The relationship between the network depth of PDBN and the performance.

### Table 11 | Training results of PDBN with different numbers of neurons in the hidden layer

<table>
<thead>
<tr>
<th>Input nodes</th>
<th>Hidden layers</th>
<th>Number of neurons in the hidden layer</th>
<th>MAPE (%)</th>
<th>RMSE</th>
<th>DC</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>3</td>
<td>4</td>
<td>25.24</td>
<td>0.2867</td>
<td>0.7349</td>
<td>17.62</td>
</tr>
<tr>
<td>5</td>
<td>30.36</td>
<td>0.3635</td>
<td>0.6548</td>
<td>19.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>28.08</td>
<td>0.3268</td>
<td>0.7012</td>
<td>20.41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>22.14</td>
<td>0.3043</td>
<td>0.6801</td>
<td>21.83</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>18.67</td>
<td>0.1986</td>
<td>0.8012</td>
<td>23.11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>24.29</td>
<td>0.2898</td>
<td>0.7134</td>
<td>24.40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>15.25</td>
<td>0.1826</td>
<td>0.8413</td>
<td>25.86</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>18.43</td>
<td>0.2326</td>
<td>0.7896</td>
<td>27.42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>13.28</td>
<td>0.1932</td>
<td>0.8846</td>
<td>29.89</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>19.48</td>
<td>0.2245</td>
<td>0.8112</td>
<td>31.34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>21.22</td>
<td>0.2876</td>
<td>0.7335</td>
<td>33.87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>31.12</td>
<td>0.3425</td>
<td>0.6123</td>
<td>39.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>35.43</td>
<td>0.3916</td>
<td>0.5712</td>
<td>49.66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>40.36</td>
<td>0.4256</td>
<td>0.5467</td>
<td>57.68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>42.12</td>
<td>0.4416</td>
<td>0.5368</td>
<td>68.31</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PDBN for the mid- to long-term runoff prediction in the Yalong River Basin is determined (13-12-12-12-1).

**Parameter optimization of PDBN.** According to the optimization algorithm, the number of iterations of every RBM in the unsupervised stage is 300, \( D \) is 1.4, and \( d \) is 0.7. It is necessary to determine the number of extracted components \( c = \{c_4, c_3, c_2, c_1\} \) of each layer from top to bottom to improve the fine-tuning precision of PDBN. Through training, the relationship between the number of extracted components \( c \) and the limited value \( L \) is as shown in Figure 11. When \( L = 0.02 \), the fine-tuning precision of PDBN is the highest, and the corresponding number of extracted components \( c = \{c_4 = 12, c_3 = 10, c_2 = 13, c_1 = 4\} \).

**Evaluation of forecasting performance**

In order to efficiently demonstrate the superior modeling performance of PDBN, we compare the performance of BP, SVM, and DBN models in runoff prediction. The forecasting evaluation results are comprehensively shown in Figures 12–14 and Table 12 from different aspects. The predicted and observed runoff during model calibration using different data-driven models is shown in Figure 15. The
relative percent error using the different methods is shown in Figure 16.

Among the evaluation results based on the dataset, the data-driven models have good prediction results on the dataset constructed by COM, rainfall factors, and climate factors, but different evaluation indicators show diverse results. Among them, for the MAPE index, the PDBN model is the best (BP is 24.75%, SVM model is 24.64%, and DBN model is 41%), which indicates the minor error between the predicted value and the observed value adopting the PDBN. For the RMSE index, the PDBN model indicates the higher prediction accuracy (PDBN is 229.7 m$^3$/s, BP model is 326.29 m$^3$/s, SVM model is 360.02 m$^3$/s, DBN is 277.5 m$^3$/s), and the RMSE from the SVM model is larger. This index is proportional to the annual flow of the river basin. For the DC index, the four data-driven models are better and the PDBN model is the best (PDBN is 0.9393, BP model is 0.8775, SVM model is 0.8508, and DBN is 0.9114), which shows these models have better fitting effect. The main reason is that the PMI method has the advantages of characterizing relationships of linear and nonlinear among factors, and thus might be appropriate for the factor selection of hydrological process with...
complex spatio-temporal characteristics. For the RT index, the PDBN model is the best (PDBN is 10.32 s, BP model is 30.92 s, SVM model is 29.15 s, and DBN is 16.58 s), the BP and SVM models have lower scores for this index, which demonstrates the advantage of deep learning in computation speed.

Table 12 | Performance comparison of different methods

<table>
<thead>
<tr>
<th>Model</th>
<th>Structure</th>
<th>MAPE (%)</th>
<th>RMSE (m$^3$/s)</th>
<th>DC</th>
<th>RT (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP</td>
<td>13-12-1</td>
<td>24.75</td>
<td>326.29</td>
<td>0.8775</td>
<td>30.92</td>
</tr>
<tr>
<td>SVM</td>
<td>–</td>
<td>24.64</td>
<td>360.02</td>
<td>0.8508</td>
<td>29.15</td>
</tr>
<tr>
<td>DBN</td>
<td>13-12-12-1</td>
<td>41.00</td>
<td>277.50</td>
<td>0.9114</td>
<td>16.58</td>
</tr>
<tr>
<td>PDBN</td>
<td>13-12-12-1</td>
<td>19.98</td>
<td>229.70</td>
<td>0.9393</td>
<td>10.32</td>
</tr>
</tbody>
</table>

Among them, the COM and factor reduction provide early data processing for runoff prediction, and the PDBN method is proposed to improve the accuracy of runoff prediction. Also, the four data-driven models have an efficient prediction for the dataset tested, and the comprehensive performance of the PDBN model is better than that of BP, SVM, and the DBN models for MAPE, RMSE, DC, RE, and RT.

CONCLUSIONS

In general, the characterization and determination of watershed runoff trends and their key factors are a difficult task. Aiming to provide a solution, this study studied the correlation between multiple factors that affect changes in mid- to long-term runoff in the watershed. A method to construct comprehensive runoff factors was proposed, and

Figure 14 | The prediction results using the SVM model compared with PDBN.

Figure 15 | Predicted and observed runoff using different methods (m$^3$/s).
the comprehensive characterization of runoff changes in the whole watershed was realized. Then, we investigated a method for generating runoff impact factors based on PMI to obtain the key feature factors related to the runoff in the watershed, which achieved the factor selection for the mid- to long-term runoff forecasting in the Yalong River Basin.

Aiming to solve the problems caused by the highly nonlinear complex characteristics of mid- to long-term runoff forecasting in the watershed, this study proposes a mid- to long-term runoff forecasting method based on deep learning to overcome the drawback of traditional artificial neural networks and other shallow learning networks. At the same time, the dynamic optimization of structure and parameters of the DBN is studied for improving the forecasting accuracy and generalization. The novelty of our proposed method lies in the PMI-based key factor selection and the PDBN-based forecasting method.

Experimental results on the data collected in the Yalong River Basin demonstrate that our method can obtain the most consistent and correct results in mid- to long-term runoff forecasting. Apart from this, future work also includes investigating more robust deep learning models.

ACKNOWLEDGEMENTS

This research is supported by ‘the Fundamental Research Funds for the Central Universities’ (Grant Nos. 2017B616X14 and 2018B610X14), ‘the National Natural Science Foundation of China’ (Grant Nos. 51420105014 and 61976118), and ‘the Postgraduate Research & Practice Innovation Program of Jiangsu Province’ (Grant No. KYCX18_0583).

REFERENCES


Chu, H., Wei, J., Li, J., Qiao, Z. & Cao, J. 2017 Improved medium- and long-term runoff forecasting using a multimodel approach in the Yellow River Headwaters region based on large-scale and local-scale climate information. *Water* 9 (8), 608.


