Streamflow forecasting based on the hybrid decomposition-ensemble model

Xiaomei Sun

Shaanxi Provincial Land Engineering Construction Group Co., Ltd.

Haiou Zhang

Shaanxi Provincial Land Engineering Construction Group Co., Ltd.

Jian Wang

Shaanxi Provincial Land Engineering Construction Group Co., Ltd.

Chendi Shi

Shaanxi Provincial Land Engineering Construction Group Co., Ltd.

Dongwen Hua

Shaanxi Provincial Land Engineering Construction Group Co., Ltd.

Juan Li (✉ 757258509@qq.com)

Shaanxi Provincial Land Engineering Construction Group Co., Ltd.

Research Article

Keywords: Streamflow forecasting, Variational mode decomposition, Long short-term memory, Ensemble tree

Posted Date: August 17th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-800297/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Streamflow forecasting based on the hybrid decomposition-ensemble model

Xiaomei Sun^{1,2,3}, Haiou Zhang^{1,2,3}, Jian Wang^{1,2,3}, Chendi Shi^{1,2,3}, Dongwen Hua^{1,2,3}, Juan Li^{1,2,3,*}

¹ Shaanxi Provincial Land Engineering Construction Group Co., Ltd., Xi'an 710075, China

² Institute of Land Engineering and Technology, Shaanxi Provincial Land Engineering Construction Group Co., Ltd., Xi'an 710021, China

³ Shaanxi Provincial Land Consolidation Engineering Technology Research Center, Xi'an 710075, China

* Corresponding author: Juan Li

E-mail: 757258509@qq.com

Abstract

Reliable and accurate streamflow forecasting plays a vital role in the optimal management of water resources. To improve the stability and accuracy of streamflow forecasting, a hybrid decomposition-ensemble model named VMD-LSTM-GBRT, which is sensitive to sampling, noise and long historical changes of streamflow, was established. The variational mode decomposition (VMD) algorithm was first applied to extract features, which were then learned by several long short-term memory (LSTM) networks. Simultaneously, an ensemble tree, a gradient boosting tree for regression (GBRT), was trained to model the relationships between the extracted features and the original streamflow. The outputs of these LSTMs were finally reconstructed by the GBRT model to obtain the forecasting streamflow results. A historical daily streamflow series (from 1/1/1997 to 31/12/2014) for Yangxian station, Han River, China, was investigated by the proposed model. VMD-LSTM-GBRT was compared with respect to three aspects: (1) Feature extraction algorithm; ensemble empirical mode decomposition (EEMD) was used. (2) Feature learning techniques; deep neural networks (DNNs) and support vector machines for regression (SVRs) were exploited. (3) Ensemble strategy; the summation strategy was used. The results indicate that the VMD-LSTM-GBRT model overwhelms all other peer models in terms of the root mean square error (RMSE=36.3692), determination coefficient ($R^2=0.9890$), mean absolute error (MAE=9.5246) and peak percentage threshold statistics (PPTS(5)=0.0391%). The addressed approach based on the memory of long historical changes with deep feature representations had good stability and high prediction precision.

Keywords: Streamflow forecasting; Variational mode decomposition; Long short-term memory; Ensemble tree

1. Introduction

Streamflow forecasting is of great significance for the optimal management and effective operation of a water resources system. Therefore, it has been investigated by many researchers, and numerous forecasting models have been developed in the past decades. Among these models, forecasting techniques based on statistical modeling, data-driven models, seem to be in fashion for their simplicity and robustness (Huang et al., 2014; Lima et al., 2016; Shiri and Kisi, 2010).

These data-driven models can be classified into time series models and artificial intelligence (AI) models. Many previous researchers have applied time-series models to forecast streamflow, including autoregression (AR), moving average (MA), autoregressive moving average (ARMA), and autoregressive integrated moving average (ARIMA) models (Castellano-Méndez et al., 2004; Mohammadi et al., 2006; Valipour et al., 2013). However, due to the linear hypothesis of these models, they are not suited to forecast streamflow with non-linear and non-stationary characteristics. Therefore, AI models that have the ability of non-linear mapping are applied to streamflow forecasting, i.e., support vector machines for regression (SVRs) (Huang et al., 2014; Yu et al., 2018), fuzzy inference systems (FIS) (He et al., 2014; Yaseen et al., 2017), Bayesian regression (BR) (Humphrey et al., 2016; Wang et al., 2017), and artificial neural networks (ANNs) (Tan et al., 2018; Wang et al., 2009). However, most of the AI models, which belong to the “shallow” learning category, cannot sufficiently represent instinctual information (Bai et al., 2016). The deep learning models, e.g., the deep belief network (DBN) and recurrent neural networks (RNNs), can overcome this drawback due to their deeper representation ability (Bai et al., 2016). However, these deep learning models completely rely on historical observed data, and some of the earlier changes of streamflow may or may not influence future streamflow. It is entirely possible for the gap between the streamflow information from further back in time and the current point where it is needed to become large. Therefore, using a deep learning model that can automatically “remember” or “forget” previous information should be able to enhance the accuracy of streamflow forecasting. Fortunately, LSTM (Hochreiter and Schmidhuber, 1997), one of the deep learning models, has the ability to tackle this task. LSTM has been successfully used in some fields, e.g., accident diagnosis (Yang and Kim, 2018), electricity price prediction (Peng et al., 2018), water table depth forecasting (Zhang et al., 2018), and others.

Unfortunately, due to the complicated non-linearity, extreme irregularity and multiscale variability of natural streamflow, the models directly built on original streamflow cannot appropriately identify streamflow change patterns (Huang et al., 2014). For this reason, the processes of transformation, data pre-processing and feature extraction have attracted the attention of many researchers. In addition, feature extraction can efficiently improve the capability of these models (Bai et al., 2016; Huang et al., 2014; Karthikeyan and Nagesh Kumar, 2013; Seo et al., 2015). Huang et al. (2014) applied a modified empirical model decomposition (EMD) method to remove the most nonlinear and disorderly noise from the original series and then established one SVR-based model that computed a summation of all prediction results of all sub-series as an ensemble result. Wang et al. (2015) used the EEMD technique to develop insight into the runoff characteristics and forecast each characteristic by the ARIMA model; the forecast results were summed to formulate an ensemble forecast for the raw runoff. Bai et al. (2016) used EEMD to extract multi-scale features

and reconstructed three deep neural networks (DNNs) by a summing strategy to forecast reservoir inflow. Yu et al. (2018) exploited both Fourier transform (FT) and SVR models to extract and learn features and to forecast monthly reservoir inflow by adding all feature learning results. Obviously, feature representation of original data can contribute to the performance improvement of streamflow forecasting because of the advantage of feature extraction, which removes noise components and detect the hidden structures inside the raw time-series.

However, some recent commonly-used feature extraction methods, e.g., EMD, EEMD, and wavelet transforms (WT), suffer from drawbacks with respect to different aspects of actual signal decomposition. For example, EMD has limitations of sensitivity to noise and sampling (Dragomiretskiy and Zosso, 2014), EEMD is not theoretically well-founded (Liu et al., 2018b), and the effectiveness of WT heavily relies on the choice of the basic wavelet function (Naik et al., 2018). Recently, a theoretically well-founded and robust VMD model (Dragomiretskiy and Zosso, 2014) has been successfully applied to container throughput forecasting (Niu et al., 2018), vibro-acoustic feature extraction (Mohanty et al., 2018), chatter detection in milling processors (Liu et al., 2018a), and other applications. This model is much more sensitive to noise and sampling than existing decomposition algorithms, such as EMD and EEMD (Dragomiretskiy and Zosso, 2014).

Moreover, the ensemble strategy plays a vital role for integrating feature forecasting results to predict original streamflow. A straightforward and frequently used ensemble technique is summation, although it may cause error accumulation problems due to summation errors of the sub-results of streamflow prediction. Sometimes, the gap between the summation of extracted features and the original values may not be small. Even a model that can achieve great performance in the prediction of sub-features may still not be able to accurately forecast the original time series. Therefore, building another supervised model (Shiri and Kisi, 2010), such as GBRT, for ensembles seems to be a good choice to avoid an accumulation of errors and obtain better performance.

Based on the above outline, this study addresses a decomposition-ensemble-based multiscale feature deep learning method to forecast streamflow. Our goal is to plug a memory framework into the process of deep feature learning that is robust to noise and sampling as well as long historical changes of streamflow, integrate an ensemble tree model with the capability to remove impacts caused by error accumulation and model the relationship between decomposition results and the original series to exploit the sophisticated nature of streamflow with a long history. To this end, VMD was used to extract smooth features, LSTM was applied to learn features sensitive to long historical streamflow changes, and GBRT was used to obtain an ensemble model to forecast the streamflow. This approach was evaluated by observed daily streamflow of the Yangxian station, Han River, China.

The rest of this paper is organized as follows: Section 2 describes the methodologies mentioned above, including VMD, LSTM, GBRT, and the proposed model VMD-LSTM-GBRT. Section 3 provides a brief introduction to the study site, data and data pre-processing techniques as well as model evaluation criteria. Section 4 presents the results and discussions of the case study, and the conclusions are presented in Section 5.

2. Methodologies

2.1. Variational mode decomposition (VMD)

The VMD algorithm, an entirely non-recursive variational mode decomposition model proposed by Dragomiretskiy and Zosso (Dragomiretskiy and Zosso, 2014), is used to concurrently decompose a sophisticated signal into several band-limited intrinsic modes.

The VMD model assumes each mode u_k to be mainly compact around a center pulsation ω_k calculated with the decomposition. The following scheme proposed by Dragomiretskiy and Zosso (2014) is applied to assess the bandwidth of a mode. The related analytic signal of each mode u_k is first computed by the Hilbert transform to acquire a unilateral frequency spectrum. Then, the frequency spectrum of each model is shifted to “baseband” by mixing with an exponential tuned to the respective evaluated center frequency. The bandwidth is finally assessed by the H^1 Gaussian smoothness of the demodulated signal, i.e., the squared L^2 -norm of the gradient (Dragomiretskiy and Zosso, 2014). To solve the decomposition problem of time series f , the constrained variational problem can be equivalently solved by the following equation:

$$\min_{\{u_k\}, \{\omega_k\}} \left\{ \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \right\} \quad (1)$$

$$s.t. \quad \sum_k u_k(t) = f(t)$$

where $\{u_k\} := \{u_1, \dots, u_k\}$ and $\{\omega_k\} := \{\omega_1, \dots, \omega_k\}$ denote the set of modes and their center frequencies, respectively. To solve this variational problem, a Lagrangian multiplier λ and a quadratic penalty term are introduced to render the problem unconstrained. The augmented Lagrangian ℓ is defined as follows:

$$\ell(\{u_k\}, \{\omega_k\}, \lambda) := \alpha \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 + \left\| f(t) - \sum_k u_k(t) \right\|_2^2 \quad (2)$$

$$+ \left\langle \lambda(t), f(t) - \sum_k u_k(t) \right\rangle$$

in which α indicates the balancing parameter of the data-fidelity constraint. In VMD, the Alternate Direction Method of Multipliers (ANMM) is used to solve Eq. (2). Eq. (3) is used to update the mode $u_k(\omega)$ in the frequency domain. The center frequencies ω_k are updated by Eq. (4), and λ is simultaneously updated by Eq. (5). In the time domain, the mode $u_k(t)$ can be obtained as the real part of the inverse Fourier transform of $u_k(\omega)$ expressed by Eq.(3).

$$\hat{u}_k^{n+1}(\omega) = \frac{\hat{f}(\omega) - \sum_{i \neq k} \hat{u}_i + \frac{\hat{\lambda}(\omega)}{2}}{1 + 2\alpha(\omega - \omega_k)^2} \quad (3)$$

$$\hat{\omega}_k^{n+1} = \frac{\int_0^\infty \omega |\hat{u}_k(\omega)|^2 d\omega}{\int_0^\infty |\hat{u}_k(\omega)|^2 d\omega} \quad (4)$$

$$\hat{\lambda}^{n+1}(\omega) = \hat{\lambda}^n(\omega) + \tau \left(\hat{f}(\omega) - \sum_k \hat{u}_k^{n+1}(\omega) \right) \quad (5)$$

The implementation process of the VMD model is summarized as Algorithm 1.

Algorithm 1: The process of VMD

Initialize $\{\hat{u}_k^1\}, \{\hat{\omega}_k^1\}, \hat{\lambda}^1, n \leftarrow 0$

Repeat

$n \leftarrow n + 1$

for $k = 1 : K$ **do**

Update u_k for all $\omega \geq 0$ using Eq. (3).

Update $\hat{\omega}_k$ using Eq. (4).

end for

Update $\hat{\lambda}^n(\omega)$ for all $\omega \geq 0$ using Eq. (5)

until convergence: $\sum_k \|\hat{u}_k^{n+1} - \hat{u}_k^n\|_2^2 / \|\hat{u}_k^n\|_2^2 < \varepsilon$.

Obtain $u_k^{n+1}(t)$ by the fast Fourier transform of $\hat{u}_k^{n+1}(\omega)$.

2.2. Long short-term memory (LSTM)

Long Short-Term Memory Networks (LSTMs) are a very specific kind of Recurrent Neural Networks (RNNs) for modeling sequential data. Therefore, it is essential to first introduce a normal version of an RNN. RNNs have chain-like structures of repeating modules that produce an output at each time step and have recurrent connections from the output at one time step to the hidden units at the next time step, illustrated in **Fig. 1** (a). The chain-like structure with self-connected hidden units can help RNNs to “remember” the previous information, which allows the RNNs to build a model for an arbitrarily-long time sequence.

The forward propagation algorithm is used to calculate the output for the RNN pictured in **Fig.**

1 (a). Begin with a specification of the initial state $h_0 = 0$ for each time step from $t = 1$ to $t = \tau$;
the following equations are used (Goodfellow et al., 2018):

$$h_t = \tanh(W h_{t-1} + U X_t + b_h) \quad (6)$$

$$o_t = V h_t + b_o \quad (7)$$

where the parameters are the bias vectors b_h and b_o along with the weight matrices U , V and W
for input-to-hidden, hidden-to-output and hidden-to-hidden connections, respectively. X_t
represents the input vector at time step t and h_{t-1} denotes the hidden cell state at time step $t-1$.

Back-Propagation Through Time (BPTT) is used to compute the gradients of the RNNs
(Werbos, 1990). However, owing to the gradient vanishing or exploding problem, it is difficult and
inefficient for BPTT to learn long-term dependencies in RNNs (Bengio et al., 1994; Goodfellow et
al., 2018; Hochreiter, 1998). LSTMs are explicitly designed by Hochreiter and Schmidhuber (1997)
to avoid this long-term dependency problem. LSTMs also have chain-like repeating modules,
although with complicated structures. Each repeating module of LSTMs includes a memory block
called a “cell”. This memory block help LSTMs to store or remove information over a long duration.

Fig. 1 (a) Chain-like structure of the RNN. Because of the connections between hidden units, information can be passed from one time step to the next. (b) A graphical representation of the LSTM recurrent network with the memory cell block.

The LSTM memory block diagram is illustrated in **Fig. 1** (b). The LSTM memory block contains four parts, a cell state in addition to three special structures called gates. The horizontal

line running through the top of the diagram is the cell state, which runs straight down the entire chain without any activation function; it is very easy for information to just flow along it unchanged. Therefore, the gradient does not vanish or explode when training an LSTM by BPTT. Moreover, the LSTM does have the ability to add or remove information to the cell state, regulated by the input, forget and output gates. Each gate is composed of a sigmoid unit and a pointwise multiplication operation, which can optionally pass information.

The corresponding forward propagation equations of LSTM are expressed for time steps from $t = 1$ to $t = \tau$ with initial state $C_0 = 0$ and $H_0 = 0$ as:

$$i_t = \sigma(W_i X_t + U_i h_{t-1} + b_i) \quad (8)$$

$$f_t = \sigma(W_f X_t + U_f h_{t-1} + b_f) \quad (9)$$

$$o_t = \sigma(W_o X_t + U_o h_{t-1} + b_o) \quad (10)$$

$$\tilde{C}_t = \tanh(W_c X_t + U_c h_{t-1} + b_c) \quad (11)$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t \quad (12)$$

$$h_t = o_t * \tanh(C_t) \quad (13)$$

where W , U and b are input weights, recurrent weights and biases, respectively, and the subscripts i , f and o represent the input, forget and output gates, respectively. The activation function *logistic sigmoid* is indicated by σ . i_t , f_t , o_t and C_t are the input, forget, output gates and cell state vectors at time step t , respectively. h_t is the output vector of the memory cell block, and $\otimes$ denotes element-wise multiplication.

2.3. Gradient Boosting Regression Trees (GBRTs)

Gradient boosting is a powerful machine learning strategy to efficiently produce highly robust, competitive, interpretable procedures for both regression and classification (Friedman, 2001; Friedman, 2002). The key to boosting is to combine the output of many weak prediction models ("learners"), typically decision trees, into a single strong ensemble model. Gradient boosting builds models in a forward stage-wise fashion. Therefore, for each stage m , $1 \leq m \leq M$, of gradient boosting:

$$F_m(x) = F_{m-1}(x) + h_m(x) \quad (14)$$

in which $h_m(x)$ are the basic estimators referred to as weak prediction models (small regression trees in the case of GBRT) and $F_m(x)$ is the summation of m small regression trees for GBRT. For iterations from $m=1$ to $m=M$, the GBRT algorithm improves on F_m by adding a new regression tree h_m to its predecessor to provide a better model. Simultaneously, the procedure estimates the target value y based on the perfect h_m from the training set, which would imply:

$$F_m(x) = F_{m-1}(x) + h_m(x) = y \quad (15)$$

which is equivalent to

$$h_m(x) = y - F_{m-1}(x) \quad (16)$$

Therefore, h_m is the regression tree model that fits the current residuals $\gamma_m = y - F_{m-1}(x)$, and the residuals $y - F_{m-1}(x)$ for a given model are the negative of the squared error loss function, i.e.:

$$-\frac{\partial \frac{1}{2}(y - F_{m-1}(x))^2}{\partial F_{m-1}(x)} = y - F_{m-1}(x) \quad (17)$$

Gradient boosting is thus a gradient descent algorithm obviously proved by Eq. (17), and generalizing it entails substituting the squared error with a different loss function and its gradient. For a more detailed description, see Friedman (2001) and Hastie et al. (2009).

Moreover, the implicit idea behind gradient boosting is to apply a steepest-descent step to minimize the loss values between the response values and estimates to find an optimal approximation $\hat{F}(x)$. Therefore, for a training set $\{(x_1, y_1), \dots, (x_n, y_n)\}$, the ensemble model would be updated in accordance with the following equations (Friedman, 2001; Hastie et al., 2009):

$$F_m(x) = F_{m-1}(x) - \gamma_m \sum_{i=1}^n \frac{\partial L(y_i - F_{m-1}(x_i))}{\partial F_{m-1}(x_i)} \quad (18)$$

$$\gamma_m = \arg \min_{\gamma} \sum_{i=1}^n L \left(y_i - F_{m-1}(x_i) - \gamma \frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)} \right) \quad (19)$$

where the derivatives are obtained with respect to the function F_i for $i \in \{1, 2, \dots, m\}$. In the m -th iteration of the GBRT model, the gradient boosting algorithm fits a regression tree $h_m(x)$ to the

226 pseudo-residuals. Let J_m be the number of tree leaves; the regression tree splits the input space
 227 into J_m disjoint regions $R_{1m}, \dots, R_{J_m m}$ and obtains a constant value for each region. The output
 228 of $h_m(x)$ for input x can thus be written as the sum (Friedman, 2001):

$$229 \quad h_m(x) = \sum_{j=1}^{J_m} b_{jm} 1_{R_{jm}}(x), (x \in R_{jm}) \quad (20)$$

230 where b_{jm} is the constant value predicted for the region R_{jm} and $1(\cdot)$ is an indicator function
 231 that has the value 1 if its argument is true and zero otherwise. Then, each coefficient b_{jm} is
 232 multiplied by an optimal value γ_{jm} (Friedman, 2001), and the model is then updated by the
 233 following rules:

$$234 \quad F_m(x) = F_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} 1_{R_{jm}}(x), (x \in R_{jm}) \quad (21)$$

$$235 \quad \gamma_{jm} = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + \gamma) \quad (22)$$

236 The implementation process of the generic gradient boosting tree is summarized as Algorithm
 237 2.

238 **Algorithm 3: The process of GBRT.**

Input the loss function $L(y, F(x))$, iteration number M and training set $\{(x_i, y_i)\}_{i=1}^n$.

1. Initialize model with a constant value:

$$F_0(x) = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, \gamma)$$

2. For $m = 1$ to M :

2.1. For $i = 1, 2, \dots, n$, compute pseudo-residuals:

$$\gamma_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}$$

2.2. Calculate the multiplier γ_{jm} by solving the following problem:

$$\gamma_{jm} = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + \gamma)$$

2.3. Fit a weak learner $h_m(x)$ to the pseudo-residuals using the training set $\{(x_i, y_i)\}_{i=1}^n$.

$$h_m(x) = \sum_{j=1}^{J_m} \gamma_{jm} 1_{R_{jm}}(x), (x \in R_{jm})$$

2.4. Update the model:

$$F_m(x) = F_{m-1}(x) + h_m(x)$$

3. Output $F_M(x)$.

2.4. The decomposition-ensemble model VMD-LSTM-GBRT

After discussing each key constituent separately, the approach of the proposed model VMD-LSTM-GBRT can be concluded as follows and is diagrammed in **Fig. 2**.

Step 1. Collect raw daily streamflow data $X = \{x_1, x_2, \dots, x_N\}$.

Step 2. Use VMD to decompose the raw series X into several components.

Step 3. Plot the partial autocorrelation coefficient figure of each component obtained in step 2 to select optimal numbers of inputs for it. Divide each of the components into three sub-sets: the training set (80%) for training multiple LSTM structures, the development set (10%) for searching optimal structure, and the test set (10%) for validating the ensemble model VMD-LSTM-GBRT.

Step 4. Given the test set, predict each component based on the optimal LSTM structure of each mode obtained in step 3.

Step 5. Build the ensemble tree model GBRT using the components obtained in step 2 as input and the original series obtained in step 1 as output. Use GBRT to reconstruct the predictions given by step 4.

Step 6. Output the forecasting streamflow results and perform error analysis.

Fig. 2 Application of the proposed model VMD-LSTM-GBRT

3. Case Study

3.1. Study site and observational data

In this paper, historical daily streamflow of the Yangxian hydrological station on the upstream of Han River were collected to assess the proposed model. The location of Yangxian station is illustrated in **Fig. 3**. The Han River, the biggest tributary of the Yangtze River, lies within 30°-34.5°N and 107°-114.5°E in the middle of China and has a total drainage area of 151,000 km². The location of this basin is in a subtropical monsoon zone that has a humid climate and differentiated seasons. The water resources in this area are rich, with an annual average precipitation of more than 900 mm/yr. The precipitation in the rainy season (July to September) accounts for 75% of the annual total, and the runoff has a similar seasonality. The study area of this paper is the upper source area of the Han River, which lies between BoZhong Mountain located in the south of the Qinling Mountains and the Yangxian hydrological station. The drainage area controlled by the Yangxian hydrological stations is 14,484 km². As the main stream control station, forecasting the daily runoff of the Han River at Yangxian hydrological station can evaluate the short-term water production at the source of this basin.

Fig. 3 Location of the Yangxian hydrological station

As shown in **Fig. 4**, daily streamflow records (total of 6,574 samples) of the Yangxian hydrological station from 1/1/1997 to 31/12/2014 were used to develop the present model. For simplicity, the observation dates on the horizontal axis have been replaced with series numbers. These records were collected from the hydrological information datacenter of Shaanxi Hydrographic and the Water Resources Survey Bureau. The instantaneous value (m³/s) observed at 8 a.m. was selected as the average daily streamflow.

Fig. 4 Daily Streamflow of the Yangxian station from 1/1/1997 to 31/12/2014

3.2. Data pre-processing

Since the range of the streamflow time series and its decomposition sequences vary widely, in some cases of building machine learning models, the optimization algorithms applied for the loss function will not work well without feature normalization. Therefore, all the variables used for the model developed in the present study were normalized to the same scale. This pre-processing strategy can ensure that the optimization algorithm converges much faster than without normalization (Ioffe and Szegedy, 2015). The normalization formula is as follows:

$$X_{normalized} = 2 * \frac{X - X_{min}}{X_{max} - X_{min}} - 1 \quad (23)$$

where $X_{normalized}$ is the normalized vector and X is the raw vector. X_{max} and X_{min} are the maximum value and minimum value of X , respectively, and $X_{normalized}$ is calculated by element-wise mathematical operations. Once we have finished the simulation, the predictions can be re-normalized following the inverse procedure of Eq. (23).

3.3. Model evaluation criteria

To evaluate the performance of the proposed model based on the decomposition-ensemble strategy, four error analysis criteria were applied. The expression of these criteria is shown in **Table 1**. The RMSE evaluates the performance of predicting high streamflow values, whereas the MAE accesses the average performance of the entire data set. The coefficient of determination R^2 indicates how well the observations are replicated by the proposed model. The peak percentage of threshold statistics, PPTS, denotes the ability to forecast peak flow (Bai et al., 2016; Stojković et al., 2017). The lower the PPTS, the better the capability to forecast peak flow. Note that the records are arranged in descending order to compute the PPTS and that the threshold level γ denotes the percentage of bottom data removed from this order; the parameter G is the number of top data at the threshold level γ .

Table 1 Formulas for error analysis criteria

Error analysis criteria	Definition
Root mean square error	$RMSE = \sqrt{\frac{\sum_{t=1}^N (x(t) - \hat{x}(t))^2}{N}}$
Mean absolute error	$MAE = \frac{\sum_{t=1}^N x(t) - \hat{x}(t) }{N}$
Determination coefficient	$R^2 = 1 - \frac{\sum_{t=1}^N (x(t) - \hat{x}(t))^2}{\sum_{t=1}^N (x(t) - \bar{x}(t))^2}$

Peak percentage threshold statistics (%)

$$PPTS(\gamma) = \frac{1}{100 - \gamma} \frac{1}{N} \sum_{t=1}^G \left| \frac{x(t) - \hat{x}(t)}{x(t)} \cdot 100 \right|$$

Note: N is the number of samples, $x(t)$ is the original series, $\bar{x}(t)$ is the average of the original series and $\hat{x}(t)$ is the predicted series.

4. Results and discussion

4.1. Data decomposition with VMD

As mentioned in Section 2.4, when building the decomposition-ensemble based model, we first decomposed the raw daily streamflow data of the Yangxian hydrological station via VMD. The raw series and its decomposition results and the frequency spectra are shown in **Fig. 5**. However, it is hard to tell how many components the original series should be decomposed into. Too few components may not properly extract features inside raw data, whereas too many may be computationally expensive for training the model. By experiments, we found that the optimal decomposition mode number can be determined by the obvious aliasing phenomenon of the center frequency for the last component. It was first found in this study that when $k=10$, the frequency spectrum of the 10th mode had obvious aliasing phenomena (area surrounded by a red rectangular border shown in **Fig. 6**). To make the decomposition result satisfy orthogonality and avoid the spurious components as much as possible, the number of components was chosen to be 9.

Fig. 5 VMD decomposition results: (a) the decomposition sequence waveform and (b) the frequency spectrum representation

Fig. 6 Schematic diagram of the center frequency aliasing of the last IMF: (a) the last two sequence waveforms and (b) the frequency spectrum representations. The area surrounded by the red rectangular border indicates the aliasing

4.2. Multiscale deep feature learning with LSTM

The numbers of inputs, network structure and other hyper-parameters are vital variables in an LSTM model. Therefore, according to the input selection method introduced by He et al. (2014), Huang et al. (2014), and Wang et al. (2009), the input variables could first be easily obtained by observing the plot of Partial Autocorrelation Functions (PACFs) illustrated by Fig. 7, in which $PACF_1$ - $PACF_9$ denotes the PACFs of each component. In other words, we assume that the output is $X(t+1)$ and $X(t+1-k)$ is then selected as one of the input variables under the condition that PACF at lag k is out of the 95% confidence interval indicated by the blue lines in Fig. 7. Fig. 7 shows that almost all the PACFs of each component are out of the range. Therefore, we select the 20 days of lag form $X(t-19)$ to $X(t)$ as the input variables of each sub-series.

Fig. 7 PACFs of subseries of daily streamflow during the period 1997/01/01 to 2014/12/31 for the Yangxian hydrological station

Since there is no mature approach in theory to determine the number of hidden units and hidden layers, an experimental method can be used to select these two parameters for the LSTM network structure. In the experiment of this study, the sub-series shown in Fig. 6, first normalized by Eq. (23), were split into three parts: the training data set (1/1/1997-27/5/2011), the development set (28/5/2011-14/3/2013) and the testing set (15/3/2013-31/12/2014). The number of hidden units for each hidden layer was designed for 11 levels ranging from 15 to 25, and the number of input variables equaled the median of this interval. The hidden layers of LSTM were initialized from 1 to 5. Therefore, for each component of the original series, there were 55 LSTM network structures in

this experiment. Training and developing the LSTM model to predict the first component, IMF₁, is an example used to describe the experimental process.

Fig. 8 PPTS(5) of different LSTM structures for predicting streamflow of IMF₁ during the training and development period: (a) line chart plots of PPTS(5) for different hidden layers and (b) boxplots of the optimal structure of each hidden layer

For each network structure mentioned in the previous paragraph, we first initialized an LSTM model with 20 input units and then trained these 55 structures using the training set. Then, the streamflow during the development period was forecast based on the trained structures. To find the optimal model structure, PPTS(5), illustrated in **Fig. 8**, was calculated for the training and development set. **Fig. 8** (a) shows the changes of PPTS(5) of the training and development set for different levels of the hidden layers. According to the rule of bias and variance tradeoff (Valentini and Dietterich, 2004), when the PPTS(5) of the development set is close to the PPTS(5) of the

training set and both of these values are small, the model structure will obtain a great generalization and predicting ability. Therefore, we selected 20-15-1, 20-19-19-1, 20-19-19-19-1, 20-17-17-17-17-1 and 20-21-21-21-21-21-1 as the optimal structure of the 1 to 5 hidden layers, respectively. 20-19-19-1 means that the structure has 20 input features, 1 output target, and two hidden layers, with 19 hidden units for each layer. **Fig. 8** (b) shows the boxplots of optimal structure for each level of the hidden layers, where the upper and lower quartiles are determined by the PPTS(5) of the training and development set. The range between the upper and lower quartiles indicates the degree of bias and variance tradeoff; the smaller the range, the better the tradeoff. From **Fig. 8** (b), one can find that the structure 20-15-1 has the best bias and variance tradeoff. Therefore, the model structure 20-15-1 was selected as the optimal model to predict IMF₁.

To validate the optimal model for forecasting each sub-series, the predictions during the test period were renormalized to the original scale and are plotted in **Fig. 9**. The PPTS(5) and R² of the whole components during the training and development period are listed in **Table 2**. From **Fig. 9** and **Table 2**, we can see that all the trained LSTM structures of all the sub-series have good accuracy.

Fig. 9 Forecasting result of sub-series during the testing period

Table 2 Results of evaluation criteria with different hidden layers and hidden units for sub-sequences

Sequence	Hidden layers	Hidden Units	Training		Developing	
			PPTS(5)(%)	R ²	PPTS(5)(%)	R ²
IMF1	1	15	0.0025	1.0000	0.0024	0.9999
IMF2	2	22	0.0396	0.9999	0.0227	0.9999
IMF3	1	16	0.2684	0.9999	0.2163	0.9997
IMF4	3	20	0.7646	0.9997	0.7156	0.9966
IMF5	4	20	0.8728	0.9993	0.2523	0.9983
IMF6	2	21	0.5031	0.9986	1.9367	0.9960
IMF7	2	23	1.2368	0.9973	0.4885	0.9943
IMF8	5	20	1.8371	0.9952	4.4562	0.9832
IMF9	1	25	1.5380	0.9972	2.1116	0.9897

4.3. Training ensemble tree model GBRT

We can simultaneously build an ensemble tree, the GBRT model, to represent the relationship between the original series and the sub-series decomposed by VMD; the GBRT algorithm can learn an ensemble function to reconstruct all the sub-series into a streamflow series. To find the optimal hyper-parameters, Bayesian optimization based on Gaussian processes was applied (Bergstra et al., 2011; Shahriari et al., 2016). The entire data set for building the GBRT mode, consisting of 9 sub-series as input and raw streamflow as the output target, was divided into two parts: the training-validating set (1/1/1997 - 14/3/2013) and the testing set (15/3/2013 - 31/12/2014). The famous machine learning toolkit scikit-learn (Pedregosa et al., 2011) was applied to train the GBRT model by use of the training-validating set. To improve the performance of GBRT, 6-fold cross-validation was used. The optimal value of the hyper-parameters for GBRT, i.e., learning rate, maximum depth, maximum features, minimum sample split and minimum sample leaf, were 0.08, 25, 9, 9 and 10, respectively. The performance evaluation results of RMSE=45.9766, R²=0.9825, MAE=11.4565 and PPTS(5)=0.0438% indicate that GBRT had a good precision for multiscale feature ensembles.

4.4. Predicting results of VMD-LSTM-GBRT

Once the feature learning models based on LSTM were built, we could ensemble the sub-results to forecast the original streamflow. There are two ensemble methods: summation and GBRT. We first summed all 9 sub-results obtained in Section 4.2 to obtain the forecasting results of the original streamflow. The streamflows predicted by the model with ensemble summation, VMD-LSTM-SUM, are illustrated in **Fig. 10**. It can be observed from **Fig. 10** (a) that the predictions of a summation method can follow the changes of records during the test period but the accuracy of the peak streamflow forecasting is poor. Moreover, the scatter plot as shown in **Fig. 10** (b) indicates that the peak predictions are not appropriately concentrated near the ideal fit. The detailed plot of the predictions during the period 07/09/2014-28/09/2014 illustrated in **Fig. 10** (c) can also prove that point. By experiment, we found that the summation results of the 9 components decomposed by

VMD were quite different from the original streamflow at the peak. Therefore, we could not simply sum the predictions of the 9 components to forecast the original streamflow; the ensemble tree model obtained in Section 4.3, GBRT, was applied to ensemble the sub-predictions predicted by LSTM. The final forecasting results forecast by the proposed model, VMD-LSTM-GBRT, are also shown in **Fig. 10**. From **Fig. 10** (a), one can find that the peak predictions obtained by VMD-LSTM-GBRT are closer to the original streamflow. Moreover, the scatter plot as shown in **Fig. 10** (b) indicates that the predictions forecast by VMD-LSTM-GBRT concentrated near the ideal fit and agreed better with the records, which could also be proved by the small angle between the linear fit and the ideal fit. The predictions of the sub-set (07/09/2014-28/09/2014) of the test set shown in **Fig. 10** (c) denotes that the proposed model has better performance at the peak flow. Therefore, the proposed model, VMD-LSTM-GBRT, has a better capability of peak streamflow forecasting than VMD-LSTM-SUM.

Fig. 10 Comparison of prediction results of the test dataset using different ensemble techniques, GBRT and summation: (a) plots of the prediction results and records, (b) scatters for the test set (15/3/2013-31/12/2014), and (c) plots of the prediction results for the period 07/09/2014-28/09/2014

To assess the forecasting performance of the proposed model, a different decomposition algorithm, EEMD, and two different feature learning models, DNN and SVR, were applied for the comparisons using the identical dataset shown in **Fig. 4**. The building process of these models is the same as the approach mentioned in Section 2.4, except that the decomposition algorithm and the feature learning model are replaced, respectively.

Fig. 11 Comparison of prediction results of the test dataset using the different decomposition algorithms VMD and EEMD: (a) forecasting results, (b) scatters for the testing data (15/3/2013-31/12/2014), and (c) plots of the prediction results for the period 07/09/2014-28/09/2014

Fig. 11 plots the streamflow predictions of Yangxian station by the proposed model VMD-LSTM-GBRT and the decomposition method-substituted model EEMD-LSTM-GBRT. As shown in **Fig. 11** (a), the proposed model performed better for peak flow forecasting than the traditional decomposition method EEMD, which can be validated by **Fig. 11** (c). From the scatter plot illustrated by **Fig. 11** (b), one can find that the recorded predicted values of the proposed model are much more concentrated than the model using EEMD. Moreover, the comparison of prediction performance between the proposed feature learning model LSTM and the other two machine learning models, DNN and SVR, were conducted and are indicated by **Fig. 12**. However, from the forecasting results shown in **Fig. 12** (a) and the scatters shown in **Fig. 12** (b), one can observe that the difference between the three feature learning models is not that obvious. However, we can still determine that the best feature learning model is the LSTM from the quantitative evaluations given in **Table 3** and the detailed predictions shown in **Fig. 12** (c). As shown in **Table 3**, the proposed model VMD-LSTM-GBRT has the lowest RMSE, MAE, PPTS(5) and the highest R^2 among these decomposition-ensemble-based models, which illustrates the proposed model superiority for both peak streamflow forecasting and global changes.

Fig. 12 Comparison of prediction results of the test dataset using the different feature learning models LSTM, DNN and SVR: (a) forecasting results, (b) scatters for the testing data (15/3/2013-31/12/2014), and (c) plots of the prediction results during the period 07/09/2014-28/09/2014

Table 3 Comparison of the forecasting performances using different models based on the decomposition-ensemble strategy

Model	Performance criteria			
	RMSE	R ²	MAE	PPTS(5)(%)
VMD-LSTM-GBRT	36.3692	0.9890	9.5246	0.0391
VMD-LSTM-SUM	67.8297	0.9619	27.8412	0.1500
EEMD-LSTM-GBRT	87.4506	0.9366	22.0321	0.0883
VMD-DNN-GBRT	44.9735	0.9832	12.1853	0.0451
VMD-SVR-GBRT	47.0555	0.9816	12.8919	0.0472

In the light of the above comparisons, all results fully indicate that the proposed model based on the decomposition algorithm, VMD, the multiscale deep feature learning model, LSTM, and the ensemble tree model, GBRT, performs very well for streamflow forecasting.

5. Conclusions

In this paper, a decomposition-ensemble-based multiscale feature learning approach with hybrid models was developed for forecasting daily streamflow, and the approach was evaluated based on a historical river streamflow dataset for Yangxian station, Han River, China. To improve the accuracy and stability of forecasting, three aspects were considered in a comprehensive way: (1) Multiscale feature extraction by the algorithm with much more robustness with respect to sampling and noise; nine feature terms were extracted by VMD in this paper. (2) Deep feature learning with

a model that can predict streamflow depending on the long historical changes of river flow; nine LSTMs were applied to sufficiently learn each feature. (3) An ensemble model with supervised learning; an ensemble tree GBRT was used to reconstruct the sub-results to obtain the final forecasting results. The daily streamflow forecasting capability of this decomposition-ensemble-based approach was compared with respect to three aspects: comparison with an approach that has the same feature learning model and ensemble technique but uses the traditional decomposition algorithm EEMD, comparison with an approach with the same decomposition algorithm and feature learning model but using the summation ensemble strategy, and comparison with an approach that has the same decomposition algorithm and ensemble technique but uses two different models, DNN and SVR. The results denote that the proposed model VMD-LSTM-GBRT exhibits the best forecasting performance among all the peer approaches for both global changes and peak streamflow forecasting.

This study reports an approach to gain insight into the sophisticated features of natural river streamflow by designing a decomposition-ensemble framework. The three segments of this approach, i.e., using a robust model, VMD, to extract features that adequately represent natural river flow; using a long dependency model, LSTM, to remember or forget the historical changes of river flow; and reconstructing the extracted features by a tree model to remove the effect of error accumulation, are combined to determine what the values of the future streamflow should be. Note that the three segments of this framework can be replaced by other models, e.g., VMD can be replaced by an algorithm that is much more robust to sampling and noise but that still uses a regression strategy and replaces GBRT with DNN. Therefore, the present approach has value for river flow forecasting.

Funding

This work was supported by the Fund Project of Shaanxi Key Laboratory of Land Consolidation (2019-JC08) and Shaanxi Provincial Land Engineering Construction Group Internal Research Project (DJNY2020-15). Sincere gratitude is extended to the editor and anonymous reviewers for their professional comments and corrections.

Conflict of Interest

Not applicable.

Ethics Approval

Not applicable.

Consent to Participate:

Not applicable.

494 **Consent for Publication**

495 Not applicable.

496 **Availability of data and materials**

497 Not applicable.

498 **Code availability**

499 Not applicable.

500 **Authors' Contributions**

501 Conceptualization, **X.S. and J.L.**; Data curation, **X.S., J.W., C.S. and D.H.**; Formal analysis,
502 **X.S. and H.Z.**; Funding acquisition, **D.H.**; Investigation, **J.W., D.H. and C.S.**; Methodology, **X.S.**;
503 Project administration, **J.L.**; Resources, **J.L. and X.S.**; Supervision, **J.L.**; Validation, **H. Z. and**
504 **J.W.**; Writing – original draft, **X.S.**. All authors reviewed the manuscript.

505 **Acknowledgments**

506 This work was supported by the Fund Project of Shaanxi Key Laboratory of Land
507 Consolidation (2019-JC08) and Shaanxi Provincial Land Engineering Construction Group Internal
508 Research Project (DJNY2020-15). Sincere gratitude is extended to the editor and anonymous
509 reviewers for their professional comments and corrections.

510 **References**

- 511 Bai, Y., Chen, Z., Xie, J., Li, C., 2016. Daily reservoir inflow forecasting using multiscale deep
512 feature learning with hybrid models. *Journal of Hydrology*, 532: 193-206.
513 DOI:10.1016/j.jhydrol.2015.11.011
- 514 Bengio, Y., Simard, P., Frasconi, P., 1994. Learning Long-Term Dependencies with Gradient
515 Descent is Difficult. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 5(2): 157-166.
- 516 Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for Hyper-Parameter Optimization,
517 25th Annual Conference on Neural Information Processing Systems (NIPS 2011). Neural
518 Information Processing Systems Foundation, Granada, Spain.
- 519 Castellano-Méndez, M.a., González-Manteiga, W., Febrero-Bande, M., Manuel Prada-Sánchez, J.,
520 Lozano-Calderón, R., 2004. Modelling of the monthly and daily behaviour of the runoff of
521 the Xallas river using Box–Jenkins and neural networks methods. *Journal of Hydrology*,
522 296(1-4): 38-58. DOI:10.1016/j.jhydrol.2004.03.011

523 Dragomiretskiy, K., Zosso, D., 2014. Variational Mode Decomposition. *IEEE Transactions on*
524 *Signal Processing*, 62(3): 531-544. DOI:10.1109/tsp.2013.2288675

525 Friedman, J.H., 2001. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals*
526 *of Statistics*, 29(5): 1189-1232.

527 Friedman, J.H., 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):
528 367-378.

529 Goodfellow, I., Bengio, Y., Courville, A., 2018. *Deep Learning*. Adaptive Computation and Machine
530 Learning. The MIT Press.

531 Hastie, T., Tibshirani, R., Friedman, J., 2009. *The Elements of Statistical Learning*. Springer, 745
532 pp.

533 He, Z., Wen, X., Liu, H., Du, J., 2014. A comparative study of artificial neural network, adaptive
534 neuro fuzzy inference system and support vector machine for forecasting river flow in the
535 semiarid mountain region. *Journal of Hydrology*, 509: 379-386.
536 DOI:10.1016/j.jhydrol.2013.11.054

537 Hochreiter, S., 1998. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and
538 Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based*
539 *Systems*, 6(2): 107-116.

540 Hochreiter, S., Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735-
541 1780.

542 Huang, S., Chang, J., Huang, Q., Chen, Y., 2014. Monthly streamflow prediction using modified
543 EMD-based support vector machine. *Journal of Hydrology*, 511: 764-775.
544 DOI:10.1016/j.jhydrol.2014.01.062

545 Humphrey, G.B., Gibbs, M.S., Dandy, G.C., Maier, H.R., 2016. A hybrid approach to monthly
546 streamflow forecasting: Integrating hydrological model outputs into a Bayesian artificial
547 neural network. *Journal of Hydrology*, 540: 623-640. DOI:10.1016/j.jhydrol.2016.06.026

548 Ioffe, S., Szegedy, C., 2015. Batch normalization: accelerating deep network training by reducing
549 internal covariate shift, *Proceedings of the 32nd International Conference on International*
550 *Conference on Machine Learning - Volume 37*. JMLR.org, Lille, France, pp. 448-456.

551 Karthikeyan, L., Nagesh Kumar, D., 2013. Predictability of nonstationary time series using wavelet
552 and EMD based ARMA models. *Journal of Hydrology*, 502: 103-119.
553 DOI:10.1016/j.jhydrol.2013.08.030

554 Lima, A.R., Cannon, A.J., Hsieh, W.W., 2016. Forecasting daily streamflow using online sequential
555 extreme learning machines. *Journal of Hydrology*, 537: 431-443.
556 DOI:10.1016/j.jhydrol.2016.03.017

557 Liu, C., Zhu, L., Ni, C., 2018a. Chatter detection in milling process based on VMD and energy
558 entropy. *Mechanical Systems and Signal Processing*, 105: 169-182.
559 DOI:10.1016/j.ymssp.2017.11.046

560 Liu, H., Mi, X., Li, Y., 2018b. Smart multi-step deep learning model for wind speed forecasting
561 based on variational mode decomposition, singular spectrum analysis, LSTM network and
562 ELM. *Energy Conversion and Management*, 159: 54-64.
563 DOI:<https://doi.org/10.1016/j.enconman.2018.01.010>

- Mohammadi, K., Eslami, H.R., Kahawita, R., 2006. Parameter estimation of an ARMA model for river flow forecasting using goal programming. *Journal of Hydrology*, 331(1-2): 293-299. DOI:10.1016/j.jhydrol.2006.05.017
- Mohanty, S., Gupta, K.K., Raju, K.S., 2018. Hurst based vibro-acoustic feature extraction of bearing using EMD and VMD. *Measurement*, 117: 200-220. DOI:10.1016/j.measurement.2017.12.012
- Naik, J., Dash, S., Dash, P.K., Bisoi, R., 2018. Short term wind power forecasting using hybrid variational mode decomposition and multi-kernel regularized pseudo inverse neural network. *Renewable Energy*, 118: 180-212. DOI:<https://doi.org/10.1016/j.renene.2017.10.111>
- Niu, M., Hu, Y., Sun, S., Liu, Y., 2018. A novel hybrid decomposition-ensemble model based on VMD and HGWO for container throughput forecasting. *Applied Mathematical Modelling*, 57: 163-178. DOI:10.1016/j.apm.2018.01.014
- Pedregosa, F. et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825-2830.
- Peng, L., Liu, S., Liu, R., Wang, L., 2018. Effective long short-term memory with differential evolution algorithm for electricity price prediction. *Energy*, 162: 1301-1314. DOI:<https://doi.org/10.1016/j.energy.2018.05.052>
- Seo, Y., Kim, S., Kisi, O., Singh, V.P., 2015. Daily water level forecasting using wavelet decomposition and artificial intelligence techniques. *Journal of Hydrology*, 520: 224-243. DOI:10.1016/j.jhydrol.2014.11.050
- Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., Freitas, N.d., 2016. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1): 148-175. DOI:10.1109/JPROC.2015.2494218
- Shiri, J., Kisi, O., 2010. Short-term and long-term streamflow forecasting using a wavelet and neuro-fuzzy conjunction model. *Journal of Hydrology*, 394(3-4): 486-493. DOI:10.1016/j.jhydrol.2010.10.008
- Stojković, M., Kostić, S., Plavšić, J., Prohaska, S., 2017. A joint stochastic-deterministic approach for long-term and short-term modelling of monthly flow rates. *Journal of Hydrology*, 544: 555-566. DOI:10.1016/j.jhydrol.2016.11.025
- Tan, Q.-F. et al., 2018. An adaptive middle and long-term runoff forecast model using EEMD-ANN hybrid approach. *Journal of Hydrology*. DOI:10.1016/j.jhydrol.2018.01.015
- Valentini, G., Dietterich, T.G., 2004. Bias-variance analysis of support vector machines for the development of svm-based ensemble methods. *Journal of Machine Learning Research*, 5: 725--775.
- Valipour, M., Banihabib, M.E., Behbahani, S.M.R., 2013. Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir. *Journal of Hydrology*, 476: 433-441. DOI:10.1016/j.jhydrol.2012.11.017
- Wang, H., Wang, C., Wang, Y., Gao, X., Yu, C., 2017. Bayesian forecasting and uncertainty quantifying of stream flows using Metropolis–Hastings Markov Chain Monte Carlo algorithm. *Journal of Hydrology*, 549: 476-483. DOI:10.1016/j.jhydrol.2017.03.073

- Wang, W.-C., Chau, K.-W., Cheng, C.-T., Qiu, L., 2009. A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. *Journal of Hydrology*, 374(3-4): 294-306. DOI:10.1016/j.jhydrol.2009.06.019
- Wang, W.-c., Chau, K.-w., Xu, D.-m., Chen, X.-Y., 2015. Improving Forecasting Accuracy of Annual Runoff Time Series Using ARIMA Based on EEMD Decomposition. *Water Resources Management*, 29(8): 2655-2675. DOI:10.1007/s11269-015-0962-6
- Werbos, P.J., 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* 78(10): 1550-1560.
- Yang, J., Kim, J., 2018. An accident diagnosis algorithm using long short-term memory. *Nuclear Engineering and Technology*, 50(4): 582-588. DOI:<https://doi.org/10.1016/j.net.2018.03.010>
- Yaseen, Z.M. et al., 2017. Novel approach for streamflow forecasting using a hybrid ANFIS-FFA model. *Journal of Hydrology*, 554: 263-276. DOI:10.1016/j.jhydrol.2017.09.007
- Yu, X., Zhang, X., Qin, H., 2018. A data-driven model based on Fourier transform and support vector regression for monthly reservoir inflow forecasting. *Journal of Hydro-environment Research*, 18: 12-24. DOI:10.1016/j.jher.2017.10.005
- Zhang, J., Zhu, Y., Zhang, X., Ye, M., Yang, J., 2018. Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas. *Journal of Hydrology*, 561: 918-929. DOI:10.1016/j.jhydrol.2018.04.065